Designing Visual Explanations of Deep Learning Classifiers Decisions in Medical Image Analysis

Martin Charachon

université
PARIS-SACLAY

Designing Visual Explanations of Deep Learning Classifiers Decisions in Medical Image Analysis

Élaboration d'Explications Visuelles des Décisions de Classifieurs entrainés par Apprentissage Profond en Imagerie Médicale

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573 : interfaces : matériaux, systèmes, usages (INTERFACES)
Spécialité de doctorat: Mathématiques appliquées
Graduate School : Sciences de l'ingénierie et des systèmes, Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche Mathématiques et Informatique pour la Complexité et les Systèmes (Université Paris-Saclay, CentraleSupélec), sous la direction de Paul-Henry **COURNÈDE**, Professeur des Universités, la co-direction de Céline **HUDELOT**, Professeure des Universités, la co-supervision de Roberto **ARDON**, Docteur et Lead Data Scientist chez Incepto Medical

Thèse soutenue à Paris-Saclay, le 18 Janvier 2023, par

Martin CHARACHON

Composition du jury

Hugues TALBOT Professeur, Université Paris-Saclay	Président
Diana MATHEUS Professeure, Ecole Centrale Nantes	Rapportrice & Examinatrice
Nataliya SOKOLOVSKA Professeure, Université Paris 6	Rapportrice & Examinatrice
Laurence ROUET Docteure, Philips	Examinatrice
Damien GARREAU Maître de conférences, Université Côte d'Azur	Examinateur

THESE DE DOCTORAT
NNT : 2023UPAST018

université PARIS-SACLAY

ÉCOLE DOCTORALE
Interfaces:
matériaux, systèmes, usages

Titre: Élaboration d'Explications Visuelles des Décisions de Classifieurs entrainés par Apprentissage Profond en Imagerie Médicale

Mots clés: IA Explicable, Apprentissage profond, Imagerie médicale, Classification, GANs, Transposition de domaine.

Résumé: Les solutions d'intelligence artificielle (IA) révolutionnent le protocole de travail en radiologie, de l'acquisition des images au diagnostic. Parmi ces solutions, les modèles d'apprentissage profond atteignent des performances de plus en plus précises et rivalisent avec les cliniciens sur certains problèmes d'imagerie médicale. Ces solutions visent à faire évoluer la pratique des radiologues en leur apportant une aide supplémentaire. Cependant, les résultats de ces modèles sont souvent fournis aux radiologues sans aucune argumentation. Cela diffère nettement de leur pratique clinique (notamment pour les problèmes de diagnostic clinique), où les observations faites lors de l'examen d'imagerie conduisent à un diagnostic qui est communiqué à leurs pairs avec des explications. Les modèles d'apprentissage profond s'appuient sur des architectures complexes (réseaux de neurones avec des milliers voire millions de paramètres) pour atteindre des performances élevées au prix d'une perte de transparence (boîte noire), c'est-à-dire que leur raisonnement et leur résultats sont peu, voire pas explicables.

Dans cette thèse, nous développons des techniques explications visuelles pour comprendre les décisions de modèles de classification, entraînés sur un problème d'imagerie médicale. Notre outil vise les utilisateurs cliniciens et les concepteurs du modèle. Il met en évidence les régions de l'image qui sont pertinentes pour le modèle, fournit un aperçu de leurs forces et faiblesses, et des idées pour les améliorer. En s'appuyant sur certaines spécificités des images médicales, nous proposons une formulation générale pour fournir des explications visuelles en adoptant une perspective de génération d'images. Nous définissons un ensemble de propriétés qui contraignent l'optimisation de deux modèles génératifs conditionnels et garantissent les objectifs de l'explication visuelle. Notre formulation exploite les techniques de transposition de domaine pour produire deux images, une stable et une contrefactuelle, appartenant à la distribution des données. Ces images étant classées similairement et différemment de l'image étudiée, respectivement. Notre explication visuelle est composée de (i) cet exemple contrefactuel, montrant les transformations réalistes qui différencient les décisions du modèle ; (ii) et une carte d'attribution basée sur la différence entre les deux images générées (stable et contrefactuelle). Cette carte d'attribution met en avant les régions de l'image les plus pertinentes pour le modèle.

Nous proposons différentes implémentations de la formulation générale en ajoutant incrémentalement les propriétés. Nous validons notre méthodologie par des expériences exhaustives sur deux problèmes d'imagerie médicale. Nous démontrons que nos méthodes (i) surpassent les techniques d'attribution de l'état de l'art sur plusieurs métriques d'évaluation, (ii) peuvent identifier les biais dans l'entraînement du modèle et fournir des indications pour l'améliorer, (iii) et peuvent être étendues à d'autres problèmes de classification satisfaisant certaines contraintes.

Title: Designing Visual Explanations of Deep Learning Classifiers Decisions in Medical Image Analysis

Keywords: Explainable AI, Deep learning, Medical image, Classification, GANs, Domain translation.

Abstract:

Artificial intelligence (AI) solutions are revolutionizing radiology workflow, from image acquisition to diagnosis. Among these solutions, deep learning models achieve increasingly accurate performance and compete with clinicians on some medical imaging problems. These solutions aim to evolve the radiologists' practice by providing them with additional assistance. However, the results of these models are often provided to radiologists without argumentation. This differs significantly from their clinical practice (especially for clinical diagnosis problems), where the observations made on the imaging examination lead to a diagnosis that is communicated to peers with explanations. Deep learning models leverage complex architectures (neural networks) to reach high performances at the cost of being black boxes, i.e., Their reasoning and results being little or not explicable.

In this thesis, we design and develop visual explanations to understand the decisions of deep classification models trained to solve medical imaging problem. Our tool targets the clinician end-users and the model's designers. It highlights image regions relevant to the model, provides insights into their forces and weaknesses, and hints for improvement. Based on the specificities of medical images, we propose a general formulation to provide visual explanations through an image generation perspective. We define a set of properties to ensure the objectives of the visual explanation in this context. We enforce them as constraints to the optimization procedure of two conditional generative models. Our formulation leverages domain translation techniques to produce an in-distribution stable and counterfactual image, classified similarly and oppositely to the input. Our visual explanation brings together (i) this counterfactual example, showing the realistic transformations that differentiate the model decisions; (ii) and an attribution map based on the difference between the two generated images (i.e., stable and counterfactual). The attribution map highlights the most relevant regions of the input to the model.

We propose different implementations of the general formulation by adding the properties incrementally. We validate our methodology through exhaustive experiments on two medical imaging problems. We demonstrate our methods (i) outperform state-of-the-art attribution techniques on several evaluation metrics, (ii) can identify confounding training bias and provide hints for improvements, (iii) and can be extended to other classification problems satisfying certain constraints.

Acknowledgements

First, I would like to thank my two academic supervisors, Paul-henry Cournède and Céline Hudelot, for their scientific supervision and support during these three years. A special thanks to Céline, who spent a lot of time proofreading the manuscript. A big thank you to my Incepto supervisor, Roberto Ardon, for his support and the time he devoted to me. I really enjoyed this collaboration made of many exchanges and shared ideas. My most sincere thanks to Nataliya Sokolovska and Diana Mateus for evaluating my work and to Laurence Rouet, Damien Garreau, and Hugues Talbot for judging this work as examiners.

Thank you, Paul-Henry, Céline, and other MICS members, for welcoming me to your research team.

I want to thank the Incepto team for its atmosphere, dynamism, environment, and values that make it the company where I wish to continue my journey. Special thanks to Florence Moreau and Benoit Bayol for allowing me to join Incepto four years ago (for an internship), then Florence and Roberto for allowing the creation of the thesis project. I would also like to thank all the people who have been involved in the project during these three years (from precious pieces of advice to computation cluster resources and paper reviewers). A big thank you to Camille and Tom, who contributed to the project by working on perturbation approaches and applying our proposed methods to Keros.

I want to thank my family and friends for their support and efforts to understand my work. Finally, thank you to Marie for being there throughout this thesis. A page is turned, and a new adventure begins!

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AGen	Adversarial Generation
AI	Artificial Intelligence
AOPC	Area Over the Perturbation Curve
BBMP	Black Box Meaningful Perturbation [Fong 17]
CyCE	Cyclic Conditioned Explanation
CyImageCE	Cyclic Image-level Conditioned Explanation
CyLatentCE	Cyclic Latent-level Conditioned Explanation
CySCGen	Cyclic Stable and Counterfactual Generations
DL	Deep Learning
DP	Dual Path
FNR	False Negative Rate
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
IG	Integrated Gradient
IoU	Intersection over Union
JS	Jenson-Shannon distance
MGen	Mask Generator
NCC	Normalized Cross Correlation
RISE	Randomized Input Sampling for Explanation [Petsiuk 18]
SAGen	Stable and Adversarial Generations
SP	Single Path
SSyGen	Single Symmetrical Generation
SyCE	Symmetrically Conditioned Explanation
SySCGen	Symmetrical Stable and Counterfactual Generations
XAI	eXplainable Artificial Intelligence

List of Symbols

Below is the list of symbols that have been used throughout this thesis:

x	Input image
f	Classification model
τ_f	Optimal threshold to provide binary classification decision
χ	Input space
χ_0	Space of inputs predicted in class 0
χ_1	Space of inputs predicted in class 1
c_f	Predicted class
g_s	Stable generator model
g_c	Counterfactual generator model
$\mathbf{x_s}$	Stable image
$\mathbf{x_c}$	Counterfactual imahe
$\mathcal{E}$	Visual explanation
$\mathcal{I}_f$	Function enforcing ordered relevance in the attribution map
d_s	Distances measuring the proximity of input x to the stable generation $g_s(x)$
d_c	Distances measuring the proximity of input x to the counterfactual generation $g_c(x)$
r_g	Penalization of errors inherited from the generation processes
g_0	Auxiliary generators with same constraints as g_c but only on sub-space χ_0
g_1	Auxiliary generators with same constraints as g_c but only on sub-space χ_1
L_d^{cy}	Cyclic reconstruction loss
L_d^{sy}	Symmetrical reconstruction loss
L_d^{st}	Stable reconstruction loss
L_d^{c}	Counterfactual reconstruction loss
L_f^{cy}	Cyclic classification loss
L_f^{sy}	Symmetrical classification loss
L_f^{st}	Stable classification loss
L_f^{c}	Counterfactual classification loss
L_D, L_{GAN}	GAN loss
L_g^w	Weight proximity loss
L_{reg}	Regularization loss

γ	Path between the input and the generated counterfactual
$\mathcal{E}_w$	Weighted visual explanation via a path-based approach
$\mathcal{E}_{FI}^{v1}$	Weighted visual explanation version 1
$\mathcal{E}_{FI}^{v2}$	Weighted visual explanation version 2
$\mathcal{E}_{w,k_\sigma}$	Regularized weighted visual explanation
$\mathcal{E}_{FI,k_\sigma}^{v2}$	Regularized weighted visual explanation version 2
$\mathcal{IG}_c^{v1}$	Integrated Gradient with counterfactual baseline
$\mathcal{IG}_c^{v2}$	Integrated Gradient with counterfactual baseline version 2
$\mathcal{IG}_{c,k_\sigma}^{v2}$	Regularized Integrated Gradient with counterfactual baseline version 2

Contents

List of Figures

List of Tables

1

Introduction

1.1 At the Crossroads of Radiology and Artificial Intelligence

In recent years, the performance of medical imaging has been steadily increasing. While it took about 30 minutes to obtain 40 medical images in 1980, machines can produce 1000 images in 4 seconds today. In parallel, the multiplication of modalities (X-ray, CT, ultrasound, MRI, PET, ...) allows clinicians to have additional and complementary data to support their diagnosis.
Reading a medical examination consists of browsing through several volumes of acquisition (images 2D or 3D) potentially available in various modalities and with different techniques. For 3D modalities, clinicians unpack the volume by navigating through the different acquisition axes, i.e., most generally axial, coronal, and sagittal (see Figure 1.1).

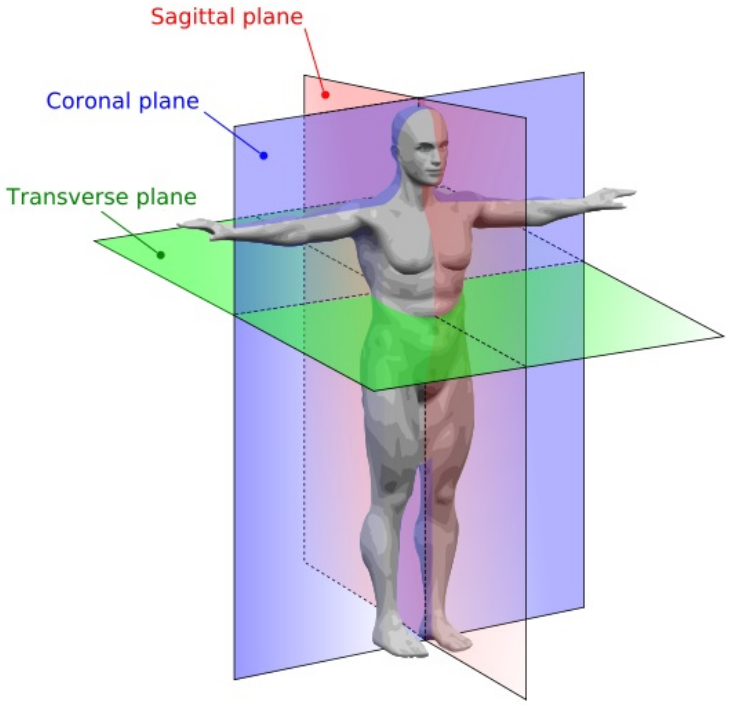


Figure 1.1: Anatomical planes in medical image analysis. The figure comes from [Mrabet 12]. Note that the transverse plane is also called the axial plane.

During this analysis phase, the radiologist reports all the elements corresponding to the clinical indication and a set of elements called "incidentalomas," which correspond to chance discoveries. Clinicians can also study previous radiologist reports if available and

recommended for decision-making, e.g., following the evolution (in time) of a disease, a treatment, or an operation. Then, the clinicians must synthesize and reformulate all of these elements in conclusion regarding the clinical indication. In real-time, clinicians often report these elements with a man-machine acquisition system through a Dictaphone. This process increases the probability of nomenclature and syntax depending strongly on the radiologist. In some cases, particularly in emergencies, a third-party radiologist may repeat these steps to confirm or invalidate the diagnosis, i.e., rereading all these elements. Analyzing all these data to diagnose is time-consuming, especially with traditional methods. It can be a source of significant errors, especially with increased fatigue and stress, or depending on who performs the examination, e.g., an emergency physician, an expert radiologist, or a non-expert radiologist (expert on the particular task at stake).

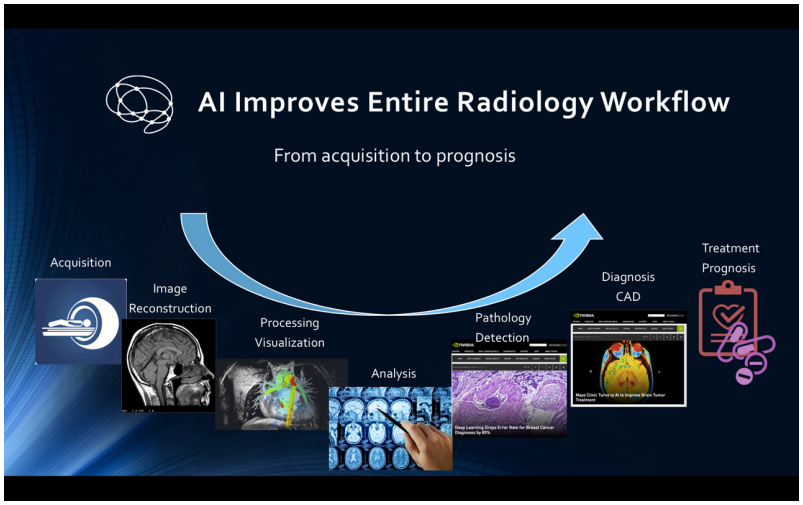


Figure 1.2: AI in the radiology workflow. Figure source: https://subtlemedical.com/ai-is-starting-to-change-radiology-for-real/

In parallel, in research and industry, Artificial Intelligence (AI) solutions are experiencing strong growth in many areas ranging from finance to medicine, including energy optimization, justice, or sports. In the medical domain and especially in radiology, AI solutions can impact the entire workflow from the image acquisition (e.g., by improving the image quality or allowing to reduce the dose of contrast agents) to the diagnosis (e.g., by detecting pathology and improving visualizations) or even to the prognosis of the treatment. Figure 1.2 illustrates the scope of AI applications in radiology.

Most of these medical applications are developed through deep learning (DL) models – a sub-domain of machine learning which is itself a domain of artificial intelligence. DL models represent the state-of-the-art for many computer vision tasks (especially in the medical image domain [Bien 18b, Esteva 17]). In medical imaging analysis, these tasks are mostly:

- Classification either to detect in a given image the presence or absence of a lesion or to categorize between different possible lesions.
- Segmentation to localize specific structures (e.g. organs, pathologies) at a pixel level. For machine learning models, it consists in classifying each pixel to establish whether it belongs to a given structure or not. At a weaker level, localization models provide boxes including the targeted structure.
- Generative modeling (using architectures such as Autoencoder, Variational autoen-

coder, Generative Adversarial Networks, Normalizing Flows). We can adopt these techniques to improve image qualities (e.g., image reconstruction, denoising); to generate "new" images to augment/enrich the training database; to translate images from one modality to another (e.g., CT to MRI), in order to transfer the capacity of a model (e.g., segmentation model) learned on the first modality into the other one. [Liu 21] describe such applications.

Deep learning models adapt well to image problems compared to previous machine learning approaches (e.g., linear model, Support Vector Machine, Decision tree, Random Forest, or rule-based model). They extract and learn relevant features (for a given task) by themself and produce better performances. The readers may refer to [Goodfellow 16, Egger 22] for an overview of deep learning methods.

1.2 Incepto: Saving Time. Saving Life. Together

Incepto[1] is a company founded in 2018 that aims to provide radiologists with technical AI tools to help them save time and to assist them in making the diagnosis. Incepto focuses on two main axes:

1. Developing AI solutions through co-creation projects with clinical partners. The physicians bring the clinical problem and the need for AI solutions. Together with Incepto's technical and clinical teams (i.e., radiologists working at Incepto), they clarify the clinical question(s) to be answered by the solution, the type and quantity of data, the type and the need for clinical annotations (of the data), the targeted users, and the potential issues (e.g., algorithm complexity, financial risk). Then, the development cycle begins, mobilizing radiologists (or other physicians) for the annotations, Incepto's data science team for the machine (or deep) learning algorithms, and the development team for the design of the product and the integration in the clinical routine. All the different teams work together, sharing feedback to identify the weaknesses of the solution (e.g., Clinicians review the algorithm output with the data-science team and give clinical insights for improvement) and to avoid divergence.
2. Distributing AI solutions from co-creation projects and AI partners (other AI companies) on a common platform. They aim to become a leader in distributing AI solutions for all radiology specialties by bringing a large community together and facilitating access to these solutions with a unified platform.

Today, Incepto distributes about 15 to 20 AI applications in more than 200 hospitals and clinics across France and has started its expansion to Europe (Switzerland, Germany, Spain, and Portugal). The AI solutions cover a large panel of radiology specialties (e.g., senology, neuroradiology, chest imaging, musculoskeletal imaging) and imaging modalities (e.g., X-ray, CT, MRI, mammography, PET). These solutions also target different objectives, e.g., pathology detection, pathology or anatomical structures measurement, or even image reconstruction. To illustrate this diversity, we first present some examples of solutions developed by Incepto's partner:

- **BoneView**, developed by Gleamer[2], analyzes radiographs (X-rays images), detects and localizes fractures on each image, then presents the results to the practitioners directly on their screen.

[1] Learn more about Incepto at https://incepto-medical.com/en/incepto/about-us
[2] https://www.gleamer.ai/

- **Transpara**, developed by ScreenPoint[3], assesses the presence or absence of breast cancer in mammography and localizes the pathology in the image.
- **Veye Chest**, developed by Aidence[4], detects and characterizes pulmonary nodules in computed tomography (CT). It also follows up on the evolution of the nodules and integrates automated results in the radiologist's reading environment.
- **SubtlePET** developed by Subtle Medical[5], applies to nuclear medicine. The solution enables regenerating high-quality images from low-quality images. The solution reduces either the injected dose of contrast agent, the acquisition time of the images, or even both. A similar solution **SubtleMR** works for magnetic resonance imaging (MRI).

Concerning co-creation projects, Incepto is distributing two products:

- **KEROS**, developed in partnership with the Swiss group 3R (Réseau Radiologique Romand), analyses the main anatomical segments of the knee on MRI images and detects lesions of ligaments, menisci, and cartilage. For this task, classification models are trained on each structure to detect lesions producing the following output decision: no pathology detected, doubt or pathology detected. Then, a report aggregates the results of all structures.
- **ARVA**, developed with the Hôpital Marie Lannelongue, automatically measures the diameter of the aorta on computed tomography (CT), points out the maximum diameter on each of the 7 anatomical sections of the aorta, and follows the evolution of these measurements if previous examinations are available. The solution targets radiologists and cardiovascular surgeons to assist them in detecting, to localize, and following vascular aneurysms. ARVA first segments the aorta, calculates its center line, and computes the diameter all along. Then, the algorithm identifies the different sections of the aorta and sets the maximum diameter for each section.

1.3 A critical need for Explainable AI (XAI)

Although increasingly accurate, deep learning models sometimes learn spurious correlations. For instance, [Zech 18] shows that models diagnosing diseases on X-rays may learn confounding information coming from the hospital department, the imaging system, e.g., the quality of images or the prevalence of diseases in the different sites. Deep learning models are still missing a general framework providing a human-readable explanation of their results compared to more transparent models (e.g., linear, rule-based, decision tree). In radiology (as in other critical fields), human conclusions on images are generally communicated to peers with explanations. It aims to increase confidence by stimulating criticism or approval. Similarly, it is imperative to provide explanations for deep learning results. Their adoption in sensitive fields such as clinical practice is at stake. As such, the notions of explainability and interpretability [Gilpin 18, Arrieta 20] –aiming to understand and clarify the decision of an AI system – have arisen and developed in the field of artificial intelligence.

In particular, given an input, deep learning classification models provide a decision by predicting one (or several) specific class(es) among all possible classes of the problem at stake. However, this decision comes without any justification or argumentation (see

[3] https://screenpoint-medical.com/
[4] https://www.aidence.com/
[5] https://subtlemedical.com/

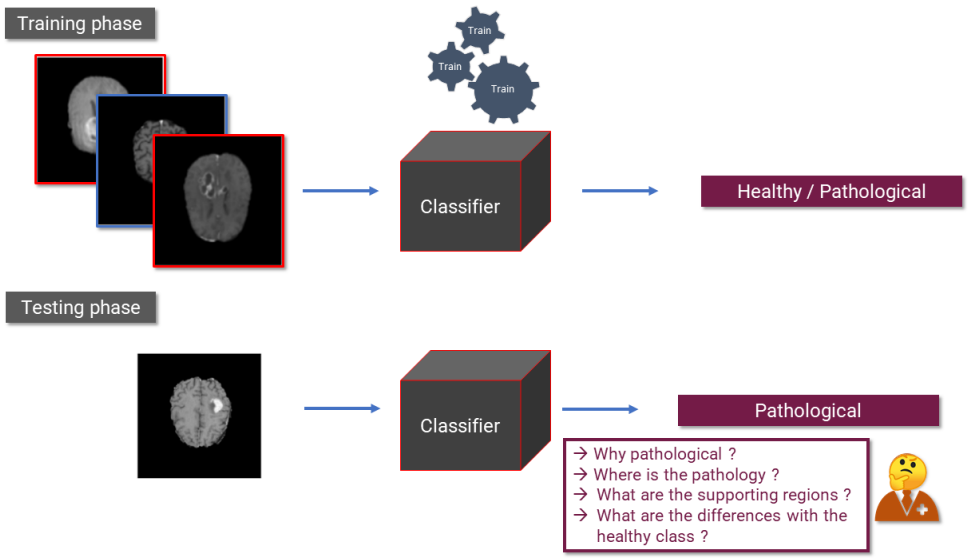


Figure 1.3: Training and testing phase for a classification model. Top: During the training phase, a classification model learns to identify pathological against healthy brain MRI images. Bottom: At test time, given a new input image, the classifier provides a decision (here pathological) but without any justification.

illustration in Figure 1.3), contrary to clinician usage. Deep learning classification models appear as black boxes because the user does not know the model reasoning, what features (relevant or confounding) of the input support its decision, or what type of patterns it has learned during its training. In contrast, segmentation or localization models generate visual outputs that the user (i.e., the clinician in medical imaging) can check to accept or reject the results. Explaining the model's prediction is therefore less critical in these other vision tasks. However, It is much simpler to obtain classification annotations than segmentation masks (and less time-consuming). This explains why classification is the most common task. In medical image analysis, classification models are especially applied to detect abnormalities in images (e.g., pathology, lesions, metallic artifact) or to identify the type of pathology (among different classes). At Incepto, KEROS belongs to the first set of methods as it aims to detect the presence or absence of lesions on each structure of the knee (training one specific classification model per structure). For this type of task, we would like the explanation method (i) to highlight if the model relies on relevant clinical features; and (ii) to show what patterns are more supportive (for the model) of one class against another.
In order to better position our work, we present a general snapshot of XAI in the next section.

1.4 A general snapshot of explainability for data-driven AI

The set of approaches and studies that aims at explaining an artificial intelligence system, in a large sense refers to the XAI (eXplainable Artificial Intelligence) domain. It is a very vast notion and has multiple meanings and applications. Many contributions have been made in this topic ranging from post-hoc explanations of trained DL models (see Subsection 1.4.2), to network dissection, or various approaches to build interpretable DL models (see Subsection 1.4.3).

This section proposes a general overview of explainability and interpretability in machine learning systems. We mainly focus on deep learning models in the case of image classification tasks (our primary concern), and we emphasize applications in the medical domain in blue boxes.

1.4.1 Definitions and problems

First, the literature on explainability and interpretability does not reach a real consensus. We find different definitions and classifications of these notions. Yet, similarities exist, and we introduce the different concepts based on several reviews [Doshi-Velez 17, Gilpin 18, Zhang 18c, Guidotti 19, Murdoch 19, Arrieta 20, Xie 22] –to which readers may refer:

1. **Explainability** describes how the model works accurately and the mechanisms involved in generating either a specific prediction or a type of prediction.
2. In contrast, **interpretability** tries to describe the system in a way understandable for the end-users; without analyzing all the details that lead to the model's decision. It is more related to the application domain and users' knowledge and biases.

Medical domain: More recently, [Patr'icio 22] reviewed the advances in explainable deep learning applied to the medical domain.

Different explanation methods exist and are used for different tasks and objectives. We illustrate the general taxonomy of explainable AI in Figure 1.4.

We first separate **explainable models** from **post-hoc explanations**. The first models are explainable by design or at least generate their explanation mechanism that clarifies either the model's outcome or the reasoning process. On the opposite, post-hoc explanations rely on a trained black box model. For this type of explanations, the available resources also condition the type of explanation. We consider different explanation methods according to available resources: do we have access to the model structure (especially in the case of deep neural networks), to a database or a single data, to a set of explaining features (from domain experts), or does it depend on a particular type of model (i.e., **model-agnostic** or not).

Then, we distinguish **global** from **local** explanations. Global explanations display the common features a model has learned through a database to define all its possible decisions, e.g, for a classification model detecting if an MRI image contains tumors, this type of explanation would show patterns learned by the model that characterize tumors in the database: intensity, texture. Local explanation only focuses on a single model's decision e.g., we aim to explain why the classification model detects a tumor on a given MRI image.

The type of explanation depends on the type of users and applications. Different approaches are considered for the clinician, the patient, or the data scientist because they have different needs for an explanation. For instance, the explanation could provide the patient with a simple textual explanation describing the key point of the medical image and the model's diagnostic. In contrast, in a decision-aided perspective, the clinician could need more complex details pointing out specific clinical signs in the images. In addition, the use of the explanation also drives its design. For instance, in clinical routine, the clinician can not spend time investigating the model's decision compared with data scientists who want to detail and understand all model operations (e.g., to detect issues and to plan a strategy to correct them).

According to this taxonomy, described in Figure 1.4, we present the main approaches from the literature: post-hoc explanations and explainable models by design. Except for Figure 1.4, all the following figures in this chapter come from the works they describe (citation in their title).

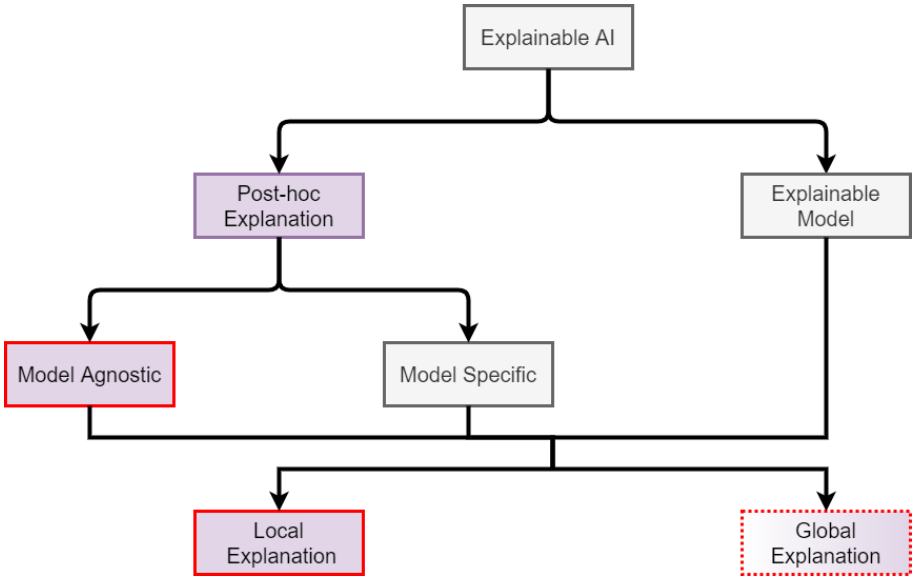


Figure 1.4: Taxonomy of explainable AI. The subdomains addressed in our work are shown in red. Dotted lines mean the work is still in progress.

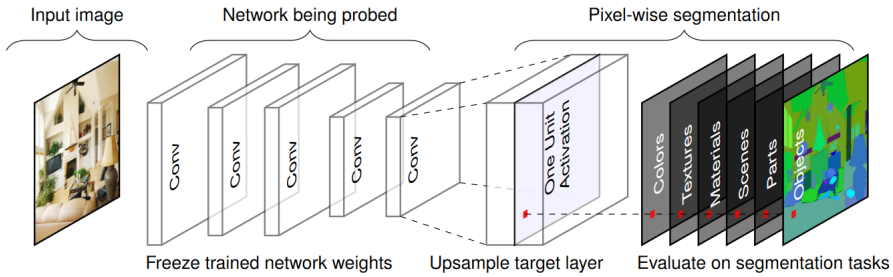
1.4.2 Post-hoc Explanation

Post-hoc explanations provide relevant information or visualization to improve models' interpretability. In these cases, the models are often not transparent or explainable by design. We study a model without modifying its structure and parameters in this situation. The objective is to produce insights into what the model has learned. It is particularly suited to explain black-box models such as deep neural networks, where the models have complex structures and many parameters (e.g., 1k to 1bn), especially for vision tasks. The processing of data and the relationships between parameters remain opaque in such models. We can analyze this black box differently depending on our objective (as mentioned in the previous section).

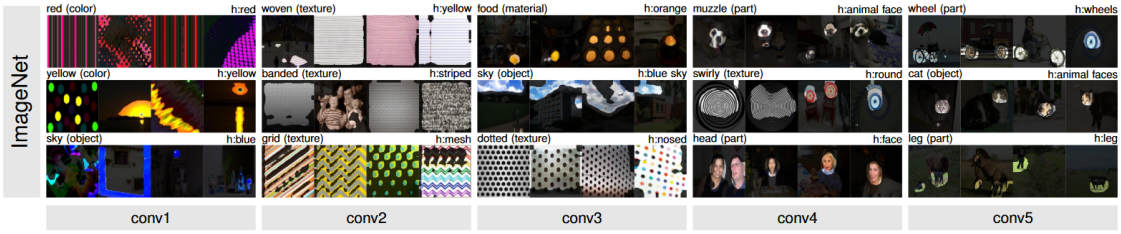
We can produce a surrogate transparent model (and interpretable) to mimic and approximate the behavior of the black box. It provides a global explanation of the trained black-box model and how it processes data. The black box at stake and a dataset are required for such an approach. In this spirit, decision trees [Zilke 16, Schmitz 99] or rule extractors [Andrews 95] can be used to decompose the neural network decision process while trying to remain as faithful as possible. Although improving the transparency, these methods often scale poorly and do not capture all the complexity of the trained model.

Other approaches inspect the internal structures and representations of the trained model. They assume that we have total access to the model and its parameters. They study the role and impact of the different components of the neural networks, leading to specific decisions from layers to individual neurons. Layers are often analyzed by studying their capacity to be reused for other tasks, i.e., transfer learning [Yosinski 14]. To analyze

the encoding representation of an image, [Zeiler 14] propose a deconvnet that reverses a convolutional network using the same components (without any learning process). The feature maps obtained for different layers of the convolutional network (for a given input) are then passed to the corresponding layers of the deconvnet. This action is repeated until the input pixel space. In a similar spirit, [Mahendran 15] try to invert the representation of different layers to reconstruct the image. They show that several CNN layers retain relevant information about the input. The transfer technique can also be used for individual units (neurons or convolution filters) at a lower granularity. In [Bau 17], the authors propose a network dissection approach to assess the alignment between individual hidden units and a set of semantic concepts in the latent representations of convolutional neural networks (see Figure 1.5a). They show that units of an image classifier can be dedicated to identifying (or locating) particular objects, colors, or textures in images (see Figure 1.5b). Based on these techniques, [Olah 18] propose an interface to navigate through internal structures of neural networks.



(a) Illustration of the network dissection



(b) Segmentations for several units.

Figure 1.5: Network dissection [Bau 17]. (a) To study convolutional unit (here of the last convolutional layer), the neural network weights are frozen, the studied layer is upsampled to the input size, and the resulting output activation is evaluated on segmentation tasks. (b) Segmentations produced by different units of AlexNet [Krizhevsky 12] (trained on ImageNet [Deng 09]) on the three images with higher activations. Different types of visual concepts are identified at each level of the network.

[Zhou 15] show that neural networks trained to perform scene or object classification can also achieve object detection. Certain objects in images maximize the activation of specific units in the classification model. Illustrations are provided in Figure 1.6.
[Nguyen 16] build upon Activation Maximization techniques and learn a generative model to produce images that maximize the activation of neurons. [Zhang 18a] disentangle features in convolutional layers, then use a graph model to give an overview of how visual knowledge is distilled in the convolutional model. In another work [Zhang 19a], the authors used a decision tree to point out which filters are used for a model's prediction and what their contributions are. A visualization method is proposed in [Simonyan 14] to gen-

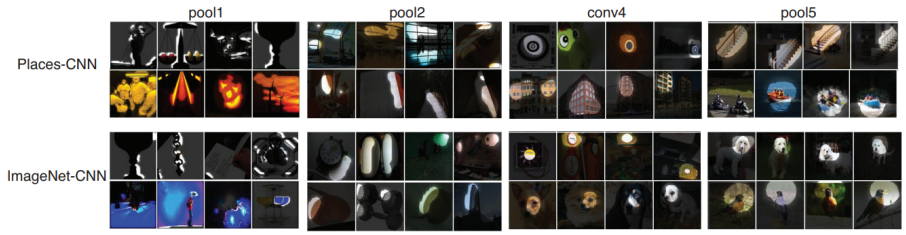


Figure 1.6: Segmentation based on maximal activation for several units [**Zhou 15**]. Segmentations are computed on images that maximize the activation of chosen units on both Places [Zhou 14] and ImageNet datasets.

erate an L_2 regularized image representative of a given class for the neural network. This image highlights some semantic patterns of the class that are understandable to humans.

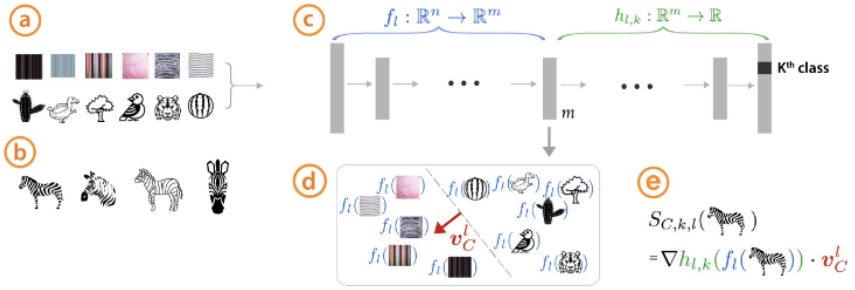


Figure 1.7: Overview of testing with Concept Activation vectors [**Kim 18**]. (a) Set of visual concepts either random or defined by the user. (b) training data for the zebras class. (c) the trained classifier. (d) A linear classifier learns to separate the activations of training and concept data (at a given layer). (d) The directional derivatives quantify the sensitivity of the class of interest against the visual concepts.

Other works analyze vector directions in the trained neural network representation space, which align with human-understandable concepts. [Kim 18] introduce Concept Activation Vectors by learning a linear classifier –in the activation space of a given layer of the model– to identify examples of a concept given by the user from random examples. The Concept Activation Vectors are thus the orthogonal vector to the hyperplane separating the two sets of examples. Then, for a given input, they test the sensitivity of the prediction against the selected concepts. The overall framework is described in Figure 1.7.
In the same spirit, [Ghorbani 19] automatically discover concepts as image segments, while [Wu 20] first attack the model to derive the importance of different features for different classes. Then, they map the feature importance to concept importance.

Medical domain: [Singla 21] use counterfactual generation techniques (see Section 2.1.5) to discover associations between medical concepts –extracted from medical reports– and the hidden units of a trained classification model (see Figure 1.8).

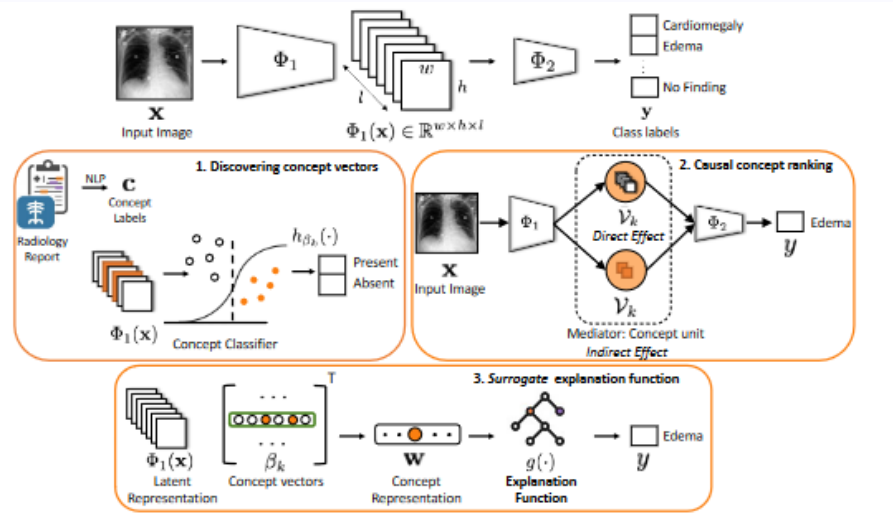


Figure 1.8: Explaining DNNs decisions with medical concepts through causal analysis [Singla 21]. On the 1st row: the trained classification model. On the 2nd row: (Left) Medical concepts are extracted from radiology reports, then the intermediate representations $\Phi_1(x)$ of the model are used to learn simple classifiers separating each concept. (Right) The non-zero coefficients of the concept vector β_k (learn via concept classification) form the set of units $\mathcal{V}_k$ the most relevant for concept k. Causal mediation analysis quantifies the relation between a concept and the model's outcome. On the last row: a decision tree maps medical concepts to class labels.

Finally, we can also explain the output of the trained model on a given input without analyzing the whole internal process, i.e., model-agnostic. Based on input perturbations, LIME [Ribeiro 16], KernelSHAP [Lundberg 17], or their bayesian versions [Slack 21] explain the prediction of a black box classifier by learning a local linear model for each prediction. They approximate the trained model's behavior in the input neighborhood. A theorical analysis of LIME is provided in [Garreau 21]. [Koh 17] instead analyze the impact of a training point on the model's prediction. They use influence functions to modify the training data slightly.
Visual explanation methods –often called attribution methods– produce a saliency map (or heat map) highlighting relevant regions from the input for the model. We further describe these approaches in Section 2.1.

1.4.3 Explainable Model

Another line of work aims to build explainable models. Here, we only present methods applying to deep neural networks and not transparent models by design (e.g., rule-based models, linear models, or decision trees). These models are locally and/or globally interpretable rather than producing an explanation for a trained black box (see the previous section). The idea is to integrate the generation of understandable explanations into the training of the model.

Some methods [Vaswani 17, Wang 17a] incorporate attention mechanisms into the deep neural network to focus features learning at a specific location. Although the modules used for attention are not explicitly trained to produce explanations understandable by humans, they provide visual maps pointing out the retained information used by the model.

Some approaches learn to generate a sentence to justify the decision taken by the model. [Hendricks 16] combine a fine-grained classifier [Gao 16] and an LSTM model [Hochreiter 97] to produce a textual answer that predicts the class label of an input image and explains why this prediction is appropriate for this image. Examples are given in Figure 1.9.

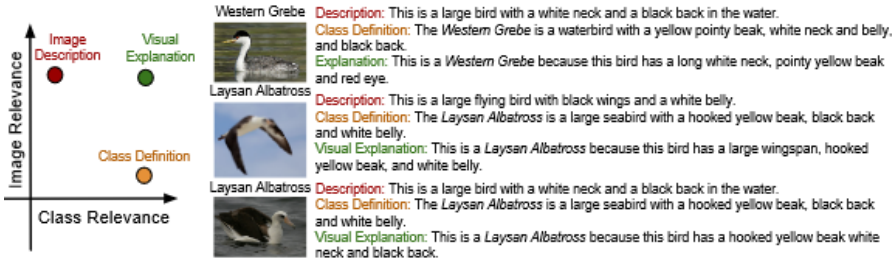


Figure 1.9: Generating textual explanations [Hendricks 16]. The generated explanation provides information related to the given input and relevant to discriminate between classes. In contrast, the image description only focuses on the image details (not always relevant for classification), while the class definition is not attached to a specific input.

In the same spirit, other self-explainable models attempt to explain the rationale for a classification decision using multi-task learning. For instance, [Park 18] produce a classification decision and then generate both visual evidence on the input image and a textual justification of the prediction. [Xu 20b] focus the model attention on discriminative image regions (attributes) and generate attribute-based textual explanations.

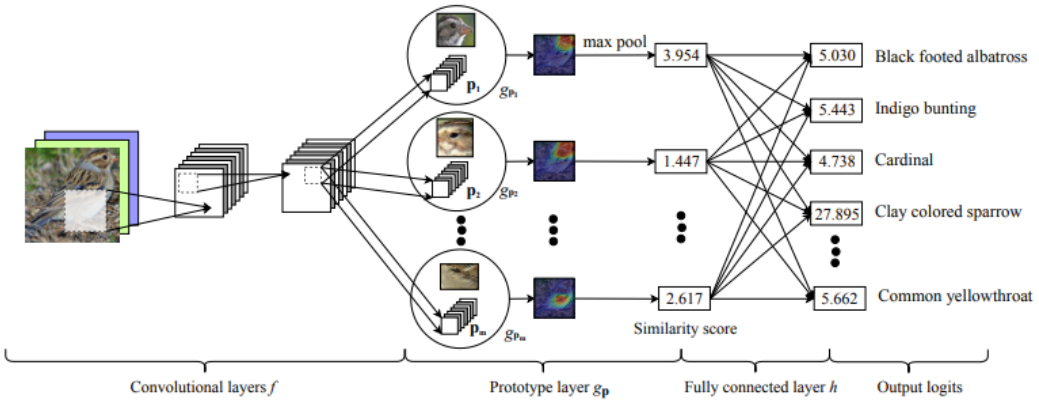
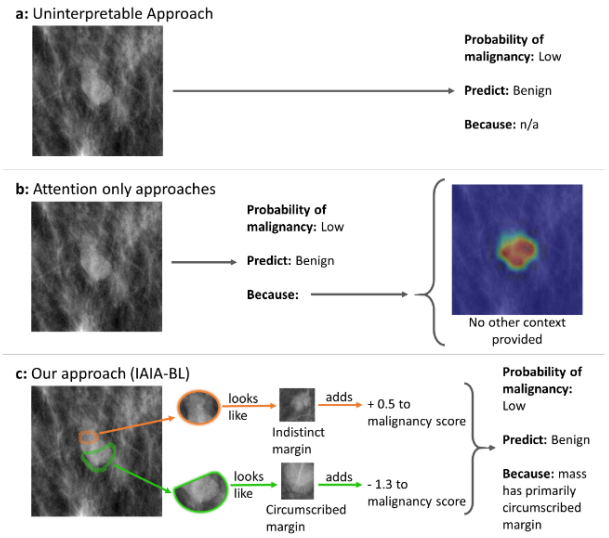


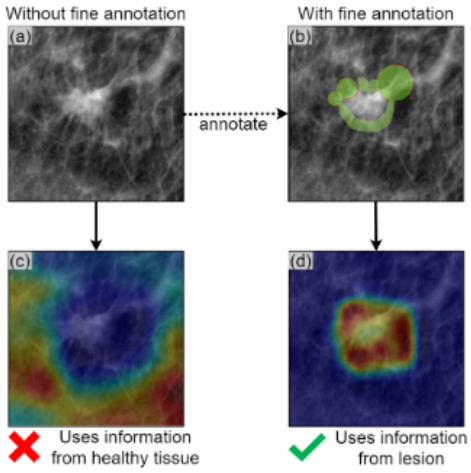
Figure 1.10: Overview of ProtoPNet architecture [Chen 19]. A first structure composed of convolution layers learns representation; then, multiple prototype layers learn discriminative parts of the inputs by decomposing the extracted representation in patches. Then, a similarity score is computed between each patch of the current input and the learned prototypes. For each input, certain prototypes are more relevant (given by the similarity scores). The model's outcome is produced by comparing the relevant prototypes of the input against relevant prototypes of each class (learned during training).

Another line of work proposes neural networks that reason similarly to humans by finding either prototype image [Li 18] or prototypical parts of the input image [Chen 19, Barnett 21], and then by combining evidence (from those prototypes) to produce the decision. Such model architecture is described in Figure 1.10. These methods combine part attention (producing visual attributions) and compare input regions with learned prototypes.

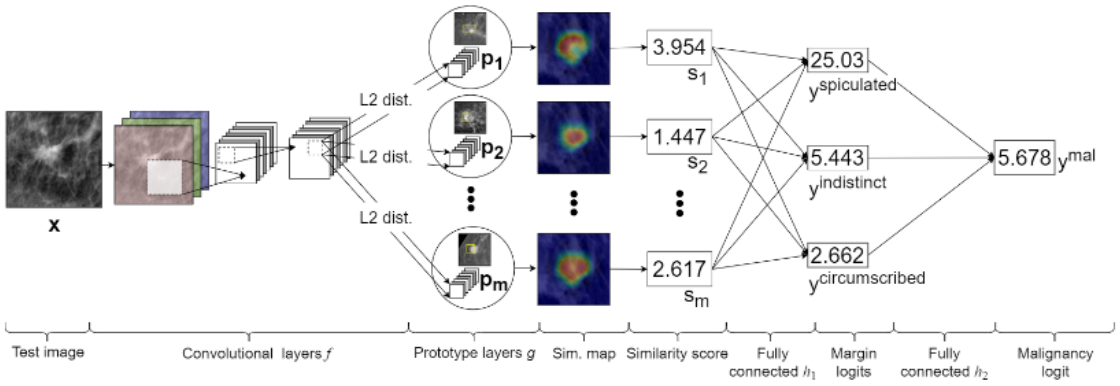
Medical domain: Built upon [Chen 19], [Barnett 21] provide a neural network using case-based reasoning for mammography analysis. They also incorporate pixel-wise annotations to guide the model's attention during training (see Figures 1.11). [Kim 21] learn dynamic prototypes (i.e., not within a patch of predefined size) of each disease from X-ray images, make a prediction on a given image based on the patterns, then produce both global (through prototype comparison) and local (visualization) explanations.



(a) IAIA-BL method vs common DNN approaches



(b) IAIA-BL additional annotations



(c) IAIA-BL architecture

Figure 1.11: Overview of IAIA-BL framework [Barnett 21]. (a) Illustration of common deep neural network decisions and explanations: **a** Black box DNNs provide a decision without any justification; **b** Post-hoc visual explanations produce an attribution map highlighting the relevant region for the decision but give no information about why attributes of this region are important; **c** IAIA-BL both puts forward relevant regions, compares them to (learned) medical signs, and then produce a decision. (b) Introduction of clinical annotations to constrain the model attention: (Top-Left) The initial input showing the lesion; (Top-Right) Clinicians marked the margin of the lesion; (Bottom-Left) Activation map obtained without further annotations; (Bottom-Right) Activation maps obtained with fine-grained annotation, highlighting relevant regions. (c) The architecture of the prototype network (similar to [Chen 19] described in Figure 1.10).

In another way, [Alvarez-Melis 18] build complex interpretable models that keep the behavior of a linear model around each input. They construct a basis of interpretable concepts in which each prediction could be decomposed. Similarly, [Boehle 21] introduce neural networks that model the classification decision as a series of input-dependent linear transformations, which allows a decomposition into individual input contributions. In contrast,

[Zhang 18b] modify the latest convolutional layer of a convolutional neural network and disentangle their representations. They add a loss to each layer's filter to enforce the feature map to encode a specific object part.

Finally, some methods leverage domain translation techniques (see Section 2.2.1) to build interpretable classifiers. Domain translation consists in learning a mapping between two different image domains (e.g., photography and painting, horse and zebra, or between diverse modalities in the medical domain). We give further details about these approaches in Section 2.2.4.

While producing a more interpretable reasoning process for each decision, these models are more complex to train and often need complex annotations (e.g., textual descriptions) or additional annotations [Xu 20b], especially when applied to medical tasks [Barnett 21]. Finally, they also face a trade-off between accuracy and interpretability.

1.5 Problems and Contributions

In this thesis, we consider the problem of explaining the decisions of a trained DL classifier in the context of medical images. Our objective is to generate post-hoc visual explanations. Numerous methods (see chapter 2), based on back-propagation techniques [Simonyan 14, Springenberg 15, Smilkov 17, Sundararajan 17], last network layers analysis [Zhou 16a, Selvaraju 17, Rajpurkar 17] or input perturbations [Fong 17, Dhurandhar 18, Hsieh 20, Fong 19], produce visual explanation maps through a process that only depends on the single input image fed to the classifier. These methods are often not model-agnostic and need regularization heuristics to produce visualization maps acceptable to humans. They require significant manual adaptations when changing the DL model, especially if the application domain changes (e.g., from natural to medical images).

In the particular case of the medical domain, the images of the different classes often share similar content (e.g., background, body structures) and differ somewhat on localized patterns.

Leveraging these specificities, we seek to build a visual explanation method (see Figure 1.12), intended for clinicians and data scientists, that:

1. highlights what the relevant regions of the input image that support the classifier's decision are.
2. shows how these regions should change to modify the decision.
3. identifies common errors and confounding training biases. For this objective, we need access to a dataset (e.g., the dataset used to train the model or a dataset used for external validation)
4. gives insights to improve the model (e.g., performance, attention, or interpretability).
5. remains the most model-agnostic (i.e., no access to the internal model structure) likely to validate partners' solutions ourselves.

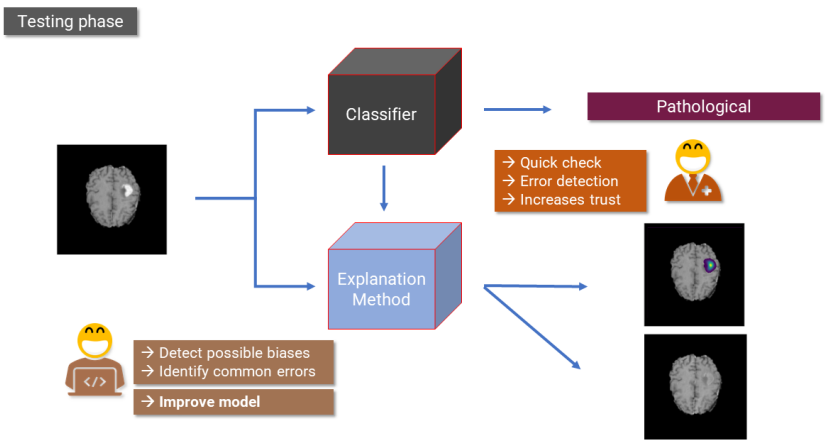


Figure 1.12: Visual explanation objectives. During the test phase, the classifier (gray box) provides a decision (e.g., "Pathological") given an input image (on the left). In addition, the explanation method produces an attribution map highlighting the relevant regions of the input (heatmap overlay on the top right image), and a counterfactual image showing how to transform the input to change the model decision (e.g., replacing tumor with healthy brain tissue). In orange and brown boxes are pointed out the intended benefits (of the visual explanations) for clinicians and data scientists.

To achieve these objectives, we turned to the field of image generation. We assume that the model's decision lean towards one class over another (for a given input) depends on the presence or absence of some specific patterns of this class. This assumption is well suited to medical imaging problems (in particular for pathology detection problems) where clinicians are looking for clinical signs to describe the images and make a diagnosis (e.g., detection of tear in the menisci or tumor tissues in the brain).
We thus introduce a method defining a visual explanation based on the difference between a stable and a counterfactual generation. In our approach, we train two generators to produce images within the distribution of real input images. In particular, the counterfactual is searched as the closest element to the input image within the distribution of real images that are classified differently. Thus, it captures crucial patterns for the classifier (see Figure 1.13). In addition, the visual explanation should translate the importance of the input features for the classifier from coarse to fine-grained details. We propose a set of properties that apply to both the generation procedure and the visual explanation computation to reach these objectives.

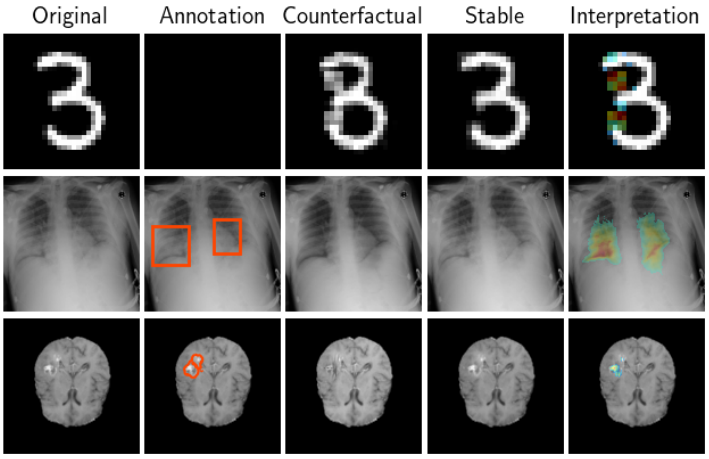


Figure 1.13: Counterfactual, stable and visual explanation generations. From left to right for three binary classification problems (i.e., MNIST digit "3 vs. 8" classification, Chest X-rays pneumonia detection, and Brain MRI tumor localization): the original image; the ground annotations are pointing out the relevant regions for humans (not available for MNIST); the counterfactual image (for the classifier to explain); the stable image; our resulting explanation map of the classifier's decision. The Intensity of the explanation map (i.e., the absolute difference between the stable and the counterfactual images) ranges from blue (values close to zero) to red (i.e., the highest intensity differences).

Compared to many previous works that only produce a map that localizes the regions of the input image relevant to the classifier, our proposed method also highlights (through counterfactual generation) how these regions should be transformed to change the classifier's decision to another class. Our contributions are as follows:

1. A formal definition of the visual explanation based on four properties and translated into a constrained optimization problem (Chap. 3).
2. Three embodiment of the general formulation by successively combining the different properties (Chap. 4, 5 and 6).
3. A novel counterfactual-based visual explanation method leveraging (i) domain translation techniques to produce counterfactuals within the distribution of real images

and (ii) path-based methods to emphasize the importance of features for any detail scales.
4. Several approximations and implementations for each embodiment. All being competitive or outperforming state-of-the-art techniques on several evaluations and for diverse medical imaging problems.
5. Publications [Charachon 20, Charachon 22, Charachon 21] illustrating the different steps of the method improvement, as well as the integration of the different properties.

1.6 Plan

The present thesis contributes to the large field of deep learning models' explainability and interpretability. We designed a novel visual explanation method intended for clinicians and algorithm developers to provide insights into the model's decisions and behavior (locally and globally) to facilitate AI tool adoption in radiology. The manuscript is organized as follows:

- A related work section (see Chapter 2) presents the different contributions on visual explanation. We first describe methods providing attribution maps that highlight the relevant regions of the input. To bridge the gap with the counterfactual generation, we introduce the main literature techniques used for domain translation. We point out some works leveraging these ideas to explain classifier decisions or build interpretable models. Then, we describe strategies to assess the quality of visual explanation and the different data sets on which we validated our work.
- A method section introduces our general formulation and describes the different embodiments. In Chapter 3, we relate the targeted objectives and outcomes of a visual explanation to three properties (i.e., Relevance, Regularity, and Realism) applying to a generation procedure and one property for the resulting feature attributions (i.e., Ordered by importance). These conditions translate into a constrained optimization problem we describe for a binary classification task, then adapt for a multi-classification setting. Using adversarial generation techniques (see Chapter 4), we describe frameworks that demonstrate the importance of both Relevance and Regularity but also point out limitations and the need for the other properties. Leveraging different frameworks and constraints (e.g., cycle consistency) from domain translation works, we propose several optimizations and architectures to generate counterfactual images (see Chapter 5). Then, we bridge counterfactual generations and path-based approach (see related work section 2.1.1) to enforce the correlation between the attribution values and their importance for the classifier (see Chapter 6).
- An experimental section (see Chapter 7) describes how we validated our methods. We present the classification tasks associated with the different data sets (described in Chapter 2). We show practical implementations of our methods (e.g., architectures building blocks, optimization steps) and comparison (i.e., state-of-the-art) techniques. Then, we introduce the experiments and evaluations to assess the performance of both the attributions and the counterfactual generations produced by our method.
- A results section (see Chapter 8) first performs an exhaustive evaluation of the generated attributions on two medical image data sets (i.e., Pneumonia detection on X-rays and Brain tumor detection on MRI slices). Then, we study the quality of the generated counterfactuals on medical and non-medical tasks; and we show

how this additional information can assist the user in identifying common errors or confounding biases of the classification model.

- A perspectives section (see Chapter 9.2) puts forward possible directions for future work and improvement of the method. We briefly describe works in progress concerning the generation of diverse counterfactuals. The objective is to point out the relative impact of different image attributes in the model's local decision on given input and to inspect what the model has learned globally.

2

Related Works on visual explanation of classifiers decisions

In this chapter, we first describe the main visual explanation approaches that we divide into two sets, based on feature attributions (see Section 2.1) or counterfactual generations (see Section 2.2). We detail the reference techniques for each set of methods and mention similar ones. Second, we present the baseline techniques to assess the quality of the visual explanations (see Section 2.3). Then, we introduce the datasets on which we evaluate our methods in this thesis (see Section 2.4). Finally, we summarize the related work's main contributions and introduce our work's preliminaries in Section 2.5.

2.1 Visual Explanation as Feature Attribution

In this section, we present feature attribution methods –especially for image classification tasks– which are more related to our work. These techniques produce a visual explanation by highlighting the relevant regions from the input for the model (in our case, a classification model). As introduced in Section 1.4.2, these approaches only focus on the relationship between the inputs and the outputs of the model without analyzing the whole internal process of the model. In general, they provide a visualization in the form of a saliency (or heat) map that attributes the importance of the different regions from the input (or the feature space) in the model's decision. Numerous methods have been proposed in the literature to produce such visualization. In the following, we describe the main state-of-the-art approaches that apply to neural networks for image classification tasks.

2.1.1 Saliency Methods

These methods leverage gradient backpropagation of small variations of the model's prediction for a trained classification neural network f and for a given input image x. They advocate that the magnitude of the derivatives translates how the change of each pixel would impact the classification output $f(x)$.

Gradient baseline- The base method [Simonyan 14] directly computes the class score derivatives w.r.t the input x. The attribution at pixel i is given by:

$$\mathcal{E}_i(x) = \frac{\partial f_c}{\partial x_i}(x) \tag{2.1}$$

Where $\mathcal{E}_i$ represents the attribution value at pixel i, the visual explanation (or attribution map) is denoted as $\mathcal{E}$. x_i is the value of the input x at pixel i. $f_c(x)$ is the output of model f for the class c (e.g. the predicted class). For multi-classification tasks, f is a

vector of size C, the number of classes, while in a binary task, f either outputs a vector of size 2 or a scalar value (in general, in $[0, 1]$). While providing interesting results, as neural networks are nonlinear functions, this method faces vanishing gradient and singular point issues. In addition, these direct derivatives can miss part of the information being processed through the network and tend to produce noisy explanation maps since any variations of the model's output are considered. Many contributions thus focus on building sharper and smoother explanation maps (see Figure 2.1).

Sharpening and Regularizing the Gradient method- [Shrikumar 16] leverage the sign and the strength of the input by multiplying the signed partial derivatives of the model's output (w.r.t the input) with the input : $\mathcal{E}_i(x) = x_i . \frac{\partial f}{\partial x_i}(x)$.
[Bach 15] rather back-propagate relevance scores instead of the gradient. Compared with gradients that describe how changes in each pixel impact the prediction, the relevance score (R_i at pixel i) assumes a decomposition property of the prediction: $f(x) = \sum_i R_i$, i.e., it shows the contribution of each pixel to the prediction. By assuming such decomposition property, the relevance score should respect a redistribution process at every layer of the neural network, i.e., the relevance is conserved at every step; the procedure is named Layer-wise Relevance Propagation (LRP). The explanation algorithm thus proceeds layer by layer from the model's output to the input and follows some rules (ϵ-rule in [Bach 15]) that are comparable to discrete gradients with particular considerations on non-linearities and adapted to each layer. Other works built upon LRP propose other rules [Montavon 18] to propagate the relevance through the layers.
DeepLIFT [Shrikumar 17] proposes an alternative to LRP and propagates a signal of importance through the network from the output to the input by analyzing the differences between the input and reference input. This reference input is defined as a neutral image (often set to the black image). This setting allows DeepLIFT to avoid artifacts related to discontinuities in the gradient. In practice, it assigns an attribution to each neuron layer by layer (as in LRP). For each unit, the attribution translates the activation difference (in the neural network) between the reference values and the input values (from x). The reference values for all hidden units of the neural network are obtained when feeding it with the reference input.

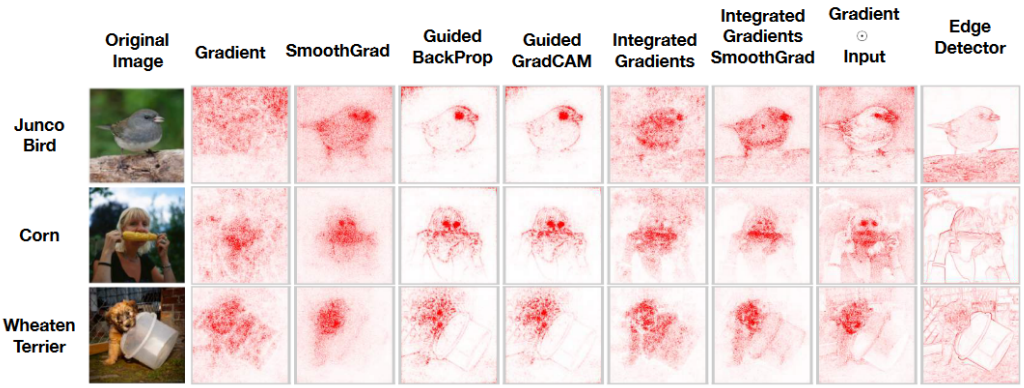


Figure 2.1: Examples of saliency maps [Adebayo 18]. Comparison between common saliency maps (computed for an Inception v3 model trained on ImageNet) and an edge detector for three inputs. Guided variants produce maps very similar to the edge detector.

Path-based methods- [Sundararajan 17] propose the integrated gradient method which mitigates the vanishing gradient and the singular point issues of [Simonyan 14] (as DeepLIFT). They first compute the gradient contribution of different intermediate inputs taken along a linear path γ between a baseline input $\bar{x}$ and the input image x i.e. $\gamma(\lambda) = \bar{x} + \lambda(x - \bar{x})$; then they average all the contributions. After simplification, the visual explanation at pixel i reads

$$\mathcal{E}_i(x) = \int_0^1 \frac{\partial f_c(\gamma(u))}{\partial \gamma(u)} \frac{\partial \gamma(u)}{\partial u} du = (x_i - \bar{x_i}) \int_0^1 \frac{\partial f_c}{\partial x_i}(\gamma(u)) du \tag{2.2}$$

The baseline is a neutral input and is often set at the zero image (in the image classification task). It is comparable to the reference input introduced in DeepLIFT. The authors identify some axioms that an attribution method should satisfy: sensitivity, implementation invariance, completeness, linearity, and symmetry-preservation. They advocate that previous methods break some axioms and bridge the gap between the implementation invariance of gradient methods and the sensitivity and completeness[1] of LRP or DeepLIFT. A comparison of these methods is found in [Ancona 18].

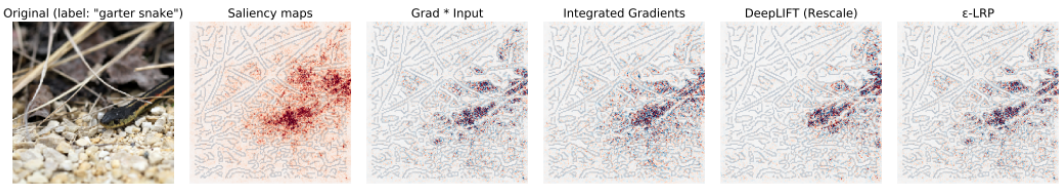


Figure 2.2: Comparison between backpropagation-based methods [Ancona 18]. Attribution maps computed for an Inception v3 network (trained on ImageNet) comparing gradient backpropagation techniques with DeepLift and ϵ-LRP that backpropagate a modified gradient (via rules chain).

Some methods build on [Sundararajan 17] by defining a more appropriate reference baseline (than the basic black image). [Jha 20] use a variational autoencoder to project images in a latent space. They set the baseline of the integrated gradient in the latent space (e.g., all zero points, the median between classes, k nearest neighbors strategy) and then map it back to the input space through the decoder. [Kapishnikov 19] first compute the integrated gradient for two reference baselines: a black and a white image. Then, all pixels should have an equal chance to contribute regardless of the baseline. For instance, if the object to detect is in black, the attribution would be null using a black image as a reference. It may introduce spurious contributions due to the closeness of pixels to the chosen baseline. Then, they combine their integrated gradient method with a segmentation algorithm that produces multiple input segments. Starting with an empty mask, they iteratively add the segment region, increasing the total attribution the most. The objective is to produce smoother and more bounded explanation maps. [Xu 20a] produce attribution in both scale (or frequency) and space which allows identifying coarse and large-scale relevant features against fine-grained features. They consider a path between the input image and the blurred version of the input (computed by a Gaussian blur filter). Here the baseline is equivalent to the maximum blurred image that is information-less. [Pan 21] revisit the integrated gradient approach by considering what makes the model

[1] The completeness means that the sum of all attributions is equal to the difference between the model's prediction for the input and the baseline.

discriminate the predicted class from all other classes; rather than explaining what makes the model choose this prediction. They establish that the gradient for the predicted class is equivalent to the negative sum of the gradient for all adversarial classes. Then, they integrate the gradient along with a set of paths between the input and adversarial examples (from all other classes or a representative sample of other classes) in the neighborhood of the input. The path is no longer linear and follows techniques used to produce adversarial attacks [Madry 18], with signed gradient steps toward the adversarial class.

Regularization by adding noise- In a different way, [Smilkov 17] point out that the derivatives of model f may vary at small scales and induce meaningless local variations in the gradient, which produce a noisy explanation map. To mitigate this issue and produce smoother maps, they propose to compute the average contribution of gradients in a neighborhood of the input x by adding Gaussian noise to the input. The visual explanation at pixel i is then

$$\mathcal{E}_i(x) = \frac{1}{N}\sum_{k=1}^{N}\frac{\partial f_c}{\partial x_i}(x+\eta_k) \tag{2.3}$$

where $\eta_k \sim \mathcal{N}(0,\sigma^2)$ is the noise sampled in a Gaussian distribution with standard deviation σ. N is the number of samples.

Alternative saliency maps- Introduced in Section 1.4.2, [Zeiler 14] can plug a deconvolutional model (not trained) from the latest features maps of the deep neural network and progressively (through the network) maps them back to the input space to create a visualization. [Bojarski 18] similarly average ReLU features maps at all scales of the model, then using deconvolution operations, multiply the averaged features map with the upsampled previous one; the operation is repeated until reaching the input space. [Springenberg 15] combine ideas (on gradient backpropagation) of the deconvnet [Zeiler 14] and the gradient [Simonyan 14]; and prevent the backward flow of negative gradients.

Despite producing sharper explanation maps (compared with [Simonyan 14]), these methods still produce noisy explanation maps, except for [Kapishnikov 19] when combined with a segmentation process. While gradient-based methods [Simonyan 14, Sundararajan 17, Smilkov 17] only need access to the gradients to backpropagate them through the neural network, other techniques require total access to the architecture of the model in order to compute specific discrete gradients [Bach 15, Shrikumar 17] or construct an adapted deconvolutional network [Zeiler 14, Springenberg 15, Bojarski 18]. In addition, [Kindermans 19] show that a transformation with no effect on the model (such as adding a shift to the input) can impact some saliency methods. Tables 2.1 and 2.2 sum up the main specificity of the different saliency techniques. Figures 2.1 and 2.2 illustrates some explanation maps produced by the different backpropagation techniques.

Table 2.1: Global comparison of Saliency methods. For each method, we indicate if the visual explanation is local or global, whether the user has access to the model internal structures (and which) or not, and whether we need additional data (e.g., to train the explanation method or to compute the explanation) or only the tested data. Note that NN refers to a neural network.

	Expl. Type	Model Agnostic	Data Agnostic
Gradient baseline [Simonyan 14]	Local	Gradients Access	✓
Input x Grad. [Shrikumar 16]	Local	Gradients Access	✓
LRP [Bach 15]	Local	Gradients + NN Access	✓
DeepLIFT [Shrikumar 17]	Local	Gradients + NN Access	✓
IG [Sundararajan 17]	Local	Gradients Access	✓
Enhanced IG [Jha 20]	Local/Global	Gradients Access	Database (VAE training)
XRAI [Kapishnikov 19]	Local	Gradients Access	✓
Blur IG [Xu 20a]	Local	Gradients Access	✓
AIG [Pan 21]	Local	Gradients Access	✓
Smooth Grad. [Smilkov 17]	Local	Gradients Access	✓
Deconvnet. [Zeiler 14]	Local	NN Access	✓
Guided Backprop. [Springenberg 15]	Local	Gradients + NN Access	✓
Visual Backprop. [Bojarski 18]	Local	NN Access	✓

Table 2.2: Specific comparison of saliency methods. This table summarizes the specificities of backpropagation methods. We give relative indications about the noisiness of the generated attribution map (e.g. +++ means very noisy); the computational cost of training the explanation method (if required e.g. ++ $\sim$ 5-10 times the classifier f training) and generating the map in inference (e.g. - - means a low computational cost $\sim$ the model's prediction generation i.e. $f(x)$, while + $\sim$ 10-100 times $f(x)$). We also indicate the baseline (or reference) image type for path-based approaches.

	Noisy	Computational Cost (Tr./Inf.)	Baseline type
Baseline [Simonyan 14]	+++	NA/- -	NA
Input x Grad. [Shrikumar 16]	++	NA/- -	NA
LRP [Bach 15]	+	NA/- -	NA
DeepLIFT [Shrikumar 17]	+	NA/- -	Synthetic (black img.)
IG [Sundararajan 17]	+	NA/-	Synthetic (black img.)
Enhanced IG [Jha 20]	+	++/-	In Distrib. (VAE based)
XRAI [Kapishnikov 19]	-	NA/+	Synthetic (black + white img.)
Blur IG [Xu 20a]	+	NA/-	Synthetic (blurred img.)
AIG [Pan 21]	+	NA/+	Adv. Example (diverse)
Smooth Grad. [Smilkov 17]	+	NA/-	NA
Deconvnet. [Zeiler 14]	+	NA/- -	NA
Guided Backprop. [Springenberg 15]	+	NA/- -	NA
Visual Backprop. [Bojarski 18]	+	NA/- -	NA

2.1.2 Class Activation Map Methods

Units of various layers of classification Convolutional Neural Networks (CNN) capture object localization information [Zhou 15]. However, these information are lost in the final fully connected layers when producing the classification output. Using global average pooling as the final layer of the neural network (instead of a fully connected layer) [Zhou 16b] mitigate this information loss and compute the class activation map for a given class c (e.g., the predicted class) at the last convolutional layer. They show that the model's score (for a given class) could be written as a sum of the importance of the activation over the spatial grid of the last convolutional layer (see Figure 2.3). Then, the explanation map is produced by upsampling the class activation map from the last convolutional layer to the input image size.

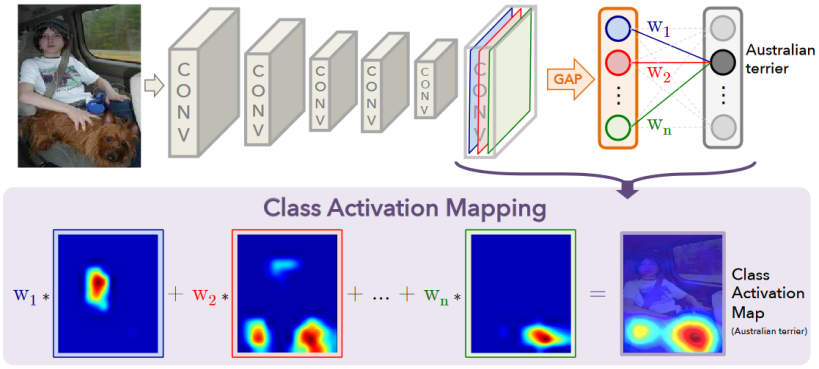


Figure 2.3: Overview of Class Activation Map [Zhou 16b]. For a neural network using a Global Average Pooling (GAP) operation to compute the model's outputs, the predicted class score is mapped back and combined with the last convolutional layer to produce class activation maps (CAM). Then, the low resolution map is upsampled to the input size to generate the attribution map.

To understand the importance of each neuron, GradCAM [Selvaraju 17], build on this work by computing the gradient of the model's output (for a given class) with respect to the last convolutional layer. They put forward that the last convolutional layer is the best compromise between spatial information and high-level semantics in any neural network. Their method generalizes the work of [Zhou 16b] (for any convolutional network) and generates compelling results. Figure 2.4 compares GradCAM attributions against backpropagation or naive perturbation techniques (see next section).

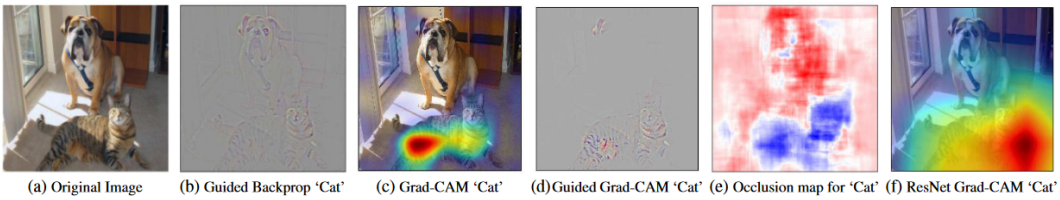


Figure 2.4: Illustration of GradCAM and Guided GradCAM against other attribution techniques [Selvaraju 17]. (a) the input image with a cat and a dog. (b-f) attributions relevant for "cat" classification. (b) Guided backprop attributions (for a VGG-16 classification model) highlighting the edges of both the cat and the dog. (c) GradCAM (VGG-16). (d) Guided GradCAM (VGG-16). (e) Occlusion map (VGG-16) produced by iteratively perturbing the input and measuring the impact on the model's output. (f) GradCAM for a ResNet-18 model.

Medical domain: [Rajpurkar 17] applies GradCAM on a chest X-rays disease classification problem (see attributions in Figure 2.5).

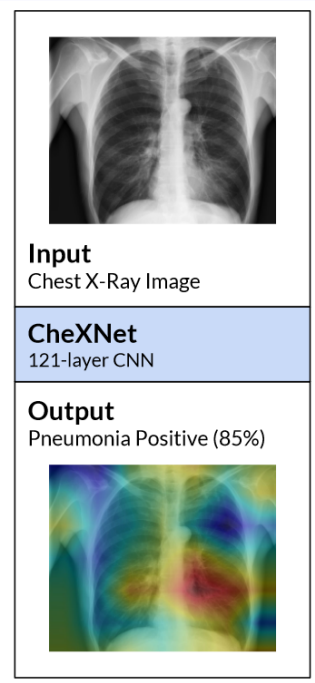


Figure 2.5: Illustration of GradCAM attributions applied on ChestXNet [Rajpurkar 17]

However, these methods often fail to localize multiple occurrences of the same class in an input image or do not capture the entire object. [Chattopadhyay 18] address these limitations by introducing a pixel-wise weighting of the gradients (of the model's output) w.r.t the last convolutional layer.
All these approaches provide reasonable visualizations in some settings, especially when localizing objects, yet they often produce coarse maps and do not show fine-grained relevant regions. Indeed, the visual explanation is generated by upsampling the class activation map from the last convolutional layer to the input size. To address this issue, the authors from [Selvaraju 17, Chattopadhyay 18] combine their class activation map technique with

guided backpropagation from [Springenberg 15].
CAM [Zhou 16b] is not model-agnostic (depending on particular architecture); the other class activation map techniques require access to the gradients and layers of the deep neural networks. Comparison between the different methods are given in Tables 2.3 and 2.4 sum up the main specificity of the different saliency techniques.

Table 2.3: Global comparison of CAM methods. For each method, we indicate if the visual explanation is local or global, whether the user has access to the model's internal structures or not, and whether we need additional data or only the tested data.

	Expl. Type	Model Agnostic	Data Agnostic
CAM [Zhou 16b]	Local	NN specific	✓
GradCAM [Selvaraju 17]	Local	Gradients + NN Access	✓
GradCAM++ [Chattopadhyay 18]	Local	Gradients + NN Access	✓
Guided GradCAM [Selvaraju 17]	Local	Gradients + NN Access	✓

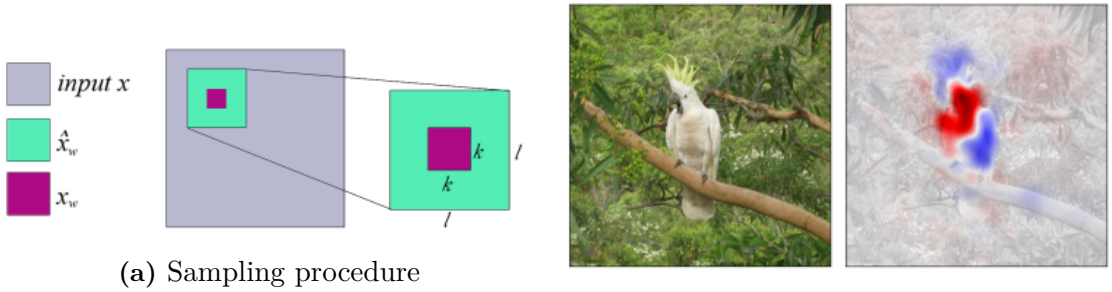
Table 2.4: Specific comparison of CAM methods. This table summarizes the specificities of class activation map methods. We report if the explanation highlights fine-grained details or coarse regions of the input. Relative indications about the computational cost of generating the map is shown (e.g. - - means a low computational cost $\sim f(x)$).

	Fine-grained/Coarse	Computational Cost (Inf.)
CAM [Zhou 16b]	Coarse ++	- -
GradCAM [Selvaraju 17]	Coarse ++	- -
GradCAM++ [Chattopadhyay 18]	Coarse +	- -
Guided GradCAM [Selvaraju 17]	Fine-grained	- -

2.1.3 Perturbation Methods

Perturbation-based methods study the impact on the model's output of perturbations applied on the input image [Zeiler 14]; they are based on prediction difference analysis (PDA). They typically compute an optimal mask M (often binary) that determines where a perturbation function Φ should act on the input image to change the classifiers' output. M then gives the visual explanation map. Various approaches have been proposed, we present the main contributions in the following.

Iterative Region Impact- The occlusion technique [Zeiler 14] successively removes patches of the input (i.e., replace input pixels with black pixels) and measures the change in the prediction to set the importance of this input region. [Zintgraf 17] also measure the class evidence difference using a patching technique (see Figures 2.6a and 2.6b for illustrations). However, each pixel value from the patch is sampled from a larger surrounding patch (as pixel strongly depends on their local neighborhood.



(a) Sampling procedure

(b) Example of attributions.

Figure 2.6: Explanations through prediction difference analysis [Zintgraf 17]. Illustration of the sampling procedure of the iterative perturbation approach. For each input image x, they sample every patch x_w and consider a larger patch $\hat{x}_w$ for perturbation conditioning of x_w. (b) Example of attribution maps showing evidence for (red) and against (blue) the model's decision.

These methods have a high computational cost as the perturbation operation has to be repeated at least for each pixel or patch to cover all the input. They are also impacted by the size chosen for the occlusion patch and the potential overlapping ratio. Different methods build upon these works, adopting different regions or perturbation types; and trying to reduce the computational cost.
In [Wei 18], the authors also consider a prediction difference analysis but rather use superpixel regions. The input is first segmented into salient regions consistent with the image's content. Then, for each superpixel, they compute an average prediction. They assign a random RGB value multiple times to the superpixel (sampled from the input histogram) and then apply the trained model. Figure 2.7 describes the overall approach.
[Seo 20] adopt multiple scales superpixel segmentation of the input from 2 to 2^r ($r = 5$) segments (see Figures 2.8). For each segment of the different scales, they measure the prediction difference, when replacing the superpixel values with a constant value sampled in a normal distribution that estimates the input values.

Medical domain: Figure 2.8b illustrates application of [Seo 20] to the classification of Alzheimer disease on brain MRI images.

More recently, [Shitole 21] divides the input image into patches at a lower scale. The

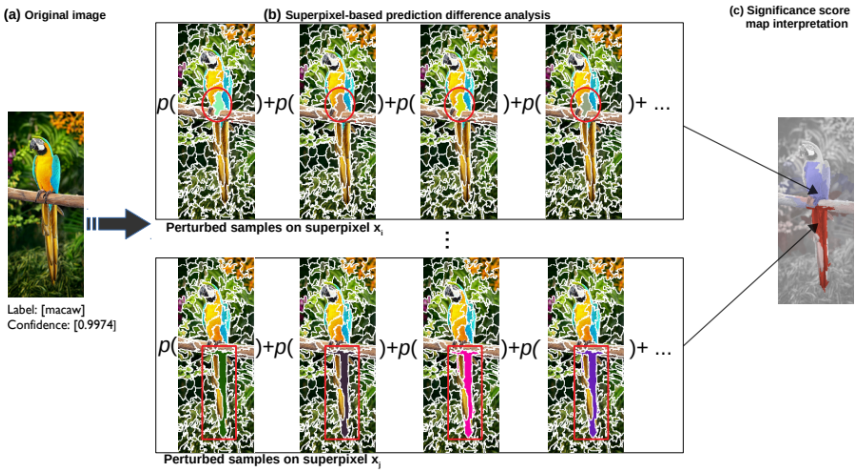
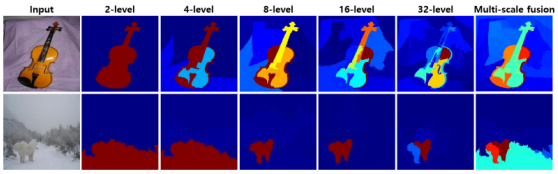
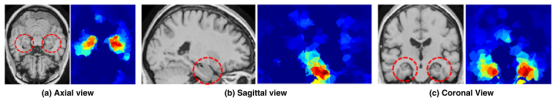


Figure 2.7: Explanation through superpixels perturbation approach [Wei 18]. (a) The input image and the corresponding classification prediction. (b) Prediction difference analysis is computed for each input superpixel (illustration for two of them in the figure). The relevance score of each segment is the average of different marginalization operations. (c) The resulting explanation map displaying supportive and unsupportive regions toward the model's decision.



(a) Multi-scales attributions



(b) Example of attributions on medical image problem.

Figure 2.8: Explanation through multi-scales perturbation approach [Seo 20]. (a) From left to right: Attribution maps from 2-level to 32-level superpixels segmentation, and the fusion of all scales (last column). (b) Expert annotations (red circles) compared with attribution maps in the classification Alzheimer disease on brain MRI images. Visualizations are shown for the three observation views (axial, sagittal, and coronal).

patches are then upsampled to the input size. Compared to previous works, the objective is to find all combinations of regions –composed of upsampled patches– that preserve the model's output. These regions are left intact while the rest of the input is set to black pixels; except at the border of the upsampled mask (taking values in $[0, 1]$) used to select the regions. Figure 2.9 shows the different steps of the explanation method.
The selected regions are studied, via Structured Attention Graphs (SAGs), by iteratively deleting the remaining sub-regions –the upsampled patches that compose the "preserving" region– and measuring the impact on the model's output.

Multiple Region Sampling- These methods sample multiple mask regions of the input, measure the impact of perturbing the input in these regions, then average all the contributions (from the different sampling). For instance, LIME [Ribeiro 16] perturbs random superpixels of the input image and propose a local explanation by training a linear classifier to predict the importance of each segment for the classifier's prediction. RISE [Petsiuk 18] samples multiple masks (M_i), then runs the model f on masked inputs

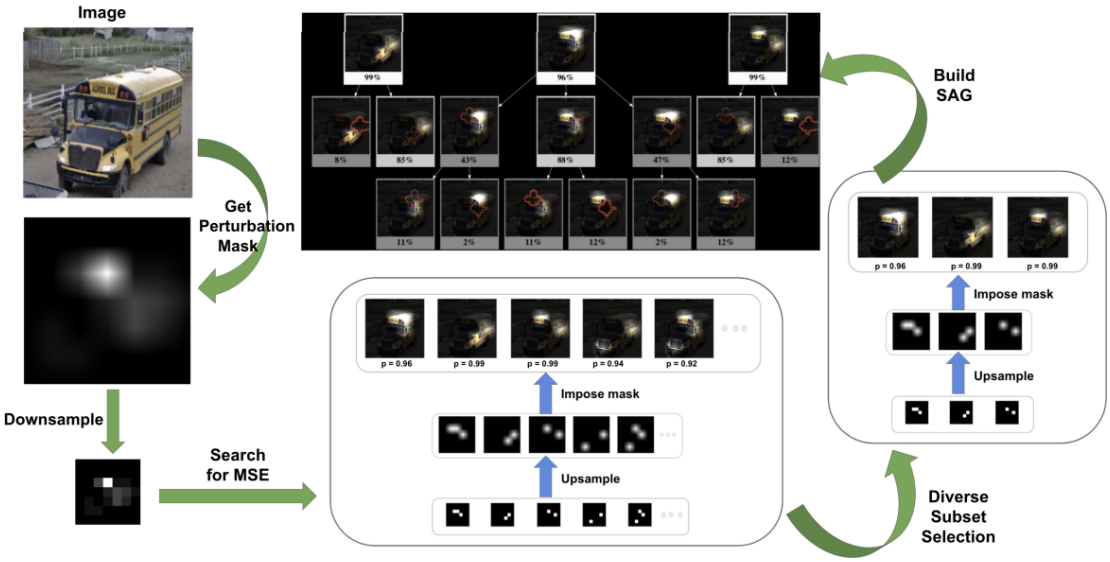


Figure 2.9: Structured Attention Graphs to explain a classifier's decision [Shitole 21]. Different steps for getting the structured attention graph: from searching for diverse minimal sufficient regions (at a low-resolution scale) to generating the structured graph by studying each "parent" attention (the "three" image on the top of the graph).

$x \odot M_i$ $(i = 1, ..., N)$.

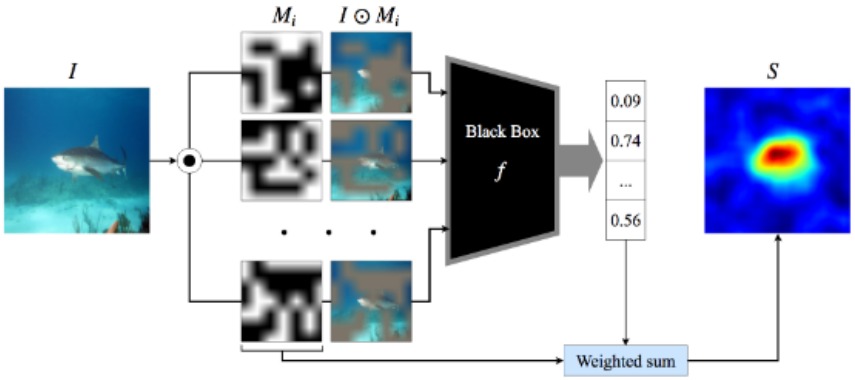


Figure 2.10: Overview of RISE [Petsiuk 18]. The input image I is multiplied with multiple sampled random masks M_i. The resulting images are given to the classification model that provides new scores. The explanation map is computed by taking the sum of the sampled masks weighted by the score of the target class.

The explanation map for the class c is obtained by computing the weighted average of the masks :

$$\mathcal{E}(x) = \frac{1}{\mathbb{E}[M].N} \sum_{i=1}^{N} f_c(x \odot M_i).M_i \tag{2.4}$$

where $\odot$ is the element-wise multiplication; M_i is generated by first sampling a binary mask at a lower resolution, then by upsampling it. f_c is again the prediction of model f towards class c. Figure 2.10 provides a schematic of the method. In the same spirit, MFPP [Yang 21b] computes the visual explanation as a weighted average of sampled masks, where

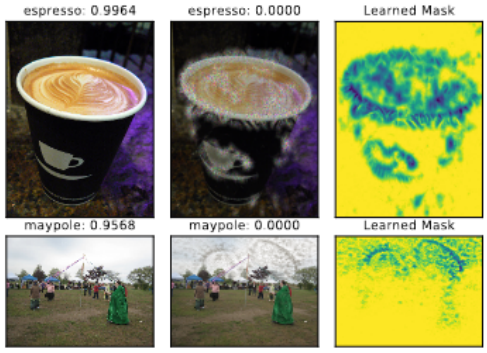
the weights are also the class prediction of the masked input. Compared with [Petsiuk 18], which ignores image structure when generating mask regions, this method uses a segmentation algorithm to produce superpixel masks at different scales. The final explanation map is obtained by averaging the contribution of all the masks at all scales.

Iterative Mask Optimization- Given an input image x, [Thakur 21] use empirical risk minimization to iteratively build and update a mask M. At each step, they perturbed masked and unmasked pixels based on the mask from the previous step and the prediction score of the corresponding masked input ($p = f_c(x \odot M)$), i.e., $n_1 p$ and $n_0(1-p)$ pixels are randomly selected from respectively the set of unmasked (Λ_1) and masked (Λ_0) pixels. The mask values are updated as follows:

$$M_i(\lambda) = \begin{cases} M_{i-1}(\lambda).p & \text{if } \lambda \in \Lambda_1 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where i is the iteration step and λ is the pixel position.
In contrast, for each image, [Fong 17] introduce an optimization setting to find the minimal perturbation region with the greatest impact on the classifier's output.

(a) BBMP learned mask



(b) Mask learned with artifact.

Figure 2.11: BBMP examples [Fong 17]. (a) From left to right: an input image where the classification model detects a "flute"; the perturbated image obtained by blurring the input on the learned mask region inducing a significant decrease (deletion objective) of the model's probability score toward the "flute" class; and the learned mask. (b) Examples of mask learned without the regularization techniques. The masks add high-resolution structures (or artifacts) typical of adversarial attacks.

They propose to solve either a "deletion" game, where the objective is the find the minimal mask that makes the prediction (for class c) to drop significantly when perturbing the input x inside the masks with a perturbation function Φ (e.g. black pixels, constant, noise, blur); or a "preservation" game, where the objective is rather to find the minimal region that should be preserve to retain the initial prediction. The two optimization problems read

$$M^* = \underset{M \in [0,1]^\Omega}{\operatorname{argmin}} \begin{cases} f_c(\Phi(x,M)) + \lambda_1 ||1-M||_1 + \lambda_2 \mathcal{R}(M), & \text{For "deletion" game} \\ -f_c(\Phi(x,M)) + \lambda_1 ||M||_1 + \lambda_2 \mathcal{R}(M), & \text{For "preservation" game} \end{cases} \tag{2.6}$$

where Φ produces a perturbed image usually defined by $\Phi(x,M) = M \odot x + (1-M) \odot p$ (p being a completely perturbed image that should contain very few information for f); $\mathcal{R}$ is a regularization term to smooth the mask and to avoid adversarial evidence (or artifacts).

Figure 2.11a shows the optimized mask and the associate blurred image produced by the method. Indeed, neural networks have been shown to be sensitive to adversarial examples [Goodfellow 15, Kurakin 17, Madry 18] which typically add a small perturbation to the input image to fool the classification model. As the optimization procedures of generating an adversarial example or an optimal mask share similarities, adversarial evidence is likely to appear in the latter if no action is taken [Fong 17]; the resulting explanation would not only rely on true evidence of the input (see Figure 2.11b). To avoid this issue, they rely on additional regularization: total variation (TV) or Gaussian filtering to smooth the mask; stochastic techniques such as adding random noise to the input of geometric transformations (e.g., translation); and optimize a mask at a lower resolution, upsampling it at each step (through bilinear interpolation). Total variation and upsampling operation are also used in [Thakur 21] for similar reasons.

Building on this work, [Fong 19] propose slightly different image-wise optimization problems, adopting an extremal perturbation strategy and enforcing the mask to match a predefined size (based on the problem). Figure 2.12 shows examples of masks learned from the method for different region sizes.

Figure 2.12: Explanation mask learned from extremal perturbation for different target areas [Fong 19].

[Qi 19] optimize the mask by computing the integrated gradient at each step (using a blurred image as reference input). Using similar regularization as [Fong 17], they show their method is more likely to achieve a global optimum and converges faster. [Du 18] combine deletion and preservation objectives and use a different regularization, assuming the mask could be decomposed as the combination of channels at a high-level layer of the trained CNN model. This idea comes from observations made in [Yosinski 14, Zhou 15]. The mask is also upsampled at the input size to compute the perturbated image. In [Wagner 19], these heuristic regularizations are replaced by a stronger control on gradients backpropagated during the mask optimization. Only neurons activated by the input image x can be activated in the explanation mask. ThiIts produces a more fine-grained explanation compared with [Fong 17, Fong 19, Qi 19, Du 18] (which generates coarse maps because of the regularizations used) but tends to be noisy.

Mask Generator Optimization- Rather than optimizing a mask for each input, a neural network is trained on a given database to generate a mask region for each image that solves the deletion or the preservation game (or both simultaneously). The training procedure captures patterns on a whole database; it enables regularization of the mask generation and reduces the risk of adversarial evidence. These methods are better suited for real-time situations as the explanation is computed as an inference of the generator model. [Dabkowski 17] use a pre-trained encoder – that should extract relevant features from inputs– and train a decoder model to produce a mask at a lower scale compared with the input size. The mask is then upsampled and should minimize a similar problem as in

[Fong 17]. The optimization problem thus reads:

$$g^* = \underset{g}{\text{argmin}} \ \mathbb{E}_x \left[\begin{array}{l} -\log(f_c(\Phi(x, M_g(x)))) + \lambda_1 f_c(\Phi(x, 1 - M_g(x)))^{\lambda_2} + \\ \lambda_3 TV(g(x)) + \lambda_4 ||g(x)||_1 \end{array} \right] \tag{2.7}$$

where $M_g(x) = \text{Up}(g(x))$: the upsampling of the generated mask $g(x)$. Φ is the perturbation function computed as in previous works. To prevent g from adapting to a particular perturbation, the perturbation is randomly sampled at each call of Φ (between Gaussian blur and a constant with additive Gaussian noise). A slightly different optimization is proposed by [Fu 19], where the generator only minimizes the classification term, i.e., they do not explicitly optimize regularization terms (the mask is still generated at a lower scale). They introduce a distribution controller module to guide the distribution of the relevance scores (of the generated mask) for each input. In their formulation, the trained classifier is used as the encoder part of the mask generator (and kept fixed during training).

Information Bottlenecks for Attribution (IBA)- The method proposed in [Schulz 20] adapts the information bottleneck concept from [Alemi 17] for the context of attribution. The idea of this information bottleneck is to restrict the flow of information (to the minimum) at a specific layer of the neural network – by inserting a bottleneck at this layer (see Figure 2.13)– while preserving the initial predicted class c.

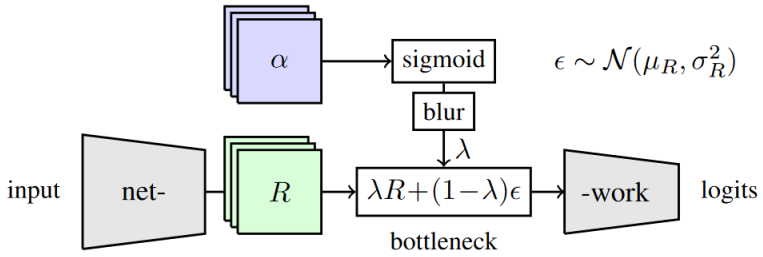


Figure 2.13: Bottleneck structure [Schulz 20]. At specific feature maps R, masks α control the flow of information passed to the end of the network. The masks λ used in the optimization are regularized versions of α (to avoid adversarial artifacts).

To do so, a random variable z is introduced such that $z = \lambda R(x) + (1 - \lambda)\epsilon$, where R is the feature map (at a specific layer l) of the input x; λ is a mask with same dimension as R ($\lambda_i \in [0, 1]$); and ϵ is a noise sampled in the normal distribution $\mathcal{N}(\mu_R, \sigma_R)$ which is estimated on the values of the feature map R. The mask λ is optimized such that the mutual information (I) between the variable z and the class c is maximal, and the mutual information between the input x and z is minimal:

$$\lambda^* = \underset{\lambda}{\text{argmax}} \ I(c, z) - \beta I(x, z) \tag{2.8}$$

The intractable problem (2.8) is approximated into :

$$\lambda^* = \underset{\lambda}{\text{argmin}} \ L_{CE}(c, f^{top}(z)) + \beta \, \mathbb{E}_R \left[D_{KL}(P(z|R) \,||\, \mathcal{N}(\mu_R, \sigma_R)) \right] \tag{2.9}$$

Where L_{CE} is the cross-entropy loss, f^{top} is the end of the model from the layer l to the output, and D_{KL} is the Kullback Leibler divergence. In this formulation, if a small region of the feature map R preserves the classification c, all other features are removed (through the term D_{KL}); the preserved region has sufficient information for the prediction ($\sim$ preservation game).

Medical domain: Using a deletion game, [Khakzar 21] propose inverse IBA to find all regions with predictive information (not only the sufficient one). In the two cases, the final explanation map is upsampled from the selected layer scale to the input size producing a coarse map. Figure 2.14 compares attributions produced by IBA and Inverse IBA for a pathological chest X-ray image.

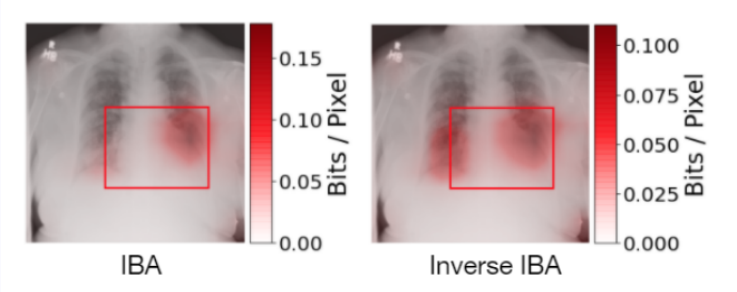


Figure 2.14: Comparing IBA against Inverse IBA on chest X-ray [**Khakzar 21**]. IBA (left) identifies a region of the input sufficient for the model to predict a Cardiomegaly pathology, while Inverse IBA detects all relevant regions.

More recently, Input IBA [Zhang 21] first computes IBA at a specific layer l of the trained model, then estimates the bottleneck variable on input z_I, such that z_I induces the same distribution of latent features (at the layer l) as the bottleneck z directly computed at the layer l. A generative model [Goodfellow 14] is used to approximate this correspondence. The technique produces fine-grained attribution compared to [Schulz 20, Khakzar 21].

Tables 2.5 and 2.6 show the main similarities and differences between all perturbation-based methods. While producing promising results and being model-agnostic (in general), perturbation methods based on iterative region impact or region sampling have a high computational cost (not suited for real-time situations). In contrast, optimization-based methods often impose strong and specific regularization directly on the explanation mask (to produce acceptable visual results and avoid adversarial artifacts). These regularizations are unrelated to the classifier to interpret and tend to produce coarse explanation maps. In addition, these methods use synthetic perturbations, e.g., black or constant pixels, Gaussian noise, or blur. This weakens and biases the visual explanation when the perturbated image deviates from the data distribution on which the model is supposed to work. In natural image classification (e.g., ImageNet [Deng 09]), the number of different classes combined with the large variability of objects to detect and image contents (e.g., background, part of objects present in the image, or location in the image) tend to reduce the impact of such type of perturbation in average.

Medical domain: In contrast, in the medical domain, images are produced through specific procedures that minimize variability to simplify the analysis. For a specific problem analysis, input images are similar (e.g., same background, similar body structures) and the problem to detect. In these cases, synthetic perturbations produce images completely outside the distribution and often have a non-negligible effect on the model's output, thus biasing the generated explanation.

Table 2.5: Global comparison of Perturbation methods. For each method, we indicate if the visual explanation is local or global; whether the user has access to the model internal structures (and which) or not; whether we need additional data (e.g., to train the explanation method or to compute the explanation) or only the tested data; and what type of perturbation techniques. "NN as Encoder" means that the studied neural network is used as the encoder part or the generative model. "Gen." is the abbreviation for Generator model.

	Expl. Type	Model Agnostic	Data Agnostic	Method
Occlusion [Zeiler 14]	Local	✓	✓	Iter. Region Impact
Pixel PDA [Zintgraf 17]	Local	✓	✓	Iter. Region Impact
Superpixel PDA [Wei 18]	Local	✓	✓	Iter. Region Impact
Multiscale PDA [Seo 20]	Local	✓	✓	Iter. Region Impact
SAG [Shitole 21]	Local	✓	✓	Iter. Region Impact
LIME [Ribeiro 16]	Local	✓	✓	Multiple Regions Sampling
RISE [Petsiuk 18]	Local	✓	✓	Multiple Regions Sampling
MFPP [Yang 21b]	Local	✓	✓	Multiple Regions Sampling
ERM Mask [Thakur 21]	Local	✓	✓	Empirical Risk Minimization
BBMP [Fong 17]	Local	✓	✓	Mask Optim.
Extr. Pert. [Fong 19]	Local	✓	✓	Mask Optim.
Fine-grained Mask [Wagner 19]	Local	NN access	✓	Mask Optim.
I-GOS [Qi 19]	Local	✓	✓	Mask Optim.
Mask Generator [Dabkowski 17]	Local	✓	Database (Gen. training)	Mask Generator Optim.
Dist.Guided Mask [Fu 19]	Local	NN as Encoder	Database (Gen. training)	Mask Generator Optim.
IBA, inverse IBA [Schulz 20, Khakzar 21]	Local/Global	NN access	✓	Feature Maps Mask Optim.
Fine-grained IBA [Zhang 21]	Local	NN access	✓	IBA + Input Bottleneck Estim.

Table 2.6: Specific comparison of Perturbation methods. This table summarizes the specificities of perturbation methods. We point out if the method requires heuristic regularizations (and what type). we give relative indications about the computational cost of training the explanation method (e.g. ++ $\sim$ 5-10 times the classifier f training) and generating the map in inference (e.g. - - means a low computational cost $\sim f(x)$, while + $\sim$ 10-100 times $f(x)$). We also indicate what types of regions are perturbated (e.g., patches, pixels, superpixels, masks) and what perturbations are applied.

	Heuristic Regularizations	Computational Cost (Tr./Inf.)	Region type	Perturbation type
Occlusion [Zeiler 14]	✗	NA/+++	Patch	Synthetic (Black)
Pixel PDA [Zintgraf 17]	✗	NA/+++	Pixel	Synthetic (Local region sampl.)
Superpixel PDA [Wei 18]	✗	NA/++	Superpixel	Synthetic (Multiple sampl. Input Histogram)
Multiscale PDA [Seo 20]	✗	NA/++	Multiscale Superpixel	Synthetic (sampl. Estim. Input values distrib.)
SAG [Shitole 21]	Upsampling	NA/++	Upsampled Patchs	Synthetic (Black)
LIME [Ribeiro 16]	✗	NA/++	Superpixels sampling	Synthetic (Black)
RISE [Petsiuk 18]	✗	NA/++	Masks sampling	Synthetic (Black)
MFPP [Yang 21b]	✗	NA/++	Multiscale Superpixels sampling	Synthetic (Black)
ERM Mask [Thakur 21]	✗	NA/++	Mask	Synthetic (Black)
BBMP [Fong 17]	Upsampling, Blur, TV, Geom. transf.	NA/+	Mask	Synthetic (Black, Noise, Blur)
Extr. Pert. [Fong 19]	Multiple area constraints	NA/+	Multiple Masks	Synthetic (Pyramid Blur)
Fine-grained Mask [Wagner 19]	✗	NA/+	Mask	Synthetic (Black)
I-GOS [Qi 19]	Upsampling, Blur, TV, Noise add.	NA/+	Mask	Synthetic (Blur)
Mask Generator [Dabkowski 17]	Upsampling, Blur, TV	++/- -	Mask	Mix Synthetic (Black, Noise, Blur)
Dist.Guided Mask [Fu 19]	Upsampling, Blur, TV	++/- -	Mask	Mix Synthetic (Blur)
IBA, inverse IBA [Schulz 20, Khakzar 21]	✗	NA/+	Patch	Synthetic (Noise)
Fine-grained IBA [Zhang 21]	✗	NA/++	Patch	Synthetic (Noise)

2.1.4 Adversarial Examples as Explanation

Rather than using a perturbation function and optimizing a mask, adversarial explanations propose to find for each input image a close adversarial example –to be compared to the input image in the sense of the L_p distance, where $p = 1, 2, \infty$– that impacts the classifier's decision within a constrained space. These approaches are based on adversarial examples generation techniques such as [Goodfellow 15, Madry 18] which aim to produce a minimal perturbation δ to add to the input image to fool the model. The corresponding optimization problem thus reads:

$$\delta^* = \underset{\delta \in \mathcal{A}}{\operatorname{argmax}} \, L_f(x + \delta, c) \tag{2.10}$$

where $\mathcal{A}$ is the space of allowed attacks, L_f is a classification term depending on the classification model f, and c is the predicted class of the input x (or the true class of the input). In common adversarial attacks, the perturbation is often searched as the minimal perturbation in the sense of a L_p norm. In this context, the perturbation emphasizes adversarial evidence that is not optimized to reveal relevant features from the input for the model f. [Woods 19] optimize the shape of the perturbation by following the gradient $\partial f_c / \partial (x + \delta)$ up to a boundary RMSE, where the perturbation reaches a minimum while allowing a certain amount of perceptual difference with the input. Perceptual perturbations proposed in [Elliott 19] enforce the adversarial example to capture relevant patterns (and reduce adversarial artefacts) using a perceptual loss as an additional regularization for adversarial perturbations. They try to find an adversarial image x_a minimizing

$$x_a^* = \underset{x_a}{\operatorname{argmin}} \begin{bmatrix} (f_c(x_a) - \max\limits_{j \neq c} f_j(x_a) - T)^2 & + \\ \lambda_1 ||x - x_a||_2^2 & + \\ \lambda_2 \sum_{l \in \mathcal{L}} ||F^l(x_a) - F^l(x_a)||_2^2 & \end{bmatrix} \tag{2.11}$$

where $F^l(x)$ is the feature map at a layer l of the trained model f for the input x; T is a positive value used as confidence target; $\mathcal{L}$ is the set of layers on which the perceptual regularization is computed. Figure 2.15 illustrates the impact of the choice of the regularizing layers on the resulting visual explanation. The first term encourages the adversarial image to be classified differently compared with the input predicted class c; the second term enforces the proximity between the input image and the adversarial image; while the last term is a perceptual loss and compels the features map of x_a to be close to those of x w.r.t. the model f. They apply a Gaussian blur (with parameters σ) to highlight the regions with the most important differences and compute their visual explanation.

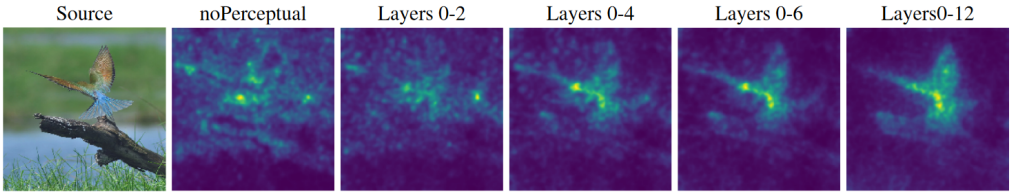


Figure 2.15: Effects of the perceptual regularization on adversarial explanations [Elliott 19]. From left to right: the input image; the difference map between the input and the generated adversary without the perceptual regularization; then the difference maps produced with perceptual regularization applied on different layers of the VGG-19 model.

In [Khakzar 19], the authors first prune unimportant neurons of the model, i.e., neurons that have a poor impact on the model's output are removed; then, they optimize an input perturbation that maximally changes the output of the pruned model. In this situation, only important neurons are affected by the adversarial attack. Figure 2.16 sums up this perturbation approach.

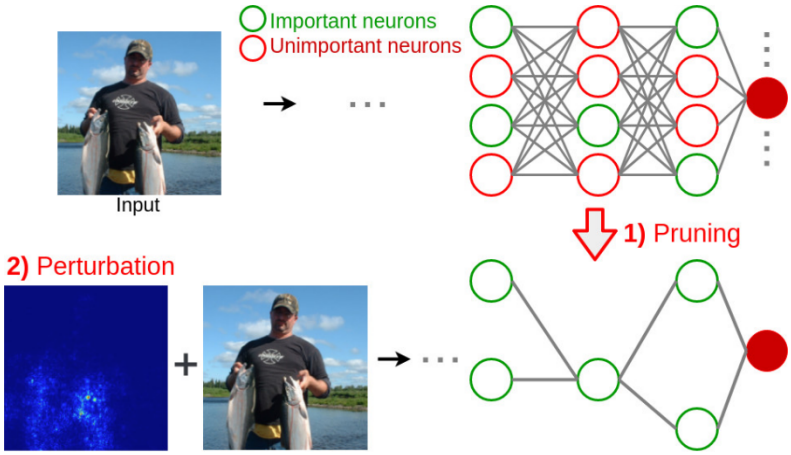


Figure 2.16: Perturbation of the important features via a pruning strategy [**Khakzar 19**]. Illustration of the attribution technique for a given input image: 1) Removal of unimportant neurons for the prediction from the classification model. 2) Generation of an adversarial perturbation that maximally changes the output of the pruned network.

Through an adversarial robustness analysis, [Hsieh 20] first propose a technique to evaluate the explanation attributions. They assume that (i) when the values of relevant features are fixed, perturbations restricted to the rest of the features have a poor impact on the model's output. (ii) On the opposite, even small perturbations applied to the relevant features (the rest left fixed) should easily impact the model's output. Then derive their visual explanations that maximize the evaluation criteria using an iterative algorithm to add an element to the targeted set of features (computationally expensive).

Compared with usual adversarial attacks, these approaches add some regularization and encourage the perturbations to apply only to essential features for the model. However, in their formulation, nothing constrains the adversarial generations to be consistent with real distributions. Then, they explicitly impose an pixel-wise distance constraint (L_p distances) between the adversary and the input image i.e. $||x - x_{adv}(x)||_p$, where $p \in \{1, 2\}$. Despite regularization, this over-constrains the adversarial generation and tends to produce adversarial artifacts. It also prevents the capture of distribution-specific patterns. We display in Tables 2.7 and 2.8 the main contributions and specificities of these adversarial explanation techniques.

Table 2.7: Global Comparison of Adversarial Generations methods. For each method, we indicate if the visual explanation is local or global; whether the user has access to the model internal structures; whether we need additional data or only the tested data (e.g., ✓means the method is data agnostic and does not required additional data); and what type of adversarial generation techniques is used.

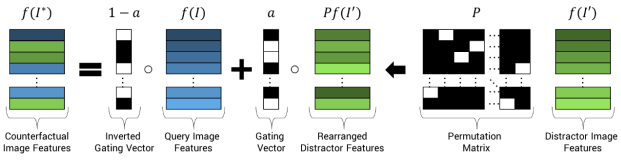
	Expl. Type	Model Agnostic	Data Agnostic	Method
RMSE Bounded Pert. [Woods 19]	Local	Gradient Access	✓	Adv. Attack w/ signif. differences
Perceptual Pert. [Elliott 19]	Local	NN Access	✓	Perceptual Perturbation
Pruning [Khakzar 19]	Local	NN Pruning	✓	Pruning Neurons
Robustness Criter.[Hsieh 20]	Local	✓	✓	Max./Min. Robustness criteria

Table 2.8: Specific Comparison of Adversarial Generations methods. This table summarizes the specificities of adversarial generation methods. We give relative indications about the noisiness of the generated attribution map (similar to backpropagation techniques in Table 2.2); the computational cost of generating the map in inference (e.g. - means a low computational cost $\sim$ 5-10 times $f(x)$, while + $\sim$ 10-100 times $f(x)$).

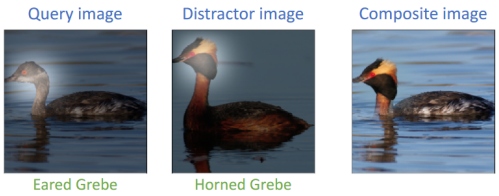
	Noisy	Computational Cost (Inf.)
RMSE Bounded Pert. [Woods 19]	++	+
Perceptual Pert. [Elliott 19]	+	+
PruningPGD [Khakzar 19]	+	+
PruningGrad [Khakzar 19]	+	-
Robustness Criter.[Hsieh 20]	+	+++

2.1.5 Counterfactual Visual Explanation

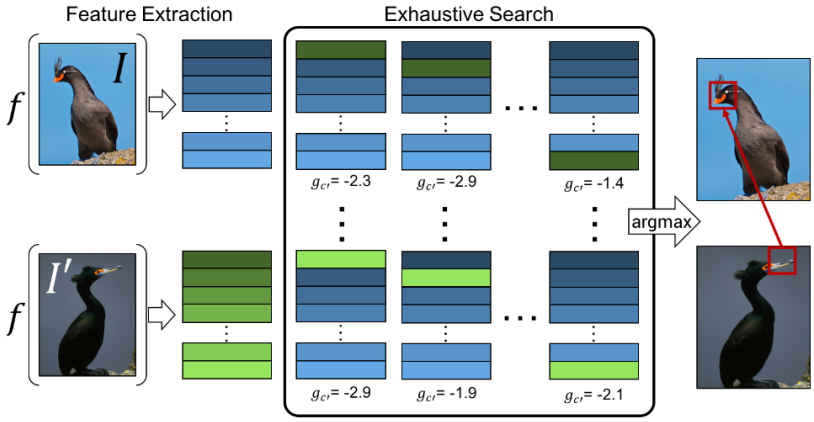
Compared with previous methods and in particular, perturbation-based or adversarial explanations, these techniques generate visual explanation by comparing the input with examples that belong to the distribution of "real" images (on which the model has been trained and is supposed to work). This explanation also answers why the input has been classified in the class c and not in another class c^{counter}. The literature tackling counterfactual examples and explanations is extensive [Verma 20, Guidotti 22]. Here we only present methods that produce visual explanations of image classification tasks.



(a) Query and Distractor images features combination



(b) Counterfactual explanation



(c) Exhaustive best-edit search.

Figure 2.17: Counterfactual visual explanation using query and distractor images [Goyal 19]. (a) Perturbation of the input (or query) feature maps $f(I)$ using the feature maps of a distractor image I', the binary mask a, and the permutation matrix P. (b) From left to right: the input image with the most discriminative region (against the Horned Grabe class) highlighted; the horned grebe distractor image (with the respective important region highlighted); and the composite image generated by combining the head of the horned grebe with the input image. (c) Check all pairs of a query(input)-distractor at different spatial locations; and selection of the pair that maximizes the model's probability score toward the distractor class. *Note: to avoid confusion in the main text between the classification model (f in the text) and the feature map at a specific layer, we replace $f(I)$ (in the figure) with $F(x)$.*

Generate counterfactual attribution maps- These methods produce visual explanations that emphasize input regions that are informative of a given class c (e.g., the input's prediction in practice) and uninformative of a counterfactual class c^{counter} chosen by the user. Using a collection images at test time and given a chosen counterfactual class c^{counter}, [Goyal 19] search for the minimal regions replacement from a counterfactual image x_{counter} to the input x (predicted in class c) to produce x_c, such that the model f predicts x_c in

class c^{counter}. To reduce the space of possible rearrangements between x_{counter} and x, they rather consider feature maps $F(x_{\text{counter}})$ and $F(x)$ (at a lower scale e.g. 16x16 instead of 256x256 in the input space); the model f applied on input x becomes $f(x) = F^{\text{top}}(F(x))$ (the output for a given class c thus reads $f_c(x) = F_c^{\text{top}}(F(x))$). Introducing a permutation matrix P –to rearrange and to align spatial regions from $F(x_{\text{counter}})$ to $F(x)$– and a binary mask a that indicates which spatial region of $F(x)$ to preserve or to replace with a region from $F(x_{\text{counter}})$. The resulting feature maps to be optimized is

$$F(x_c) = (1 - a)F(x) + aPF(x_{\text{counter}}) \tag{2.12}$$

Figure 2.17a illustrates the feature combination operation.
Then, they either propose an exhaustive search (see Figure 2.17c) over all permutations of P constraining a to be one-hot, or a relaxed version reparameterizing a and P using a softmax which allows optimization with gradient descent. The mask a is resampled to the input size (similarly as [Zhou 16b]) to produce the final explanation map (or to generate a bounding box). Although they capture relevant information for the classifier within the distribution of real images, they derive a region of interest heuristically. They still require a counterfactual image (with a completely different structure as shown in Figure 2.17) to which they can compare the input image. Building on this work, [Wang 20] generalize the generation of counterfactual visual explanation without using a collection of images at test time. Similarly, they use top layers feature maps (with a low-resolution scale) but they only study features that are informative for the input class c and uninformative for a chosen counterfactual class c^{counter}. Their proposed method is similar to GradCAM [Selvaraju 17] as they backpropagate gradients from the model's output to the specified feature maps. The attribution map at the feature map scale for x that discriminate the predicted class c against counterfactual class c^{counter} reads :

$$\mathcal{E}_F(x, c, c^{\text{couter}}) = a_F(f_c(x)).\bar{a_F}(f_{c^{\text{counter}}}(x)).a_F(s(x)) \tag{2.13}$$

where a_F is the dot-product between partial derivatives of the model's output (for a specific class c or c^{counter}) w.r.t. the feature map $F(x)$ i.e. $\partial f_c(x)/\partial F(x)$, and the feature map activation; $\bar{a_F}$ is the complement of a_F; and s attributes a confidence score ($s(x) \in [0, 1]$). The visual explanation at the input size are computed through a segmentation technique using a threshold T. Figure 2.18 describes the method.

Mask optimization with counterfactual perturbation- Built upon [Fong 17], [Dabkowski 17] or [Ribeiro 16], these methods [Chang 19, Agarwal 20] aim to optimize similar problems but propose to generate realistic perturbations with generative models attached to the input domain, e.g., perturbing by healthy tissue an image classified as pathological. [Chang 19] iteratively solve a similar problem as [Fong 17] by in-filling the perturbation region with background context of the image using a contextual attention GAN [Yu 18a]. Figures 2.19 show visual explanation of the such method and illustrate the impact of out-of-distribution perturbations (on the classifier's output).

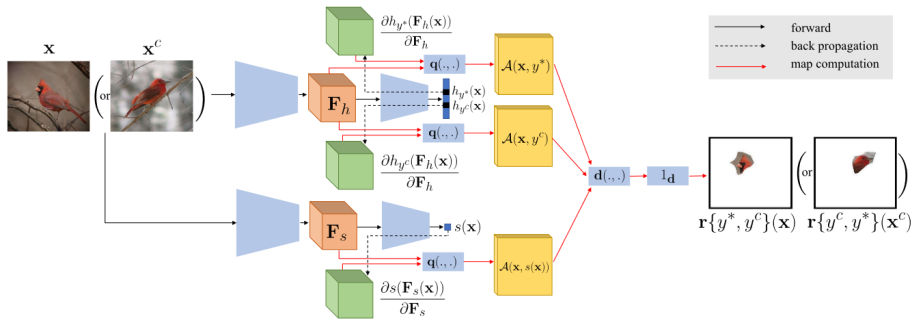
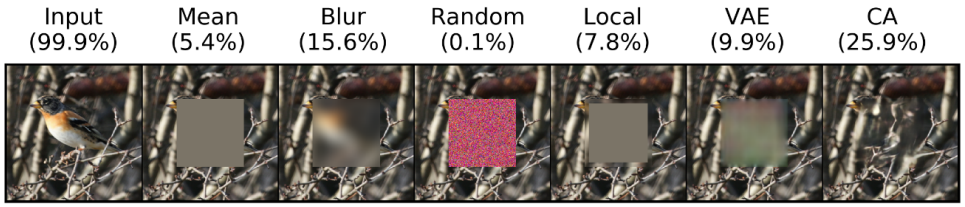
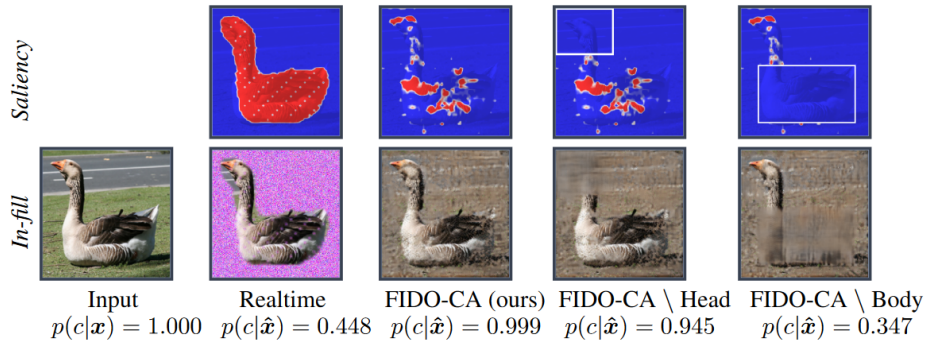


Figure 2.18: SCOUT: Discriminant explanation architecture [Wang 20]. Features maps are generated for both the classifier (F_h) and the confidence predictor (F_s) for the cardinal (c) and the summer tanager ($c^{counter}$)classes. Then, attributions $\mathcal{A}$ are computed to produce $a_F(f_c(x))$, $\bar{a_F}(f_{c^{\text{counter}}}(x))$ and $a_F(s(x))$ respectively. Finally, the counterfactual explanations produced using the thresholding technique (1_d).



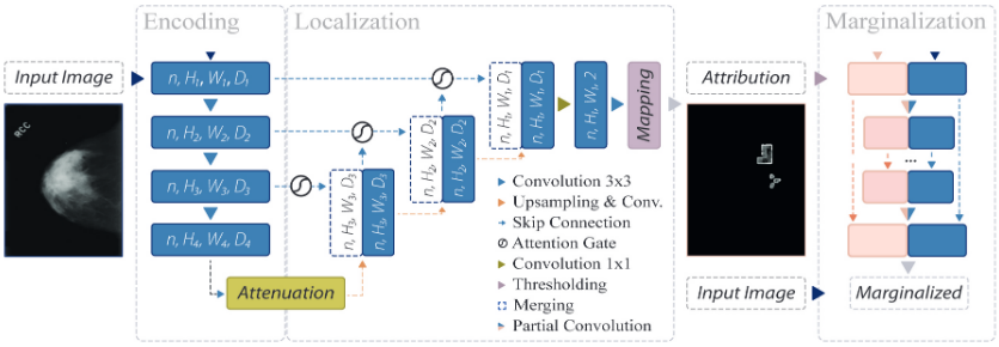
(a) Comparison of in-filling methods



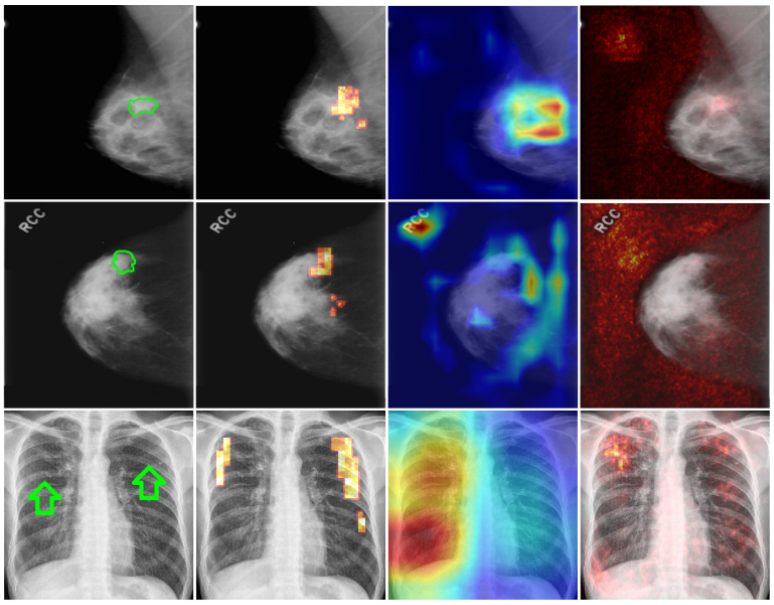
(b) Classifier confidence of perturbated input

Figure 2.19: Comparing the impact of the perturbation techniques [Chang 19]. (a) The model's output score (of the correct class) when perturbing the input with different perturbation in a centered mask. Here, CA is produced by the contextual attention GAN. (b) Top: the attribution (saliency) maps generated by the Realtime technique ([Dabkowski 17]), the method of [Chang 19] (FIDO-CA), and the method of [Chang 19] when part of the input is removed (head or body). Bottom: the corresponding perturbated inputs with the associated model's output score. [Dabkowski 17] coarsely finds the central object and decreases the classification score (while trying to preserve it), because of out-of-distribution perturbations. In contrast, FIDO-CA shows a minimal pixel region that preserves the classification score.

Medical domain: In [Uzunova 19], a variational autoencoder (VAE) [Kingma 14] is trained on a healthy image. It is then used to perturb the region when the input is predicted as pathological. Similarly, using an inpainting technique leveraging partial convolutions [Liu 18b], [Major 20] and [Lenis 20] respectively adapt [Fong 17] and [Dabkowski 17] to mammography and chest diseases classifications (see Figures 2.20). They train the inpainter to reconstruct healthy tissues (on images with holes).



(a) Architecture overview



(b) Comparing attributions between different techniques

Figure 2.20: Mask generator optimization applied to medical domain [Lenis 20]. (a) The architecture of the mask generator coupled with the marginalization module. The encoder part uses the classifier features (kept fixed). (b) Attribution maps for mammography and chest X-ray pathology detection. From left to right: the input image with expert annotations showing the pathologies; then attribution maps produced by their method; GradCAM [Selvaraju 17]; and Gradient [Simonyan 14].

In binary classification, through domain transposition technique (see Section 2.2.1), [Samangouei 18] train a generative model to produce for all inputs a reconstructed and a transposed image as well as a binary mask region. The binary classifier should assign a different class to the transposed image. Our work on counterfactual generation (see Section 5) considers similar ideas. However, in their framework, using DCGAN architecture [Radford 16], input images are encoded in vectors that pass through a generator model to produce images. Compared to our proposition, the generated images are generally of poor quality with lots of reconstruction errors compared with the input. To compute the final counterfactual image, they need to use a binary mask to combine the input and the transposed image (to reduce residual errors due to their generation process). However, for all these approaches, strong and heuristic regularizations on the perturbation regions (i.e. the mask) are still needed to produce smooth and acceptable explanation maps.

Iterative Counterfactual Generation- For each input image, these methods iteratively generate a counterfactual image assigned with a different class (by the classifier), and that should belong to the distribution of images of this different class. In [Dhurandhar 18], two perturbations δ are optimized to respectively highlight what minimal regions of the input are sufficient to yield the same classification (i.e., Pertinent Positives) or what minimal regions should not be added to the input to prevent the classification from changing (i.e. Pertinent Negatives). To generate realistic perturbations in the manifold of real images, they first trained a convolutional autoencoder [Mousavi 17], then used it to constrain the perturbation during the optimization. Another line of work explains the decisions of a trained classification model f by directly generating examples along the data distribution. Using a pre-trained encoder model e mapping input images x to a latent representation z; coupled with a pre-trained generative model g (GANs) (computing the inverse mapping) representation, xGEMs [Joshi 18] optimize the latent vector z to produce counterfactual images that belong to the approximated distribution of real images (through the trained generative modeling). They aim to find z that minimizes:

$$z^* = \underset{z}{\operatorname{argmin}}\ L_d(x, g(z)) + \lambda L_f(f(g(z), t) \tag{2.14}$$

where L_d enforces the proximity between the input x and the generated image $g(z)$, and L_f enforces the generated image to be classified in a given target class t by the model f. Then, the counterfactual example is $x_c = g(z^*)$.
In the same spirit, [Yang 21a, Liu 19c] search for optimal latent vectors composed of raw features z and available attributes features a (e.g., hair color, sex, mustache for facial attributes). As described in Figure 2.21, manipulating attribute features requires specific generative models.

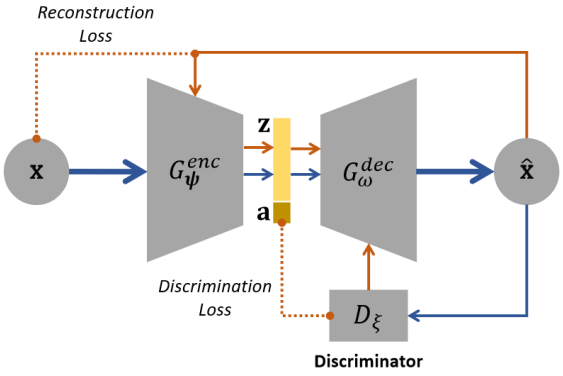


Figure 2.21: Optimization of an attribute-informed latent space in a generative model [Yang 21a]. At each step of the optimization, a batch of inputs x are encoded into a latent space divided into raw features z and attribute features a. Then, the decoder part learns to reconstruct the input image while learning the attributes a. To do so, an additional module D_{ξ} learns to classify each attribute. Blue arrows (resp. red) show the forward (resp. backward) pass, while dashed lines indicate the loss functions.

Medical domain: [Cohen 21] rather follow the trajectory of the partial derivatives of the model f w.r.t the latent vector z i.e. $\partial f(g(z))/\partial z$. Figures 2.22 describes the principles of the method.

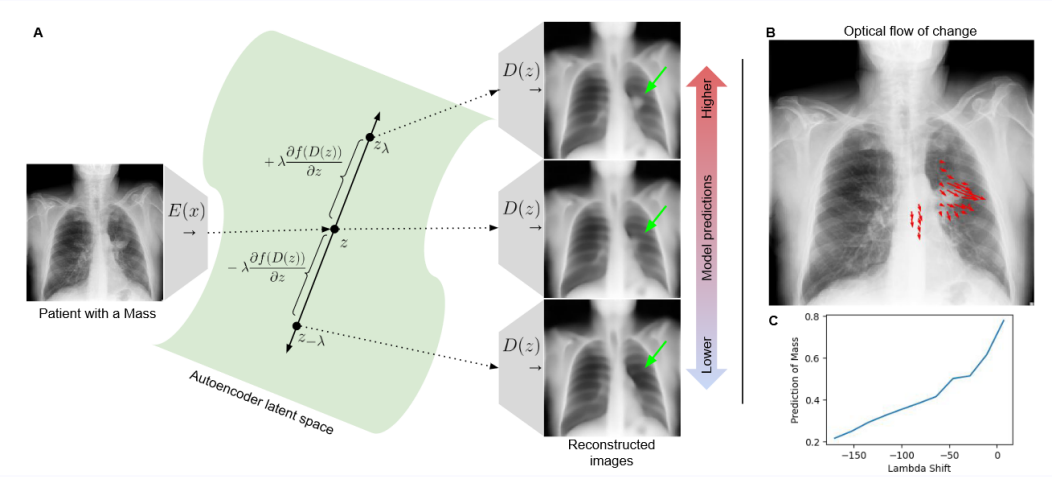


Figure 2.22: Latent shift method [Cohen 21]. **(A)** The input image is projected into a vector z in the latent space via an encoder model. Multiple input versions are generated with a decoder by shifting the latent vector z along the derivatives $\partial f(D(z))/\partial z$ in the latent space. **(B)** Optical flow translating the change in the input image given the generated ones. **(C)** The prediction of the generated image toward the "Mass" class according to the λ shift.

More recently, DiVE [López 21] optimizes multiple sparse perturbations, in the latent space of a trained generative model, to produce diverse counterfactuals. To encourage the diversity of the generations, they use a β-TCVAE [Chen 18], which is known to produce disentangled latent representation to produce diverse counterfactuals. In addition, they

enforce the different counterfactuals to rely on non-trivial attributes w.r.t. the class to explain. These methods produce insights on how to modify the input to change the classification, either to produce a prototype of the predicted class (an increase of the confidence score of the model) or to generate counterfactual examples. However, they are not suited to compute an attribution map. Indeed, by manipulating vectors in the latent space (with loss of spatial information and fine-grained details), the generated examples often differ from the input with important reconstruction errors (e.g., blurry images, different backgrounds or textures, lack of details). In addition, methods optimized on known attributes allow studying the impact on the classifier of each attributes separately. However, these attributes constitute additional annotations to perform, which is not always an option, particularly in medical image analysis.

Table 2.9: Global Comparison of Counterfactual methods. For each method, we indicate if the visual explanation is local or global; whether the user has access to the model internal structures (and which) or not; whether we need additional data (e.g., to train the explanation method or to compute the explanation) or only the tested data (i.e., ✓); and what type of counterfactual techniques. "NN as Encoder" means that the studied neural network is used as the encoder part or the generative model. "Gen." is the abbreviation for Generator model; "CAE" for convolutional autoencoder.

	Expl. Type	Model Agnostic	Data Agnostic	Method
Counterfactual Ex. [Goyal 19]	Local	Access NN	Data collection (test phase)	Replacement of relevant region from Counter. Img
Counterfactual GradCAM [Wang 20]	Local	Access NN & Gradient	✓	∼ GradCAM x Counter. GradCAM
CA-FIDO [Chang 19]	Local	✓	Database (CA-GAN training [Yu 18a])	Mask Optim.
BBMP Healthy-VAE [Uzunova 19]	Local	✓	Database healthy images (VAE training)	Mask Optim.
BBMP Healthy-Inpainter [Major 20]	Local	✓	Database healthy images (Inpainter training)	Mask Optim.
Mask Generator Healthy-Inpainter [Lenis 20]	Local	NN as Encoder	Database healthy images (Inpainter training)	Mask Generator Optim.
ExplainGAN [Samangouei 18]	Local/Global	✓	Database (Gen. training)	Mask Optim
Pertinent +/- [Dhurandhar 18]	Local	✓	Database (CAE training)	Additive Perturbation Optim.
xGEMs [Joshi 18]	Local/Global	✓	Database (Gen. training)	Iterative Counterfactual Ex. Optim.
Attribute-Informed [Yang 21a, Liu 19c]	Local/Global	✓	Database (Gen. training) + attributes annotations	Iterative Counterfactual Ex. Optim.
Gifsplanation [Cohen 21]	Local/Global	✓	Database (Gen. training)	Iterative Counterfactual Ex. Optim.

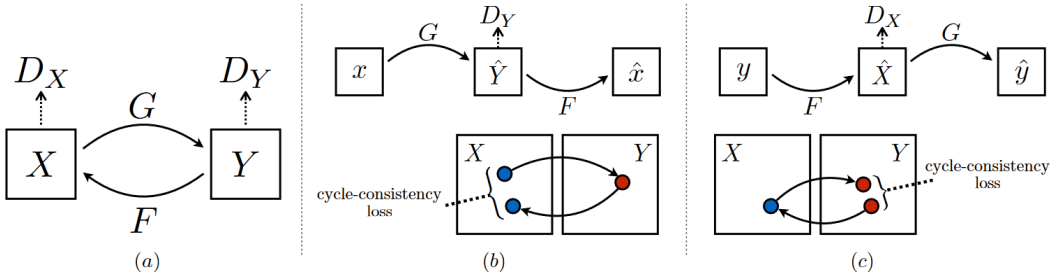
Table 2.10: Specific Comparison of Counterfactual methods. This table summarizes the specificities of counterfactual methods. We point out if the method requires heuristic regularizations (and what type). we give relative indications about the computational cost of training the explanation method (e.g. ++ $\sim$ 5-10 times the classifier f training) and generating the map in inference (e.g. - - means a low computational cost $\sim f(x)$, while + $\sim$ 10-100 times $f(x)$). We indicate the output of the explanation method / model, and what types of perturbations are applied. We also put forward if the method returns an attribution map and/or a counterfactual example.

	Heuristic Reg.	Computational Cost (Tr./Inf.)	Output type	Perturbation type	Expl. Map / Counterfactual Gen.
Counterfactual Ex. [Goyal 19]	NA	NA/++	Feature Maps Mask	Counterfactual Img in collection	✓/ ✗
Counterfactual GradCAM [Wang 20]	Segmentation w/ threshold	NA/-	CAM	✗	✓/ ✗
CA-FIDO [Chang 19]	Upsampling, TV	++/+	Mask	Background inpainting	✓/ +/- (Object ⟶ background)
BBMP Healthy-VAE [Uzunova 19]	Upsampling, Blur, TV, Geom transf.	++/+	Mask	Healthy tissue inpainting	✓/ ✓
BBMP Healthy-Inpainter [Major 20]	TV, thresholding, size constraints	++/+	Mask	Healthy tissue inpainting	✓/ ✓
Mask Generator Healthy-Inpainter [Lenis 20]	TV, thresholding, size constraints	++/- -	Mask	Healthy tissue inpainting	✓/ ✓
ExplainGAN [Samangouei 18]	TV, size constr., binary constr.	++/- -	Mask, Counterf. Gen	Counterfactual Generation	✓/ ✓
Pertinent +/- [Dhurandhar 18]	✗	++/+	Additive Mask	On Manifold	✓/ +/- (Controlled by CAE)
xGEMs [Joshi 18]	✗	++/+	Counterfactual Gen.	Progressive Counterf. Gen	✗/ ✓
Attribute-Informed [Yang 21a, Liu 19c]	✗	++/+	Counterfactual Gen.	Progressive Counterf. Gen	✗/ ✓
Gifsplanation [Cohen 21]	✗	++/+	Counterfactual Gen.	Progressive Counterf. Gen	✗/ ✓

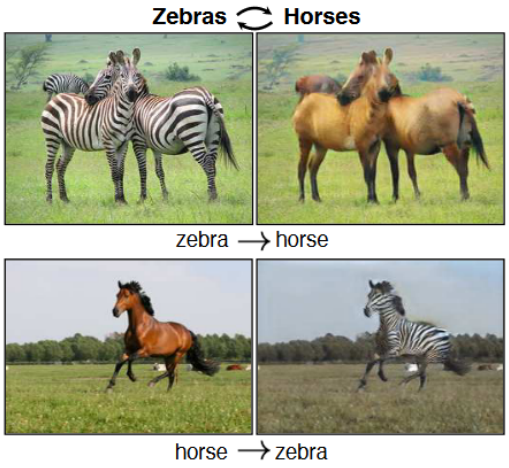
2.2 Visual Explanation via domain translation

2.2.1 Domain translation with GANs

Domain translation –also referred to as image-to-image translation– is a common and successful application of Generative Adversarial Networks (GANs) [Goodfellow 14]. It consists in learning a mapping between two different image domains [Isola 17].



(a) Description of the CycleGAN mapping functions

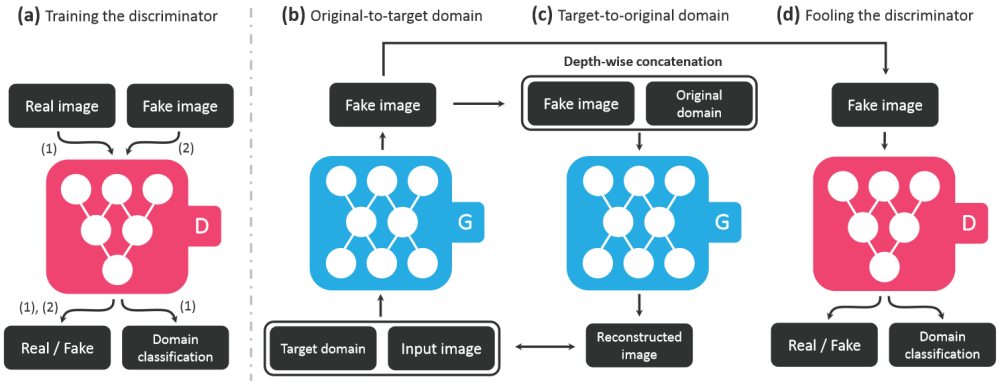


(b) Illustration of image-to-image translation

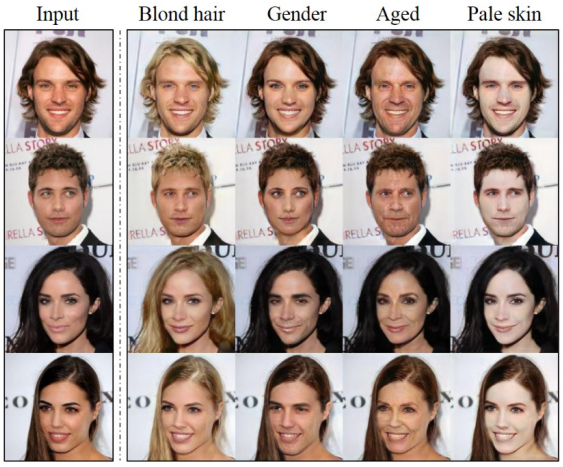
Figure 2.23: Overview of the CycleGAN method [Zhu 17]. (a) From left to right: the overall mappings between domain X and Y; where the generator G translates images from domain X to Y and tries to fool the domain-specific discriminator D_Y (resp. for F and D_X for the opposite translation); The cycle consistency illustrated for inputs in X translated to Y (via G) and mapped back to X (via F); and the opposite cycle for inputs in Y. (b) Example of translations between domains X: zebras and Y: horses.

Entangled domain translation- With paired images being available, Pix2Pix [Isola 17] learns to transform input images from a source domain into the paired images in the target domain. Compared to previous GANs architectures, which generate images from the latent variable with low dimension (e.g., Normal distribution with dimensions between 10 and 1000); domain translation frameworks use autoencoder-like, or UNet-like [Ronneberger 15] generative model to produce an image given an input image. To avoid the constraint of paired data (which are often not available), CycleGAN [Zhu 17] introduces a cycle-consistency constraint that enforces certain proximity across domains (two domains

in their case), encouraging translation and back-translation consistency. To learn a one-to-one mapping between two domains (e.g. Photos ↔ paintings, seasons transfers, horse ↔ zebra, healthy images ↔ pathological images, medical images types transfer), they consider two pairs of generator and discriminator (g_1, D_1) and (g_2, D_2). Given an input x_1 from domain 1 (resp. x_2 from domain 2), g_1 transpose x_1 in domain 2 (resp. g_2 transpose x_2 in domain 1) trying to fool the domain-specific discriminator D_2 (resp. D_1).
To enforce consistency when applying the reverse transformation –via g_2 (resp. g_1)– in order to return to the input domain 1 (resp. 2), they minimize $||x_1 - g_2(g_1(x_1))||_1$: the cycle consistency term. Figure 2.23a shows the different mappings while Figure 2.23b gives image translation examples between horses and zebras.



(a) Description of the StarGAN training



(b) Illustration of image-to-image translation

Figure 2.24: Overview of the StarGAN method [Choi 18]. (a) From left to right: the discriminator model D learns to identify real from generated (fake) images and to classify the domain from the given real input; Given an input image and a target domain, the generator G translates the input image into the target domain. Then, given this generated image and the original domain, the generator G maps the generated image back to the original one (reconstructed image). The generator G learns to fool the discriminator D. (b) Example of image translation for random input to the different target domain (e.g., blond hair, change of gender, aging, pale skin).

Similarly, UNIT [Liu 17] designs a shared latent space for the two domains and uses

two generators (with VAE architecture); it imposes a cycle consistency directly in the latent space. To increase the one-to-one mapping constraint, [Shen 20b] build upon [Zhu 17] but use a single generator i.e. enforcing a self-inverse function. Recent works [Lample 17, Choi 18, Yu 18c, Xiao 18, Yu 18b, He 19, Liu 19b] extend this one-to-one mapping to multi-domain translation (e.g., photos to paint from several artists, multiple season transfer) or to facial attributes editing (e.g., adding glasses, mustache, changing the hairstyle, hair color, age or sex). In these approaches, a unique couple of generator and discriminator allows the production of different mapping. Compared with CycleGAN [Zhu 17], the generator is conditioned with the target domain or the target attributes at image-level [Choi 18] or in the latent space [Xiao 18, He 19]; and an additional module is trained to classify the domain or attributes e.g. a classification model [He 19, Liu 19b] or an auxiliary classification head introduced in the discriminator model [Choi 18] (see Figures 2.24).

Another line of work [Yu 18b, Romero 19, Wang 19] is the multi-modal translation, which aims to produce diverse translations in the target domain given an input image, i.e., one-to-many mappings. The domain translation is enforced by using conditional normalization [Yu 18b, Romero 19] or by providing the domain class (or attribute classes) at the image level as an additional channel [Wang 19]. To encourage generation diversity, they add latent random code (sampled from a normal distribution as in VAE). The training procedure is presented in Figure 2.25.

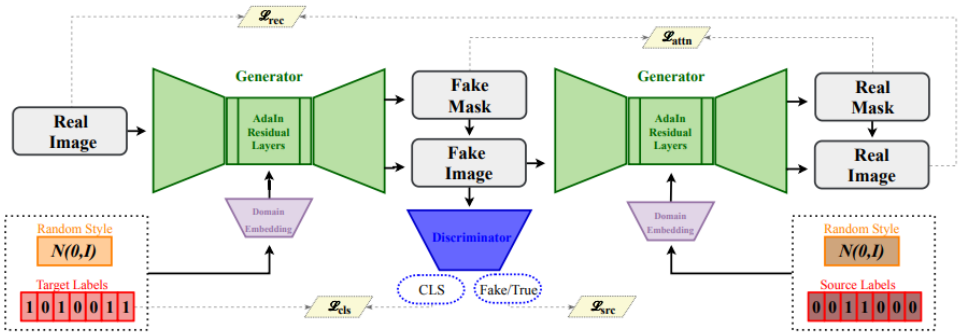


Figure 2.25: Overview of SMIT framework [Romero 19]. The input image is translated into another domain via a generator (green) that takes as additional inputs (at the latent space level) a random style (orange) and target attributes (red). Adaptive Instance Normalization modules condition the generation with the sampled style. The generator also outputs a mask to focus the model's attention and preserve background details. The discriminator (blue) learns to distinguish real from generated images and identify the real inputs' target attributes while the generator tries to fool it. Using another random style and the input target code, the generated image is mapped back to the input image (cycle consistency).

To enforce the generator to use the latent code, they either train an additional encoder [Yu 18b] or an auxiliary discriminator head [Wang 19], to compel the sampled code and the encoding of the transposed generation to match. A KL divergence also encourages the latent code to match the Gaussian prior (as in a VAE framework [Kingma 14]).

Disentangled domain translation- Compared with previous works, these techniques aim to separate image content (e.g., coarse image features, pose of a face) from image

style (e.g., fine-grained image features, colors). An exhaustive survey on disentanglement methods is available in [Liu 22b]; we only introduce approaches for domain translation. For each domain i (in general 2), MUNIT [Huang 18] and DRIT [Lee 18] train two encoders e_c^i and e_s^i to extract the content c_i and the style s_i of the input image x respectively; then they switch content and style of images from different domain to perform the translation (via a generator g_i) e.g. $x_{1\to 2} = g_2(c_1, s_2)$. They also enforce multi-modal generation by sampling style code in a Gaussian prior ($s_i \sim \mathcal{N}(0,1)$) and minimizing both a KL divergence and the distance $||s_i - e_s^i(g_i(c_{1-i}, s_i))||_p$ ($p \in \{1,2\}$). At test time, given an input x from domain $1-i$, they generate diverse translation in domain i either by sampling multiple $s_i \sim \mathcal{N}(0,1)$ or extracting multiple style $e_s^i(x_i)$ from image collection in domain i. Figure 2.26 describes the optimization of DRIT and how to produce image translation at test time.

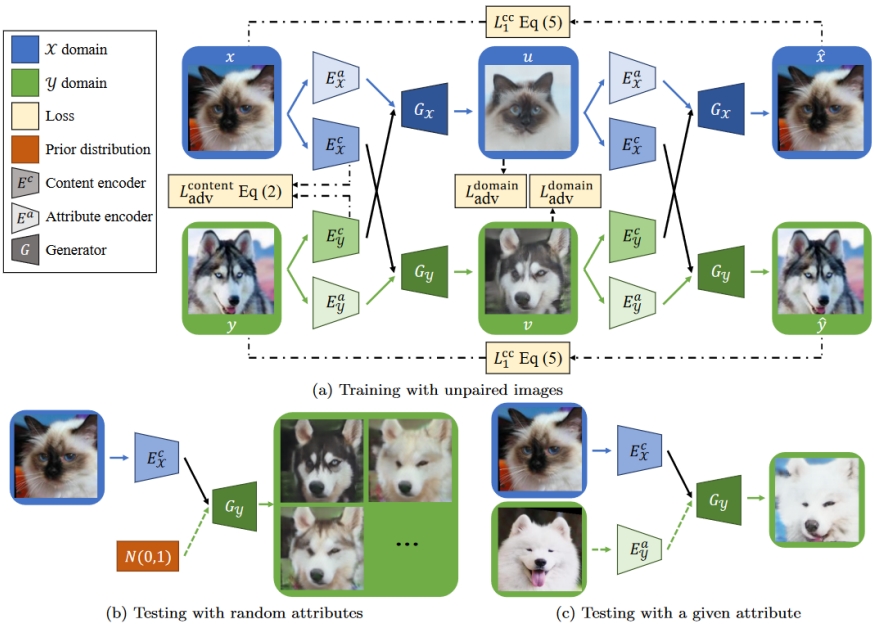


Figure 2.26: Overview of DRIT framework [Lee 18]. Top: Training procedure. At each step, content (E_i^c) and attributes (E_i^a) encoders extract latent information from inputs from coming from domain $i \in \{x, y\}$. They aim to remove attribute information from the content encoding via an adversarial content loss ($L_{adv}^{content}$). The image translation is obtained by switching attributes and content between the two domains. Domain-specific discriminator enforces the generations to belong to the target domain distribution (L_{adv}^{domain}). A cycle consistency constraint enforces the domain-specific generator to map the translated image back to the inputs. Bottom: At test time, image translations are produced by sampling random attributes in a normal distribution or using the attributes from a real image (of the target domain).

[Yu 19, Choi 20] generalize their approach for multi-domain mappings by adding a domain encoder. For each input image, DMIT [Yu 19] extracts the content, the style, and the domain, then switches them to perform multi-mappings. StarGAN v2 [Choi 20] rather introduce multiple heads to the style encoder and the discriminator: one for each domain. In addition, they use a multi-head style mapper (the same idea as [Karras 19]) to enforce generation diversity and better disentangle the style space (compared with a simple Gaussian prior).

Table 2.11: Global comparison of Domain Translation methods. We indicate if the technique works on unpaired data; if it can translate images to multiple domains; if it can change (or translate) multiple attribute features simultaneously; and if it can generate diverse generations for a single input (and a unique target domain). We also highlight if the method uses a disentanglement strategy (and what type) and what additional code is injected into the generative model to guide the image translation. "Cont." is the abbreviation for content; "Sty." for style; "Dom." for domain; and "Div." for diversity.

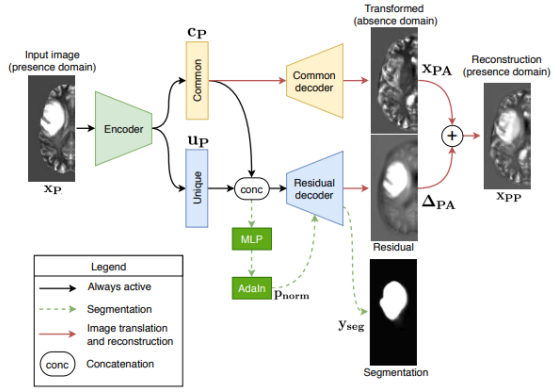
	Unpaired data	Multi-domain	Multi-attributes (Simult.)	Diversity	Disentangled	Gen. Injection
Pix2Pix [Isola 17]	-	-	-	-	-	-
CycleGAN [Zhu 17]	✓	-	-	-	-	-
UNIT [Liu 17]	✓	-	-	-	-	-
ELEGANT [Xiao 18]	-	-	-	-	-	-
StarGAN [Choi 18]	✓	✓	✓	-	-	Dom.
FaderNet [Lample 17]	✓	✓	✓	-	-	Dom.
AttGAN [He 19]	✓	✓	✓	-	-	Dom.
STGAN [Liu 19b]	✓	✓	✓	-	-	Dom.
SingleGAN [Yu 18b] (1)	✓	✓	-	-	-	Dom.
SingleGAN [Yu 18b] (2)	✓	-	-	✓	-	Dom. + Div.
SMIT [Romero 19]	✓	✓	✓	✓	-	Dom. + Div.
SDIT [Wang 19]	✓	✓	✓	✓	-	Dom. + Div.
MUNIT [Huang 18]	✓	-	-	✓	Cont./Sty.	Div
DRIT [Lee 18]	✓	-	-	✓	Cont./Sty.	Div
DMIT [Yu 19]	✓	✓	✓	✓	Cont./Sty./Dom.	Dom. + Div
StarGAN v2[Choi 20]	✓	✓	-	✓	Cont./Sty./Dom.	Div

Table 2.12: Specific comparison of Domain Translation methods. This table summarizes the specificities of domain translation methods. We point out how the generated images are enforced to be close to the inputs (e.g., "Cy-C" for cycle consistency and "Rec." for reconstruction loss); how the target domain is passed to the generative model to perform the image translation, i.e., at the image-level, in the latent space via feature map concatenation or conditional normalization, e.g., CBIN, AdaIN; and how the diversity code is passed to the generative model. We also indicate the nature of the latent space inside the generative model (e.g., feature maps, vector following or not distribution, a combination of latent spaces), the number of discriminative models, and its outputs.

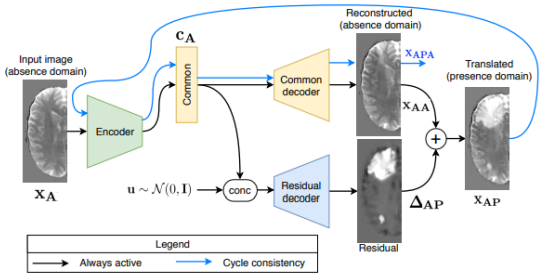
	Input proximity	Gen Dom. cond.	Gen Div. cond.	Gen. Latent space	Disc. Nb - Out.
Pix2Pix [Isola 17]	-	-	-	Fmaps	1 - R/F
CycleGAN [Zhu 17]	Cy-C	-	-	Fmaps	2 - R/F
One2One CycleGAN [Shen 20b]	Cy-C	-	-	Fmaps	2 - R/F
UNIT [Liu 17]	Cy-C (Lat.)	-	-	Prior $\mathcal{N}$	2 - R/F
ELEGANT [Xiao 18]	Rec.	Latent concat	-	Vector	1- R/F
StarGAN [Choi 18]	Cy-C	Image-level	-	Fmaps	1 - R/F + Cls
FaderNet [Lample 17]	Rec.	Latent concat	-	Vector	1 (Lat.) - Cls R/F
AttGAN [He 19]	Rec.	Latent concat	-	Fmaps	2 - R/F & Cls
STGAN [Liu 19b]	Rec.	Latent concat	-	Fmaps	2 - R/F & Cls
SingleGAN [Yu 18b] (1)	Cy-C	Latent CBIN	-	Fmaps + vect.	Nb Dom. - R/F
SingleGAN [Yu 18b] (2)	Cy-C	Latent CBIN	Sampl. concat	Fmaps + vect. + Prior $\mathcal{N}$	3 - 2 R/F & Lat.
SMIT [Romero 19]	Cy-C	Latent AdaIn	Sampl. concat	Fmaps + vect. + Prior $\mathcal{N}$	1 - R/F + Cls
SDIT [Wang 19]	Rec.	Image-level	Sampl. AdaIN	Fmaps + Prior $\mathcal{N}$	1 - R/F + Cls
MUNIT [Huang 18]	Cy-C	Latent AdaIn	Sampl. AdaIn	Fmaps/Prior $\mathcal{N}$	2 - R/F
DRIT [Lee 18]	Cy-C	Latent concat	Sampl. concat	Fmaps/Prior $\mathcal{N}$	2 - R/F
DMIT [Yu 19]	Cy-C	Latent CBIN	Sampl. CBIN	Fmaps/Prior $\mathcal{N}$	1 - R/F
StarGAN v2[Choi 20]	Cy-C	Latent AdaIn	Sampl. AdaIn	Fmaps/Style Space [Karras 19]	1 - Nb Dom. R/F

Medical domain:

Pathology disentanglement in medical image analysis- In the context of weak or semi-supervised pathology segmentation, several works [Andermatt 18, Vorontsov 20, Xia 19, Du 21, Kobayashi 21, Tardy 21, Zhang 22] leverage domain translation techniques and disentanglement strategies. Compared with [Huang 18, Lee 18, Choi 20] that separate content and style of the input image –which induces important transformation of the input after being transposed to another style (e.g., change of background and colors from medium to fine-grained details)–, these techniques rather disentangle pathology features (e.g., extract tumors from pathological images) from the rest of the input image (i.e., background, body structure, healthy tissues) left intact. For most of these works, an additional module or generator's output produces a segmentation mask; their frameworks are often asymmetrical to handle the extraction or the addition of pathology. Such framework is shown in Figures 2.27.



(a) Pathological ⟶ Healthy



(b) Healthy ⟶ Pathological

Figure 2.27: Simultaneous translation, segmentation and autoencoding [**Vorontsov 20**]. (a) The translation from pathological to healthy images, coupled with the segmentation of the tumor. The input is reconstructed by adding the translated image with the extracted tumor region. (b) Image translation from healthy to pathological images. The input is reconstructed with common code (only), and pathological images are produced by sampling random tumor code. They also use cycle consistency for this translation (blue lines).

2.2.2 Image Attribute Manipulation

Another line of research focuses on manipulating image attributes by studying the latent space of generative models. The objective is to change attributes (e.g., for human faces: hair color, glasses, age) separately, continuously, or discretely.

Unsupervised- Exploring the latent space of trained GANs, [Radford 16] show that a linear interpolation between two points of the latent space or following a given direction towards an attribute classification produces smooth and realistic transformations. First attempts try to disentangle the latent space of generative models into interpretable factorized representations. [Higgins 17] tune the importance of the KL-term in a VAE. Instead, [Chen 16] train an additional encoder network Q to maximize the mutual information between a subset c of latent code $[z, c]$ and the encoding of the generated image $Q(g(z, c))$, where g is the generative model. To manipulate the latent space of GANs, [Härkönen 20] perform a PCA on the first layer of the generator and transfer the resulting basis to the latent space (via linear regression). Then, they edit generations by moving along this latent basis. [Esser 20] rather train an invertible translation model to disentangle the latent space of a classifier or an autoencoder into interpretable semantic concepts. They show that modifications in this translated latent space induce semantic image modifications. Another work [Voynov 20] finds meaningful semantic directions in the latent space of a trained GAN. Given a set of K directions to discover, they train a weight matrix A to apply shifts of different magnitudes ϵ along a certain direction k. They also train a reconstructor model R to predict the direction k and the shift magnitude ϵ given the initial generated image, and the shifted one. The Figure 2.28 illustrates the training procedure of [Voynov 20].

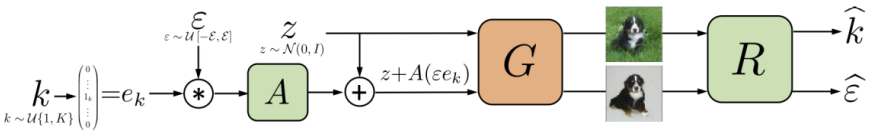


Figure 2.28: Discovering interpretable directions in the latent space of GAN [Voynov 20].

Building upon this work, [Cherepkov 21] propose a similar optimization in the parameters space of the GAN. In contrast, the StyleGAN frameworks [Karras 19, Karras 20, Karras 21] introduce an alternative generator architecture to synthesize images. Their generator starts with a constant learned input (rather than a noise sampled in $\mathcal{N}(0, 1)$), and then style vectors, generated from the latent code, are passed and updated at each convolution layer. Hence, the generator can control features of different scales when generating the image. To produce the latent code, they map the input sampled code $z \sim \mathcal{N}(0, 1)$ (commonly used in GANs) into another latent space $\mathcal{W}$. The latent code w is given to multiple MLP modules to produce the style vectors for each scale. This strategy, combined with the injection of noise, allows the separation of high-level image attributes (e.g., coarse forms, pose) from fine-grained details. The StyleGAN architecture is shown in Figure 2.29.

A series of works [Karras 19, Karras 20, Collins 20, Abdal 20, Chong 21, Wu 21, Tov 21] studies the disentanglement properties of the StyleGAN latent space $\mathcal{W}$ or the style space $\mathcal{S}$. To manipulate image attributes or to mix styles between images, different techniques project real images in the latent space $\mathcal{W}$ (or a continuous and larger version $\mathcal{W}+$), by

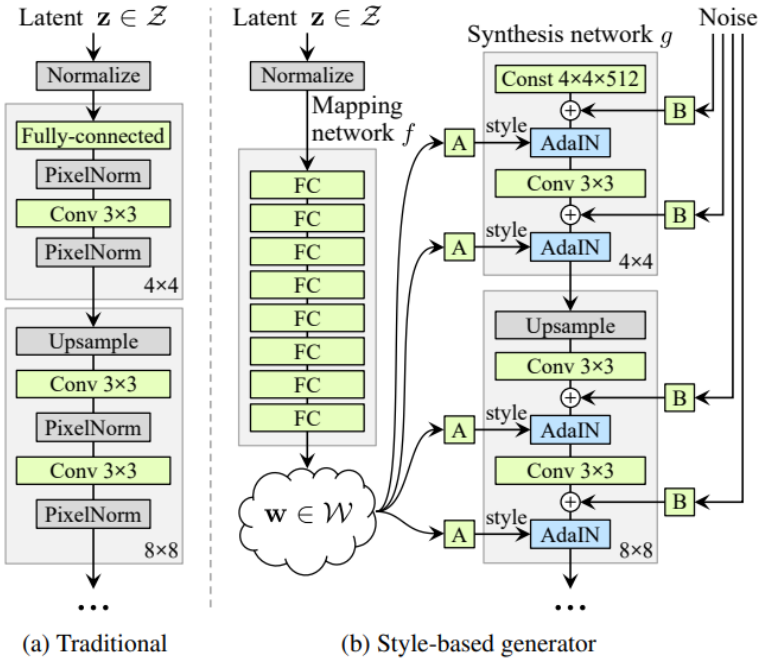


Figure 2.29: Comparison of common GAN and StyleGAN architectures [**Karras 19**]. (a) Traditional GAN architecture that generates images directly from a latent vector by successive upsampling and convolution operations. (b) The style-based generator where the latent code is first mapped to a second latent space $\mathcal{W}$ then given to a generator architecture at a different level through normalization operations (AdaIN).

training a specific encoder [Pidhorskyi 20, Richardson 21, Tov 21, Alaluf 21] or by solving an optimization problem on $\mathcal{W}+$ [Abdal 19, Abdal 20]. Built on StyleGAN [Karras 19], other works [Zhu 21, Liu 22a] slightly modify the architecture to enforce stronger disentanglement. Recently, a survey [Bermano 22] describes the state-of-the-art frameworks and applications of StyleGAN.

Supervised- Another line of work manipulates image attributes using a certain amount of supervision. Domain translation frameworks used for multi-domain translation [Xiao 18, Choi 18, He 19, Liu 19b] (see Section 2.2.1), can produce changes only impacting certain attributes that are learned during training. However, they only produce translations towards opposite attributes (e.g., the face of a young or an old person), and cannot generate an intermediate state between the two classes. In contrast, [Goetschalckx 19, Jahanian 20, Plumerault 20, Yao 21] use supervision to find interpretable directions in the latent space of GANs. For each image transformation, they optimize a linear or a non-linear model to find a path in the latent space such that moving along this path induces a continuous change in the image.

2.2.3 Visual Explanation as Counterfactual Generations

In the context of visual explanation, these approaches aim at producing an in-distribution counterfactual example. The generated image should resemble the input image while displaying patterns of a different class (or domain). Compared with attribution maps that highlight relevant region locations, counterfactual images provide relevant patterns or structures of each class and show how the input should be changed to belong to another class.

Via domain translation generators- Leveraging domain translation techniques (see Section 2.2.1), these methods first train a generative model to produce counterfactual images. At test time, the generative model produces for each input a counterfactual image that is classified in a different class than the input. This counterfactual image belongs to the distribution of "real" images from this other class. The differences between the input and the generated counterfactual give insights into the relevant patterns of each class.

Medical domain: In medical imaging, a CycleGAN framework [Zhu 17] is used in [Narayanaswamy 20] to emphasize important structures that differ between images from different classes. However, this framework does not interpret a classifier's decision but rather gives additional insights into what is different between healthy and pathological images. In a similar spirit, VA-GAN [Baumgartner 18] leverages conditional Wasserstein GAN [Arjovsky 17] to transpose pathological images into healthy images–producing an additive mask (see Figure 2.30).

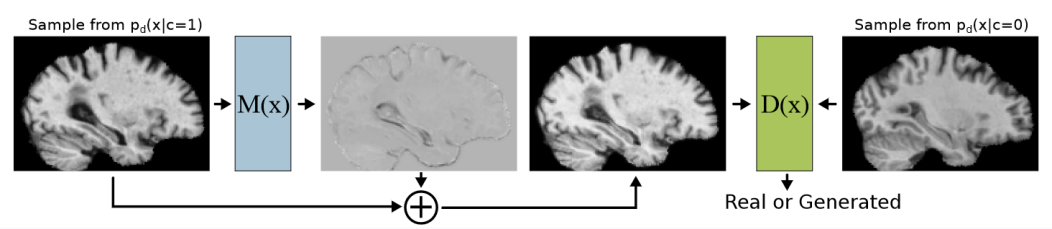


Figure 2.30: Overview of VA-GAN [Baumgartner 18]. Given a pathological input image, a mask generator M produces an additive mask that translates the input into a healthy image (counterfactual). A discriminator D encourages the generator to produce a realistic healthy image using a common GAN strategy.

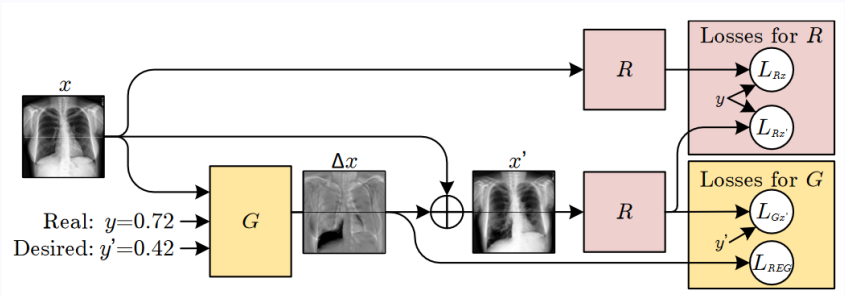


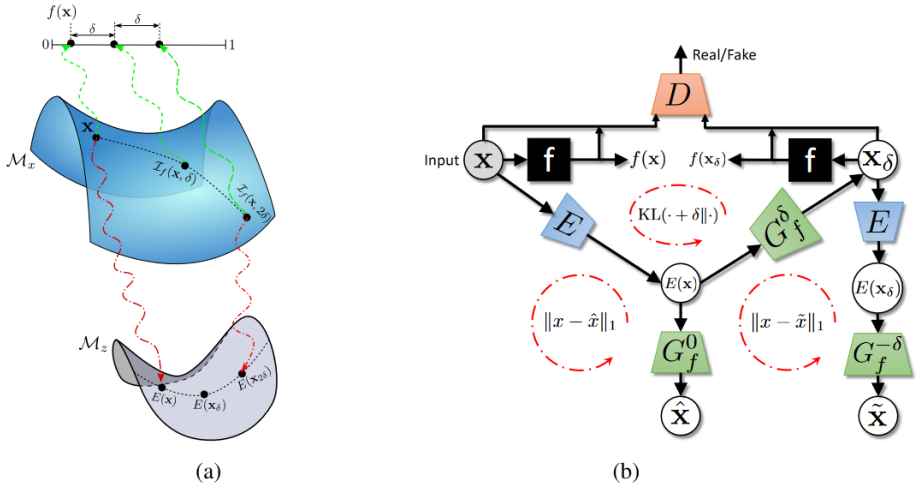
Figure 2.31: Overview of VR-GAN [Lanfredi 19]. Description of the training procedure of both G (orange) and R (red) using different loss terms.

For regression tasks in medical imaging, VR-GAN [Lanfredi 19] adapts the work of [Baumgartner 18] and replaces the discriminator model (commonly used in GANs

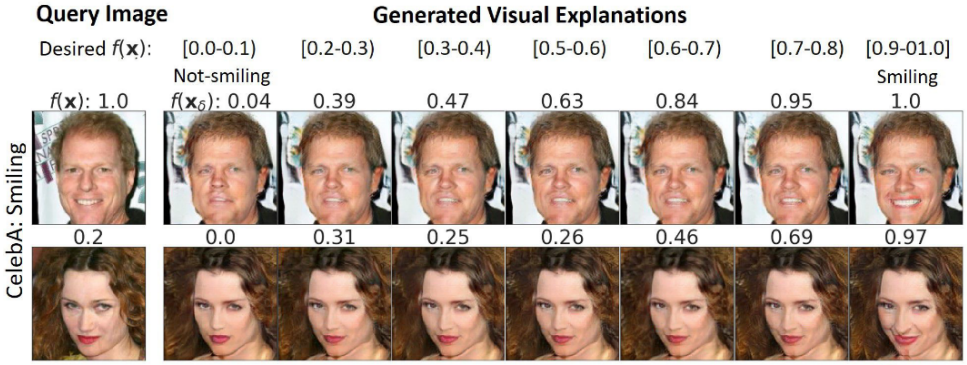
frameworks) with a regressor model that is trained to predict the same score for both the input and the generated image while the generator tries to make it predicts another target score (see Figure 2.31).

[Baumgartner 18] and [Lanfredi 19] are not directly used to explain the classifier's decision –as they do not involve a trained classifier and are trained on ground truth. As for the previous Cycle GAN framework [Narayanaswamy 20], it provides informative patterns of the different classes. More recently, [Oh 20] also propose to train a conditional generative model (composed of an encoder $e$and a generator g) to produce an additive map $\delta = g(e(x,t))$ such that $f(x+\delta) = t$ (t a target class) and the counterfactual image $x+\delta$ belongs to the distribution of images classified in class t. As in CycleGAN [Zhu 17], they use cycle-consistency constraint to enforce reverse mapping between domains. This method is not model-agnostic as they use the trained classifier (before the final classification layers) as the encoder e part of the global architecture. The encoder (i.e., the classifier backbone) remains fixed during the training. These additive mask approaches penalize the size and intensity of the mask using a L_1 norm to change the input minimally. It sometimes prevents the counterfactual examples from exhibiting localized but intense differences (compared with the input image) or taking into account the semantics of the image (e.g., adding the counterfactual pattern with poor realism). Finally, these additive maps often contain residual differences irrelevant to classification.

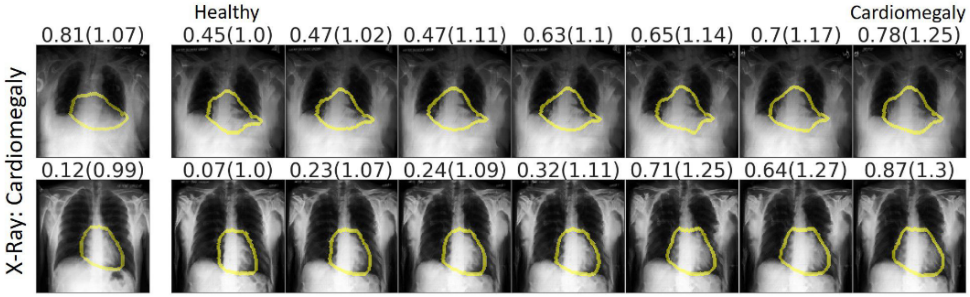
In [Singla 20], authors propose a framework to align plausible progressive generations with changes in a binary classifier's prediction score through batch normalization conditioning. They propose a conditional GAN framework (described in Figure 2.32a) where they train an encoder e, a conditional generator g and a conditional discriminator D to produce images $x_\delta = g(e(x), \delta)$ such that $f(x_\delta) = f(x) + \delta$ and x_δ belongs to the distribution of real images classified this way (via optimizing D). They also provide a visual explanation by computing the difference between x_0 and x_1 s.t. $f(x_0) = 0$ and $f(x_1) = 1$. Progressive generations are shown in Figures 2.32b and 2.32c.



(a) Framework overview



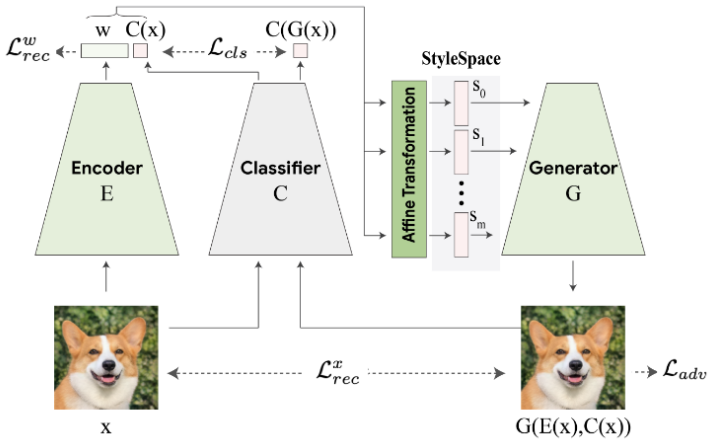
(b) Progressive generations on the smiling attribute



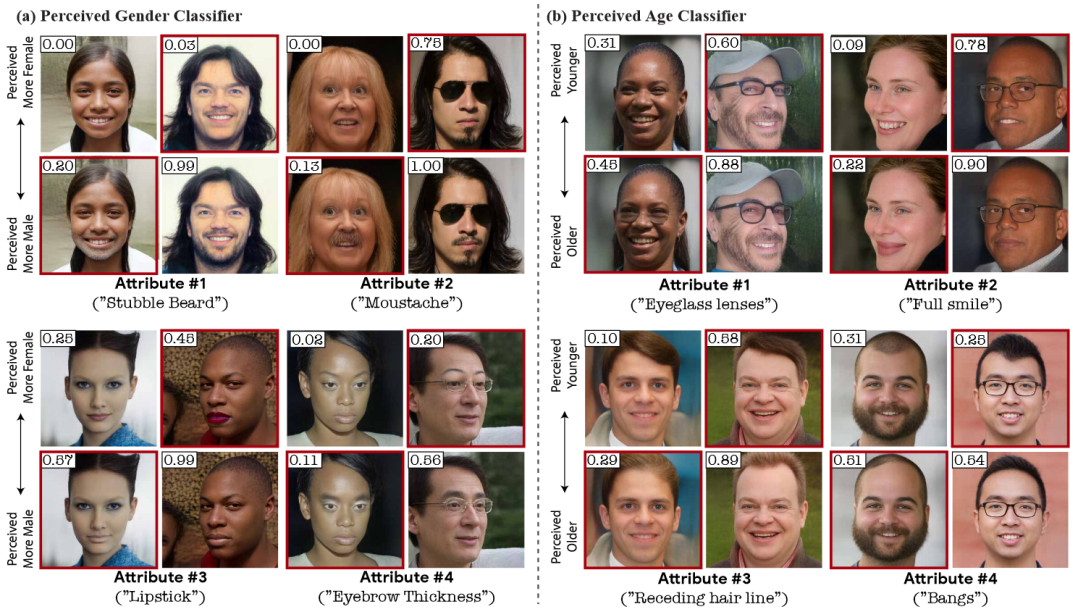
(c) Progressive generations on chest X-rays

Figure 2.32: Explanation by progressive exaggeration [Singla 20]. (a) *Left*: Schematic of the generations. They aim to generate an image that changes the model's output $f(x)$ (1st green line) from δ (2nd green line), or 2δ (3rd green line). The model E encodes (red lines) the data manifold (or space) $\mathcal{M}_x$ into a latent space $\mathcal{M}_z$. *Right*: The optimization framework of their model; red circles show the loss functions. (b) and (c) display input images and corresponding predictions (1st column), then images generated by the method for a given range of target score ($f(x)$). Note that (i) the generations are not always realistic (e.g., the eyes or the mouth for the smiling problem) or blurrier than the input (e.g., X-ray problem). (ii) Background details are sometimes modified.

Via GANs latent space Manipulation- In contrast, [Goetschalckx 19, Jahanian 20] produce counterfactual images by learning directions in the latent space of a GAN (via logistic regression, support vector machine classifier, or a more complex transformation model) that allow changing semantic and relevant features of the input image towards a given class or scoring function. Other works [Shen 20a, Schutte 21, Yao 21] leverage the disentangled latent space or even the style space [Wu 21] of the StyleGAN [Karras 19] to enforce the generations to only change the image features relevant for the classification of specific attributes, e.g., for facial attributes: hair color, mustache vs. no mustache, male vs. female, glasses vs. no glasses. All the attributes are available to train the latent transformation model. However, for a classification task involving multiple image attributes (e.g., old vs. young classification), as for domain translation and iterative counterfactual generation techniques, these methods change all relevant image attributes at once when (progressively) generating a counterfactual example. More recently, StyleEx [Lang 21] trains a StyleGAN model conditioned on a binary classification model's output to encourage the disentangled Style space to better capture relevant attributes for the classification (see the optimization framework in Figure 2.33a).
Then, they search for dimensions of the style space vectors that mostly impact the classification score. They show that these selected style vectors translate into disentangled image attributes important for classification (see Figure 2.33b). Using an encoder model, they project input images in the style space and change one relevant attribute at a time to assess its impact on the classifier's decision.
Although the StyleGAN framework produces high-quality images compared with previous GAN techniques, and despite efforts to project real images in the StyleGAN latent space, some image contents (e.g., elements in the background, object textures, or details) differ. The latent classification does not translate perfectly what the image classification model has learned. It induces a bias in the explanation. These methods are more adapted to provide insights on relevant image attributes of each class rather than producing a post-hoc explanation of a classifier's decision (on a specific input).



(a) Training framework



(b) Manipulation of the top detected attributes.

Figure 2.33: StyleEx: Explaining classifier's decision in the style space [Lang 21]. (a) Overview of the Training procedure: E encodes the input x in the latent space of the styleGAN $\mathcal{W}$. The encoded vector w is concatenated with the prediction score provided (by the studied classifier) for the input x. The resulting vector undergoes affine transformations to produce style vectors for each scale of the generative path. The generator G aims to reconstruct the input and preserve the classification's decision. (b) For both gender (left) and age classifier (right), the transformations along the different detected attributes direction are visually realistic and relevant for the classification task.

Table 2.13: Global Comparison of Domain Translation Counterfactual methods. For each method, we indicate if the visual explanation is local or global (Note: "Insights only" means the technique does not apply to a classifier but rather gives information about the task); whether the user has access to the model internal structures (and which) or not; whether we need additional data or only the tested data; and what is the sort of technique used. "NN as Encoder" means that the studied neural network is used as the encoder part or the generative model. "Gen." is the abbreviation for Generator model.

	Expl. Type	Model Agnostic	Data Agnostic	Method
Insights via CycleGAN [Narayanaswamy 20]	Insights only	✓	Database (Gen. training)	Not Explaining a Classifier
VA-GAN[Baumgartner 18]	Insights only	✓	Database (Gen. training)	Not Explaining a Classifier
VR-GAN[Lanfredi 19]	Insights only	✓	Database (Gen. training)	Not Explaining a Regressor
BIN[Oh 20]	Local/Global	NN as Encoder	Database (Gen. training)	Additive Counterfactual Mask Generator Optim.
PE [Singla 20]	Local/Global	✓	Database (Gen. training)	Counterfactual Generator Optim.
Latent Classif. in StyleGAN[Shen 20a, Schutte 21]	Insights	✓	Database (Enc, StyleGAN training)	Latent Manipulation (Not really an explanation)
StylEx [Lang 21]	Insights	✓	Database (Enc, StyleGAN training)	Latent Manipulation

Table 2.14: Specific Comparison of Domain Translation Counterfactual methods. This table summarizes the specificities of these methods. We point out if the method requires heuristic regularizations. we give relative indications about the computational cost of training the explanation method (e.g. ++ $\sim$ 5-10 times the classifier f training) and generating the map in inference (e.g. - - means a low computational cost $\sim f(x)$). We indicate the output of the explanation method / model, and what types of generation is used. We also emphasize if the method returns an attribution map and/or a counterfactual example.

	Heuristic Reg.	Computational Cost (Tr./Inf.)	Output type	Generation type	Expl. Map / Counterf. Gen.
Insights via CycleGAN [Narayanaswamy 20]	✗	++/- -	Counterfactual Gen.	Counterf. Gen	✗/ ✓
VA-GAN[Baumgartner 18]	✗	++/- -	Additive Counterfactual Mask	Counterfactual Gen.	✗/ ✓
VR-GAN[Lanfredi 19]	✗	++/- -	Additive Counterfactual Mask	Counterfactual Gen.	✗/ ✓
BIN[Oh 20]	✗	++/- -	Additive Counterfactual Mask	Counterfactual Gen.	✓/ ✓
PE [Singla 20]	✗	++/- -	Counterfactual Gen.	Counterfactual Gen.	✓/ ✓
Latent Classif. in StyleGAN[Shen 20a, Schutte 21]	✗	++/- -	Progress. Counterfactual Gen	Counterfactual Gen.	✗/ ✓
StylEx [Lang 21]	✗	++/- -	Progressive Multi-Counterfactual Gen	Single Attribute manipulation	✗/ ✓

2.2.4 Interpretable classifiers using domain translation

Leveraging domain translation techniques, some works directly build interpretable classifiers that produce their explanation maps. They typically optimize a multi-tasks model that learns to classify the input and to generate either reconstructed input or a counterfactual image.
To do so, these techniques adopt:

- A generative model structure with an additional classification sub-module inside. The sub-module is often connected to the encoder part of the generative model.
- A common domain translation generative model that is coupled with a two-heads discriminative model. The first head identifies real from generated images (usual discriminator task); the second head classifies the domain.

Classification head inside a generative model-

Medical domain: [Seah 19] first train a GAN generator g coupled with an encoder e to project chest x-rays images in a latent space and then map them back to the image space. Second, they learn a multilayer perceptron (f_{MLP}) to classify the latent code as a healthy or pathological case.

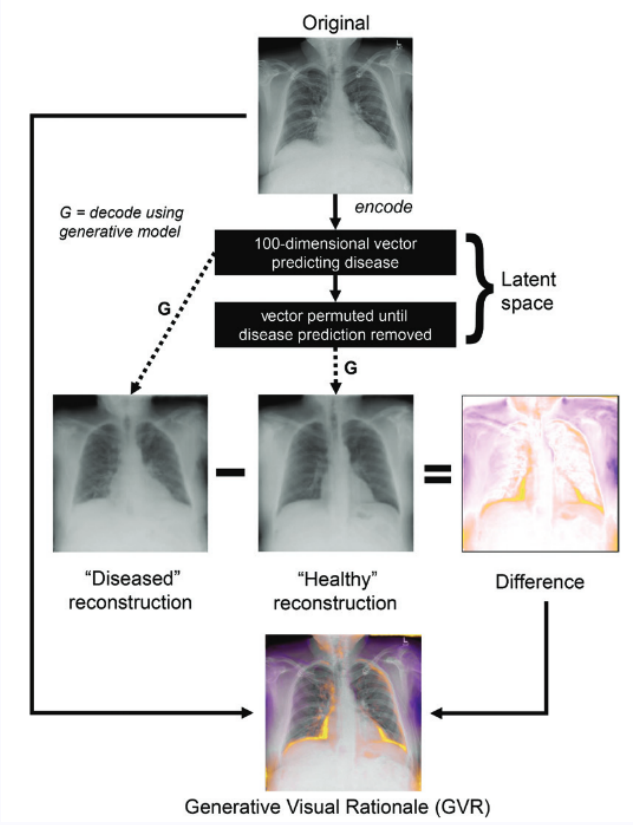


Figure 2.34: Overview of the visual explanation generative process [Seah 19]. *Translation*: The input image is encoded in a 100-dimensional vector and permuted until the decoded image is no longer predicted as pathological. *Reconstruction*: the decoder module reconstructs the input given the initial encoded vector. The difference between these two generated images provides the visual rationale (or explanation) for the input.

At test time, an image x is projected in the latent space (via e), then they change the latent encoding until the classification changes i.e. z_c such that $f_{MLP}(z_c) \neq f_{MLP}(e(x))$; and finally reconstruct the corresponding counterfactual image (via g). The visual explanation is defined as the difference between the decoded initial latent vector $g(e(x))$ and the decoded counterfactual latent vector $g(z_c)$. Figure 2.34 illustrates the procedure.

In a similar spirit, [Biffi 18] train a variational autoencoder (VAE) [Kingma 14] to learn a distribution of left ventricular segmentation (region of the heart). Simultaneously, an MLP module is connected to the mean encoded vector of the VAE; and is trained to identify healthy from hypertrophied heart segmentation. The main issue of this approach is the poor reconstruction quality producing blurry images and losing details from the inputs.
The ICAM framework [Bass 20] (see Figure 2.35) proposes an interpretable classifier for brain MRI images, built upon DRIT++ [Lee 18], which disentangles content and style of images to perform domain translation.

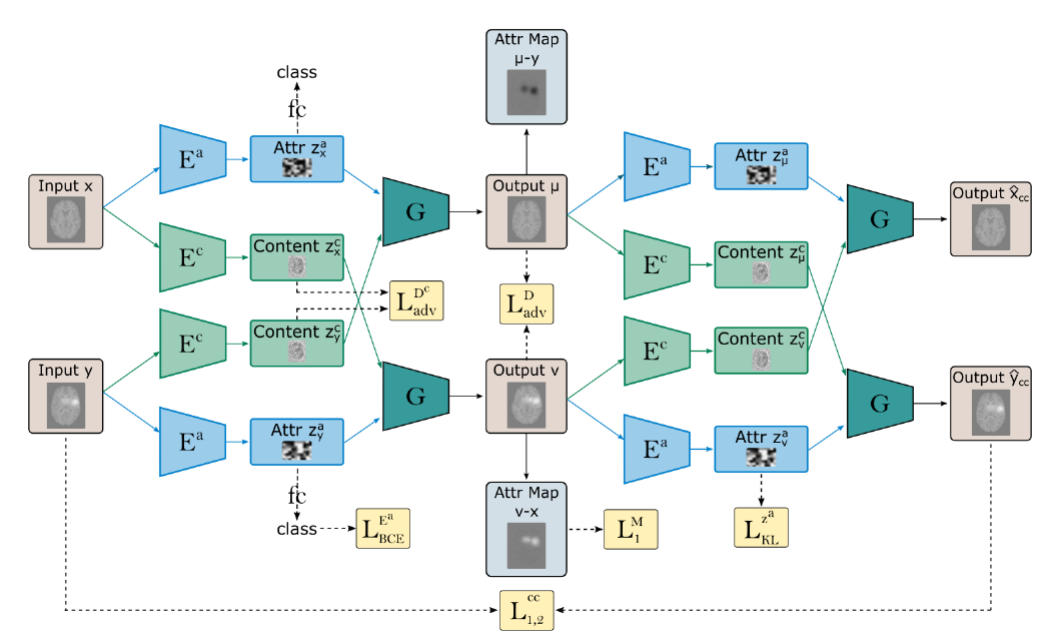


Figure 2.35: Overview of ICAM optimization framework [Bass 20]. At each step, content (E^c) and attributes (E^a) encoders extract latent information from inputs from coming from domains x and y. The attribute information (i.e., brain pathology vs. healthy) is removed from the content (i.e., brain structure, background) encoding via an adversarial content loss ($L^{D^c}_{adv}$). The image translation is obtained by switching attributes and content between the two domains (through the generator decoder G). A discriminator enforces the generations to belong to the target domain distribution (L^{D}_{adv}). A cycle consistency constraint enforces the generative model to map the translated image back to the inputs. An additional loss L^M_1 enforces the difference between the input and its generated translation to be minimal.

The idea is to first encode separately content $c = e_c(x)$ (e.g. background, gen-

eral structures or textures) and class attributes $a = e_a(x)$ (features relevant for the domain or the classification task) of the images; then to mix encodings of images from different domain to generate transposed images $g(c_i, a_j)$ (i.e. counterfactual images). In their framework, the class attributes encoder is followed by a multilayer perceptron f_{MLP} that performs classification and guides the choice of attributes to generate the counterfactual image. Using a VAE-GAN architecture (as in DRIT++), they place a Gaussian prior over the attribute latent space to encourage variability in the generation. This emphasizes the relative impact of different features on the classification task. At test time, for a given input x, the attribute encoder provides a prediction (i.e. $y = f_{MLP}(e_a(x))$); They sample multiple counterfactual attributes (i.e. $a_i \sim \mathcal{N}(0, 1)$ s.t. $f_{MLP}(a_i) \neq y$) which allows to generate multiple counterfactual images (i.e. $x_c^i = g(e_c(x), a_i)$). They compute the visual explanation by taking the mean and the variance of the difference between the input x and the generated counterfactuals.

Classification head in the discriminative model- Based on StarGAN [Choi 18] (see Section 2.2.1), [Siddiquee 19] use a double head discriminator; the first output identifies real from generated images, the second is for domain classification. As for StarGAN, [Siddiquee 19] produce a counterfactual image, for a given input x, by conditioning a generator g with the target class domain c_t i.e. $g(x, c_t)$.

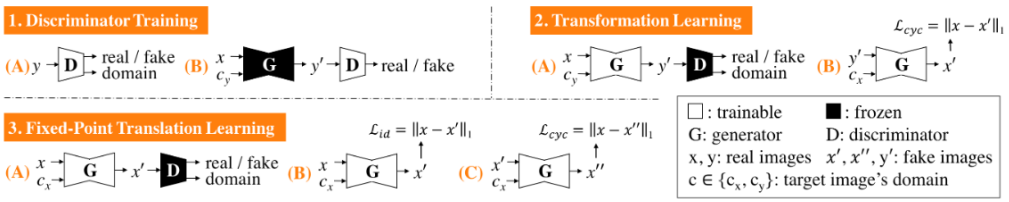


Figure 2.36: Overview of Fixed Point GAN [Siddiquee 19]. *1. Discriminator training*: the discriminator D learns to identify real from generated images and to classify the domains of the input images (Similar to StarGAN). *2. Image translation training*: given the input and a target domain, the generator learns to perform cross-domain translations using cycle-consistency constraints (similar to StarGAN). *3. Fixed point translation training*: the generator also learns same-domain translation given the input image and the corresponding domain code.

Then, they enforce minimal transformations in the counterfactual generation, which better serves localization purposes. They add an identity penalization term that encourages identical reconstruction between the input x and a stable reconstruction produced by conditioning the generator with the input class c. We provide the optimization framework in Figure 2.36.

Medical domain: For pathology detection, at test time, [Siddiquee 19] generate a counterfactual image towards the healthy class only and compute a difference map between the input and the healthy generated image. The classification is obtained by measuring the maximum value across all pixels in the difference map, i.e., high absolute value for pathological cases and near zero for healthy cases because of the identity term. The difference map gives the explanation map or the anomaly

localization map (in this case). Note that the classification head of the discriminator could also produce the classification.
For specific disease detection, [Wolleb 20] propose a similar optimization, using a generator with two heads producing healthy and pathological generation, respectively (rather than a conditioned generator). Using skip connections, they produce more detailed localization maps.

iCaps [Jung 20] considers a similar framework as DRIT++ [Lee 18] by disentangling latent features of an input x into two complementary subspaces : class-relevant and irrelevant. They leverage Capsules Networks [Sabour 17] to show intra-class variation as a set of concepts for the class-relevant subspace. Although a classification head also produces the classification in the discriminator model, no visual explanation is provided.

Despite efforts, these classification models often perform poorer than common deep neural networks and are more complex to train.

2.3 Evaluation of visual explanations

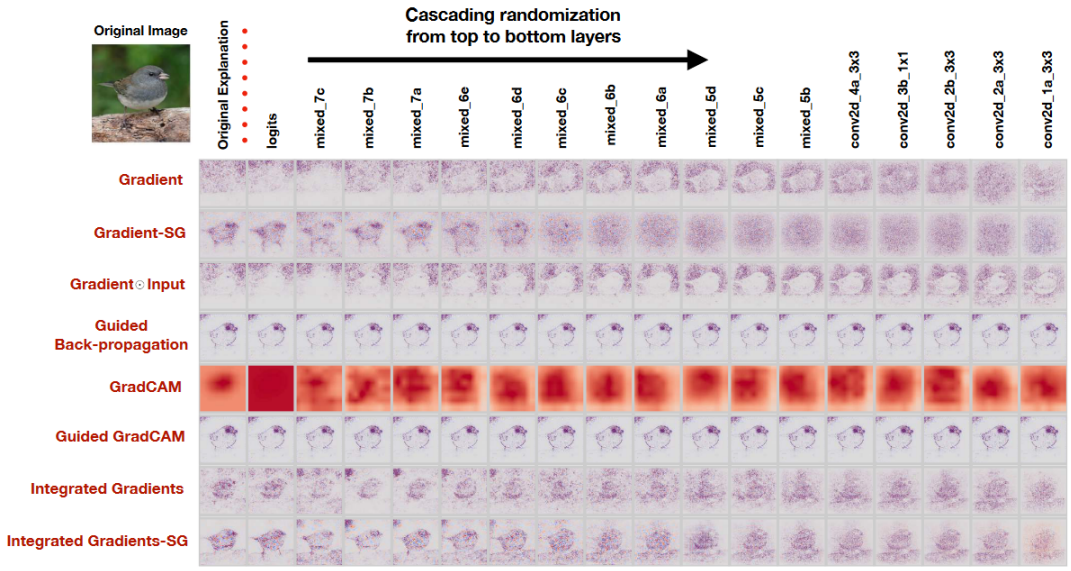
Several methods have been proposed to evaluate visual explanations (especially attribution maps), but there is still no consensus. Indeed, visual explanation methods are complex to assess as no ground truth explanation is available. It may be difficult to separate errors from the classification model or the attribution technique. Through randomization tests, [Adebayo 18] propose sanity checks to assess if an attribution method is related (or not) to the model and the data used for training. They first randomize the model weights starting from the top layer (and continuing until the input layer) and compute attribution maps at each step. Second, they retrain the same model on the same database but permute the labels. Then, they provide an attribution map for this new model. An attribution describing a model's behavior depends on the model's parameters and the training database; the attribution is expected to differ significantly in the two randomization tests.
In their study, they show that methods such as Guided Backprop [Springenberg 15] or Guided GradCAM [Selvaraju 17] are independent of both model parameters and label randomization and rather detect the edges of the input image (see Figures 2.37). Another common method to evaluate visual explanation techniques is correlating the feature attributions with human annotations to evaluate how well they match human expectations. Weak localization or pointing game are thus proposed for instance in [Zhou 16b, Fong 17, Dabkowski 17] to illustrate the performance of attribution maps to localize objects. Localization metrics enable human experts to assess the quality of the visual explanation. However, the model may use input features outside the annotation. Input degradation techniques thus measure the importance of features for the classifier's prediction. First [Samek 17] propose to progressively perturb the input image starting by the most relevant region first (MoRF) w.r.t. the attribution map, and compute the classification score (applying f). Then, they compute the area over the MoRF perturbation curve (AOPC):

$$AOPC = \frac{1}{L}\mathbb{E}_x\left[\sum_{i=0}^{L} f(x) - f(x^i_{MoRF})\right] \tag{2.15}$$

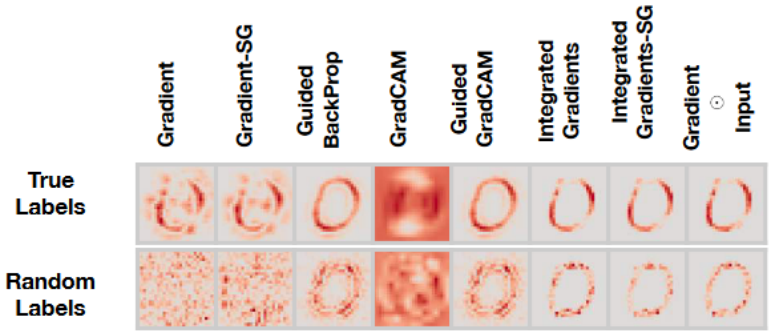
where L is the number of perturbation steps and x^i_{MoRF} is the perturbated image at step i. Suppose an attribution map translates the importance of region w.r.t. f (by order of importance), in that case, the classification score should quickly change along with the perturbation steps, and the AOPC score is expected to be high. Similarly, [Petsiuk 18] introduces a deletion (D) and an insertion (I) score by removing (resp. adding) relevant features to the input images (resp. a baseline perturbated image, e.g., blurred image), and measuring the impact on the classification's output. These two metrics are combined in [Lim 21] giving a feature relevance R such that

$$\frac{1}{R} = \frac{1}{2}(\frac{1}{I} + \frac{1}{1-D}) \tag{2.16}$$

In these evaluations, synthetic perturbations (e.g., black pixels, blur, noise) are applied to the input image. However, as stated in section 2.1.3, synthetic perturbations may have an undesired and uncontrolled impact on the classification. To mitigate this effect, [Samek 17] also propose a second metric that measures the impact of perturbing the most relevant region first w.r.t. the impact of the perturbation on the classifier. For this second term, they first measure the impact on f of perturbing the least relevant region. [Chang 19] use their in-filling technique (via contextual attention GAN) to perturb the input and enforce proximity to the database distribution. To prevent any impact of the perturbation, ROAR [Hooker 19] removes the most relevant features (w.r.t. the attribution map) from



(a) Weight randomization test



(b) Label randomization test

Figure 2.37: Sanity checks through randomization tests [Adebayo 18]. (a) In the 1st column: the original explanation map for different visual explanation methods. Then from left to right: the evolution of the explanation maps along with progressive randomization of the model's weights (up to complete randomization). (b) The absolute value of the explanation map generated by different techniques for the initial classification model (top) and a model trained on random labels (bottom).

all inputs of the training database and retrains the model on these new inputs. If the attributions highlight important input features, the model accuracy should drop. This last method is computationally costly.

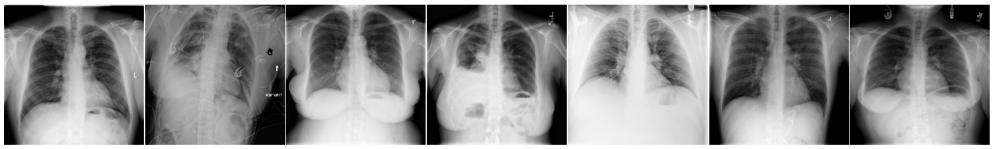
Visual explanation can be evaluated through substasks such as localizing object (mentioned above), detecting biases [Joshi 18, Singla 20] the model has learned (especially for counterfactual visual explanations), showing adversarial robustness [Hsieh 20], or through a human study [Singla 20, Yeh 20, Lang 21].

2.4 Datasets

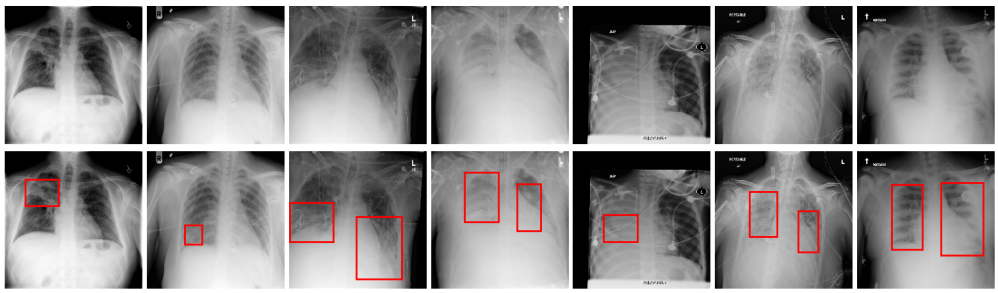
In this section, we introduce the different datasets we used to validate our methods (see chapters 7 and 8). Note that we did not provide all types of qualitative and quantitative evaluations for all the datasets.

2.4.1 Chest X-Rays from RSNA Pneumonia Detection Challenge

We created a chest X-Rays dataset from the available RSNA Pneumonia Detection Challenge, which consists of 26684 X-Ray Dicom exams extracted from the NIH CXR14 dataset [Wang 17b]. This medical imaging technique exploits the different densities of body structures. X-rays pass through the body region to study, are impacted depending on the density of the tissue, and a specific camera produces a 2-dimensional image given the resulting rays it receives. Dense tissues absorb these rays (e.g., bone tissue, opacity in the lung), while soft tissues let them through (e.g., skin, air, and fat). In an X-ray image, dense and compact tissues are the lighter regions (while the soft tissues or the air are darker). Clinicians often used x-rays images to detect bone pathologies (e.g., fracture) or lung pathologies (e.g., pneumonia, mass, nodule, cardiomegaly). This dataset only contains images of healthy patients, patients with pneumonia, or patients with other pathology than pneumonia. Pneumonia manifests as one or multiple opacity regions in the lung, i.e., lighter areas in the lung which is dark in x-rays (soft tissue filled with air).



(a) Healthy cases



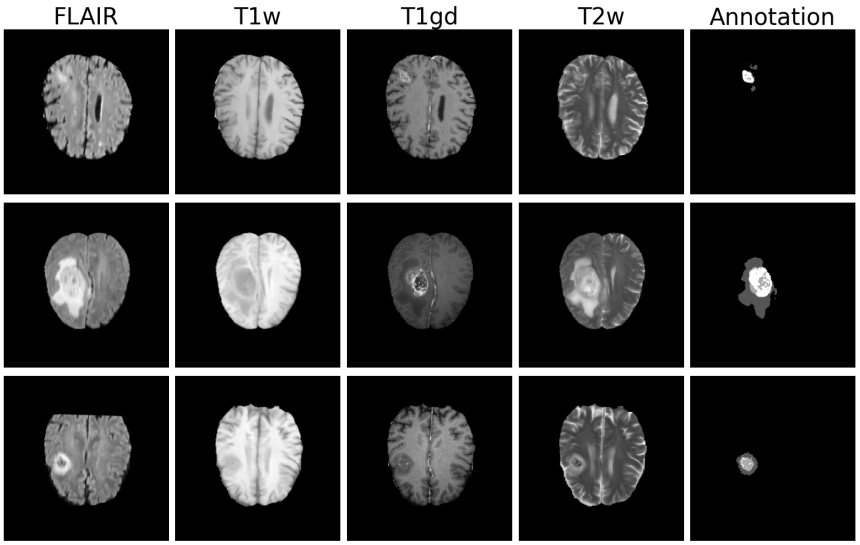
(b) Pneumonia cases

Figure 2.38: Examples of chest X-ray images from the Pneumonia Detection Challenge. (a) Exams of patients with healthy lungs. (b) Top: Exams of patients with pneumonia; Bottom: The same exams with the experts' annotations contouring the pneumonia opacity.

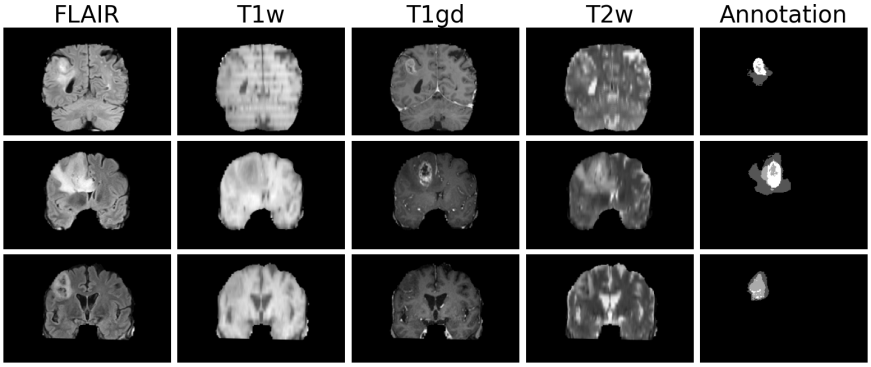
The 2D images are found at the scale 1024x1024 in Dicom format, the standard format for medical imaging data. Pneumonia cases are provided with expert bounding box annotations around opacities.

2.4.2 Brain MRI from Medical Segmentation Challenge

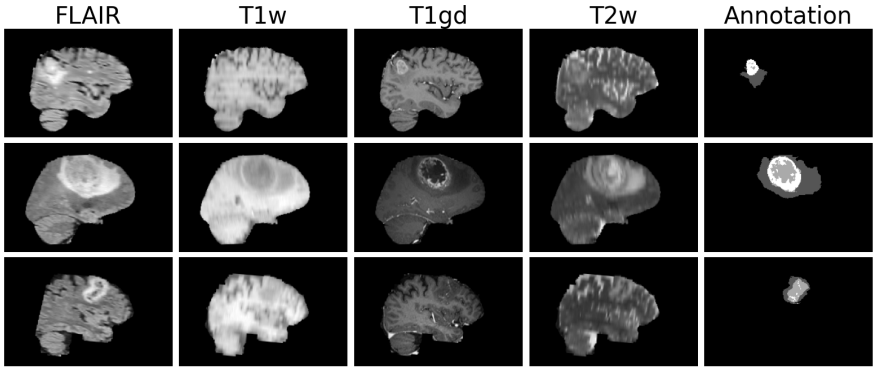
Magnetic Resonance Imaging (MRI) is a medical application of nuclear magnetic resonance that generates 3D volumes of images (compared with X-rays images in 2D). It leverages the excitation and the relaxation of hydrogen atoms subjected to a magnetic field. The brain MRI dataset comes from the Medical Segmentation Decathlon Challenge [Simpson 19] which is composed of 750 exams (484 for training and 266 for testing) from patients diagnosed with two types of tumors: lower-grade glioma or glioblastoma. The exams and corresponding annotations are extracted from the data used in the Brain Tumour Image Segmentation (BraTS) challenge [Menze 15, Bakas 17, Bakas 18]. Each exam is anonymous and comes with 4 MRI modalities: T2 Fluid-Attenuated Inversion Recovery(FLAIR), native T1-weighted (T1w), post-Gadolinium contrast T1-weighted (T1gd), and T2-weighted (T2w). In addition, segmentation masks are provided with 4 levels of annotations: background, edema, non-enhancing tumor, and enhancing tumor. Expert neuroradiologists approved all annotations. We display some examples in Figures 2.39a, 2.39b and 2.39c for the 3 different views of the 3D MRI volumes (see the different planes in the Figure 1.1). All the image volumes are centered, min-max scaled to the same range, and at the resolution 155 x 244 x 244 (axial x coronal x sagittal).



(a) Axial view



(b) Coronal view



(c) Sagittal view

Figure 2.39: Examples of MRI images from the Medical Segmentation Decathlon Challenge. Different patient exams are displayed in different lines. For each patient, we show a slice in (a) the axial (or transverse), (b) the coronal, and (c) the sagittal view. The different MRI modalities are shown in the first four columns, and the expert segmentation in the last column. From darker to lighter regions in the annotations: background, edema, non-enhancing tumor, enhancing tumor.

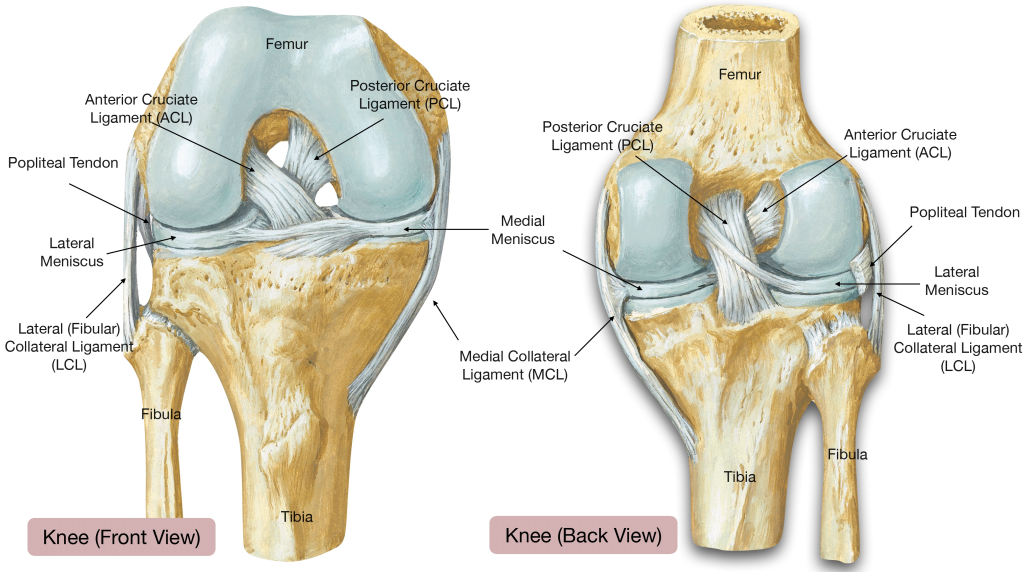
2.4.3 Knee MRI from Incepto's project KEROS

Knee pathologies and the role of MRI- Knee injuries are among the most common musculoskeletal injuries and affect many patients, such as athletes and young and older patients. Knee injuries can result in pain, knee instability, or arthrosis. The functional consequences limit sports or even the daily activities of the patients. The most effective way to diagnose meniscus, ligament, and cartilage injuries and fractures with little or no displacement is achieved through magnetic resonance imaging (MRI). First, we review the main elements of knee anatomy (see also the Figure 2.40):

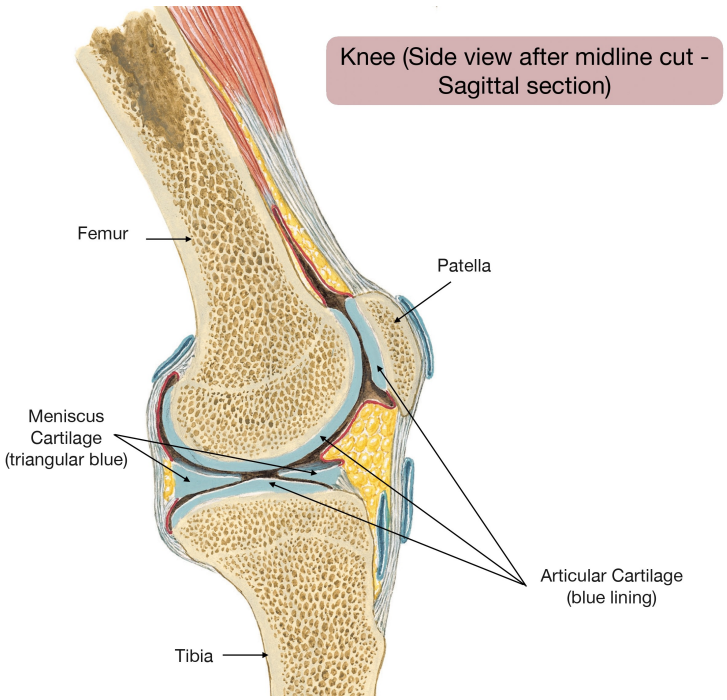
- The **anterior** and **posterior cruciate ligaments** are the main stabilizing ligaments of the knee. Their role is to connect the femur and the tibia while allowing the knee to move. Their rupture leads to instability of the knee, and if not treated, this instability can lead to the development of early osteoarthritis of the knee. Rupture of the anterior cruciate ligament (ACL) is common following sports injuries and should be detected early for appropriate management.
- The **menisci** are two fibro-cartilaginous structures located between the femur and the tibia in the shape of a crescent. They play both the role of stabilizers of the joint and shock absorbers. They allow the bones to fit together and slide smoothly, thus avoiding premature cartilage wear. They can tear and, from the age of 40, become damaged. They can also suffer traumatic lesions in younger subjects. A tear can appear in the meniscus, and a fragment of variable size can detach and sometimes get stuck in the knee. In addition, a meniscal tear causes pain and can lead to osteoarthritis.
- **Cartilage** protects the bone and allows the surfaces of the joint sections to slide easily against each other. Cartilage degrades not only over the years but also due to its use. The regenerative capacity of cartilage is limited. This is due to the absence of blood vessels that generally allow a high metabolism. The scar tissue consists mainly of fibrous cartilage, which is of inferior quality to the original hyaline cartilage. With time, cartilage damage occurs.
- The **medial collateral ligament** is one of the other major ligaments of the knee. Its function is to resist the outward rotational forces of the knee. It is frequently injured and has an excellent ability to heal if treated properly.

Keros project- Due to the frequency of knee injuries and the good diagnostic performance of the exams, the knee MRI is one of the most frequently ordered and performed examinations. For this reason, it is interesting to have a diagnostic aid to reduce the interpretation time of this examination and to help non-musculoskeletal radiologists interpret knee MRI. Indeed, studies have shown that musculoskeletal radiologists have better diagnostic accuracy than non-musculoskeletal radiologists in detecting ACL tears [Challen 07] and surgeons in detecting cartilage injuries [Figueiredo 18]. Those with more years of experience are intuitively more accurate among musculoskeletal radiologists, especially in diagnosing meniscal tears and cartilage lesions [Krampla 09]. Therefore, an artificial intelligence-based tool could help non-expert or inexperienced physicians interpret knee MRIs.

Several studies have shown the feasibility of a deep neural network to detect abnormalities in different knee structures with high performance. In particular, [Bien 18a] proposes a convolutional neural network that detects ACL and meniscus abnormalities on knee MRI with AUC scores of 0.965 (95% CI: 0.938, 0.993) and 0.847 (95% CI: 0.780, 0.914). [Liu 19a] proposed several convolutional neural networks, including one capable of de-



(a) Menisci and Ligaments parts



(b) Cartilage sections

Figure 2.40: Illustration of the principal components of the knee anatomy.(a) Meniscus and ligament components shown in front and back views (along the coronal view). (b) Cartilage regions in sagittal view of the knee. These figures are extracted from http://thekneeworld.com/anatomy/.

tecting ACL tears (AUC 0.98) [Liu 19a] and cartilage injuries (AUC 0.917) [Liu 18a]. [Fritz 20] model is capable of detecting meniscal tears (AUC 0.961). However, there are no turnkey AI applications on the market today (except for Keros) tackling all the major knee injuries. We describe the main pipeline later in section **??**.

2.4.4 MNIST

The MNIST [LeCun 10] dataset consists of white written digits on a black background. The digits are centered, min-max normalized to the range [0, 255], and have a fixed size of 28 x 28. It is a common database to test machine learning techniques. Figure 2.41 below shows a sample of digits.

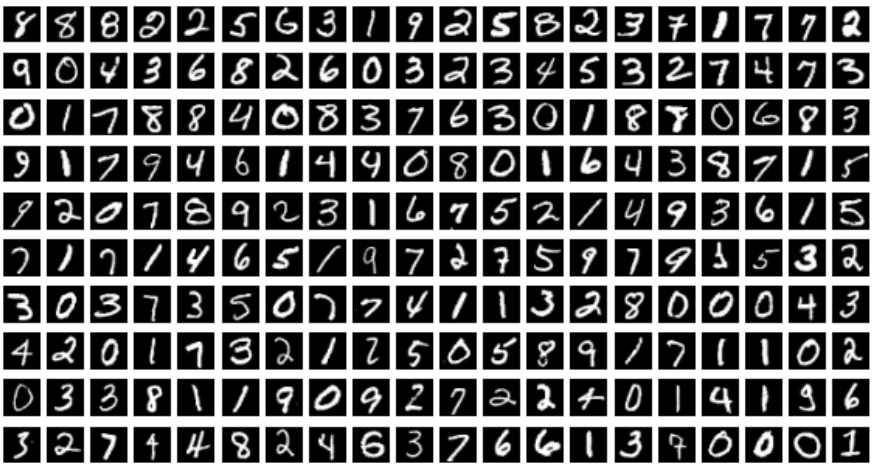


Figure 2.41: Examples of MNIST digits.

2.4.5 CelebA

We also conduct experiments on a colored image dataset: CelebA [Liu 15]. This is a large-scale face attributes dataset with more than 200K celebrity-colored images in RGB format (more than 10K different identities), each including 40 attribute annotations, e.g., eyeglasses, bangs, mustache, smiling, bold. The authors also provide a second dataset version ("Align&Crop"). All the images are cropped around the celebrity's face and centered. As many computer vision works, we used this preprocessed dataset. The Figure 2.42 shows some samples.

Figure 2.42: Examples of CelebA images. The samples are extracted from the "Align&Crop" dataset [Liu 15].

2.4.6 Synthetic squares dataset

We generated synthetic 28x28 images with two possible classes: healthy (class 0) and pathological (class 1). The healthy images contain two large gray squares on a black background. The pathological cases contain these squares, and pathology is represented as a smaller and brighter square either in the middle right (R), in the middle left (L), or both (RL). The two possible positions of the square translate two types of pathology (i.e., in different regions). We add a random offset to the position of the large and the small squares in each direction and impose small variations of the sizes of the squares. Examples are provided in Figure 2.43.

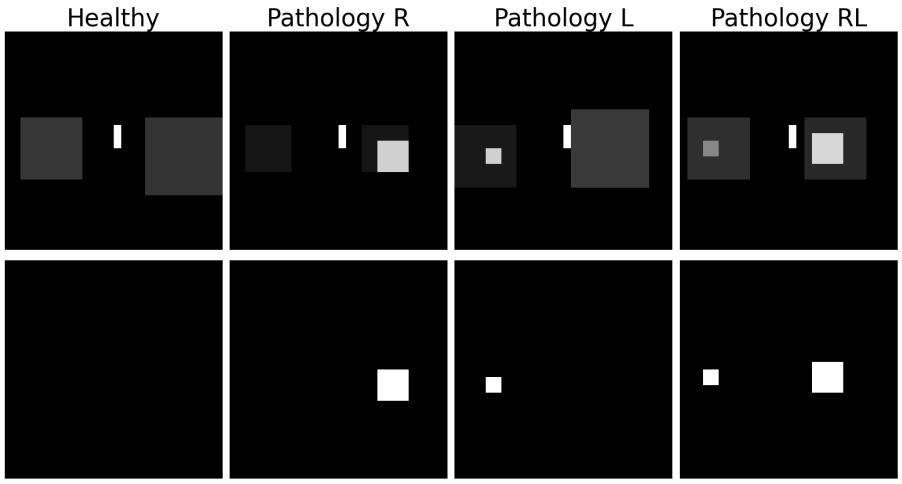


Figure 2.43: Examples of Synthetic images Top: Examples of synthetic images. Bottom: The corresponding ground truth annotation of the synthetic pathology. From left to right: a healthy case, then pathological cases displaying the different types of synthetic pathology.

2.5 Related work summary and Preliminary works

Visual explanation- Many contributions tackle the problem of visual explanation to explain deep learning classification model decisions. The different approaches differ according to the type of users who will use the AI solution, how black box the model is (i.e., more or less restricted access to the internal structures of the model); and what kind of visualizations are expected.
In the literature, visual explanations generate either attribution maps highlighting the impacting regions for the model or counterfactual examples. These counterfactuals show how the studied image would have to change to be classified differently, but also, more globally, what are the structures on which the model relies to decide. In this last case, we can then ask ourselves if the model bases its decisions on the right structures (e.g., related to clinical practice). If it only recognizes specific structures but does not focus on other clinical signs that are important for the clinician, or if it has learned biases (e.g., due to the composition of the training database).

Preliminary on attribution techniques- Backpropagation-based and GradCAM methods produce attribution maps that give first clues about the model's attention. However, they are often

1. Noisy (e.g., Gradient) or coarse (GradCAM);
2. Less applicable to medical images (and even more so to 3D images) where the structures to be detected may be diffuse or resemble other surrounding structures. In contrast, the studied objects often stand out from the background in natural images.
3. These methods also require access to the gradients or even to the model's structures.

For perturbation methods (or path-based methods), the choice of the perturbation (resp. of the reference) strongly impacts the explanation output.

> **Medical domain:** Indeed, the images of the same medical classification problem are similar, i.e., we observe the same structures (e.g., tissues, organs, bones) with the same intensities, framing, and small variability. Synthetic perturbations (e.g., zeros, constant, noise, blur) produce new images completely outside the distribution of images on which the model is supposed to work. We cannot control this impact and thus quantitatively distinguish what comes from the selected region (i.e., explanation map) and what comes from the perturbation type. In addition, for binary classification, we expect that the perturbation applied to the relevant features either changes the classification prediction toward a neutral prediction (not always clear for a binary task) or toward the opposite class. The second objective better suits detection problems as it points out what should be present (or not) to produce one decision or the other. However, we observed that certain perturbations either poorly impact the classifier's decision (e.g., Gaussian blur on chest X-rays), are biased toward one decision (e.g., a full blurred rain MRI image is predicted healthy), or are not applicable to the problem, e.g., On both chest X-rays and brain MRI, a constant perturbation (e.g., zeros, ones, mean) applied on a random attribution map has more effect than any other attribution methods (showing high sensitivity of the model to such artifact patterns in images).

In comparison, the objects in natural images can be very different their position in the image and everything around them (e.g., background, other objects). In this case, the image diversity mitigates the impact of synthetic perturbations.

The same problem arises when we want to evaluate the importance of features highlighted in the explanation map (see sections 2.3 and 7.3.2).

While providing promising results and mainly model-agnostic, perturbation methods suffer from heuristic regularization used to smooth the explanation map (or mask in this case) and avoid adversarial artifacts.

Medical domain: These techniques often generate explanation maps that are not attached to the image structures because of regularizations (e.g., upsampling, blur). Thus, they point out coarse regions that contain both relevant and irrelevant features for the model, especially when the abnormality to detect is diffuse (e.g., pneumonia in chest X-rays). We also noticed that the same regularization is not always adapted for different medical problems and even for a different image in the same problem. In addition, iterative perturbation methods such as BBMP [Fong 17], RISE [Petsiuk 18], or worse PDA techniques [Zeiler 14, Zintgraf 17], are computationally expansive especially in 3D image problems (e.g. Keros on MRI), and thus not adapted for real-time situations.

To avoid (or at least reduce) the regularization problems associated with masks, we proposed to generate directly the perturbed image that changes the model's decision. We thus join the adversarial generation approaches. As we have seen in section 2.1.4, some contributions have been proposed to generate adversaries to explain the models' decisions by adding regularity constraints on the space in which the perturbation is sought. However, these approaches are iterative (one optimization problem for each image) and therefore computationally expensive, but also not adapted to binary or a few class problems. Indeed, in natural image problems with 100 to 1000 classes and significant inter and intra class variability, the attacks (with these regularizations) will tend to target the objects studied (Yet, this is not always the case [Woods 19]).

Medical domain: In our medical case, we do not observe this mitigation effect because of the few classes, combined with the low variability, and the similarity between the abnormality and the rest of the structures. The attacks are constrained enough and easily produce a pattern (anywhere or everywhere in the image) that changes the model's decision. It is comparable to an overfitting effect during the model's training.

To avoid this issue, we thus combine (see chapter 4) the ideas from adversarial generation for visual explanation [Elliott 19] with trainable mask model [Dabkowski 17] to regularize the attack (and so the adversarial generation) on a database (decreasing the overfitting effect). Then, given the benefits and flaws of this first attempt, we progressively add conditions on the generation optimization and the resulting visual explanation. The next chapter introduces these different properties in a general formulation of the visual explanation and bridges the gap between counterfactual and attribution techniques.

3

Problem Formulation

Our research focused on medical image classification tasks and, more specifically, on deep neural network classifiers. We aim to produce a post-hoc visual explanation of the classifier's decision for each image input.

Targeted users: First of all, this specific type of explanation is designed for both

1. The clinical end-users
2. Researchers working in collaboration with clinicians to build, validate, and improve the classification solution

Objectives: The explanation method

1. Should **highlight the relevant input regions** for the classifier's decision (i.e., **attributions**).
2. Show how these regions should be changed to produce a different classification decision while remaining in the data distribution (i.e., **counterfactual**).
3. While addressing the local explanation for each input, the explanation should also **point out global patterns, and possible biases** learned by the classifier.
4. Should be **model-agnostic** (as much as possible) to inspect and analyze black-box solutions (e.g., partners or competitors) with no access to the internal structures of the classifier.

To achieve these objectives, we rely on a image generation perspective. In medical image problems, and in particular for pathology detection, clinicians search for clinical signs to describe the images and make their diagnosis (e.g., tear in the menisci, opacity in the thorax or tumor tissues in the brain). The classification depends on the presence or absence of these specific patterns. We assume that the model's decision also depends on the presence or absence of some specific patterns (e.g., clinical, correlated or confounding signs). To change the model's decision, our explanation should transform the impacting input regions, e.g., generating or removing tumor in the brain. However, to generate such structures while keeping the resulting input within the distribution of real images, we cannot use synthetic perturbations. our explanation method requires access to a database to learn how to generate these counterfactual patterns.

Then, we formulate our visual explanation method as a generation process relying on a database and the trained deep classifier at stake. Let χ denote the space of real images. We define the visual explanation $\mathcal{E}$ of a classifier f on $x \in \chi$ as the function $\mathcal{I}_f$ of two generated images $g_s^\star(x)$ and $g_c^\star(x)$. $g_s^\star(x)$, the stable generation, is built to be classified as x; $g_c^\star(x)$, the counterfactual generation, is built to be classified in a different class and to belong to the distribution of real images from this different class. As such, $g_c^\star$ and $g_s^\star$ clearly depend on f. Since this will always be the case, we omit f in their notation. The

visual explanation $\mathcal{E}$ then reads:

$$\mathcal{E}(x) = \mathcal{I}_f(g_s^\star(x), g_c^\star(x)) \tag{3.1}$$

To better understand how the visual explanation is built and the different properties it should satisfy, $\mathcal{I}_f$ could be set in its simplest form: $\mathcal{I}_f : \ (x, y) \longrightarrow |x - y|$. $\mathcal{E}$ is defined as the pixel difference between the two generated images and now reads:

$$\mathcal{E}(x) = |g_s^\star(x) - g_c^\star(x)| \tag{3.2}$$

Properties: To compel the visual explanation method to satisfy the objectives stated above, $\mathcal{E}$ should capture relevant, regular, consistent, and ordered information from x, impacting the decision of the classifier f. This translates into the 3 following properties on the generators $g_s^\star$ and $g_c^\star$ (illustrated in Figure 3.1):

1. **Relevance**
2. **Regularity**
3. **Realism**

And one property explicitly on the visual explanation $\mathcal{E}$ (and more specifically impacting the term $\mathcal{I}_f$): it should be **Ordered by importance**.

We describe these properties in the following.

3.1 Properties

3.1.1 Relevance

The visual explanation $\mathcal{E}$ should highlight the important and impacting input regions for f. Then, $g_s^\star(x)$ and $g_c^\star(x)$ should only differ in regions that are relevant to the classification of x by f. If these regions or input features are replaced or removed [1], they impact the classifier's decision.

3.1.2 Regularity

To ensure the relevance property, both generation processes should be comparable to avoid differences independent from the classifier f. Especially, residual noise imputable to the generation process should be minimized. We adopt a specific model to illustrate the generation process error and to support this property:

Additive Image Independent Model (AIIM). *For both generators, every generated image is the sum of a term containing all the relevant information for f and a reconstruction error which is independent from x: $g_s(x) = x_f^s + e_{g_s}$ and $g_c(x) = x_f^c + e_{g_c}$. Visual explanation* (3.1) *then reads: $\mathcal{E}(x) = |x_f^s - x_f^c + e_{g_s} - e_{g_c}|$.*

In order to capture only the relevant information for f, we aim to get rid of the generation errors $e_{g_s} - e_{g_c}$ in the explanation definition $\mathcal{E}$ above. If the two generation processes are comparable, so should be their generation errors i.e. $e_{g_s} \approx e_{g_c}$. Note that with this formulation and given the property of Regularity, the stable generation term $g_s^\star(x)$ is the

[1] In the specific case of medical image, we cannot replace or remove input regions just like that, using synthetic perturbations (e.g., black or white pixels, noise, blur). These transformations should respect the realism property described in 3.1.3.

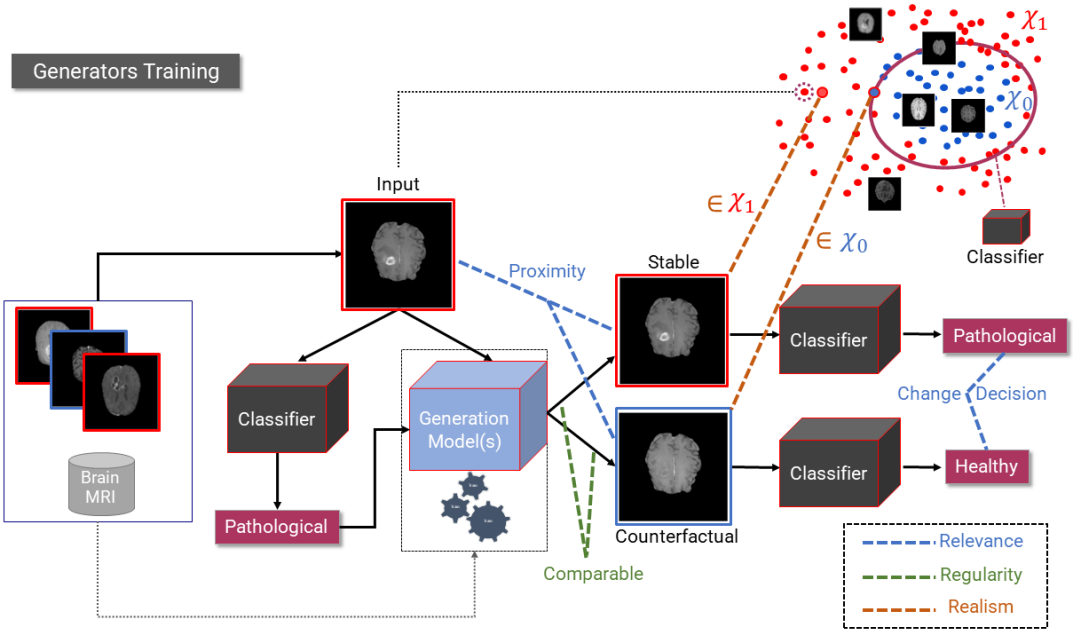


Figure 3.1: Enforcing properties during generators training. Illustration in the case of a binary classification task. Given a trained classifier and an image database (on which the classifier is supposed to work), for each input, the generator models (in light blue), i.e., $g_s^\star(x)$ and $g_c^\star(x)$ respectively learn to produce a stable and a counterfactual image. **Enforcing Relevance** (blue dotted line): The counterfactual (resp. stable) generation should be classified in a different (resp. the same) class and remain very close to the input. **Enforcing Regularity** (green dotted line): The generation of the stable and counterfactual images are comparable. It eliminates errors imputable to the generation process (independent from the classifier). In the top right-hand corner, we illustrate the space of real images. Red and blue points represent respectively real pathological and healthy images. The violet circle shows the classification boundary obtained by the classifier; the inside (resp. the outside), denoted χ_0 (resp. χ_1), are images being predicted as healthy (resp. pathological). **Enforcing Realism** (orange dotted line): The counterfactual generation should belong to the distribution of real images classified in the class opposite to the input (e.g., healthy in the example). Similar ideas for the stable image.

reconstruction of the input x; and thus should be almost equal to the input image x within one generation error. Without $g_s^\star(x)$, the visual explanation becomes $\mathcal{E}(x) = |x - g_c^\star(x)| = |x - (x_f^c + e_{g_c})|$ (under the AIIM), and a residual generation error would remain. We illustrate the impact of the regularity in Figure 3.2 comparing visual explanation results from the left of the figure against the other cases.

3.1.3 Realism

The generations $g_s^\star(x)$ and $g_c^\star(x)$ should be realistic in the sense that they should belong to the distribution of real images of the problem at stake. This property is essential to avoid (i) adversarial generated artifacts (typical of adversarial attacks [Goodfellow 15, Madry 18]); (ii) synthetic perturbations that would produce images completely outside the distribution of real images. It also reveals distribution-specific patterns influencing f only visible if generations and transformations are coherent with the distribution of real images. Moreover, f being a high dimensional function, this property also enforces the

study of f only on the subset of images it is expected to work (similar to those it has been trained), not on all possible images. Compared with natural image problems where the objects of interest and the global content of the input (e.g., background, foreground) vary significantly in the database, it is less the case of classification problems in medical images, especially for objects/pathology detection. Indeed, for pathology detection in chest X-Rays, all images in the database display similar structures, e.g., a dark background, the thorax in the center of the image, and quite similar for all patients. In this case, synthetic perturbations have more impact on the generated image, which does not belong to the distribution of real images. It damages the visual explanation because we can not assess whether the perturbation or the highlighted regions impact the classifier f. The bottom cases of Figure 3.2 show the impact of enforcing realism.

3.1.4 Order

The visual explanation values should translate into importance values for f at the pixel level or any higher scale, i.e., The regions with higher values in the explanation map should be the most relevant for the classifier f (see the illustration in the middle of Figure 3.2). Indeed, a single region or several regions can impact the classifier's decision. The visual explanation should highlight the relative importance of each region or sub-regions, e.g., up to the pixel level.

In the following sections, we first introduce the general optimization problem of the stable and the counterfactual generators $g_s^\star$ and $g_c^\star$ for a binary classification task. Then, we adapt the formulation for a multi-classification problem.

3.2 Binary classification

Here we aim to learn a function f (i.e., binary classifier) mapping each element x from the space of real images χ to the classification space $\mathcal{Y} = \{0, 1\}$. In practice, the function f rather maps χ to a continuous space $\mathcal{Y}^*$ – often equals to $[0, 1]$ or $\mathbb{R}$ depending of the final layer of f (e.g. logits, sigmoid or softmax functions). In the following, we denote $c_f(x) \in \{0, 1\}$ the class of x predicted by f and obtained from $f(x)$ by a threshold τ.
Let τ be the threshold to binarise the output of f ($\tau \in [0, 1]$ if $\forall x \in \chi, f(x) \in [0, 1]$). The set χ of real images may then be partitioned into two sets, $\chi_0 = \{x \in \chi \,|\, f(x) < \tau\}$ (images classified as class **0**) and $\chi_1 = \{x \in \chi \,|\, f(x) \geq \tau\}$ (images classified as class **1**). To summarise Relevance, Regularity, and Realism conditions, $g_s^\star$ and $g_c^\star$ may be searched as a solution couple to the optimization problem

$$(g_s^\star, g_c^\star) = \underset{g_s, g_c}{\operatorname{argmin}} \overbrace{\Big[\underbrace{r_g(g_s, g_c)}_{\textbf{Regularity}} + \mathbb{E}_x \left(d_s(x, g_s(x)) + d_c(x, g_c(x)) \right) \Big]}^{\textbf{Relevance}} \quad (3.3)$$

$$\underbrace{\text{s.t.} \left\{ \begin{array}{l} g_c(\chi_0) \subset \chi_1, g_c(\chi_1) \subset \chi_0 \\ g_s(\chi_0) \subset \chi_0, g_s(\chi_1) \subset \chi_1 \end{array} \right\}}_{\textbf{Realism}}$$

This problem is dependent on classifier f (through sets χ_0 and χ_1) and a database of images over which the expectation $\mathbb{E}$ is taken. Functions (d_c, d_s) are distances measuring the proximity of image x to each of its generated counter-parts i.e. $g_c(x)$ and $g_s(x)$.

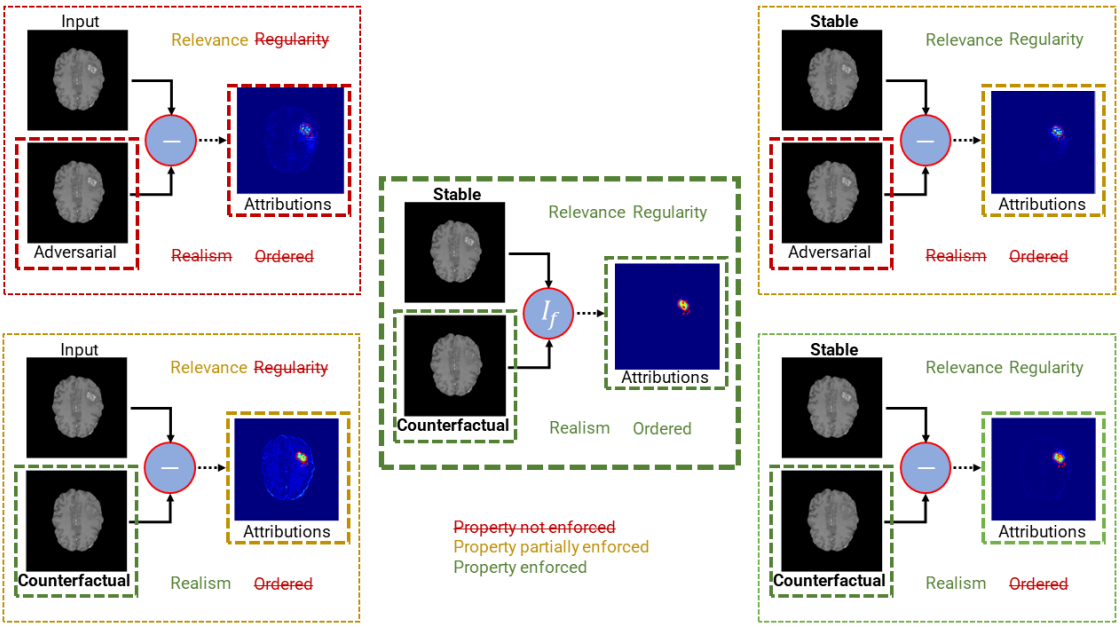


Figure 3.2: Illustrations of the properties impact on the visual explanation. The global objective is to produce visual explanations (counterfactual and attribution) satisfying the formulation objectives (all green) through enforcing the different properties. ***Top-Left: Relevance only***: we only enforce that $g_c^\star$ generates images classified in the class opposite to the input and remain close to it. Without Regularity, many irrelevant details (i.e. reconstruction errors) appear in the attribution map, affecting the Relevance property. Without realism, the method generates an adversarial image rather than a counterfactual. Differences with the input are either difficult to perceive or reveal unnatural artifacts. This generated image can not translate relevant patterns learned by the classifier. We do not enforce the correlation between the explanation map values and the importance for the classifier (i.e., Not ordered by importance) as the explanation map is the difference map between the input and the generated image. ***Top-Right: Relevance and Regularity***: Both $g_s^\star$ and $g_c^\star$ are optimized. The reconstruction errors are mainly removed (improving Relevance). Similar observations as for *Top-Left* for Realism and Ordered properties. ***Bottom-Left: Relevance and Realism***: As for the case above, removing the Regularity increases reconstruction errors when computing the difference between the two images. With realism, the counterfactual generation changes the texture of the tumor tissue into healthy tissue. ***Bottom-Right: Relevance, Regularity, and Realism***: Both the counterfactual and the attribution map are relevant. The three properties strongly reduce residual errors in the attributions (improving the Relevance). ***Center: All properties***: By enforcing that attribution values correlate with importance for the classifier, the method further improves the properties of Relevance and Regularity (compared to *Bottom-Right*)

Function r_g aims to penalize errors inherited from the generation processes and irrelevant to f.

3.3 Multi-classification

This section proposes adapting the binary general problem (3.3) for a multi-classification setting. The classifier f is expected to output a vector of dimension equal to the number of classes. The predicted class for an image x is then given by $\arg\max f(x)$. Different strategies have been proposed to explain the classifier's decision in a multi-class setting visually. Saliency [Simonyan 14, Sundararajan 17] and class activation maps [Zhou 16a, Selvaraju 17] can generate visual explanation for each class on a single input; but they often provide similar attribution regardless of the class (highlighting the edges or the main object in natural images). The user may decide to confront several explanations, given the proximity of the model's decision to other classes. Perturbation methods [Fong 17, Dabkowski 17] suppress the prediction score instead and so only focus on the decision disregarding all other possibilities. Adversarial explanations [Woods 19, Elliott 19] rather produce two types of visual explanations: targeted and untargeted. Targeted explanations show what input regions need to be changed (and how) to make the classifier predict another target class. The untargeted version is closer to perturbation approaches and aims to suppress the prediction score by enforcing the model to predict its second choice. Similarly, our approach produces targeted and untargeted visual explanations in the multi-classification setting.

3.3.1 Untargeted: Explain one class against All others

In this case, visual explanation highlights important regions when f predicts a particular class i instead of any other. For instance, in chest X-rays analysis, the method would highlight regions influencing the prediction 'pneumonia' against all other pathologies (e.g., the combination of cardiomegaly, emphysema, pneumothorax, effusion, atelectasis, nodules) The visual explanation method exposed for the binary case can then be applied. The optimization problem to solve for each predicted class i reads as (3.3) after replacing

$$\begin{array}{rcl} \chi_0 & \rightarrow & \chi_i = \{x \in \chi \,|\, \arg\max f(x) = i\} \\ \chi_1 & \rightarrow & \bar{\chi_i} = \{x \in \chi \,|\, \arg\max f(x) \neq i\} \end{array} \tag{3.4}$$

Although it may seem expensive to train one visual explanation for each class i, the number of classes in medical classification problems is generally limited (e.g., from 2 to roughly 20).

3.3.2 Untargeted: Explain one class against the closest one

As introduced by adversarial explanation techniques (see above), another approach is to generate an explanation map pointing out what regions need to be changed to make the classifier predicts its second choice j. In this setting, for each couple (i, j), the binary visual explanation methods can be applied by replacing in (3.3):

$$\begin{array}{rcl} \chi_0 & \rightarrow & \chi_i = \{x \in \chi \,|\, \arg\max f(x) = i\} \\ \chi_1 & \rightarrow & \chi_j = \{x \in \chi \,|\, \arg\max f(x) = j\} \end{array} \tag{3.5}$$

where the multi-classification model f satisfies $f = [f_k]_{k \in [\![1\ ;\ C]\!]}$ (C the number of classes); and $j = \underset{k \neq i}{\arg\max} f(x)$.

3.3.3 Targeted: Explain each class against Each other

Another perspective is to produce a visual explanation reflecting the attribution of class i instead of another specific target class j ('pneumonia' instead of 'cardiomegaly'). Similar binary visual explanation methods can again be applied with the same adaptation as in subsection 3.3.2, except that j could be any target class:

$$\begin{array}{rcl} \chi_0 & \rightarrow & \chi_i = \{x \in \chi \,|\, \arg\max f(x) = i\} \\ \chi_1 & \rightarrow & \chi_j = \{x \in \chi \,|\, \arg\max f(x) = j \neq i\} \end{array} \tag{3.6}$$

In the following chapters, we propose embodiments of the general formulation (3.3), progressively integrating the different properties introduced in 3.1. From chapter 4 to chapter 6, we progressively tackle the conditions of **Relevance**, **Regularity**, **Realism** and **Order**.
We first describe the method in the binary case; then, we propose an adaptation to the multi-classification setting. It may be easier to answer multi-binary questions in medical image classification problems rather than all together.

4

Embodiments 1: Adversarial Explanation

In this chapter, we study the properties of **Relevance** and **Regularity** and how they are related. We propose embodiments using adversarial generations. We presented, then published the following work at the International Conference on Pattern Recognition (ICPR 2020).

In Sections 4.1 and 4.2, we introduce the embodiments and the corresponding optimization frameworks in the case of a binary classification problem. An adaptation to the multi-classification settings is given in Section 4.3. In this chapter, we refer to $g_c^\star(x)$ as an adversarial generation rather than a counterfactual generation as the Realism condition is omitted; it better aligns with state-of-the-art.

4.1 Adversarial Generation (AGen)

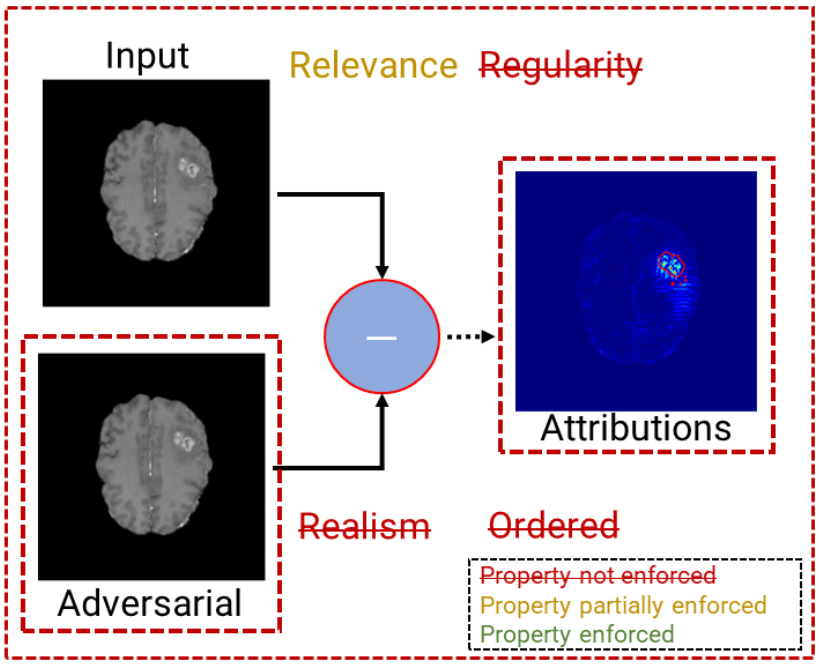


Figure 4.1: Illustrations of the properties impact on the AGen visual explanation. The global objective is to produce visual explanations (counterfactual and attribution) satisfying the formulation objectives (all green) through enforcing the different properties. ***Enforced properties: Relevance only***: We only enforce that $g_c^\star$ generates images classified in the class opposite to the input and remain close to it. Without Regularity, many irrelevant details (i.e. reconstruction errors) appear in the attribution map, affecting the Relevance property. Without realism, the method generates an adversarial image rather than a counterfactual. Differences with the input are either difficult to perceive or reveal unnatural artifacts. This generated image can not translate relevant patterns learned by the classifier. We do not enforce the correlation between the explanation map values and the importance for the classifier (i.e., Not ordered by importance) as the explanation map is the difference map between the input and the generated image.

4.1.1 Relevance via adversarial attack

A naive approach to reaching the Relevance property is to consider a unique generative function $g_c^\star$. $g_c^\star$ learns to produce an adversarial image classified oppositely as the input x by the classifier f and remaining close to x (in the sense of a given L_p norm). In this formulation we do not learn a stable generator $g_s^\star$ i.e. Here, $g_s^\star = Id : x \longrightarrow x$. We aim to generate adversarial images that attack input images only in relevant regions for f. Figure 4.1 (extracted from Figure 3.2) illustrates this approach in term of formulation goals achieved and properties enforced. To produce an attack consistent with a whole database and a visual explanation method suited for real-time situations, we train the adversarial generator $g_c^\star$ on a database. In practice, we combine the idea of the trainable masking model from [Dabkowski 17] with adversarial perturbations for visual explanations of [Elliott 19]. Our generator thus learns to produce images close to the input that change

the classifier's decision (i.e., adversarial generation idea from [Elliott 19]) without most constraint from perturbation mask approaches [Dabkowski 17] (i.e., mask regularization and perturbation choice). Inversely, by training the adversarial generator on a database (i.e., [Dabkowski 17]), adversarial images are generated on the fly at test time (compared to [Elliott 19]) and capture impacting features present in the database, i.e., not input specific only. It acts as an implicit regularization of the adversarial generation. The difference between the input image and its generated adversary gives the visual explanation map. For an input image x, we define the "naive" visual explanation as

$$\mathcal{E}(x) = |x - g_c^\star(x)| \tag{4.1}$$

In this formulation, we omit the terms d_s and r_g from Equation (3.3). The distance d_c enforcing the proximity between the generated adversarial image and the input is explicitly minimized, setting

$$d_c(x, g_c(x)) = ||x - g_c(x)||_{1,2} \tag{4.2}$$

where $||.||_{1,2} = \frac{1}{2}(||.||_1 + ||.||_2)$. The adversarial generative model $g_c^\star$ is obtained via a training process with the goal of "fooling" the classifier f while producing an image "very close" to x, written as follows:

$$g_c^\star = \underset{\substack{g_c \\ \text{s.t. } f(g_c(x))=1-c_f(x)}}{\operatorname{argmin}} \mathbb{E}_x \left[\|x - g_c(x)\|_{1,2} \right] \tag{4.3}$$

The mean value is taken over a training data set. Note that we also replace the conditions on distributions from Equation (3.3) with a constraint on the classification of $g_c(x)$ by f i.e., we remove the Realism condition.

4.1.2 Optimization and objective function

We can solve the previous optimization problem (4.3) by minimizing

$$\min_{g_c} \left[\lambda_f^c L_f^c(x, g_c) + \lambda_d^c L_d^c(x, g_c) \right] \tag{4.4}$$

where L_f^c enforces the adversarial generation $g_c^\star(x)$ to be classified oppositely by f (i.e., $1 - c_f(x)$); L_d^c minimizes the distance between the input x and its generated adversary; it is directly equal to $d_c(x, g_c(x)) = ||x - g_c(x)||_{1,2}$. The parameters λ_f^c and $\lambda_d^c \in \mathbb{R}^+$ balance the importance of each term. We elaborate more on the term L_f^c in Section 4.2, which builds on this naive approach. The optimization framework is illustrated in Figure 4.2.

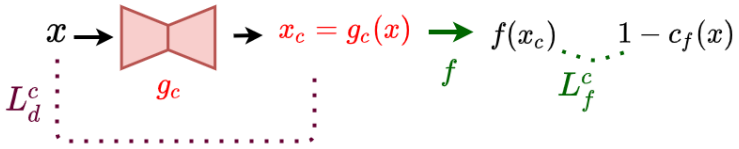


Figure 4.2: Overview of AGen. A training step of g_c. The input image x is given to the adversarial generator g_c (black arrow), which produces an adversarial image x_c. This adversarial image is enforced (L_f^c) to be classified in the opposite class $1 - c_f(x)$ by f, while being close (L_d^c) to x.

Generating a visual explanation using (4.1) and (4.3) effectively counterbalances drawbacks of [Dabkowski 17] as the method is no longer dependent on the choice of a perturbation function since the adversarial sample "learns" the perturbation; and of [Elliott 19] as it does not iterate over inputs at test time which is more suited for clinical routine and regularizes the attack over a database. Nevertheless, despite the regularization expected from the learning process, visual explanations are often corrupted by noise, highlighting regions that clearly should not impact the classifier's decision (see the attribution map in Figure 4.1).

4.2 Stable - Adversarial Generation (SAGen)

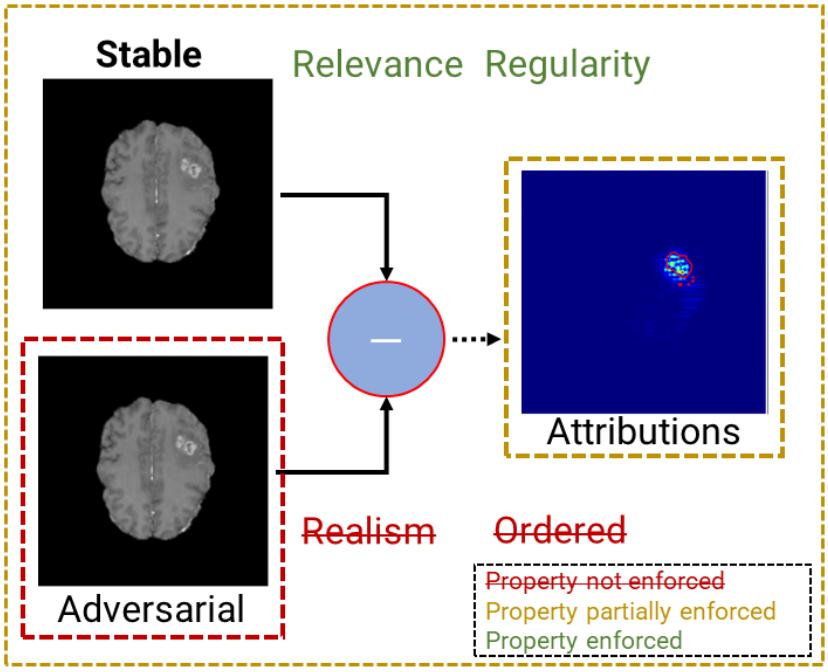


Figure 4.3: Illustrations of the properties impact on the SAGen visual explanation. The global objective is to produce visual explanations (counterfactual and attribution) satisfying the formulation objectives (all green) through enforcing the different properties. ***Enforced properties: Relevance and Regularity*** : Both $g_s^\star$ and $g_c^\star$ are optimized. We enforce that $g_c^\star$ (resp. $g_s^\star$) generates images classified in the class opposite (resp. equal) to the input and remain close to it. Using the stable image improves the regularity, and therefore the reconstruction errors are mainly removed (improving Relevance). Without realism, the method generates an adversarial image rather than a counterfactual. Differences with the input are either difficult to perceive or reveal unnatural artifacts. This generated image can not translate relevant patterns learned by the classifier. We do not enforce the correlation between the explanation map values and the importance for the classifier (i.e., Not ordered by importance) as the explanation map is the difference map between the input and the generated image.

Why does the "naive" formulation generate incoherent visual explanations? As introduced in the AIIM in chapter 3, we argue that the flaw resides in the explanation definition as expressed in Equation (4.1). Comparing the original image with its generated adversarial sample exposes the method to a risk of reconstruction errors. Some details of the original image can be absent from the generated adversarial sample and vice-versa. As introduced in 3.1.2, we propose to model $g_c(x) = x_f^c + e_{g_c}$, where x_f^c contains all the relevant information for f and e_{g_c} defines the reconstruction error. However, these details are not discriminating for the classifier in the sense that their sole presence would not change the classification prediction. We can also say that the adversarial image belongs to the target space of $g_c^\star(\chi)$ which is different from the space of input images (χ) (e.g., the space of generated images by a deep generative model against the space of medical images produced by medical machines). The comparison between x and $g_c^\star(x)$ inherits from the differences between χ, and $g_c^\star(\chi)$ and these differences are not explicitly related to the explanation

problem by Equation (4.3).

4.2.1 Introduction of a stable generation

Since we do not have control over the input image space χ, we propose to mitigate the reconstruction risk by defining the visual explanation as the difference between the generated adversary $g_c^\star(x)$ and the closest element to x in the generation space on which f returns the same value as x. Here, we introduced the stable[1] generation $g_s^\star(x)$ from the general formulation of chapter 3. $g_s^\star(x)$ is the function mapping input images to their stable counterparts. The rationale is to reduce the reconstruction error so that $\mathcal{E}$ only contains values related to the classifiers' decision and reads

$$\mathcal{E}(x) = |g_s^\star(x) - g_c^\star(x)| \tag{4.5}$$

Figure 4.3 (extracted from Figure 3.2) illustrates SAGen approach in term of formulation goals achieved and properties enforced. This optimization problem of this embodiment is close to the one defined in Equation (3.3), but only retains relevance and regularity properties. we omit the realism condition; the generations are not encouraged to belong to the distribution of real images but should rather satisfy classification constraints $f(g_s^\star(x)) = c_f(x)$ and $f(g_c^\star(x)) = 1 - c_f(x)$ (similarly as the naive adversarial version in Section 4.1). The general optimization problem (3.3) becomes:

$$(g_s^\star, g_c^\star) = \underset{g_s, g_c}{\operatorname{argmin}} \Big[r_g(g_s, g_c) + \mathbb{E}_x \left(d_s(x, g_s(x)) + d_c(x, g_c(x)) \right) \Big] \tag{4.6}$$

$$\text{s.t.} \left\{ \begin{array}{l} f(g_c(x)) = 1 - c_f(x) \\ f(g_s(x)) = c_f(x) \end{array} \right\}$$

4.2.2 Weaker formulation: Objective function

Compared with the naive formulation (4.3), the optimization problem (4.6) can not be minimized directly. We need to define the function r_g that penalizes reconstruction errors inherited from the generation process. In our setting, the generative models g_c and g_s are neural networks. Minimizing r_g consists in considering the same architecture for the two networks and enforcing their weights to be close in the L_p sense. Under the AIIM, $g_s(x) = x_f^s + e_{g_s}$ and $g_c(x) = x_f^c + e_{g_c}$. The visual explanation is $\mathcal{E}(x) = |x_f^s - x_f^c + e_{g_s} - e_{g_c}|$. Since neural networks are completely defined by their architecture and their weights, such choice for r_g minimizes $e_{g_s} - e_{g_c}$.
We propose to solve a weak formulation of the previous constrained optimization problem (4.6). In the following, we detail the different terms of this new optimization problem.

Adversarial Constraint: $\mathbf{f(g_c(x)) = 1 - c_f(x)}$-
For a real input image $x \in \chi$, $g_c(x)$ should be classified in the opposite class $(1 - c_f(x))$ by f. We thus introduce the standard classification term:

$$L_f^c(x, g_c) = \mathbb{E}_{x \in \chi} L_{bce}(1 - c_f(x), f(g_c(x))) \tag{4.7}$$

Where L_{bce} is the binary cross-entropy loss function, a common choice to optimize a binary classifier and thus attack it. This term constitutes an adversarial attack on classifier f.

[1] In our work [Charachon 20], the stable generation is defined as a similar image. We have renamed this term to be more consistent with the general formulation from chapter 3

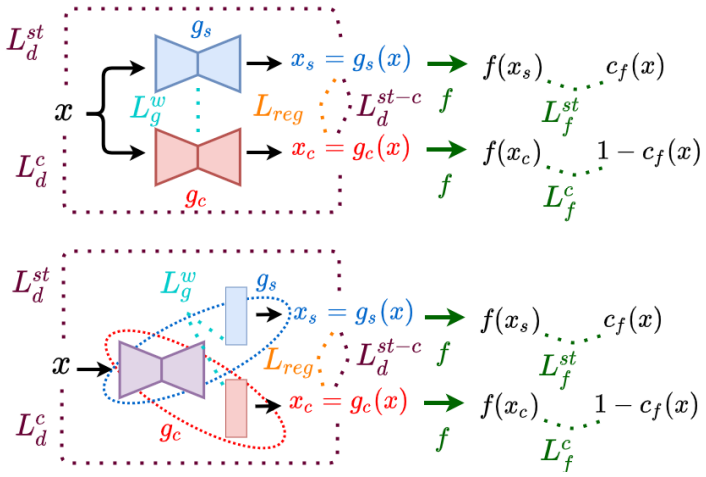


Figure 4.4: Overview of Duo SAGen and Single SAGen. Top: Duo SAGen, two generative models with the same architectures, are used to define g_s and g_c. Bottom: Single SAGen. The two generative models share a common network; they only differ in the final layers (but have the same architecture). In both cases, the figure describes a training step of both g_s and g_c. The input image x is given to both the stable and the adversarial generator (black arrow), producing a stable image x_s and an adversarial image x_c. The stable image is enforced (L_f^{st}) to be classified as x and be very close to x (L_d^{st}); while the adversarial image is enforced (L_f^c) to be classified in the opposite class, and still being close (L_d^c) to x. L_g^w encourages the weights of the two networks (or unshared part of the networks) to be close; L_d^{st-c} compels the outputs of the two generators to be close while L_{reg} regularizes their differences.

Stable proximity: $d_s(x, g_s(x))$-
The proximity d_s (in Equation (4.6)) between the input x and the stable generation $g_s(x)$ is explicitly enforced by minimizing:

$$L_d^{st}(x, g_c) = \mathbb{E}_{x \in \chi} \|x - g_s(x)\|_{1,2} \tag{4.8}$$

Using the average of L_1 and L_2 distances ($||\cdot||_{1,2} = (||\cdot||_1 + ||\cdot||_2) * 1/2$) produces slightly better results in our experiments. Additionally, to ensure the classification of $g_s(x)$ to be equal to $c_f(x)$, we also minimize:

$$L_f^{st}(x, g_s) = \mathbb{E}_{x \in \chi} L_{bce}(c_f(x), f(g_s(x))) \tag{4.9}$$

Explicit Adversarial proximity: $d_c(x, g_c(x))$-
The distance d_c between the input x and the adversarial generation $g_c(x)$ is explicitly minimized as in the naive formulation (4.3) by:

$$L_d^c(x, g_c) = \mathbb{E}_{x \in \chi} \|x - g_c(x)\|_{1,2} \tag{4.10}$$

We also add a term to minimize the distance between the two generations g_s and g_c, which increases the relevance condition by reducing differences between outputs of the two generators:

$$L_d^{st-c}(x, g_s, g_c) = \mathbb{E}_{x \in \chi} \|g_s(x) - g_c(x)\|_{1,2} \tag{4.11}$$

Combining L_1 and L_2 norms to enforce similarity between $g_s(x)$ and $g_a(x)$ also produces better results experimentally (as in [Zhang 19b]). Compared with L_d^{st} or L_d^c where the

choice of the norm (between L_1, L_2 or a combination) has not a significant impact; here, the norm L_1 is essential to allow sparse but intense differences between the adversarial and the stable image.

Weight penalization: $r_g(g_s, g_c)$-
To minimize the error inherited from the generation process, and encourage the two generators g_s and g_c to produce similar generation errors, we use a measure of the distance between the weights of the two generators. In the particular case where both g_s and g_c are neural networks (parameterized by w_s and w_c respectively), first, we use the same architecture for the two models, then we aim to minimize the following metric

$$L_g^w(g_s(.,w_s), g_c(.,w_c)) = \sum_k \left\| w_s^k - w_c^k \right\|_2 \tag{4.12}$$

Note that for this embodiment, other choices on g_s and g_c can be made (see in Section 8.1, the counterfactual generations without the realism constraint).

Regularization- In addition to the terms of (4.6), L_{reg} acts on the difference $(g_s(x) - g_a(x))$ to enforce regularity (here in the sense of smoothness). It improves the relevance and the regularity properties by minimizing local reconstruction errors, favoring proximity between the two generators, and by regularizing the explanation map $\mathcal{E}$. L_{reg} is defined as the total variation applied on the difference $g_s(x) - g_a(x)$:

$$L_{reg}(x, g_s(x), g_c(x)) = \sum_{i \in \mathbb{R}^d} \left\| \nabla \left({g_s}^i(x) - {g_c}^i(x) \right) \right\|_2 \tag{4.13}$$

where d is the dimension of the output space of the generators.
In summary, we search for both similar and adversarial generators as minimizers of the following problem

$$\min_{g_s, g_c} \left[\begin{array}{l} \lambda_f^s L_f^{st}(x, g_c) + \lambda_f^c L_f^c(x, g_c) + \\ \lambda_d^{st} L_d^{st}(x, g_s) + \lambda_d^c L_d^c(x, g_c) + \lambda_d^{st-c} L_d^{st-c}(x, g_s, g_c) + \\ \lambda_g^w L_g^w(g_s, g_c) + \lambda_{reg} L_{reg}(x, g_s, g_c) \end{array} \right] \tag{4.14}$$

The parameters $\lambda_i \in \mathbb{R}$ controls the relative importance of each term in the global objective function (4.14). We show an overview of the optimization framework in Figure 4.4 for two choices of generative models g_s and g_c.

4.3 Multi-classification setting

The two embodiments introduced in Sections 4.1 and 4.2 can be adapted to the multi-classification problem. We just have to respectively change the classification terms in Equations (4.4) and (4.14) i.e. L_f^{st} and L_f^c. The classifier f outputs a dimension vector equal to the number of classes. The predicted class for an image x is then given by $\arg\max f(x)$).
Different strategies could be used for modifying the terms L_f^{st} and L_f^c. First, it depends on the type of visual explanation we are looking for: targeted or untargeted (see Section 3.3). In the targeted version, at each step, we sample a classification target for the adversarial generation; we compute the cross entropy loss function for the stable and the adversarial images enforcing them to be classified as the input image or as the target class, respectively. In the untargeted version[2], we can either compute a cross-entropy loss function between the stable generation and the input prediction (L_f^{st}), then between the adversarial generation and the second choice of f (L_f^c); or we can adapt the CW loss [Carlini 17] used in[Elliott 19, Zhang 19b] into

$$L_f^{st}(x, g_s) = \mathbb{E}_{x\in\chi} \max(\max_{i\neq l}(f_i(g_s(x))) - f_{c_l}(g_s(x)), -\kappa) \tag{4.15}$$

and

$$L_f^{c}(x, g_c) = \mathbb{E}_{x\in\chi} \max(f_{c_l}(g_c(x)) - \max_{i\neq l}(f_{c_i}(g_c(x))), -\kappa) \tag{4.16}$$

where index l is defined by $\arg\max_i [f_{c_i}(x)]$ corresponding to the class selected by the classifier on input x. κ is a strictly positive margin.

[2]We propose an adaptation of the binary case for the untargeted explanation described in 3.3.2 because the other proposed in 3.3.1 is equivalent to the binary case.

5

Embodiments 2: Counterfactual Explanation

In chapter 4, we proposed to generate a stable image and an adversarial image (w.r.t f) to encourage both Relevance and Regularity of the visual explanation. Although producing promising results (see Chapter 8), this method faces some limitations:

1. When images of the different classes display very distinct patterns (with different sizes and intensity), the adversarial attacks fail (i.e., $f(g_c(x)) = c_f(x)$) on average on the test set, or show a weak successful attack rate.
2. The method does not capture realistic patterns of the given class, as the generated adversary is searched as the closest element to the input x in the sense of an $L_{1,2}$ norm (applied pixel-wise). The differences between the input and the adversary are almost imperceptible (see Figure 4.3 and results from Section 8.3).

There are two main reasons for these limitations:

1. The explicit minimization of d_c (distance between the input x and the counterfactual generation $g_c(x)$)compels the generated adversary to be pixel-wise close to the input and prevents intense but localized differences in relevant regions. Compared to the classic adversarial attack setting where we optimize the attack on a single image, here, the adversarial generator is optimized on a database. It regularizes the attack toward more relevant regions but weakens it, because the generator is expected to produce an adversary for all images. In the scenario described in limitation 1, combining a pixel-wise constraint on the generation and learning g_c on a database may fail.
2. No constraint enforces the adversarial generation to belong to the distribution of real images (realism property). We only encourages the classification output of $g_c(x)$ to be different from $c_f(x)$. Thus, we miss realistic patterns of the different classes. Finally, note that enforcing realism in the generation process would produce intense differences between the input and the adversary (e.g., healthy brain tissues show very different textures and intensity compared with tumors in MRI images). Using a pixel-wise constraint between x and $g_c(x)$ does not go in this direction and would even penalize it.

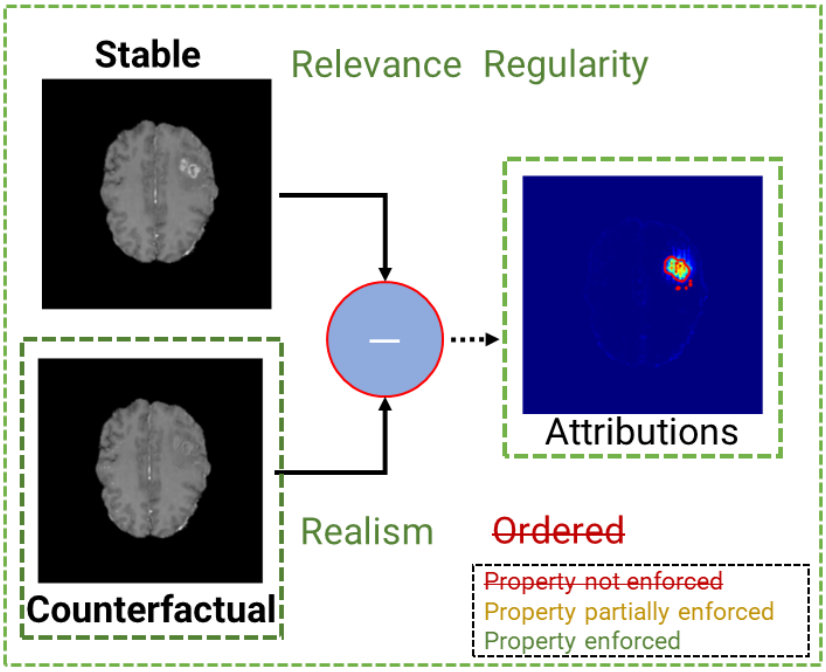


Figure 5.1: Illustrations of the properties impact on the counterfactual visual explanation. The global objective is to produce visual explanations (counterfactual and attribution) satisfying the formulation objectives (all green) through enforcing the different properties. ***Enforced properties: Relevance, Regularity and Realism***: Both $g_s^\star$ and $g_c^\star$ are optimized. We enforce that $g_c^\star$ (resp. $g_s^\star$) generates images classified in the class opposite (resp. equal) to the input and remain close to it. Using the stable image improves the regularity, and therefore the reconstruction errors are mainly removed (improving Relevance). With realism, the counterfactual generation changes the texture of the tumor tissue into healthy tissue. The three properties strongly reduce residual errors in the attributions (improving the Relevance). We do not enforce the correlation between the explanation map values and the importance for the classifier (i.e., Not ordered by importance) as the explanation map is the difference map between the input and the generated image.

In this chapter, we encourage the realism of the generations through counterfactual generation methods based on domain translation. We also reconsider the distance d_c. We propose different embodiment versions that satisfy the properties of **Relevance, Regularity, and Realism**. Figure 5.1 (extracted from Figure 3.2) summarizes our counterfactual approach, pointing out the formulation goals achieved and the properties enforced. Parts of the following work have been published in a special issue on "Explainable AI for Healthcare" in the journal Future Generation Computer System (2021).

5.1 Single Generator using Symmetry (SSyGen)

5.1.1 Built-in Regularity

In the following embodiment, we propose a built-in suppression of the generation process error under the AIIM. We force counterfactual and stable generations to be outputs of a single unique generator. To this end, we propose to derive the stable generator from the counterfactual one:

$$g_s = g_c \circ g_c = g_c^2 \tag{5.1}$$

Both generated images result from the same generation process, and their difference is less subject to purely reconstruction errors. Each application of g_c produces the same reconstruction error e_{g_c} which is suppressed in the difference $\mathcal{E}(x)$. We thus get rid of g_s and r_g terms in Equation (3.3) under the AIIM assumption.

5.1.2 Relevance via Symmetry (or Self-inversion)

In problem formulation (see Chapter 3), both generated images are expected to be as close as possible to the input image (via the minimization of d_s and d_c). However, they should be classified differently by f: the stable as x and the counterfactual oppositely. Since the stable generation belongs to the same subspace as x (χ_0 or χ_1), we naturally set

$$d_s(x, g_s(x)) = ||x - g_s(x)||_{1,2} \tag{5.2}$$

where $||.||_{1,2} = \frac{1}{2}(||.||_1 + ||.||_2)$. Using Equations (5.1) we derive from (5.2),

$$d_s(x, g_s(x)) = ||x - {g_c}^2(x)||_{1,2} \tag{5.3}$$

The explicit minimization of d_s then recalls a fundamental property of linear symmetries $s^2(x) = x$ –also referred to as involutions or self-inversion functions in linear algebra. As supported by empirical results in [Zhu 17, Shen 20b], it compels g_c to transpose elements from one classification space to the other, changing only restricted regions of the original image, in order to "easily" return ($g_c : \chi_0 \to \chi_1$ and ${g_c}^2 : \chi_1 \to \chi_0$). The distance between x and $g_c(x)$ is thus implicitly penalized and we can obviate in (3.3) the distance term d_c. The proposed embodiment of problem (3.3) is then

$$\begin{aligned} g_c^\star = \underset{g_c}{\text{argmin}} \;\; & \mathbb{E}_{x\in\chi}\left(||x - {g_c}^2(x)||_{1,2}\right) \\ \text{s.t.} \;\; & \left\{ \begin{array}{ll} g_c(\chi_0) \subset \chi_1 \,, & g_c(\chi_1) \subset \chi_0 \\ {g_c}^2(\chi_0) \subset \chi_0 \,, & {g_c}^2(\chi_1) \subset \chi_1 \end{array} \right\} \end{aligned} \tag{5.4}$$

Note that this form satisfies all Relevance, Regularity, and Realism properties stated in chapter 3. We use implicit constraints to minimize generation process errors (r_g) and the distance between the original image and the counterfactual image (d_c). Adding explicit terms such as $||x - g_c(x)||_{1,2}$ within the minimization increases the difficulty to satisfy constraints $g_c(x) \in \chi_i$ if $x \in \chi_i$ (for $i \in \{0, 1\}$), specially if elements of χ_0 and χ_1 are L_1-L_2 distant.

5.1.3 Weak Formulation

We approximate the previous constrained minimization problem (5.4) as a non-constrained minimization problem. We next describe its different terms and their relation with (5.4).

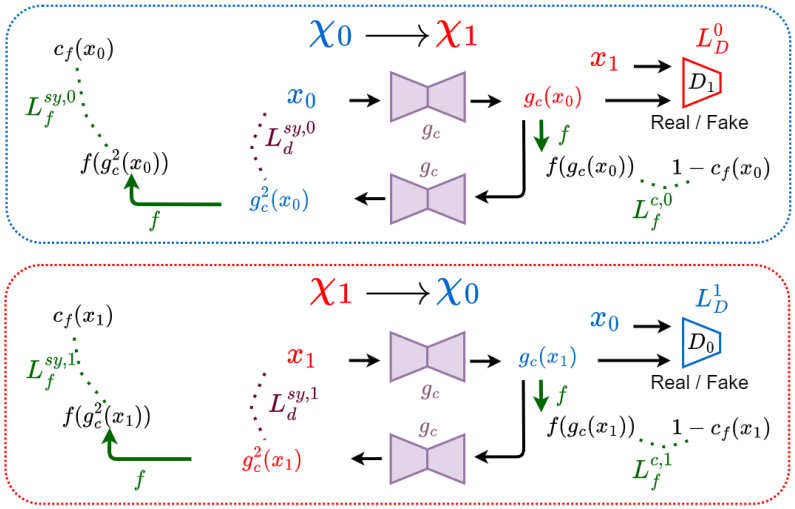


Figure 5.2: Overview of SSyGen optimization framework. Top: training step of g_c for an original image $x_0 \in \chi_0$. Bottom: training step for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) illustrated through dashed lines, are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_c (black arrow) which produces a counterfactual image $g_c(x_0)$. This generated image is enforced ($L_f^{c\,0}$) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the generated image $g_c(x_0)$ to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Symmetrical path***: $g_c(x_0)$ is then mapped back to χ_0 through g_c (black arrow starting above $g_c(x_0)$). The resulting symmetrical image ${g_c}^2(x_0)$ is enforced to be pixel-wise close to x_0 ($L_d^{sy,0}$) and classified the same way ($L_f^{sy,0}$). Similar procedures for the other transposition (**Bottom**).

The problem formulation (5.4) is an optimization problem constrained by the conditions of realism (applied on distributions) i.e.

- Counterfactual constraints: $g_c(\chi_0) \subset \chi_1$ and $g_c(\chi_1) \subset \chi_0$
- Stable constraints: ${g_c}^2(\chi_0) \subset \chi_0$ and ${g_c}^2(\chi_1) \subset \chi_1$

Although problem (5.4) is hard to solve directly, the constraints applied on distributions χ_0, and χ_1 (above) can be approximated into an unconstrained min-max optimization problem commonly used to optimize GANs [Goodfellow 14]. As in [Zhu 17, Narayanaswamy 20], we introduce two domain-specific discriminators D_0 and D_1 that enforce generated images to belong to a specific domain distribution. D_0 (resp. D_1) is in charge of distinguishing real images in χ_0 (resp. χ_1) from outputs of g_c. D_0 (resp. D_1) thus applies on $g_c(x)$ for $x \in \chi_1$ (resp. χ_0) as $g_c(x)$ should be translated to χ_0 (resp. χ_1)). Another possibility is to use a single discriminator conditioned by the domain target as in [Mirza 14, Miyato 18].
The different components of this approximated problem are now detailed.

Counterfactual Constraint: $\mathbf{g_c(\chi_0) \subset \chi_1}$ and $\mathbf{g_c(\chi_1) \subset \chi_0}$-
For a real image $x \in \chi_0$ (resp. $x \in \chi_1$), $g_c(x)$ should be classified as of class **1** (resp. class **0**) by f. We thus introduce the term:

$$L_f^c(x, g_c) = \mathbb{E}_{x \in \chi} L_{bce}(1 - c_f(x), f(g_c(x))) \tag{5.5}$$

Where L_{bce} is the binary cross-entropy loss function. This term constitutes a typical "attack" on classifier f and does not enforce generated images to belong to the distributions of "real" images in χ_0 or χ_1. To cope with this problem, we introduce a classical GAN term as in [Liu 17, Lee 18, Bass 20]

$$\begin{array}{l} L_D(x, g_c, D_0, D_1) \; = \\ \quad \mathbb{E}_{x\in\chi_0}\left[L_{bce}(1, D_0(x)) + L_{bce}(0, D_1(g_c(x)))\right] \\ \quad \mathbb{E}_{x\in\chi_1}\left[L_{bce}(1, D_1(x)) + L_{bce}(0, D_0(g_c(x)))\right] \end{array} + \tag{5.6}$$

Where D_0 and D_1 are trained to minimize it while g_c intends to maximize it.

Symmetry: $||\mathbf{x} - \mathbf{g_c}^2(\mathbf{x})||$ and $\mathbf{g_c}(\chi_\mathbf{i}) \subset \chi_\mathbf{i}$ for $\mathbf{i} \in \{0, 1\}$ -
Symmetry objectives of problem (5.4) are directly enforced by training g_c to minimize

$$L_d^{sy}(x, g_c) = \mathbb{E}_{x\in\chi}\|x - {g_c}^2(x)\|_{1,2} \tag{5.7}$$

Using the average of L_1 and L_2 distances ($||\cdot||_{1,2} = (||\cdot||_1 + ||\cdot||_2) * 1/2$) produces better results in our experiments. Additionally, to drive the classification of generated elements ${g_c}^2(x)$ towards $c_f(x)$, we add the loss term:

$$L_f^{sy}(x, g_c) = \mathbb{E}_{x\in\chi} L_{bce}(c_f(x), f({g_c}^2(x))) \tag{5.8}$$

The global min-max optimization problem then reads

$$\min_{g_c} \max_{D_0, D_1} \left[\begin{array}{l} \lambda_f^c L_f^c(x, g_c) + \lambda_D L_D(x, g_c, D_0, D_1) + \\ \lambda_f^{sy} L_f^{sy}(x, g_c) + \lambda_d^{sy} L_d^{sy}(x, g_c) \end{array} \right] \tag{5.9}$$

where parameters $\lambda_d^{sy}, \lambda_f^c, \lambda_f^{sy}$ and $\lambda_D \in \mathbb{R}^+$ control the relative importance of the different terms. In Section 7.2, we give insights into the relative importance of these parameters and how to choose them. Figure 5.2 shows an overview of the whole framework. In this Figure, we dissociate the optimization step of an input image $x \in \chi_0$ from the step for $x \in \chi_1$. We could have merged the two steps as the single symmetrical generator g_c does not depend on the input domain χ_0 or χ_1. In this setting, we should use a conditional discriminator (as [Mirza 14, Miyato 18]) rather than two domain specific discriminators. The corresponding framework is given in the Appendix: Figure A.1. First, separating the two steps improves the comparison with the following counterfactual embodiments (see the next sections); then, it better works in practice (for the binary case).
Table 5.1 shows which loss term enforces which conditions from Equation (3.3).

Table 5.1: Conditions and loss terms relationship in SSyGen. Contribution of each loss term from SSyGen (weak formulation) to encourage the three conditions (Relevance, Regularity, and Realism) from the general formulation (3.3).

	L_f^c	L_D	L_f^{sy}	L_d^{sy}
Relevance	✓		✓	✓
Regularity				✓
Realism		✓		

The experiments (see Chapter 8) show that counterfactual ($g_c^\star(x)$) and stable generations ($g_c^{\star 2}(x)$) are, in general, too close. The counterfactual generation fails to fall into the opposite distribution. Realism constraint is thus seldom satisfied, inducing a corrupted visual explanation map.

5.2 Symmetrically Conditioned Explanation (SyCE)

5.2.1 Specific counterfactual generators on χ_0 and χ_1

To weaken the too strong constraints in Equation (5.4) induced by

$$\forall x \in \chi \left\{ \begin{array}{l} (g_c(x) \in \chi_1,\ {g_c}^2(x) \in \chi_0) \ \text{ if } x \in \chi_0 \\ (g_c(x) \in \chi_0,\ {g_c}^2(x) \in \chi_1) \ \text{ if } x \in \chi_1 \end{array} \right.$$

we search for two auxiliary generators g_0 and g_1 defined on χ that have the same constraints as g_c (in Section 5.1) but only on the specific sub-spaces χ_0 and χ_1, respectively. By considering two generators for sub-tasks, we impose weaker constraints on each of them. It reads:

$$\forall x \in \chi \left\{ \begin{array}{l} (g_0(x) \in \chi_1,\ g_0^2(x) \in \chi_0) \ \text{ if } x \in \chi_0 \\ (g_1(x) \in \chi_0,\ g_1^2(x) \in \chi_1) \ \text{ if } x \in \chi_1 \end{array} \right.$$

We then redefine counterfactual and stable generations:

$$g_c(x) = \left\{ \begin{array}{l} g_0(x) \text{ if } x \in \chi_0 \\ g_1(x) \text{ if } x \in \chi_1 \end{array} \right. \quad g_s(x) = \left\{ \begin{array}{l} g_0^2(x) \text{ if } x \in \chi_0 \\ g_1^2(x) \text{ if } x \in \chi_1 \end{array} \right. \tag{5.10}$$

To explicitly minimize the distance d_s (between x and $g_s(x)$) and implicitly minimize the distance d_c (between x and $g_c(x)$), as in (5.4), we shall minimize quantities $||g_0^2(x) - x||$, $||g_1^2(x) - x||$ and $||g_c^2(x) - x||$. From (5.10) we have $g_c^2(x) = g_1 \circ g_0(x)$ and $g_c^2(x) = g_0 \circ g_1(x)$ for $x \in \chi_0$ and $x \in \chi_1$ respectively. The new embodiment of (3.3) then reads

$$\begin{array}{rl} (g_0^\star, g_1^\star) = & \underset{g_0, g_1}{\text{argmin}} \left[+ \begin{array}{l} \mathbb{E}_{x \in \chi_0} \left(||x - g_0^2(x)|| + ||x - g_1(g_0(x))|| \right) \\ \mathbb{E}_{x \in \chi_1} \left(||x - g_1^2(x)|| + ||x - g_0(g_1(x))|| \right) \end{array} \right] \\ \text{s.t.} & \left\{ \begin{array}{l} g_0(\chi_0) \subset \chi_1 \ , \ g_1(\chi_1) \subset \chi_0 \\ g_0^2(\chi_0) \subset \chi_0 \ , \ g_1^2(\chi_1) \subset \chi_1 \end{array} \right\} \end{array} \tag{5.11}$$

Figure 5.3 gives an illustration of the mappings built by $(g_0^\star, g_1^\star)$ after optimization. The expression of visual explanation is given by

$$\mathcal{E}(x) = \left\{ \begin{array}{l} |{g_0^\star}^2(x) - g_0^\star(x)| \ \text{ if } \ x \in \chi_0 \\ |{g_1^\star}^2(x) - g_1^\star(x)| \ \text{ if } \ x \in \chi_1 \end{array} \right\} \tag{5.12}$$

Problem (5.11), further referred to as SyCE, gives the best results in our experiments on binary classification problems described in Section 7. In the following section, we give details on its implementation.

5.2.2 Weak Formulation

As in Section 5.1, we approximate the constrained minimization problem (5.11) as a non constrained minimization problem. In this case, formulation (5.11) is an optimization problem constrained by the following conditions of realism :

- Counterfactual constraints: $g_0(\chi_0) \subset \chi_1$ and $g_1(\chi_1) \subset \chi_0$
- Stable constraints: $g_0^2(\chi_0) \subset \chi_0$ and $g_1^2(\chi_1) \subset \chi_1$

We approximate the optimization problem (5.11) into an unconstrained min-max optimization problem (as in Section 5.1.3). The domain-specific discriminator D_0 (resp. D_1) is in charge of distinguishing real images in χ_0 (resp. χ_1) from outputs of g_1 (resp. g_0). We detail the different components of this approximated problem in the following.

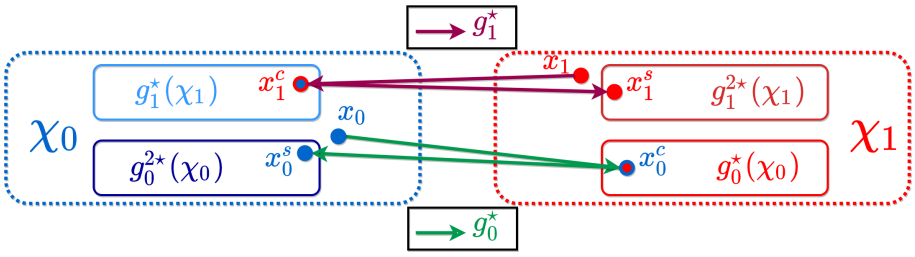


Figure 5.3: Generators mappings in SyCE. $g_0^\star$ maps each element $x_0 \in \chi_0$ to an element $x_0^c \in g_0^\star(\chi_0) \subset \chi_1$. By reapplying $g_0^\star$, given the symmetry constraint, this element $x_0^c \in g_0^\star(\chi_0)$ is mapped back to an element $x_0^s \in g_0^{\star 2}(\chi_0) \subset \chi_0$ very close to the original image x_0. $g_1^\star$ acts similarly on $x_1 \in \chi_1$.

Counterfactual Constraint: $\mathbf{g_0}(\chi_0) \subset \chi_1$ and $\mathbf{g_1}(\chi_1) \subset \chi_0$-
For a real image $x \in \chi_0$, $g_0(x)$ should be classified as of class **1**, and reciprocally, if x is a real image in χ_1, $g_1(x)$ should be classified as of class **0** by f. We thus introduce the term:

$$L_f^c(x, g_0, g_1) = \begin{array}{l} \mathbb{E}_{x\in\chi_0} L_{bce}(1 - c_f(x), f(g_0(x))) \\ \mathbb{E}_{x\in\chi_1} L_{bce}(1 - c_f(x), f(g_1(x))) \end{array} + \tag{5.13}$$

Where L_{bce} is the binary cross-entropy loss function. This term constitutes a typical "attack" on classifier f and does not enforce generated images to belong to the distributions of "real" images in χ_0 or χ_1. To cope with this problem, we use the common GAN term (as in [Liu 17, Lee 18, Bass 20])

$$\begin{array}{l} L_D(x, g_0, g_1, D_0, D_1) \ = \\ \quad \begin{array}{l} \mathbb{E}_{x\in\chi_0} \left[L_{bce}(1, D_0(x)) + L_{bce}(0, D_1(g_0(x)))\right] \\ \mathbb{E}_{x\in\chi_1} \left[L_{bce}(1, D_1(x)) + L_{bce}(0, D_0(g_1(x)))\right] \end{array} + \end{array} \tag{5.14}$$

where D_0 and D_1 are trained to minimize it while g_0 and g_1 to maximize it.

Symmetry: $||\mathbf{x} - \mathbf{g_i^2}(\mathbf{x})||$ and $\mathbf{g_i}(\chi_i) \subset \chi_i$ for $\mathbf{i} \in \{0, 1\}$ -
Symmetry objectives of problem (5.11) are directly enforced by training g_0 and g_1 to minimize

$$L_d^{sy}(x, g_0, g_1) = \mathbb{E}_{x\in\chi_0} \|x - g_0^2(x)\|_{1,2} + \mathbb{E}_{x\in\chi_1} \|x - g_1^2(x)\|_{1,2} \tag{5.15}$$

Using the average of L_1 and L_2 distances also produces slightly better results in our experiments. Additionally, to drive the classification of generated elements $g_0^2(x)$ and $g_1^2(x)$ towards $c_f(x)$, we add the loss term:

$$L_f^{sy}(x, g_0, g_1) = \begin{array}{l} \mathbb{E}_{x\in\chi_0} L_{bce}(c_f(x), f(g_0^2(x))) \\ \mathbb{E}_{x\in\chi_1} L_{bce}(c_f(x), f(g_1^2(x))) \end{array} + \tag{5.16}$$

Cyclic Consistency: $||\mathbf{x} - \mathbf{g_1}(\mathbf{g_0}(\mathbf{x}))||$ and $||\mathbf{x} - \mathbf{g_0}(\mathbf{g_1}(\mathbf{x}))||$-
Cyclic constraints (as in [Zhu 17]) are enforced by minimizing

$$L_d^{cy}(x, g_0, g_1) = \begin{array}{l} \mathbb{E}_{x\in\chi_0} \|x - g_1(g_0(x))\|_{1,2} \\ \mathbb{E}_{x\in\chi_1} \|x - g_0(g_1(x))\|_{1,2} \end{array} + \tag{5.17}$$

This term enforces a strong relation between g_0 and g_1, increases convergence speed, and represents an embodiment of the minimization of d_c in (3.3).

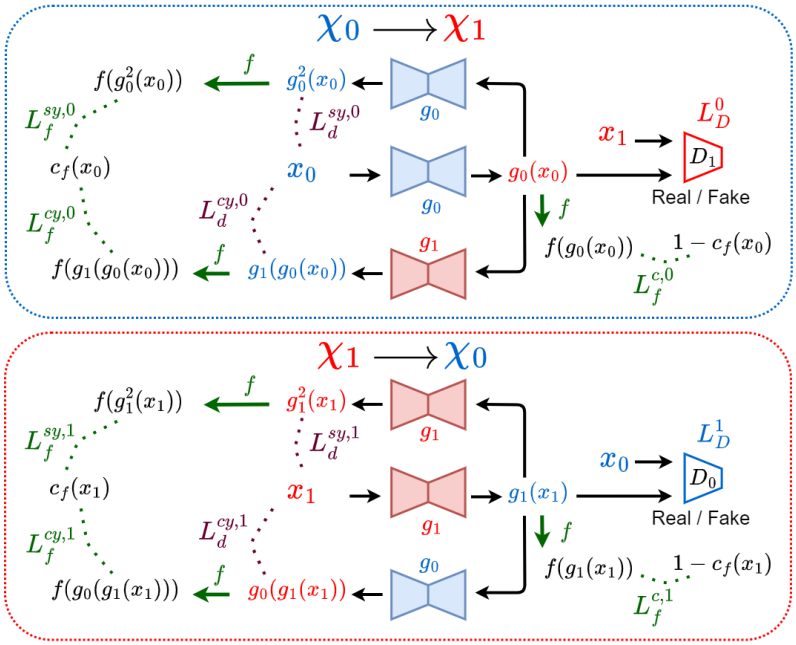


Figure 5.4: Overview of SyCE optimization framework. Top: training step of g_0 and g_1 for an original image $x_0 \in \chi_0$. Bottom: training step for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) illustrated through dashed lines, are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_0 (black arrow) which produces a counterfactual image $g_0(x_0)$. This generated image is enforced ($L_f^{c\,0}$) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the generated image $g_0(x_0)$ to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Symmetrical path***: $g_0(x_0)$ is then mapped back to χ_0 through g_0 (black arrow starting above $g_0(x_0)$). The resulting symmetrical image $g_0^2(x_0)$ is enforced to be pixel-wise close to x_0 ($L_d^{sy,0}$) and classified the same way ($L_f^{sy,0}$). ***Cyclic path***: $g_0(x_0)$ is also mapped back to χ_0 through g_1 (black arrow bellow $g_0(x_0)$). Similar to the symmetrical image, distance ($L_d^{cy,0}$) and classification ($L_f^{cy,0}$) constraints are used. Similar procedures for the other transposition (**Bottom**).

Finally, to also ensure that cyclic terms are classified like x: we add a consistency loss

$$L_f^{cy}(x, g_0, g_1) = \begin{array}{l} \mathbb{E}_{x\in\chi_0} L_{bce}(c_f(x), f(g_1(g_0(x)))) \\ \mathbb{E}_{x\in\chi_1} L_{bce}(c_f(x), f(g_0(g_1(x)))) \end{array} + \tag{5.18}$$

In experiments, we observe that L_f^{cy} has a smaller effect than $L_{f_c}^s$ yet it slightly improves the cycle consistency.
The global min-max optimization problem then reads

$$\min_{g_0,g_1} \max_{D_0,D_1} \left[\begin{array}{l} \lambda_f^c L_f^c(x, g_0, g_1) + \lambda_D L_D(x, g_0, g_1, D_0, D_1) + \\ \lambda_f^{sy} L_f^{sy}(x, g_0, g_1) + \lambda_d^{sy} L_d^{sy}(x, g_0, g_1) + \\ \lambda_f^{cy} L_f^{cy}(x, g_0, g_1) + \lambda_d^{cy} L_d^{cy}(x, g_0, g_1) \end{array} \right] \tag{5.19}$$

where parameters $\lambda_d^s, \lambda_d^{cy}, \lambda_f^c, \lambda_f^s, \lambda_f^{cy}$ and $\lambda_D \in \mathbb{R}^+$ also control the relative importance of the different terms. In Section 7.2, we give insights into the relative importance of these parameters, and we further detail how to choose them. Figure 5.4 shows an overview of

Table 5.2: Conditions and loss terms relationship in SyCE. Contribution of each loss term from SyCE (weak formulation) to encourage the three conditions (Relevance, Regularity, and Realism) from the general formulation (3.3)

	L_f^c	L_D	L_f^{sy}	L_d^{sy}	L_f^{cy}	L_d^{cy}
Relevance	✓		✓	✓	✓	✓
Regularity				✓		
Realism		✓				

the whole framework ; Table 5.2 shows which loss term enforces which conditions from Equation (3.3).

5.3 Cyclic Conditioned Explanation (CyCE)

Inspired by the CycleGAN framework used in [Narayanaswamy 20, Wolleb 20], we also propose a simpler embodiment of the general formulation (3.3). As in Equation (5.11), we introduce generative models $g_0^\star$ and $g_1^\star$ to define the conditional counterfactual generator $g_c^\star$. In contrast, we relax the formulation and define $g_s^\star$ as the identity. The visual explanation becomes

$$\mathcal{E}(x) = \left\{ \begin{array}{l} |x - g_0^\star(x)| \text{ if } x \in \chi_0 \\ |x - g_1^\star(x)| \text{ if } x \in \chi_1 \end{array} \right\} \tag{5.20}$$

In this formulation, we directly compare the counterfactual to the original image removing both constraints r_g and d_s. We trade potential reconstruction errors against better proximity of the generated elements to the counterfactual class. Figure 5.5 (extracted from Figure 3.2) illustrates CyCE properties and outcomes which differ from other counterfactual approaches.

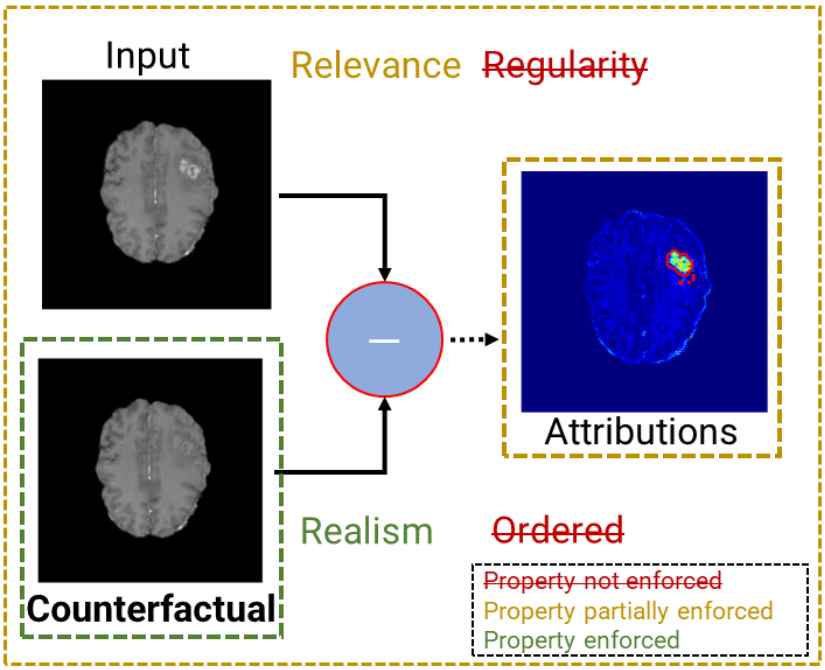


Figure 5.5: Illustrations of the properties impact on the CyCE visual explanation. The global objective is to produce visual explanations (counterfactual and attribution) satisfying the formulation objectives (all green) through enforcing the different properties. ***Enforced properties: Relevance and Realism***: Only $g_c^\star$ is optimized. We enforce that $g_c^\star$ generates images classified in the class opposite to the input and remain close to it. With realism, the counterfactual generation changes the texture of the tumor tissue into healthy tissue. Removing the Regularity increases reconstruction errors when computing the difference between the two images (and thus affect the Relevance). We do not enforce the correlation between the explanation map values and the importance for the classifier (i.e., Not ordered by importance) as the explanation map is the difference map between the input and the generated image.

Concerning d_c, the cycle consistency term encourages proximity to the input. The ap-

proximated problem reads

$$(g_0^\star, g_1^\star) = \underset{g_0, g_1}{\text{argmin}} \begin{bmatrix} \mathbb{E}_{x\in\chi_0}\left(||x - g_1(g_0(x))||\right) \\ +\mathbb{E}_{x\in\chi_1}\left(||x - g_0(g_1(x))||\right) \end{bmatrix} \quad \text{s.t. } \{g_0(\chi_0) \subset \chi_1, g_1(\chi_1) \subset \chi_0\} \tag{5.21}$$

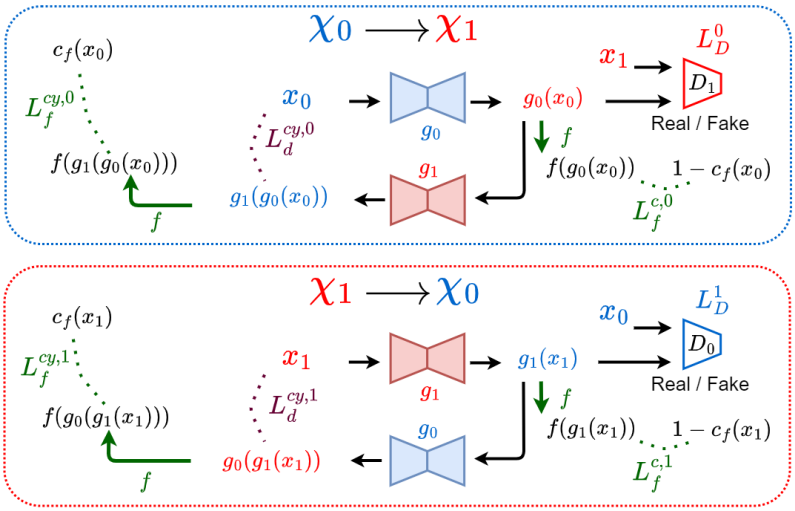


Figure 5.6: Overview of CyCE optimization framework. Top: training step of g_0 and g_1 for an original image $x_0 \in \chi_0$. Bottom: training step for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) illustrated through dashed lines, are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_0 (black arrow) which produces a counterfactual image $g_0(x_0)$. This generated image is enforced ($L_f^{c\,0}$) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the generated image $g_0(x_0)$ to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Cyclic path***: $g_0(x_0)$ is also mapped back to χ_0 through g_1 (black arrow bellow $g_0(x_0)$). Similar to the symmetrical image, distance ($L_d^{cy,0}$) and classification ($L_f^{cy,0}$) constraints are used. Similar procedures for the other transposition (**Bottom**).

In practice, the optimization framework is very close to the one presented in Section 5.2.2 and reads:

$$\min_{g_0,g_1} \max_{D_0,D_1} \begin{bmatrix} \lambda_f^c L_f^c(x, g_0, g_1) + \lambda_D L_D(x, g_0, g_1, D_0, D_1) + \\ \lambda_f^{cy} L_f^{cy}(x, g_0, g_1) + \lambda_d^{cy} L_d^{cy}(x, g_0, g_1) \end{bmatrix} \tag{5.22}$$

We only remove the terms related to symmetrical elements g_0^2 and g_1^2 i.e. L_d^{sy} and L_f^{sy} (see Figure 5.6). Table 5.3 supports the relaxation stated above in the presentation of CyCE. We remove the regularity constraint and decrease the relevance enforcement (removing the terms L_d^s and L_f^s compared to Table 5.2). This translates into poorer visual explanation results compared to SyCE, especially in terms of localization performances (see Section 8.1).

Table 5.3: Conditions and loss terms relationship in CyCE. Contribution of each loss term from CyCE to encourage the three conditions (Relevance, Regularity, and Realism) from the general formulation (3.3).

	L_f^c	L_D	L_f^{cy}	L_d^{cy}
Relevance	✓		✓	✓
Regularity				
Realism		✓		

5.4 Single conditioned generator with cycle consistency

5.4.1 Constraint relaxation through a single conditioned generator

In this embodiment, we also relax the formulation from Equation (5.4) where the constraints are too strong on the single symmetrical generator $g_c^\star$. Rather than using auxiliary generators g_0 and g_1 specific to the domain χ_0 and χ_1 respectively (see Sections 5.2 and 5.3), we consider a unique generator g_f conditioned by the output of f.
Thus we redefine counterfactual and stable generations as:

$$\forall x \in \chi, \quad g_c(x) = g_f(x, 1 - c_f(x)), \quad g_s(x) = g_f(x, c_f(x)) \tag{5.23}$$

where the condition $c_f(x)$ or $1 - c_f(x)$ indicates the domain target of the generation. In the binary case, conditioning by the classification target or the output of f (e.g., score in $[0, 1]$) produces similar results. From Equation (5.23), the realism constraints from the general formulation (3.3) become

$$\forall x \in \chi \begin{cases} (g_f(x, 1 - c_f(x)) \in \chi_1, \ g_f(x, c_f(x)) \in \chi_0) & \text{if } x \in \chi_0 \\ (g_f(x, 1 - c_f(x)) \in \chi_0, \ g_f(x, c_f(x)) \in \chi_1) & \text{if } x \in \chi_1 \end{cases} \tag{5.24}$$

Using such generator g_f to relax the single generator formulation from Equation (5.4), first introduces a new stable generations $g_f(x, c_f(x))$. The counterfactual and stable generations are the output of the same generator g_f (even if the conditioning differs) which enforces the regularity property. We can remove the penalization term r_g from (3.3) for similar reasons as in Section 5.1.1. Second, to ensure that stable generation is very close to x, we explicitly minimize the distance d_s from Equation (5.2):

$$d_s(x, g_s(x)) = ||x - g_s(x)||_{1,2} = ||x - g_f(x, c_f(x))||_{1,2} \tag{5.25}$$

As the counterfactual generations is also the output of g_f, the minimization of (5.25) also encourages $g_f(x, 1 - c_f(x))$ (by construction) to a certain proximity to the input x (i.e., it reduces the degree of liberty of g_f for the counterfactual generation).
Finally the symmetrical constraint (5.3) that implicitly compels $g_c(x)$ to be close to the input x (d_c) becomes an intermediate between a symmetrical and cyclic constraint:

$$d_c{}^{implicit}(x, g_f(x, .)) = ||x - g_f(g_f(x, 1 - c_f(x)), c_f(x))||_{1,2} \tag{5.26}$$

In the following, we refer to this constraint as cyclic; which aligns the terminology with other works using a conditioned generator [Bass 20, Singla 20, Siddiquee 19].
The two constraints (5.25) and (5.26) enforce the Relevance condition. This new embodiment of problem (3.3) reads:

$$\begin{aligned} g_f^\star \quad = \quad & \underset{g_f}{\text{argmin}} \ \mathbb{E}_{x \in \chi} \left[\begin{array}{l} ||x - g_f(x, c_f(x))|| \quad + \\ ||x - g_f(g_f(x, 1 - c_f(x)), c_f(x))|| \end{array} \right] \\ \text{s.t.} \ & \left\{ \begin{array}{l} g_f(\chi_0, 1) \subset \chi_1 \ , \ g_f(\chi_1, 0) \subset \chi_0 \\ g_f(\chi_0, 0) \subset \chi_0 \ , \ g_f(\chi_1, 1) \subset \chi_1 \end{array} \right\} \end{aligned} \tag{5.27}$$

where we use the target class 0 or 1 in the realism constraints applied to distribution. The corresponding visual explanation is given by

$$\mathcal{E}(x) = ||g_f(x, c_f(x)) - g_f(x, 1 - c_f(x))|| \tag{5.28}$$

5.4.2 Weak Formulation

As for the previous embodiments described in Sections 5.1, 5.2 and 5.3, the constrained problem (5.27) can be approximated in a similar min-max optimization. We now detail the different terms of the global objective function.

Counterfactual Constraint: $\mathbf{g_f(\chi_0, 1) \subset \chi_1}$ and $\mathbf{g_f(\chi_1, 0) \subset \chi_0}$-
Similar to previous counterfactual embodiment, a real image $x \in \chi$, $g_f(x, 1-c_f(x))$ should be classified in the class opposite to input x. We use the classification term :

$$L_f^c(x, g_f) = \mathbb{E}_{x\in\chi} L_{bce}(1 - c_f(x), f(g_f(x, 1 - c_f(x)))) \tag{5.29}$$

To avoid adversarial attacks while encouraging generated images to belong to the distribution of real images, we also introduce a GAN term:

$$\begin{aligned} &L_D(x, g_f, D_0, D_1) \ = \\ &\quad \mathbb{E}_{x\in\chi_0}\left[L_{bce}(1, D_0(x)) + L_{bce}(0, D_1(g_f(x, 1 - c_f(x))))\right] \\ &\quad \mathbb{E}_{x\in\chi_1}\left[L_{bce}(1, D_1(x)) + L_{bce}(0, D_0(g_f(x, 1 - c_f(x))))\right] \end{aligned} \quad + \tag{5.30}$$

Stability: $\mathbf{||x - g_f(x, c_f(x))||}$ and $\mathbf{g_f(\chi_i, i) \subset \chi_i}$ for $\mathbf{i \in \{0, 1\}}$ -
The stability objectives of problem (5.27) are directly enforced by training g_f to minimize

$$L_d^{st}(x, g_f) = \mathbb{E}_{x\in\chi}\|x - g_f(x, c_f(x))\|_{1,2} \tag{5.31}$$

In this formulation, L_d^{st} alone suffices as it directly reconstructs the input x without any intermediate generation. We do not need an additional term to enforce the classification toward $c_f(x)$. Adding this classification term even amplifies patterns from the class $c_f(x)$ (which is not the objective).

Cyclic Consistency: $\mathbf{||x - g_f(g_f(x, 1 - c_f(x)), c_f(x)))||}$-
The "pseudo" cyclic constraint is also enforced by minimizing

$$L_d^{cy}(x, g_f) = \mathbb{E}_{x\in\chi}\|x - g_f(g_f(x, 1 - c_f(x)), c_f(x))\|_{1,2} \tag{5.32}$$

Finally, to also ensure that the cyclic term is classified like x, we add a consistency classification loss (as embodiment 5.2 and 5.21).

$$L_f^{cy}(x, g_f) = \mathbb{E}_{x\in\chi} L_{bce}(c_f(x), f(g_f(g_f(x, 1 - c_f(x)), c_f(x)))) \tag{5.33}$$

In experiments, we observe that with L_f^{cy}, the classification of the cyclic term improves.

Content Encodings Constraint-
As other works in domain transposition [Bass 20, Lee 18, Yu 19], we also enforce the content of the input image and the generated counterfactual to be close; thus encouraging relevancy. To do so, we constrain the feature maps of g_f: we enforce the latent space encoded from the input x to be close to the latent space of the counterfactual generation. We introduce e and d respectively the encoder and decoder part of g_f. Then, the content loss is

$$L_{cont}(x, g_f) = \mathbb{E}_{x\in\chi}\|e(x) - e(g_f(x, 1 - c_f(x)))\|_{1,2} \tag{5.34}$$

Note that in Sections 5.4.3 and 5.4.4, we rename the encoder and decoder depending on the conditioning specificity of g_f.

The resulting min-max optimization problem then reads

$$\min_{g_f} \max_{D_0, D_1} \left[\begin{array}{l} \lambda_f^c L_f^c(x, g_f) + \lambda_D L_D(x, g_f, D_0, D_1) + \\ \lambda_d^{st} L_d^{st}(x, g_f) + \\ \lambda_f^{cy} L_f^{cy}(x, g_f) + \lambda_d^{cy} L_d^{cy}(x, g_f) + \lambda_{cont} L_{cont}(x, g_f) \end{array} \right] \tag{5.35}$$

In the following we present two possible ways to condition the generator g_f: either at one or multiple latent spaces (see 5.4.3), or at image level (see 5.4.4).

5.4.3 Cyclic Latent conditioned Explanation (CyLatentCE)

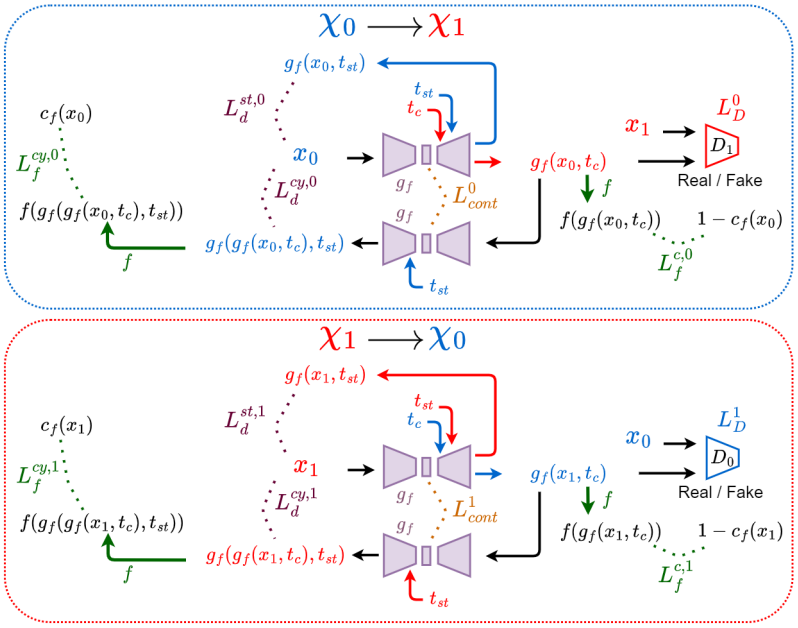


Figure 5.7: Overview of CyLatentCE optimization framework. Top: training step of g_f for an original image $x_0 \in \chi_0$. Bottom: training step for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) illustrated through dashed lines, are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_f (black arrow) and conditioned through its latent space by $t_c = c_f(x_1) = 1 - c_f(x_0)$ in the binary case (or directly $f(x_1)$). It produces a counterfactual image $g_f(x_0, c_f(x_1))$. This generated image is enforced ($L_f^{c,0}$) to be classified in the opposite class $1 - c_f$ by f (green arrow); or in the same class as x_1 (which is equivalent). We also enforce the counterfactual image to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Stable path***: the input image x_0 is given to the generator g_f and conditioned by the original prediction $t_{st} = c_f(x_0)$ (or $f(x_0)$). It produces a stable image $g_f(x_0, c_f(x_0))$ which is enforced to be pixel-wise close to x_0 by the term $L_d^{st,0}$. ***Cyclic path***: $g_f(x_0, t_c)$ is also mapped back to χ_0 through cycle consistency (black arrow bellow $g_f(x_0, t_c)$). Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L_d^{cy,0}$) and ($L_f^{cy,0}$). Similar procedures for the other transposition (**Bottom**).

We first propose to condition the latent space of the generator g_f by the target class c_f or directly the target output of f; it produces comparable generations results. Similar to

other state-of-the-art works in domain translation, we either condition the decoding path (d_f) of the g_f:

- at the lowest scale (or multiple scales) by concatenation operation (as in [Lee 18, Bass 20])
- through normalization layers (as in [Huang 18, Yu 19, Chiou 20])

We elaborate more on these conditionings in the implementation Section 7.2.1. In this formulation, the encoding path (e_{cont}) of g_f aims to extract the image's content. Decomposing g_f into its encoding and decoding path, the counterfactual and stable generation from (5.23) become

$$\forall x \in \chi, \quad g_c(x) = d_f(e_{cont}(x), 1 - c_f(x)), \quad g_s(x) = d_f(e_{cont}(x), c_f(x)) \tag{5.36}$$

The overall framework is shown in Figure 5.7.
As for the single symmetrical generator (see Section 5.1), we separate the steps for input images from χ_0 and χ_1. Using a single conditional discriminator [Miyato 18], we can merge the two steps into one that does not depend on a specific input domain (χ_0 or χ_1). We only need to update the GAN term from Equation (5.30) into:

$$L_D(x, g_f, D) = \mathbb{E}_{x \in \chi} \begin{bmatrix} L_{bce}(1, D[x, c_f(x)]) \ + \\ L_{bce}(0, D[g_f(x, 1 - c_f(x)), 1 - c_f(x)]) \end{bmatrix} \tag{5.37}$$

This single path variation in presented in the appendix (see Figure A.2). While the single path approach is suited for the multi classification case (see Section 5.6), the dual path version (Figure 5.7) performs better in average in the binary case.

5.4.4 Cyclic Image-level Conditioned Explanation (CyImageCE)

Inspired by [Choi 18, Siddiquee 19], we propose to condition an image level as an additional channel of the generative model g_f. Here, the encoding path (e_f) is conditioned, and Equation (5.23) becomes

$$\forall x \in \chi, \quad g_c(x) = d(e_f(x, 1 - c_f(x))), \quad g_s(x) = d_f(e_f(x, c_f(x))) \tag{5.38}$$

We illustrate the optimization framework with a dual path in Figure 5.8; the single path version is also shown in the appendix (Figure A.3).

Table 5.4 shows the contribution of the different loss terms to enforce the properties from the general formulation; it applies to both conditioning types. Note that we could also condition the generator:

- both at the image level and through its latent space (internal structure of g_f)
- both at the encoding and decoding path (as in [Singla 20] that use conditional batch normalization)

The same embodiment works in these cases.

Table 5.4: Conditions and loss terms relationship in CyLatentCE and CyImageCE. Contribution of each loss term (weak formulation) to encourage the three conditions (Relevance, Regularity, and Realism) from the general formulation (3.3)

	L_f^c	L_D	L_d^{st}	L_f^{cy}	L_d^{cy}
Relevance	✓		✓	✓	✓
Regularity			✓		
Realism		✓			

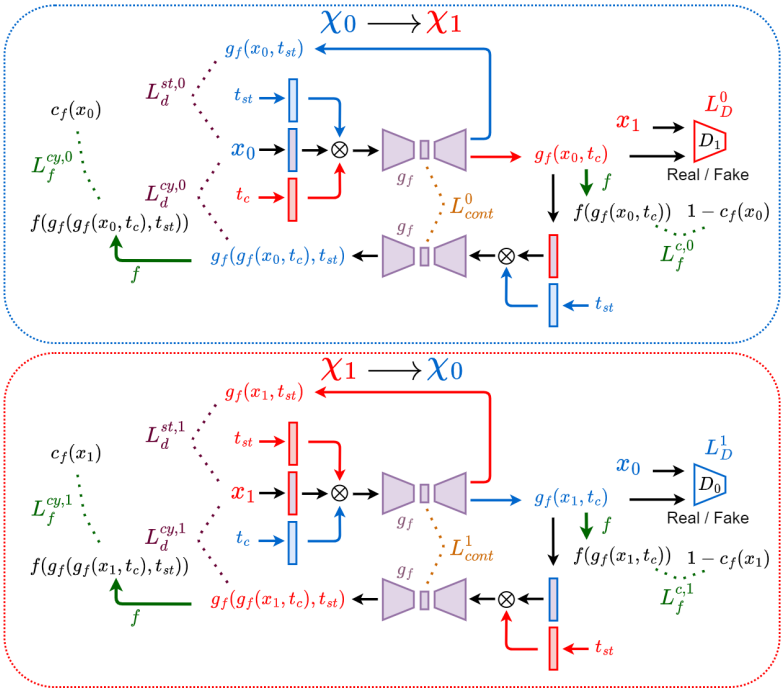


Figure 5.8: Overview of CyImageCE optimization framework. Top: training step of g_f for an original image $x_0 \in \chi_0$. Bottom: training step for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) illustrated through dashed lines, are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_f (black arrow) and conditioned at image level by $t_c = c_f(x_1) = 1 - c_f(x_0)$ in the binary case (or directly $f(x_1)$). It produces a counterfactual image $g_f(x_0, c_f(x_1))$. This generated image is enforced ($L_f^{c\,0}$) to be classified in the opposite class $1 - c_f$ by f (green arrow); or in the same class as x_1 (which is equivalent). We also enforce the counterfactual image to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Stable path***: the input image x_0 is given to the generator g_f and conditioned by the original prediction $t_{st} = c_f(x_0)$ (or $f(x_0)$). It produces a stable image $g_f(x_0, c_f(x_0))$ which is enforced to be pixel-wise to x_0 by the term $L_d^{st,0}$. ***Cyclic path***: $g_f(x_0, t_c)$ is also mapped back to χ_0 through cycle consistency (black arrow bellow $g_f(x_0, t_c)$). Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L_d^{cy,0}$) and ($L_f^{cy,0}$). Similar procedures for the other transposition (**Bottom**).

5.5 Symmetrical Stable and Counterfactual Generations (SySCGen)

5.5.1 From SAGen to its counterfactual embodiment counterpart

In this counterfactual embodiment, we revisit the adversarial generation proposed in Section 4.2. The idea is similar to the previous counterfactual embodiments (see Sections 5.1 to 5.4) in spirit, but we use an explicit couple of generators: g_s and g_c.

Thus, the stable and the counterfactual generators g_s and g_c ensure the realism property (as defined in the general formulation (3.3)):

$$\forall x \in \chi \begin{cases} (g_c(x) \in \chi_1,\ g_s(x) \in \chi_0) \text{ if } x \in \chi_0 \\ (g_c(x) \in \chi_0,\ g_s(x) \in \chi_1) \text{ if } x \in \chi_1 \end{cases} \tag{5.39}$$

With such distinct stable and counterfactual generators, we can no longer remove the penalization term r_g from Equation (3.3). This term is explicitly minimized to ensure regularity. Using these two explicit generators, we weaken the constraint from the single symmetrical generator 5.1. Thus, the stable generation is explicitly enforced to be close to the input x by minimizing d_s:

$$d_s(x, g_s(x)) = ||x - g_s(x)||_{1,2} \tag{5.40}$$

We use the symmetrical constraint from (5.3) to implicitly minimize d_c:

$$d_c^{implicit}(x, g_c(x)) = ||x - {g_c}^2(x)||_{1,2} \tag{5.41}$$

The resulting embodiment of problem (3.3) reads:

$$\begin{aligned} g_s^\star, g_c^\star \ = \ & \underset{g_s,\, g_c}{\text{argmin}}\ \mathbb{E}_{x\in\chi}\left[||x - g_s(x)|| + ||x - {g_c}^2(x)||\right] + r_g\left(g_s, g_c\right) \\ \text{s.t. } & \left\{ \begin{array}{l} g_c(\chi_0) \subset \chi_1\ ,\ g_c(\chi_1) \subset \chi_0 \\ g_s(\chi_0) \subset \chi_0\ ,\ g_s(\chi_1) \subset \chi_1 \end{array} \right\} \end{aligned} \tag{5.42}$$

5.5.2 Weak Formulation

As for the previous counterfactual embodiments, the constrained optimization problem (5.42) is approximated into a min-max optimization. We precise the different terms of the objective function below.

Counterfactual Constraint: $\mathbf{g_c}(\chi_0) \subset \chi_1$ and $\mathbf{g_c}(\chi_1) \subset \chi_0$-
We also enforce the counterfactual generation to be predicted in the class opposite to x, using a classification term :

$$L_f^c(x, g_c) = \mathbb{E}_{x\in\chi} L_{bce}(1 - c_f(x), f(g_c(x))) \tag{5.43}$$

We encourage the counterfactual image to belong to the distribution of real images with:

$$\begin{aligned} & L_D(x, g_c, D_0, D_1) \ = \\ & \quad \mathbb{E}_{x\in\chi_0}\left[L_{bce}(1, D_0(x)) + L_{bce}(0, D_1(g_c(x)))\right] \\ & \quad \mathbb{E}_{x\in\chi_1}\left[L_{bce}(1, D_1(x)) + L_{bce}(0, D_0(g_c(x)))\right] \end{aligned} \quad + \tag{5.44}$$

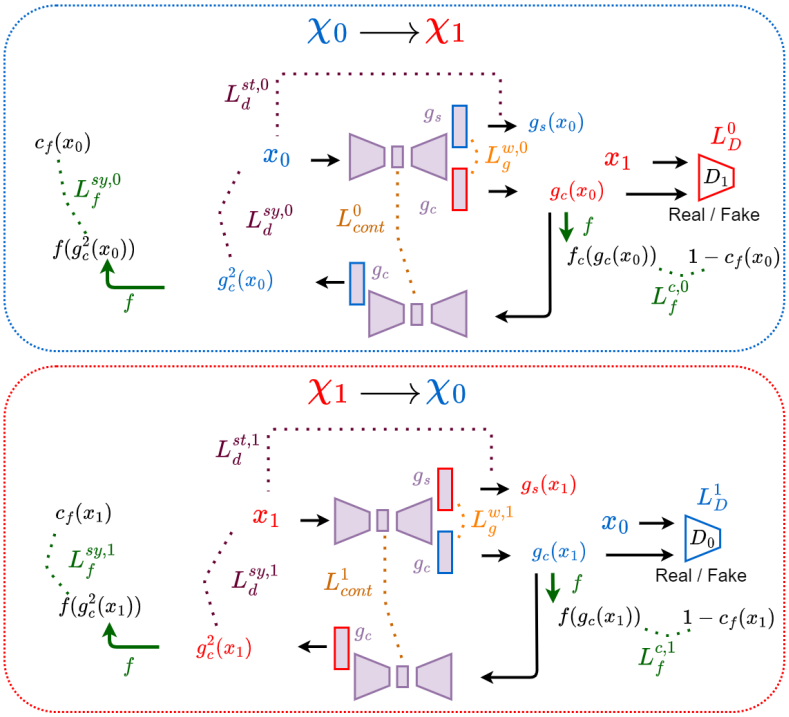


Figure 5.9: Overview of SySCGen optimization framework. Top: training step of g_s and g_c for an original image $x_0 \in \chi_0$. Bottom: training step for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) illustrated through dashed lines, are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_c which produces a counterfactual image $g_c(x_0)$. This generated image is enforced ($L_f^{c\,0}$) to be classified in the class opposite to x_0. We also enforce the generated image $g_c(x_0)$ to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Stable path***: given x_0, g_s generated a stable generation which is enforced to be pixel-wise close to x_0 by the term $L_d^{st,0}$. ***Symmetrical path***: the counterfactual image $g_c(x_0)$ is mapped back to χ_0 through symmetrical constraint by re-applying g_c. Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L_d^{sy,0}$) and ($L_f^{sy,0}$). Similar procedures for the other transposition (**Bottom**).

Stability: $||\mathbf{x} - \mathbf{g_s}(\mathbf{x})||$ and $\mathbf{g_s}(\chi_\mathbf{i}) \subset \chi_\mathbf{i}$ for $\mathbf{i} \in \{0, 1\}$ -
Minimizing the distance d_s (5.40) enforces the stability constraints. The corresponding loss term is:

$$L_d^{st}(x, g_s) = \mathbb{E}_{x \in \chi} \|x - g_s(x)\|_{1,2} \tag{5.45}$$

L_d^{st} alone suffice; no additional classification term is required to enforce the target classification ($c_f(x)$).

Symmetrical constraint: $||\mathbf{x} - \mathbf{g_c}^2(\mathbf{x})||$-
We combine a pixel-wise distance term L_d^{sy} and a classification term L_f^{sy} to encourage the proximity of ${g_c}^2(x)$ to x, and the same classification. Those terms read

$$L_d^{sy}(x, g_c) = \mathbb{E}_{x \in \chi} \|x - {g_c}^2(x)\|_{1,2} \tag{5.46}$$

$$L_f^{sy}(x, g_c) = \mathbb{E}_{x \in \chi} L_{bce}(c_f(x), f({g_c}^2(x))) \tag{5.47}$$

Content Encodings Constraint-
We also enforce the content of the input image and the generated counterfactual to be close, thus encouraging relevancy without applying explicit constraint between $g_c(x)$ and x. As in Section 5.4, we extract the latent encodings ($e(x)$) from g_c and minimize

$$L_{cont}(x, g_f) = \mathbb{E}_{x\in\chi}\|e(x) - e(g_c(x))\|_{1,2} \tag{5.48}$$

Weight penalization: $r_g(g_s, g_c)$-
Finally, we explicitly penalize the distance between the two generators g_s and g_c, and minimize the distance between the weights of the two generators (as in 4.2). Let g_s and g_c have the same architecture (or even share a part of it), and be parameterized by their weights w_s and w_c respectively; we aim to minimize the same term L_g^w as in Equation (4.12) i.e.

$$L_g^w(g_s(., w_s), g_c(., w_c)) = \sum_k \left\| w_s^k - w_c^k \right\|_2 \tag{5.49}$$

.
Figure 5.9 illustrates the dual path version of the optimization framework; a single path (step for any $x \in \chi$) could also be considered here. In the Figure, g_s and g_c share a common part of the architecture; and only differ in the final layers (as in the single SAGen introduced in Figure 4.4). In this case, L_g^w is computed only on the weights of the differing layers.
The final objective function is then

$$\min_{g_s,g_c} \max_{D_0,D_1} \left[\begin{array}{l} \lambda_f^c L_f^c(x, g_c) + \lambda_D L_D(x, g_c, D_0, D_1) + \\ \lambda_d^{st} L_d^{st}(x, g_s) + \lambda_f^{sy} L_f^{sy}(x, g_c) + \lambda_d^{sy} L_d^{sy}(x, g_c) + \\ \lambda_{cont} L_{cont}(x, g_c) + \lambda_g^w L_g^w(g_s, g_c) \end{array} \right] \tag{5.50}$$

Table 5.5 resume the contribution of each term to enforce relevance, regularity and realism.

Table 5.5: Conditions and loss terms relationship in SySCGen. Contribution of each loss term (weak formulation) to encourage the three conditions (Relevance, Regularity, and Realism) from the general formulation (3.3)

	L_f^c	L_D	L_d^{st}	L_f^{sy}	L_d^{sy}	L_g^w
Relevance	✓		✓	✓	✓	
Regularity						✓
Realism		✓				

In the appendix, we also introduced a cyclic variation of this proposition. As for SyCE and CyCE (5.2 and 5.3), we introduce domain-specific generators ${g_s}^0$, ${g_s}^1$, ${g_c}^0$ and ${g_c}^1$. In this case, we relax the symmetrical constraint into a cyclic one. Figure A.4 describes the optimization framework. Comparison results are proposed in the experimental Chapter 8.

5.6 Extension to multi-classification

Similar to Section 4.3, we can adapt some previous embodiments from the binary case to the multi-classification setting. We only consider the untargeted and targeted settings from 3.3.2 and 3.3.3 respectively, as the untargeted setting 3.3.1 transform the explanation of multi-classification task into a binary problem.

First, with multiple classes (in general between 3 and 20 in medical image problems), the proposed implementations SyCE (sec. 5.2), CyCE (sec. 5.3) and all other embodiments (e.g. CySCGen (sec. 5.5)) using domain-specific models scale poorly. The number of generators and discriminators to train increases as the square of the number of classes.
In the multi-classification setting, we rather consider the other embodiments CyLatentCE (sec. 5.4.3), CyImageCE (sec. 5.4.4) or SySCGen (sec. 5.5) which introduce a unique generative model. They could also be optimized with a single discriminator (see single path version in the appendix) conditioned with the output of the classifier f (or c_f), although using two domain-specific discriminators performs better in the binary case. Thus we alleviate the scaling issue by considering for these last embodiments the single path version, i.e., a unique couple of generator and discriminator conditioned by a classification target (as in conditional GANs [Mirza 14, Miyato 18]). Note that we do not consider SSyGen (sec. 5.1) as the multi-classification setting further increases the constraints on a single counterfactual generator g_c. It increases the lack of realism of the counterfactual generations (and results in adversarial attacks).

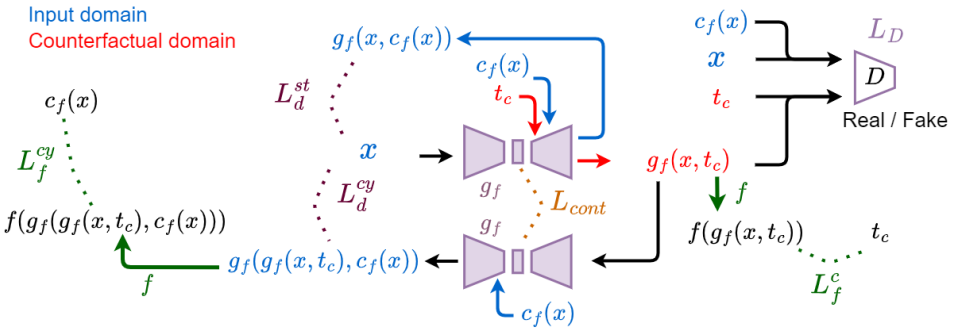
For the embodiments CyLatentCE and CyImageCE, the optimization problem from Equation (5.27) becomes:

$$\begin{aligned} g_f^\star \quad &= \quad \underset{g_f}{\text{argmin}} \ \mathbb{E}_{x\in\chi}\left[\begin{array}{l} ||x - g_f(x, c_f(x))|| \quad + \\ ||x - g_f(g_f(x, t_c), c_f(x))|| \end{array}\right] \\ \text{s.t.} \ &\left\{\begin{array}{l} g_f(\chi_{t_c}, t_c) \subset \chi_t \ , \ \forall t_c \in [\![1 \ ; \ C]\!] \\ g_f(\chi_{c_f}, c_f(x)) \subset \chi_{c_f} \end{array}\right\} \end{aligned} \tag{5.51}$$

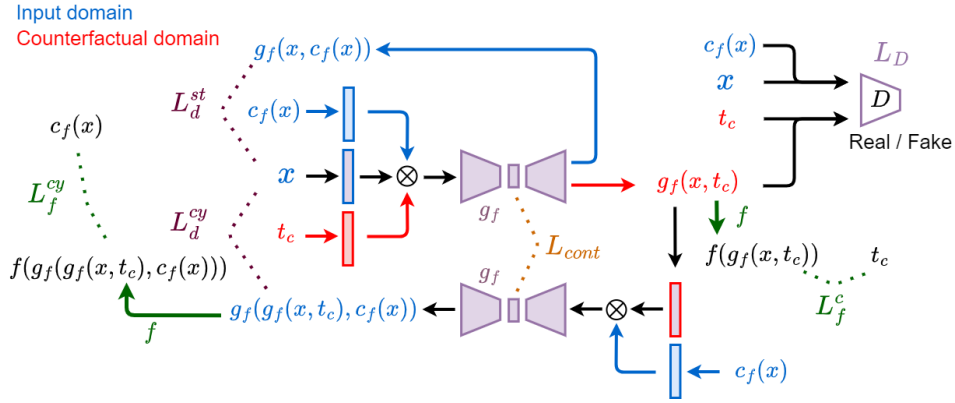
where χ_{c_f} is the subspace of the distribution of real images χ where images or classified as $c_f(x)$ i.e. the predicted class of the input x by f. χ_{t_c} is the subspace of χ where images are classified in the targeted or untargeted class t_c; depending on the type of visual explanation we seek (similar choices in Section 4.3). C is the number of classes.
Then, we also adapt the approximated min-max optimization problem from Equation (5.35) by using a single conditioned discriminator. Thus the GAN term now reads

$$L_D(x, g_f, D) = \mathbb{E}_{x\in\chi}[L_{bce}(1, D(x, c_f(x))) + L_{bce}(0, D(g_f(x, t_c), t_c))] \tag{5.52}$$

In the multi-classification setting, we replace the binary classification loss terms L_f^c and L_f^{cy} with a softmax cross entropy loss. Note that the classification terms can be adapted as described in the multi-classification extension of the adversarial generation embodiment (see Section 4.3).
Figures 5.10a and 5.10b illustrate the adapted optimization frameworks for the multi-classification setting. SySCGen framework can be adapted similarly, but it would only work for an untargeted counterfactual explanation.



(a) CyLatentCE



(b) CyImageCE

Figure 5.10: Overview of CyLatentCE and CyImageCE optimization frameworks in the multi-classification setting. (a) Training step for CyLatentCE. (b) Training step for CyImageCE. The frameworks are similar in the two cases; only the conditioning type differs. ***Counterfactual path***: an input image x is given to the generator g_f (black arrow) and conditioned –either through its latent space (a) or at image level (b)– by a target t_c . It produces a counterfactual image $g_f(x, t_c)$. This generated image is enforced (L_f^c) to be classified in the target class t_c by f (green arrow). We also enforce the counterfactual image to fool the discriminator D, trained to identify real (x) from generated images. D is conditioned by the classification target of its input: either $c_f(x)$ for the real image x or t_c the generated counterfactual $g_c(x)$. ***Stable path***: the input image x is given to the generator g_f and conditioned by the original predicted class $c_f(x)$. It produces a stable image $g_f(x, c_f(x))$ which is enforced to be pixel-wise to x by the term L_d^{st}. ***Cyclic path***: $g_f(x, t_c)$ is also mapped back to its original domain through cycle consistency (black arrow bellow $g_f(x, t_c)$). Pixel-wise proximity and classification consistency to x are encouraged by the constraints (L_d^{cy}) and (L_f^{cy}).

5.7 Conclusions of the counterfactual embodiment

We propose several embodiments of counterfactual generations satisfying the conditions of Relevance, Regularity, and Realism defined in chapter 3 and required to produce our visual explanation of a classifier's decision.
Compared with chapter 4, we brought two major modifications:

- We introduced the Realism property in the optimization. We approximated the problem constrained on distribution into a min-max optimization using a GAN framework.
- We implicitly minimize the distance d_c using either symmetric or cyclic constraints (or even both). This implicit penalization allows more freedom for the counterfactual generation that is no longer enforced to be pixel-wise close to the input image.

In all the optimizations of both chapter 4 and 5, we introduced different terms of loss that are similar for the different proposals but adapted for each. Table 5.6 sums up all of these terms for each of the proposals and points out the contribution of each term to enforce the properties of the general formulation from chapter 3.

Table 5.6: Sum up the different approaches for Counterfactual Explanation. Contribution of each loss term from the different propositions of chapter 5 to enforce the three properties: Relevance, Regularity, and Realism.

	L_d^{st}	L_d^{sy}	L_d^{cy}	L_d^c	L_f^{sy}	L_f^{cy}	L_f^c	L_D	L_g^w
Relevance	✓	✓	✓	✓	✓	✓	✓		
Regularity	✓	✓							✓
Realism				✗*				✓	
AGen (4.1)				✓			✓		
SAGen (4.2)	✓			✓			✓		
SSyGen (5.1)		✓			✓		✓	✓	
SyCE (5.2)		✓	✓		✓	✓	✓	✓	
CyCE (5.3)			✓			✓	✓	✓	
CyLatentCE (5.4.3)	✓		✓			✓	✓	✓	
CyImageCE (5.4.4)	✓		✓			✓	✓	✓	
SySCGen (5.5)	✓**	✓**			✓		✓	✓	✓
CySCGen*** (5.5)	✓**		✓			✓	✓	✓	✓

* An explicit minimization of the distance between the input x and the generated counterfactual (or adversary) penalize the realism condition as it enforces the proximity of all pixels of the two images.
** In SySCGen and CySCGen, the stable and the counterfactual generations are not outputs of the same generative model. In these cases, the terms L_d^{st} and L_d^{sy} do not enforce regularity; The term L_g^w does.
*** CySCGen is introduced at the end of Section 5.5. The optimization framework is detailed in the appendix.

6

Embodiments 3: Unification of Counterfactual Explanation and Integrated Gradient

In Chapters 4 and 5, we first proposed to enforce relevance and regularity of our visual explanation through adversarial generations; then, we encouraged the realism of the generations via counterfactual generation methods.

Adversarial or counterfactual generation methods –either proposed in previous chapters or the literature [Woods 19, Elliott 19, Singla 20]– produce a visual explanation by comparing either the input image x or a generated stable image $\mathbf{x_s}$ with a "close" generated adversarial (or counterfactual) image $\mathbf{x_c}$ and defining visual explanation by

$$\mathcal{E}(\mathbf{x}) = \begin{cases} |\mathbf{x_s} - \mathbf{x_c}| & \text{if } \exists\, \mathbf{x_s} \\ |\mathbf{x} - \mathbf{x_c}| & \text{otherwise} \end{cases} \tag{6.1}$$

Although these methods –and especially counterfactual generations– perform well in localizing relevant regions of the input and highlighting realist patterns learned by the classifier (see Section 8.1); no counterfactual generation approach explicitly enforces that visual explanation values translate into importance values for f at the pixel level or any higher scale. For instance, suppose $\mathbf{x}$ is a CT-scan and classifier f is influenced by regions containing bone tissues (which have high intensity in CT scans), these regions should then be attenuated in the generated counterfactual $\mathbf{x_c}$ and appear with high intensity in the difference $\mathcal{E} = |\mathbf{x} - \mathbf{x_c}|$. This high intensity may not be directly related to the relative importance of bone regions for f, but only result from their original intensity in $\mathbf{x}$. This case would poorly perform when assessing features importance for f, using, for instance, the AOPC metric [Samek 17] or the features relevance metric [Lim 21].
In addition, we show experimentally (see Chapter 8) that even when using $\mathbf{x_s}$, it is sometimes impossible to remove all irrelevant regions for f inherited from the generation process.

On the other hand, gradient-based methods (see Section 2.1.1), which derive visual explanation from pixel-wise derivatives of the model, are built to detect pixel regions with a high impact on the model prediction. However, these approaches generally produce very noisy outputs and are not dedicated to medical image tasks (not specific enough). Among these methods, path-based approaches –such as integrated gradient [Sundararajan 17]– compute the vanilla gradient method [Simonyan 14] at different steps of a linear path between a baseline image and the input image. Then, they integrate all the contributions along the path. They improve the vanilla approach by avoiding the gradient vanishing issue and singular points effect (where the gradient is not continuous). The resulting visual

explanation is also more focused and, to some extent, smoother. The main drawback of the method is the choice of the baseline. The baseline is an element of the input space with a neutral prediction or that does not contain the features of the studied class. For "natural" images and multi-classification problems, the black image (or a different type of synthetic image) often satisfies this property. It is not the case for medical image problems, especially binary classification (such as object detection). A synthetic baseline is completely outside the distribution of real images (of the task at stake). The variability between images of a given medical task is much smaller than for natural images. A synthetic baseline has much more impact and cannot be compensated by the diversity of context, colors, or structures of the medical images. For similar reasons, with a decreasing number of classes, this synthetic baseline has an increasing impact, and there is no reason it produces a neutral prediction. In binary classification, a neutral prediction could have no meaning as it supports total doubt on the input class rather than defining the absence of the studied relevant features.

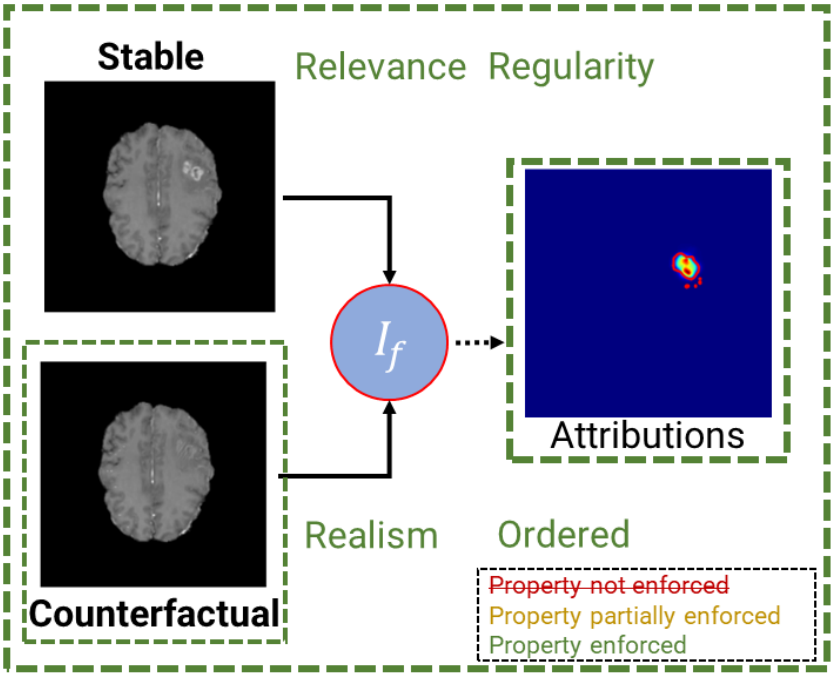


Figure 6.1: Illustrations of the properties impact on the integrated counterfactual visual explanation. The global objective is to produce visual explanations (counterfactual and attribution) satisfying the formulation objectives (all green) through enforcing the different properties. ***Enforced properties: Relevance, Regularity, Realism and Ordered by importance***: Both $g_s^\star$ and $g_c^\star$ are optimized. We enforce that $g_c^\star$ (resp. $g_s^\star$) generates images classified in the class opposite (resp. equal) to the input and remain close to it. Using the stable image improves the regularity, and therefore the reconstruction errors are mainly removed (improving Relevance). With realism, the counterfactual generation changes the texture of the tumor tissue into healthy tissue. The first three properties strongly reduce residual errors in the attributions. By enforcing that attribution values correlate with importance for the classifier, the method further improves the properties of Relevance and Regularity.

In this chapter, we unify counterfactual generation methods and path-based approach to:

- enforce that our counterfactual explanation produces a visual explanation where values reflect the importance for the classifier, i.e., relevant features found by the visual explanation are ordered by importance.
- define a baseline image for path-based methods more adapted to medical image task (especially binary classification).

Figure 6.1 (extracted from Figure 3.2) summarizes this unified approach, pointing out the formulation goals achieved and the properties enforced.

6.1 Counterfactual explanation reflecting an ordered importance of features

Given the counterfactual generation approach, we want to enforce the generated visual explanation's features to reflect its importance for the classifier f.
Consider an input image $\mathbf{x}$ –or its stable generation $\mathbf{x_s} = g_s^\star(x)$ (depending on the chosen counterfactual method)– and its generated adversary $\mathbf{x_c} = g_c^\star(x)$. Following [Singla 20, Sundararajan 17] we consider a differential path γ mapping elements $\lambda \in [0, 1]$ to the space of real images and satisfying $\boldsymbol{\gamma}(0) = \mathbf{x}$ (or $\mathbf{x_s}$) and $\boldsymbol{\gamma}(1) = \mathbf{x_c}$. From Equation (6.1) we have

$$\mathcal{E}(\mathbf{x}) = |\mathbf{x_c} - \mathbf{x}| = \left| \int_0^1 \frac{d\boldsymbol{\gamma}}{d\lambda}(u) du \right|. \tag{6.2}$$

To enforce a monotonic relationship between high-value regions of $\mathcal{E}$ and high-importance regions of f, we propose introducing weights related to the variations of f along the path integral (6.2). We define these weights (w) at every $u \in [0, 1]$ based on the variations :

$$\frac{d(f \circ \boldsymbol{\gamma})}{d\lambda}(u) = \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)) \frac{d\boldsymbol{\gamma}}{d\lambda}(u). \tag{6.3}$$

Several strategies are possible for w. The expressions studied in Section 6.2 can be summarized using a continuous function of two variables F, setting

$$w(u) = F\left(\frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)), \frac{d\boldsymbol{\gamma}}{d\lambda}(u) \right) \tag{6.4}$$

We then define the visual explanation map as

$$\mathcal{E}_w(\mathbf{x}) = \left| \int_0^1 w(u) \frac{d\boldsymbol{\gamma}}{d\lambda}(u) du \right| = \left| \int_0^1 F\left(\frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)), \frac{d\boldsymbol{\gamma}}{d\lambda}(u) \right) \frac{d\boldsymbol{\gamma}}{d\lambda}(u) du \right| \tag{6.5}$$

Weights w, path γ, and its derivative are of the same dimension as image $\mathbf{x}$. Summation and multiplication are thus computed pixel-wise.
The weighted explanation $\mathcal{E}_w$ replaces the simple definition of function $\mathcal{I}_f$ (see Equation (3.1)) with a more complex formulation.

6.2 Choice of the path and the weight

Path $\boldsymbol{\gamma}$ should be traced on the manifold of real images. In practice, this constraint induces heavy computation burdens to determine the derivative $\frac{d\boldsymbol{\gamma}}{d\lambda}$. Indeed, if we consider for our generative model an encoder(E)-decoder(D) architecture as in Chapter 5 or in [Singla 20]; E (resp. D) maps from (resp. to) the space of real images ($\subset \mathbb{R}^n$) to (resp. from) an encoding space ($\subset \mathbb{R}^k$). The *real* images path γ can, for instance, be defined as

$$\gamma : \lambda \to D(z_{\mathbf{x}} + \lambda(z_{\mathbf{x_c}} - z_{\mathbf{x}})) \tag{6.6}$$

where $z_{\mathbf{x}} = E(\mathbf{x})$ and $z_{\mathbf{x_c}} = E(\mathbf{x_c})$. It follows that $\frac{d\boldsymbol{\gamma}}{d\lambda} = \frac{\partial G}{\partial z}(z_{\mathbf{x}} + \lambda(z_{\mathbf{x_c}} - z_{\mathbf{x}}))(z_{\mathbf{x_c}} - z_{\mathbf{x}})$. However, $\frac{\partial D}{\partial z}$ is a vector of dimension $n.k$ which easily reaches a magnitude of 10^9 that is to be computed at several values of λ.
To tackle this issue, we use a similar expression as [Sundararajan 17] and define a linear path in the input space

$$\boldsymbol{\gamma} : \lambda \rightarrow \mathbf{x} + \lambda(\mathbf{x_c} - \mathbf{x}) \tag{6.7}$$

so that $\frac{d\boldsymbol{\gamma}}{d\lambda} = (\mathbf{x_c} - \mathbf{x})$. In the particular case of medical classification tasks, the intermediate images produced along the path remain close to the input distribution as our generated counterfactual aims only to modify relevant regions of the input. It transforms those regions into realistic regions of the opposite class (in the binary case). Thus, the intermediate images more or less highlight the characteristics of one or the other class. Such a case also exists in practice, e.g., pathological regions can be more or less intense or with variable sizes throughout the database.
Experimentally, even with this simplification, visual explanation maps integrating feature importance (FI) improve counterfactual explanation baselines and outperform state-of-the-art methods (see Sections 8.1.3 and 8.2.1).

Then, to define the weight w, we set the function F in two possible ways:

- **Version 1**: $F : (x, y) \rightarrow x.y$. This setting follows the formulation from [Sundararajan 17], combining the path derivative and the gradient of the output of f (w.r.t the input).
- **Version 2**: $F : (x, y) \rightarrow |x|.y$. Here we take into account all derivatives regardless of their signs.

These two settings of F respectively lead to the following expression of the visual explanation:

$$\begin{aligned} \mathcal{E}_{FI}^{v1}(x) &= (\mathbf{x_c} - \mathbf{x})^2 \left| \int_0^1 \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)) du \right| \\ \mathcal{E}_{FI}^{v2}(x) &= (\mathbf{x_c} - \mathbf{x})^2 \int_0^1 \left| \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)) \right| du \end{aligned} \tag{6.8}$$

6.3 Regularization

Despite the accumulation of gradients along the linear path between $\mathbf{x}$ and $\mathbf{x}_a$, $\mathcal{E}_{FI}^{v1}$ and $\mathcal{E}_{FI}^{v2}$ inherit from the drawback of gradient-based methods and tend to be noisy. We thus introduce a regularized version of $\mathcal{E}_w$:

$$\mathcal{E}_{w,k_\sigma}(x) = \left| \int_0^1 \left[F\left(\frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)), \frac{d\boldsymbol{\gamma}}{d\lambda}(u) \right) \frac{d\boldsymbol{\gamma}}{d\lambda}(u) \right] * k_\sigma du \right| \tag{6.9}$$

where k_σ is a centered Gaussian kernel of variance σ. Hence, for the second setting of F (which produces the better results), $\mathcal{E}_{FI}^{v2}$ becomes:

$$\mathcal{E}_{FI,k_\sigma}^{v2}(x) = \int_0^1 \left((\mathbf{x_c} - \mathbf{x})^2 \left| \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)) \right| \right) * k_\sigma du \tag{6.10}$$

The Gaussian kernel k_σ applies the contribution at each step of the path and therefore regularizes especially the most relevant contributions (the others tending toward zero).
In our experiments, $\mathcal{E}_{FI,k_\sigma}^{v2}$ is competitive with $\mathcal{E}_{FI}^{v1}$ and $\mathcal{E}_{FI}^{v2}$ for features' importance evaluation metrics while improving pathology localization performance.

6.4 Integrated Gradient with Counterfactual baseline

The other way to consider the problem is to focus on the baseline choice in the integrated gradient method. As stated at the beginning of this chapter, using a synthetic baseline (such as the black image) is not suited for classification tasks in medical images, especially for the binary case.
We propose to use the counterfactual generation as the baseline reference. Hence, the attributes of images relevant for the classification would be absent while the rest of the image (irrelevant for f) remains the same. The resulting counterfactual integrated gradient thus reads:

$$\mathcal{IG}_c^{v1}(x) = \left|(\mathbf{x_c} - \mathbf{x}) \int_0^1 \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u))du\right| \tag{6.11}$$

Then, similarly, as in Sections 6.2 and 6.3, we can also introduce a second version where all derivatives are taken into account $\mathcal{IG}_c^{v2}$, as well as a regularized version $\mathcal{IG}_{c,k_\sigma}^{v2}$ reducing the noise imputable to gradient methods (see the Appendix B.1).

6.5 Conclusions and extension to multi-classification

In Sections 6.1 and 6.4, we proposed two ways to unify counterfactual generation and path-based methods. We start with our counterfactual visual explanation and adapt the function $\mathcal{I}_f$ from the general formulation (see Chapter3) to better translate the local importance of input features for f. In contrast, we define a counterfactual baseline for the integrated gradient technique that belongs to the distribution of real (of the given task) and where only features of interest are absent from the input (more or less a residual reconstruction error).
These are two different ways of looking at the problem, but they produce very similar expressions, especially when considering a linear path γ.

As mentioned at the beginning of this chapter, despite attempts to reduce the residual error inherited from the generation process (using a stable generation), some errors sometimes remain. We show in our results (see Chapter 8) that this unification strategy improves the relevance property and, in some way, the regularity. By summing the gradient contributions (along γ), the method better focuses on impacting regions (or even pixels). The remaining residual errors do not contribute (or much less). Then, the visual explanation highlights the relevant regions for f (relevance); the residual errors imputable to the generation process decrease (indirect regularity). In addition, we notice in the experiments (see Section 8.1.3) that the stable image is no longer needed with this unification strategy.

Unifying counterfactual generations and path-based methods seem relatively straightforward in the binary case. Can we use or adapt the same operation to the multi-classification case ?
In the multi-classification case, for both the untargeted and targeted visual explanations (see Sections 3.5 and 3.6 respectively), we can consider the same proposals as above. We would produce a visual explanation translating the importance of feature for f by comparing the input to a second choice image for f (untargeted) or any other class image (targeted). In the targeted version, we could also reformulate the visual explanation similarly as in [Pan 21], and rather than explaining why the classifier takes such a decision on the input; we explain what makes the model discriminate this prediction from all other

classes. The resulting weighted visual explanation would thus read

$$\mathcal{E}_w(x) = \sum_{k=1,k\neq l}^{C} \left| \int_0^1 \left[F\left(\frac{\partial f_l}{\partial \mathbf{x}}(\boldsymbol{\gamma}_k(u)), \frac{d\boldsymbol{\gamma}_k}{d\lambda}(u) \right) \frac{d\boldsymbol{\gamma}_k}{d\lambda}(u) \right] du \right| \tag{6.12}$$

where C is the number of classes; l is the predicted class on the input x; and $\boldsymbol{\gamma}_k$ defines the linear path between the input image and the counterfactual generation from class k e.g. Considering the CyLatentCE or CyIlmageCE embodiments as in Section 5.6, $\boldsymbol{\gamma}_k : \lambda \mapsto x + \lambda(g_f(x,k) - x)$.

7

Experiments

7.1 Classification tasks

In this section, we describe (i) the classification tasks we studied in our experiments, (ii) the preprocessing of the data, and (iii) the classification model training procedure and results. We design multiple binary classification tasks to validate our approach and propose a simple case study for the extension in the multi-classification setting.

7.1.1 Binary Classification

7.1.1.1 Pneumonia detection on RSNA chest X-Rays

We created a binary chest X-Rays dataset from the available RSNA Pneumonia Detection Challenge. We only kept the healthy and pathological exams with pneumonia. It constitutes a binary database of 14863 samples (8851 healthy / 6012 pathological). We randomly split the dataset into train (80 %), validation (10 %) and test (10 %) sets. The main clinical difference between healthy and pneumonia cases is the presence (or abscence) of white opacities in the pulmonary region (that are commonly dark). However, the size, the location and the density of pneumonia opacities vary. Depending on the quality of the image, the lungs intensity also varies and some regions may be confounded with some opacities. It may also be confusing when ribs (i.e., high density appearing in white) and small opacities are superimposed.
A ResNet50 [He 16] and a DenseNet121 [Huang 17] were trained to minimize a binary cross entropy loss, on images rescaled from 1024 x 1024 to 224 x 224 and normalized to [0, 1]. We use pre-trained backbone layers from ImageNet [Deng 09] for the two models. Then after averaging feature maps, we add a dropout layer and adapt their output for binary classification, i.e., we use a dense layer with a single output (instead of 1000 for ImageNet), followed by a sigmoid activation. To use the pre-trained backbone from ImageNet, we must pass 3 channels inputs as they work on RGB images. We thus concatenate 3 times the gray-scale images to produce the required input format. They respectively achieve 0.974 and 0.978 AUC scores on the test set. We remind that the AUC score stands from the "Area Under the ROC Curve". The ROC curve is a graph plotting values of the true positive rate vs. the false positive rate at all thresholds of the classification model.

7.1.1.2 Brain tumor detection on BRATS MRI slices

The images from the Medical Segmentation Decathlon Challenge [Simpson 19] are 3D MRI volumes of the brain. All volumes contain at least one tumor region. To design a binary classification task, we consider the problem of localizing a tumor region in an MRI volume.

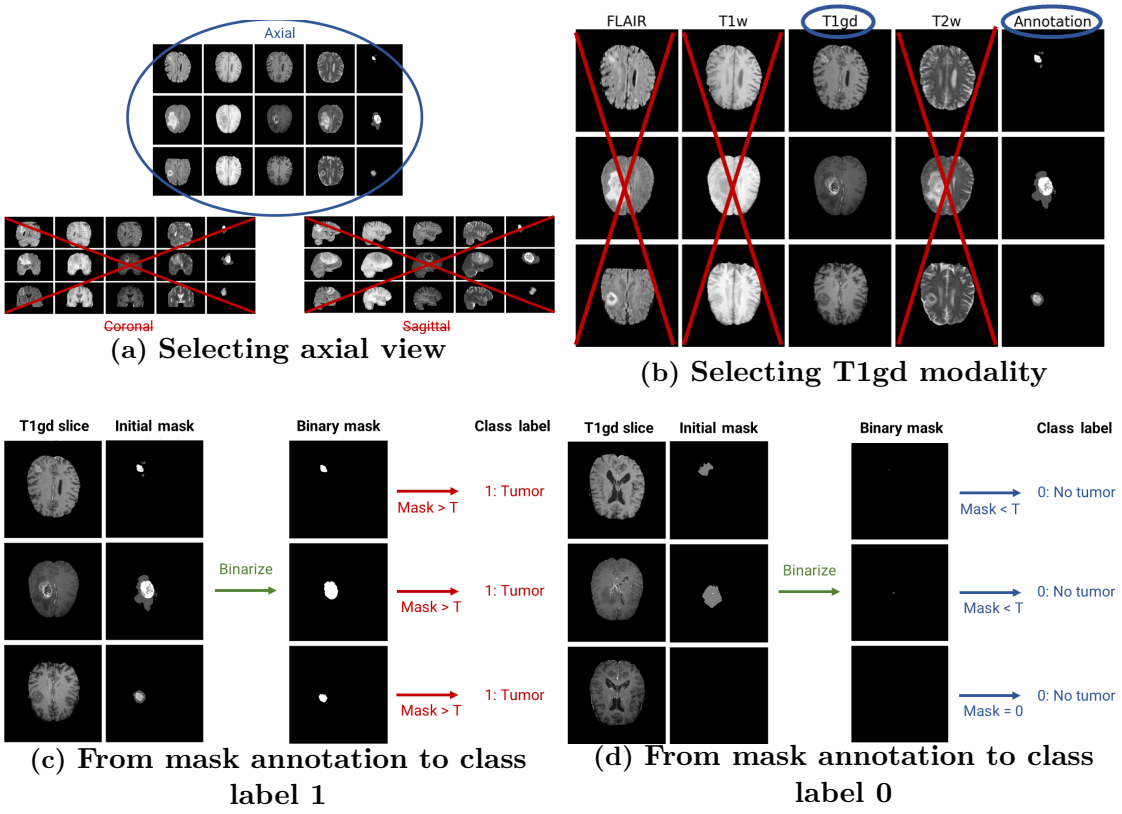


Figure 7.1: Illustration of the brain MRI dataset preprocessing. The different steps of the preprocessing – to produce a 2D binary classification dataset from a 3D multi-segmentation labels dataset– are described. We retained only (a) axial views and (b) T1gd volumes. (c) and (d) illustrate the process of obtaining the classification label of an axial slice image given the original multi-level segmentation mask. The threshold T refers to a mask's minimal size to be considered pathological (i.e., with a tumor).

We propose to classify each slice (2D image) along the axial axis as containing either one (or multiple) tumor region(s) [class **1**], or none [class **0**]. We only studied the training set of the Medical Segmentation Decathlon Challenge [Simpson 19], which is composed of 484 exams and comes with the corresponding segmentation masks (no annotations are provided for the test set). Using the four-level segmentation masks, we computed binary ground truth masks by considering edema and background as class **0**, while non-enhancing and enhancing tumors are gathered in one single tumor class (class **1**). In addition, we only consider the T1gd volumes for this binary problem. First, we resample both the 3D T1gd volumes and the binary ground truth annotation masks from size 155 x 240 x 240 to 145 x 224 x 224. We extract the slices along the axial axis (1^{st} axis), remove slices outside the brain (black images), and normalize the images to [0, 1]. Then, we attribute a class label of **1** if a tumor region larger than 10 pixels (0.02 % of the image size) exists on the corresponding annotation slice; and a label of **0** otherwise. This strategy prevents the classifier from identifying any noise as a tumor. Figure 7.1 illustrates the previous preprocessing process. The split consists of 363 training, 48 validation, and 73 test patient exams. It respectively corresponds to 46900 - 6184 - 9424 slice images and a class balancement of : **0**: 75 % - **1**: 25%.

In this problem, the pathology structures (i.e., tumors) often shows very different textures

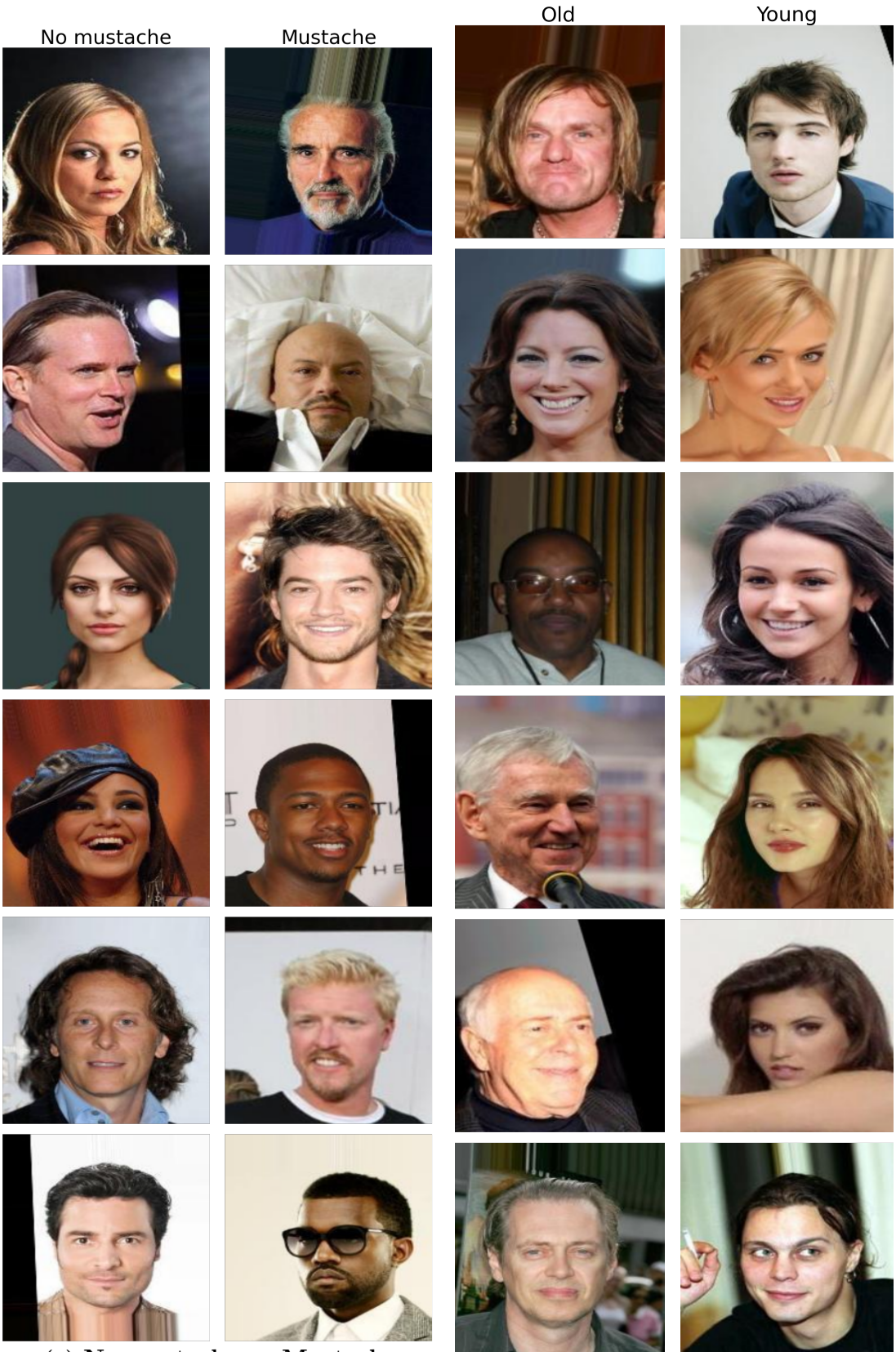
and contrasts compared with the rest of the brain tissues. In this task, the main difficulty is that the texture and contrast of tumors vary. As we aim the classify each slice of the MRI, the context of the image changes (depending where the slice is located) and the tumor sizes also vary (e.g., very small 2-dimensional tumors on first slices they appear). We also train a ResNet50 and a DenseNet121 with the same settings as the Chest X-Rays problem described above, except that they are trained to minimize a weighted binary cross entropy and achieve test set AUC values of 0.975 and 0.980.

7.1.1.3 Digits Identification on MNIST: "3 vs 8"

We designed a binary classification task on the MNIST [LeCun 10] dataset that consists in distinguishing digits **3** from digits **8**. We extracted digits **3** and **8** from the original dataset to create training, validation, and test sets of 9585, 2397, and 1003 samples. The original images of size 28 x 28 are normalized to [0, 1]. We use a convolutional network based on LeNet [Lecun 98], with an adapted output for binary classification (sigmoid activation applied on an output layer of size 1). The model is trained to minimize binary cross-entropy. The classifier reaches an AUC very close to 1.0 (and an accuracy of 0.997) on the test set.

7.1.1.4 Binary attributes classification on CelebA

Using the CelebA dataset, we design two binary classification tasks using the annotated attributes: (i) Mustache vs. No mustache; (ii) Young vs. Old. The "Mustache" dataset is composed of 13465, 1684, and 1684 images, respectively, in train validation and test sets. Each set is balanced with 50 % mustache and 50% without a mustache. We also try to decrease the gender bias by setting half men and half women in the set without a mustache. For the second task, train, validation, and test sets are balanced (between young and old) and contain 40000, 5000, and 5000 images. As for the medical image problems described previously, all images are rescaled to 224 x 224 (but with 3 channels for RGB colors) and normalized to [0, 1]. The same ResNet50 model is used and achieves a test set AUC of 0.962 on "Mustache vs. No mustache" and 0.930 on "Young vs. Old". These colored images cover large poses and background variations compared to medical images (see Figures 7.2 bellow).



(a) No mustache vs Mustache

(b) Old vs Young

Figure 7.2: Image samples for the binary attribute classification on CelebA. (a) Samples for the classification "No mustache vs. Mustache". (b) Samples for the classification "Old vs. Young".

7.1.1.5 Synthetic squares identification

We designed a classification task to identify images that contain squares (one or two) against empty images. We especially used this problem to show the capacity of our method to detect biases. We generated two databases: one biased and one not biased. The databases are described in Table 7.1 bellow:

Table 7.1: Synthetic datasets settings.

# Cases (Ratio)	Biased	Not Biased
Total	2000 (100 %)	2000 (100 %)
Healthy	1000 (50 %)	1000 (50 %)
Pathological R	900 (45 %)	333 (16.65 %)
Pathological L	20 (1 %)	333 (16.65 %)
Pathological RL	80 (4 %)	334 (16.7 %)

We then trained a convolutional network to minimize the binary cross entropy loss. For the network, we used 3 downsampling blocks composed of a convolutional layer, a ReLU activation, and a max-pooling operation. Then we flattened the last feature maps and ended the model with a dense layer (single unit). We split the images into 1000 train, 400 validation, and 600 test sets for the two databases. We achieved accuracy scores of 0.823 and 0.988 on the test set of the unbiased dataset for the biased and not biased datasets, respectively.

7.1.2 Multi-classification

we propose primary experiments in the multi-classification setting on MNIST. The dataset is split into train, validation, and test sets of 48000, 12000, and 5000 digit images with the same preprocessing as in the binary case (see Section 7.1.1.3). We also train a similar convolutional LeNet classifier with a 10-dimensional output and minimize a softmax cross entropy loss. This model achieves a categorical accuracy of 0.989 on the test set.

For all the problems, we use the Adam optimizer [Kingma 15] with an initial learning rate of 1e-4. We use a batch size of 128 for LeNet and 32 for both ResNet50 and DenseNet121. During training, random geometric transformations such as zoom, translations, flips, or rotations are introduced. To train the classifiers, we used a computation node with Intel Skylake 8 cores CPU alongside 52 Go of RAM and an NVIDIA Tesla GPU V100 as an accelerator (referred to as GPU NVIDIA Tesla V100 in the following). The classifier's training converges in about 30 minutes for MNIST and 2 to 4 hours for the other problems.

7.2 Implementation of visual explanations

7.2.1 Generators

Inspired by domain transposition approaches, our generators follow an encoder-decoder architecture (i.e., generator backbone) and end with a specific output layer depending on the embodiment and the problem (i.e., grayscale or RGB images) at stake. For instance, in a problem with grayscale images, the generator produces a single channel image concatenated three times to output the final generated image. Indeed, ResNet50 or DenseNet121 are RGB classifiers and expect RGB images as input (with 3 channels). Tables 7.3a, 7.3b and 7.3c illustrate the main generator output blocks found in our experiment. For the encoder-decoder backbone, we mainly focused on UNet-like and common domain transposition architectures.

UNet-like architecture- As introduced in [Isola 17] for image-to-image translation, we used UNet-like [Ronneberger 15] architecture for several implementation of generator models. The architecture follows an encoder-decoder architecture with skip connections, as described in Tables 7.2a and 7.2b, then ends with a specific output layer (see Table 7.3). The encoder path is composed of 3 to 4 downsampling blocks for images at scale 224x224; we used only 2 blocks for the MNIST problems (see architecture in Table B.1). Each encoding block consists of one residual block followed by a max-pooling layer. At each scale level of the decoded path, we concatenate (ResBlockSkipConcat in the Table 7.2b) the upsampled current layer with the corresponding skip connection layer from the encoder path. Then, a residual block is applied while upsampling the layer to the next scale. Dropout (rate of 0.1 - 0.2) may be added at the end of the residual blocks (better results achieved empirically).

Table 7.2: UNet-like generator architecture [Ronneberger 15]. Illustration for input images at scale 3x224x224. For both (a) the encoder and (b) the decoder, we indicate the type of layer/block, normalization inside the block, and the resampling used at the different scales.

(a) Encoder path

Layer	Resample	Norm	Output shape
Input			3x224x224
Conv	-	BN / -	32x224x224
ResBlock	MaxPool	BN / -	64x112x112
ResBlock	MaxPool	BN / -	128x56x56
ResBlock	MaxPool	BN / -	256x28x28

(b) Decoder path

Layer	Resample	Norm	Output Shape
Enc. Fmaps			256x28x28
ResBlockSkipConcat	Upsampling	BN / -	128x56x56
ResBlockSkipConcat	Upsampling	BN / -	64x112x112
ResBlockSkipConcat	Upsampling	BN / -	32x224x224

Conditional generator architectures for domain transposition- As introduced in the Chapter 5, some counterfactual embodiments consider a unique conditional generator (see Section 5.4) rather than two domain-specific generators (see Sections 5.2 or 5.3). The domain transposition literature [Lee 18, Choi 18, He 19, Yu 19, Romero 19, Choi 20] widely uses this strategy. The idea is to condition the generator with the class domain to guide the image transposition. The architecture also consists of an encoder-decoder structure but differs from the UNet-like one. The base encoding path is composed of 2 to 3 downsampling blocks –either residual or convolution blocks– followed by several residual blocks (2 to 4) at the same resolution scale. The decoder has a symmetrical structure to retrieve the input resolution. The generator also ends with a single output

Table 7.3: The Different output Layers of the generator model.

(a) Single grayscale output

Layer	Activation	Output shape
Conv1x1	-	1x224x224
Concat	-	3x224x224
Out. Activation	Sigmoid / Clip / -	3x224x224

(b) Single RGB output

Layer	Activation	Output shape
Conv1x1	-	3x224x224
Out. Activation	Sigmoid / Clip / -	3x224x224

(c) Double-head grayscale output. Same layers for the two heads.

Layer	Activation	Output shape
Conv	-	32x224x224
Conv	-	16x224x224
Conv1x1	-	1x224x224
Concat	-	3x224x224
Out. Activation	Sigmoid / Clip / -	3x224x224

layer (e.g. grayscale or RGB like; see Table 7.3). Several modules can be used to condition the generator from input [Choi 18, Siddiquee 19, Wolleb 20] or latent conditioning [Lee 18, Bass 20] to specific normalization layers [Singla 20, Huang 18]. We tested several generator architectures in our experiments; they are described in Tables 7.4 and 7.5. To avoid the collapse mode of the discriminator and stabilize the adversarial training, we also add noise after each residual block and upsampling block in the decoder path (as in [Karras 19]). Note that content-style disentanglement works [Yu 18c, Lee 18] (see Section 2.2.1) propose similar generator architectures. However, they condition the generator with image styles rather than attributes relying only on the domain class. In this case, the generated transposed image and the input image differ significantly (which is not our objective), i.e., only coarse details are retained (e.g., the object's pose). In contrast, all the fine-grained details and colors are changed.

Table 7.4: Common domain transposition generator architectures with latent conditioning.

(a) Encoder path

Layer	Norm	Output shape
Input		3x224x224
Conv	-	32x224x224
DownBlock	IN / -	64x112x112
DownBlock	IN / -	128x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56

(b) Decoder path - MUNIT-like [Yu 18c]

Layer	Norm	Output Shape
Enc. Fmaps, y		256x56x56, C
ResBlock / ModResBlock	AdaIN / -	256x56x56
ResBlock / ModResBlock	AdaIN / -	256x56x56
ResBlock / ModResBlock	AdaIN / -	256x56x56
UpBlock	IN / -	64x112x112
UpBlock	IN / -	32x224x224

(c) Decoder path - DRIT/ICAM-like [Lee 18, Bass 20]

Layer	Norm	Output Shape
Enc. Fmaps, y		256x56x56, C
ConditionBlock	-	[256 + C / 32]x256x256
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56
UpBlock	IN / -	64x112x112
UpBlock	IN / -	32x224x224

(d) Decoder path - StyleGAN2-like [Karras 20]

Layer	Norm	Output Shape
Enc. Fmaps, y		256x56x56, C
ModResBlock	-	256x56x56
ModResBlock	-	256x56x56
ModResBlock	-	256x56x56
ModUpBlock	-	64x112x112
ModUpBlock	-	32x224x224

Table 7.5: Common domain transposition generator architectures with Image-level conditioning (StarGAN-like [Choi 18]).

(a) Encoder path - StarGAN-like

Layer	Norm	Output shape
Input, Y		3x224x224, C
ConditionBlock	-	[3 + C]x224x224
Conv	-	32x224x224
DownBlock	In / -	64x112x112
DownBlock	In / -	128x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56

(b) Decoder path

Layer	Norm	Output Shape
Enc. Fmaps		256x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56
ResBlock	IN / -	256x56x56
UpBlock	IN / -	64x112x112
UpBlock	IN / -	32x224x224

Hybrid architectures- We also tested to combine these two types of architectures. We either tested a UNet with the encoder-decoder structures of domain transposition technique (but without the conditioning module), or we added skip connections in the conditional generator architectures to compel the generation to only change the most impacting regions of the input.

In Table 7.6, we show what type of generators and output layers are used for the different embodiments from Chapters 4 and 5.

Table 7.6: Generator architectures for the different embodiments. For each embodiment, we precise the type of encoder-decoder backbone used (or tested), the type of output layer, and the number of generators used in the optimization.

Method	Generator backbone	Output type	Nb. generators
Agen 4.1	UNet-like *	Unique	1
SAGen 4.2	UNet-like *	Single head	2
	""	Double head	1
SSyGen 5.1	UNet-like *	Unique	1
SyCE 5.2	UNet-like *	Unique	2
CyCE 5.3	UNet-like *	Unique	2
CyLatentCE 5.4.3	ICAM-like **	Unique	1
	StyleGan2-like **	""	""
	DRIT-like **	""	""
	MUNIT-like **	""	""
CyImageCE 5.4.4	StarGAN-like **	Unique	1
SySCGen 5.5	UNet-like *	Double head	1
CySCGen 5.5	UNet-like*	Double head	2

The table illustrates the main generator architecture tested for each embodiment but does not go further, e.g., it does not show possible variations inside a given backbone class. * The UNet-like can either follow the architecture described in Table 7.2, or an architecture inspired from Tables 7.4 and 7.5 with additional skip connections and without class conditioning y. ** All architectures are considered with and without skip connections. It depends on the constraint we want to apply to the counterfactual generator.

7.2.2 Discriminators

We adversarially optimize a generator and a discriminator model for counterfactual generation optimization (see Chapter 5). For binary classification tasks, we either consider two class-specific discriminators D_0 and D_1, or a unique conditional discriminator D. For class-specific discriminators, D_0 and D_1 consist of an encoder model composed of residual or convolutional downsampling blocks followed by a reduction operation (e.g., feature maps flattening or global average pooling) and a dense linear layer that outputs a single logit vector. For the conditional case, the discriminator strictly follows the architecture described in [Miyato 18] projecting the class embedding at the final layer. Note that only this conditional discriminator is suited for the multi-classification setting. We use Leaky ReLu activation and no normalization layers for the two cases. The two architectures are described in Table 7.7. We used either the sigmoid cross-entropy or the logistic loss [Karras 19] to optimize the discriminators.

Table 7.7: Discriminator architectures.

(a) Common discriminator

Layer	Norm	Output shape
Input		3x224x224
Conv	-	32x224x224
DownBlock	-	64x112x112
DownBlock	-	128x56x56
DownBlock	-	256x28x28
DownBlock	-	512x28x28
Opt. DownBlock	-	512x14x14
Reduction	-	512
Dense	-	1

(b) Discriminator w/ projection [Miyato 18]

Layer	Norm	Output shape
Input, Y		3x224x224, C
Conv	-	32x224x224
DownBlock	-	64x112x112
DownBlock	-	128x56x56
DownBlock	-	256x28x28
DownBlock	-	512x28x28
Opt. DownBlock	-	512x14x14
Reduction	-	512 ($\rightarrow R_x$)
Embed(Y)	-	512 ($\rightarrow E_Y$)
Projection $E_Y.R_x$	-	1 ($\rightarrow P_{XY}$)
Dense (R_x)	-	1 ($\rightarrow O_X$)
Sum ($O_X + P_{XY}$)	-	1

7.2.3 Training procedure and loss parameters

We use the optimizer Adam to train the models. Initial learning rates are set to 1e-4 for the generators and 2e-4 for the discriminators. Random geometric transformations are also applied with the same settings as for the classifier's training. In practice, we tested two optimization schemes for counterfactual generations in the binary classification setting.

Double batch optimization using class-specific discriminators- At each step, we receive two batches of images: one batch with images from χ_0 (i.e., classified in class 0) and one from χ_1 (class 1). We first optimize the unique generator (g_c or g_f) or the couple of class-specific generators (g_0 and g_1), given a batch of images x in the source domain χ_0 and target domain χ_1. We proceed symmetrically after switching the source and the target domains for a batch of images in χ_1. Then, we optimize the two discriminators.

Single batch optimization using a unique generator and a conditional discriminator- At each step, we receive one batch of images from any class (i.e., $x \in \chi$). We first optimize the unique generator (conditional or not) to produce a counterfactual image for each input. Then, we optimize the conditional discriminator to identify real from generated images. We use this second optimization procedure for multi-classification tasks, as well as for adversarial generation (AGen and SAGen) but without the discriminator optimization (no realism property) .

In practice, we adapt the GAN objective when optimizing the generator, and the discriminator, respectively, which is a common technique [Goodfellow 14, Zhu 17, Karras 19]. For instance in SyCE (see equation 5.14):

- Rather than maximizing L_D, g_0 and g_1 are encouraged to minimize the following term L_D^g with weighting parameters λ_D:

$$\begin{aligned} L_D^g(x, g_0, g_1, D_0, D_1) \; = & \; \mathbb{E}_{x \in \chi_0} \; L_{bce}(1, D_1(g_0(x))) \; + \\ & \; \mathbb{E}_{x \in \chi_1} \; L_{bce}(1, D_0(g_1(x))) \end{aligned} \tag{7.1}$$

- The discriminators D_0 and D_1 respectively minimize the term L_D^d that adds a gra-

dient penalty term L_{gp} [Gulrajani 17]:

$$L_D^d = \lambda_D^d L_D + \lambda_{gp}^d L_{gp} \tag{7.2}$$

The training parameters λ_i (for the different total loss functions) are selected through empirical trials to achieve both the optimization objectives and produce the best evaluation results. For the counterfactual embodiments (Chapter 5), the terms of greatest influence are the symmetrical distance L_d^s (or L_d^{cy} when there is no L_d^s term), the GAN term L_D, and the counterfactual classification loss L_f^c. A balance should be found between the λ_i values. In addition, the variation range of these values could be restricted (depending on the problem), especially for the term L_D. Keeping some specific ratios within a certain range also improve the convergence : $\lambda_d^s/\lambda_D = 50$ to 500 and $\lambda_f^c/\lambda_D \sim 1$. Similar ratios are found in other domain translation state-of-the-art works [Bass 20, Siddiquee 19, Wolleb 20]. The other terms help improve the convergence and the produced results, allowing a greater variation in λ_i values. Similar remark for the adversarial embodiment (Chapter 4), except that there is no L_D term. We elaborate more about the weighting parameters λ_i in the Appendix Section B.2.

7.2.4 Path-based computation

For the different variations $\mathcal{E}_{FI}^{v1}$, $\mathcal{E}_{FI}^{v2}$, and $\mathcal{E}_{FI,k_\sigma}^{v2}$ (as well as the integrated gradient variation), the integral is approximated using a Riemann sum. For instance, given the input x (or the stable image $\mathbf{x_s}$) and the counterfactual $\mathbf{x_c}$, $\mathcal{E}_{FI}^{v2}$ is computed through:

$$\mathcal{E}_{FI}^{v2}(x) \approx \frac{(x - \mathbf{x_c})^2}{M} \sum_{m=1}^{M} \left| \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}_m) \right| \tag{7.3}$$

where $\boldsymbol{\gamma}_m = x + \frac{m-1/2}{M}(\mathbf{x_c} - x)$, and M is the number of steps in the Riemann sum. For the regularized version, we apply Gaussian filtering of kernel 28x28 and σ=2.
In our experiments, we study the impact of the number of steps $M = \{2, 5, 10, 50\}$.

7.2.5 Comparison to Baselines

We compare our proposed methods against several state-of-the-art visual explanation approaches:

1. **Gradient** [Simonyan 14]: We use the official implementation and directly backpropagate the gradient of the model's output to the input.
2. **Integrated Gradient (IG)** [Sundararajan 17]: We consider a linear path between the input and a null image reference. We compute and accumulate gradient along this path (using 50 steps).
3. **GradCAM (GCAM)** [Selvaraju 17]: We use the official implementation. We backpropagate the gradients of the model's output to the last convolutional layer. Then we use bilinear upsampling to produce an attribution map at the input scale.
4. **RISE** [Petsiuk 18]: We use the official implementations but tested with either black (official) or gaussian filtering ($\sigma = 10.0$) perturbations.
5. **BBMP** [Fong 17]: We adapt the official settings to better fit with our problem. We look for a mask of size 56 x 56, and then filter it after upsampling (gaussian with $\sigma = 3$) as in [Fong 17]. We generate the explanation mask after 150 iterations. We also use a total variation regularization. The gaussian blur perturbation ($\sigma = 5$) produces better results.

6. **MGen** [Dabkowski 17]: As in [Dabkowski 17], we use a ResNet-50 backbone pre-trained on ImageNet then on the specific task for the encoder part. We adapt the architecture of MGen to a binary classification problem and remove the class embedding input. We follow the directions proposed in [Dabkowski 17] for the training. More specifically, we alternate between gaussian blur perturbation and a mix of constant and random noise. The generator model produces masks of size 112x112 (14x14 for MNIST) that are upsampled to 224x224 (28x28 for MNIST). As in BBMP, additional gaussian filtering helps to remove some artifacts.

While Gradient, Integrated Gradient, GradCAM, RISE, or BBMP do not need any training; the optimization takes about 5 hours for both MGen, AGen, and SAGen, 10 to 15 hours for counterfactual embodiments on the two medical classification tasks (on GPU NVIDIA Tesla V100). Yet, the adversarial and counterfactual methods are ideally suited for real-time situations, which is critical for adoption in daily clinical routines (see Table 7.8).

Table 7.8: Computation time. Average time (in seconds) of visual explanation generation for LeNet on MNIST digits and ResNet-50 on both X-Ray Pneumonia detection and Brain MRI tumor localization. The visual explanations are computed on GPU NVIDIA Tesla V100.

Method	Digits	Pneumonia	Tumor Loc.
Gradient	0.06	2.26	2.05
IG*	0.09	4.06	3.90
GradCAM	0.02	0.50	0.55
RISE*	-	11.85	12.00
BBMP	1.26	17.26	15.38
Mgen	0.03	0.06	0.11
Adversarial Gen.	0.03	0.04	0.04
Counterfactual Gen. w/o St. **	0.03	0.11	0.11
Counterfactual Gen.	0.05	0.14	0.14

The table displays the average time for generating visual explanations for a single instance at a time. * Note that Integrated Gradient (IG) and RISE benefit from batching here while others do not. ** w/o St. means we do not compute the visual explanation with a stable generation.

7.3 Evaluation Methods

To compare the different techniques, we evaluate the quality of both the visual attributions and the generated counterfactual examples (for suited methods). We propose qualitative and quantitative evaluations of the different methods as introduced in Section 2.3. For a good classifier, relevant regions should match the clinicians' expectations. We thus measure the localization performance of the visual explanations. This evaluation also shows if the visual explanation can be used as an additional tool to assist the clinician (pointing to relevant areas). A visual explanation should highlight relevant input regions for the classifier sorted by importance (i.e., feature importance).
To study counterfactual generation, we also assess the quality of the domain transposition techniques.

7.3.1 Localization performance

We consider three metrics to evaluate the localization performance of visual explanation techniques:

1. The Intersection over Union (IoU), the false positive rate (FPR), and the normalized cross-correlation (NCC). The IoU measures the ratio between the intersection and the union of the ground truth annotation and the visual attribution:

$$IoU = \frac{M_{GT} \cap \mathcal{E}}{M_{GT} \cup \mathcal{E}} \tag{7.4}$$

   where M_{GT} is the mask computed with the ground truth annotation. The IoU takes values in [0, 1]; the higher, the better.
2. The False Positive Rate (FPR) corresponds to the proportion of attribution maps that lays outside the ground truth annotations

$$FPR = \left(1 - \frac{M_{GT} \cap \mathcal{E}}{A(\mathcal{E})}\right) \tag{7.5}$$

   where A is the area in 2D (the volume in 3D). The FPR takes values in [0, 1]; the lower, the better.
3. The Normalized Cross-Correlation (NCC):

$$NCC = \sum_{i=1}^{n} \frac{1}{\sqrt{n-1}} \left(\frac{M_{GT,i} - \mu(M_{GT}))}{\sigma(M_{GT})}\right) \frac{1}{\sqrt{n-1}} \left(\frac{\mathcal{E}_i - \mu(\mathcal{E}))}{\sigma(\mathcal{E})}\right) \tag{7.6}$$

   where $M_{GT,i}$ (resp. $\mathcal{E}_i$) are the value the ground truth annotation mask (resp. attribution map) at pixel i. μ and σ are the sample mean and standard deviation.
   The NCC is used in several works [Baumgartner 18, Bass 20, Lanfredi 19] to evaluate the similarity between the ground truth annotation mask and the attribution map. This metric does not depend on the magnitude of the signals.

The IoU and the FPR are computed between binary ground truth annotation (e.g., filled bounding boxes for pneumonia detection and segmentation masks for tumor detection) and a thresholded binary explanation mask. The choice of the threshold depends on the representative size of the annotations (w.r.t the image size) on the dataset. Tables 7.9a and 7.9b show the main statistics of the size of expert annotations on the training set for pneumonia and brain tumor detection problems. We give the results as ratios of the size

of the image.

For Pneumonia expert annotations, bounding box annotations represent 8.8 % (median) of the image size whatever the number of pathological regions annotated; 4.4 % when there is a unique bounding box, and about 14.6% when there are at least 2 different annotations. In addition, bounding boxes are weaker annotations than segmentation masks as they also contain regions that are not included in the opacity (pneumonia signature): they overestimate the pathological regions.

Table 7.9: Annotations statistics - Some basic statistics about the size of the ground truth annotations on the **training set**. The figures are given as ratios (%) to the size of the image.

(a) Pneumonia Detection - Chest X-rays

Annotations	Nb	Mean	Median	Std	Min	Max	25^{th} Perc.	75^{th} Perc.
Bounding Box	All	11.7	8.8	9.4	0.3	60.2	4.5	16.5
	1	5.4	4.4	3.9	0.3	35.3	2.6	7.1
	> 1	16.6	14.6	9.5	1.4	60.2	9.1	22.7

(b) Brain Tumor Localization - MRI

Annotations	Nb	Mean	Median	Std	Min	Max	25^{th} Perc.	75^{th} Perc.
Segmentation	all $(= 1)$	1.5	1.2	1.2	0.04	6.3	0.5	2.2

To capture the variability of the size of the annotations (single and multiple), we choose the following thresholds: the 90th, 95th, and 98th percentiles. These choices match with the bounding box statistics (given that a box also contains non-relevant regions) and act for annotations that cover from 2 to 10% of the size of the image.
Similarly, we set thresholds for the brain tumor problem at the 98th and 99th percentiles for explanation maps. Table 7.9b shows that segmentation masks of the tumors have a much smaller variability i.e. 1.5 ± 1.2 %.
We set the same threshold for the IoU and the FPR metrics.

7.3.2 Features importance evaluation

Although localization performance enables human experts to assess the quality of the visual explanation, it is not enough to translate the importance of features for the classifier. High localization performance does not reflect the capacity of the visual explanation to order regions of the input image w.r.t their importance for the model decision. It only reports on its capacity to find these regions. To evaluate feature importance for the classifier, we use two metrics introduced in Section 2.3 and based on input degradation techniques [Samek 17]:

1. The area over the perturbation curve (AOPC) by progressively perturbing the input, starting with the most relevant regions of the explanation map first (introduced in [Samek 17]).
2. The feature relevance score (R) proposed in [Lim 21] which combines degradation (most relevant first) and preservation (least relevant first) impacts w.r.t. the classifier.

For both metrics, a perturbation method must be set. In our experiments, we use a counterfactual perturbation (as in [Chang 19]). Other perturbations (replacement by zero,

replacement by noise) generate images outside the training distribution and break down all visual explanation methods, rendering their evaluation impossible. As introduced before, the global context (e.g., background, body structures) of medical images has a lower variability compared to natural images (e.g., ImageNet [Deng 09]), and synthetic perturbations have a much greater and uncontrolled impact on classifiers in this domain. Indeed, synthetic perturbations produce images completely outside the training distribution. For instance, in the pneumonia detection problem, replacing the input pixels with black pixels following a random explanation map completely degrades the classifier score after a few steps (much more than any attribution method). Such perturbation can not be used to assess the feature's importance.
To produce fair comparisons, the counterfactual perturbation process for these metrics should be independent of counterfactual generations used to compute the visual explanations. We combine counterfactual generations from the different counterfactual implementations (except the one assessed) and the image-to-image translation approach proposed in [Siddiquee 19]. At each perturbation step, we sample multiple counterfactual examples (10 in our experiments) and average the classifier scores to produce the final score at this step. The two metrics are computed on a balanced subset of 1000 images of the test set.

7.3.3 Domain transposition assessment

Since there is no global consensus on measuring the proximity between real and generated image distributions for domain translation, we use and revisit some methods and propose novel evaluations.

Classification accuracy- We aim to generate counterfactual images in a different domain (e.g., the image distribution of the opposite class for binary classification) with respect to the classifier to explain. Generated images should belong to the distribution of real images from the opposite (or target) class (realism) and be classified in this opposite class. Then, we validate that the domain transposition is at least satisfied for the classifier by measuring the accuracy between the target prediction (i.e., the opposite of the input's prediction) and the classifier's prediction on the generated images. Good accuracy is necessary as we want to explain the classifier, but not sufficient (missing the realism property). Indeed, common adversarial attacks would achieve this objective.

Fréchet Inception Distance [Heusel 17]- This metric assesses the quality of generated images by comparing the distribution of generated images with the distribution of real images. The metrics is based on the Wasserstein-2 distance [Wasserstein 69] between two multidimensional Gaussian distributions: $\mathcal{N}(\mu_r, \Sigma_r)$ and $\mathcal{N}(\mu_g, \Sigma_g)$ for real and generated distribution respectively.

$$FID = ||\mu_r - \mu_g||_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}\right) \tag{7.7}$$

These gaussian distributions are obtained via the last feature maps (i.e., global average pooling before the classification layers) of the Inception network [Szegedy 15] trained on natural images from ImageNet. As pointed out in [Zhou 15], these latest layers tend to mimic humans as object detectors. However, as the Inception network is trained on the ImageNet dataset, the common FID may be used to assess the quality of the generated image compared to the real image but not the quality of domain translation, as the network is not trained on the classification task at stake.

A novel evaluation- Based on the idea of the Fréchet Inception Distance, we propose to learn an embedding function independent of the visual explanation method and from the classifier that can separate in the latent space the distribution of real images predicted in class **0** from real images predicted in class **1**. This embedding is built by training a variational autoencoder (VAE) –encoding images in 100-dimensional vectors μ and $\log(\sigma)$. It is coupled with a multilayer perceptron that learns to classify if the 100-dimensional mean encoded vector μ comes from an image in χ_0 or χ_1 [Biffi 18]. We illustrate the training procedure of the VAE in figure 7.3.

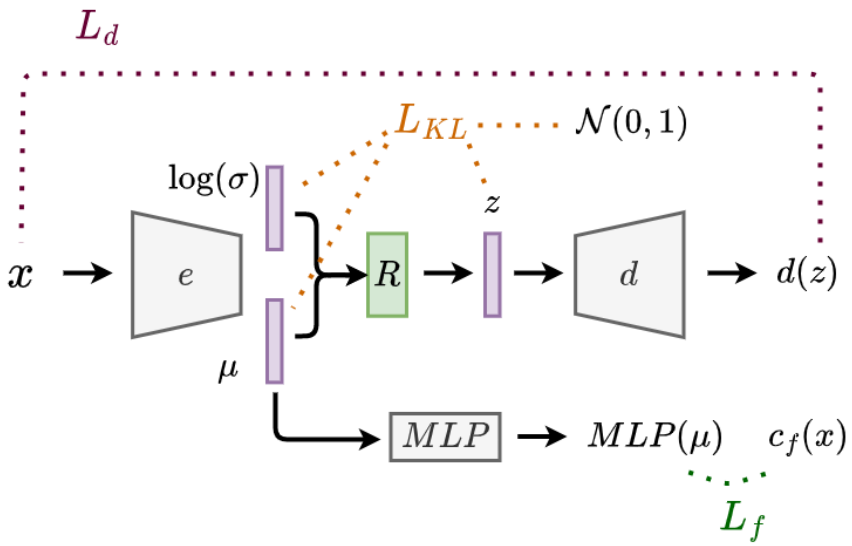


Figure 7.3: Overview of the VAE optimization framework [Biffi 18]. Training step of the VAE for any input image $x \in \chi$. ***Encoder path***: an input image x is given to the encoder e which produces a mean vector (μ) and a pseudo variance vector ($\log(\sigma)$). ***Classification path***: The mean vector is given to an MLP module that is enforced ($Ł_f$) to produce the same classification prediction as the classifier f. ***Reparameterization trick***: the module R apply the reparameterize the encoded vectors (μ and $\log(\sigma)$) into a new vector z using the VAE trick: $z = \mu + \epsilon \exp(\frac{\log(\sigma)}{2})$ ($\epsilon \in \mathcal{N}(0,1)$). The generated vector is enforced to match a Gaussian distribution $\mathcal{N}(0,1)$ through a KL divergence loss. ***Decoder path***: The vector z passes through the decoder to reconstruct the input image (L_d).

We use two different metrics to measure the distance between the encoded distributions. The first adapts the Fréchet Inception Distance [Heusel 17] to our VAE setting. The Inception Network embedding is replaced by the mean vector μ of our VAE to compute the Wasserstein-2 distance. It is denoted by FD_μ. The second is based on kernel probability density estimation [Scott 92] performed on a 2-dimensional Principal Component Analysis applied to the mean vector μ. We then compare real and generated densities using Jenson-Shannon distance (JS) [Endres 03].
In addition to these revisited measures, we also compute the FID metric from an Inception network trained to produce the same predictions as the trained classifier to explain. In this setting, we should easily separate the final layers of the network between encodings of inputs from the different classes.

7.3.4 Biases detection

As presented in [Joshi 18, Singla 20], we can use counterfactual generation to detect training biases. The counterfactual image minimally edits the input to change the classification decision. As generators are trained on a database, the transformations imputed to the input also depends on what impact the classifier, on average, on this database. As the classifier might have learned biases, the counterfactual images should highlight them, especially if these biases simplify the classification tasks.
We can either assess bias detection by qualitative observations of the counterfactual examples or by training an oracle to detect the bias attribute in particular. For instance, [Singla 20] trained a biased and not biased classifier to detect if a person is smiling. In the biased case, the classifier is trained on a database where 80 % of the smiling persons are women. Then, they trained a gender classifier as an oracle applied to the counterfactual generations to confirm this bias. This type of evaluation is not always possible, as annotations to train the oracle may lack.

In the following, we present and discuss our main results. We did not evaluate all the datasets and tasks with all the metrics introduced in this section. For instance, some classification datasets are not suited for localization evaluation (e.g., MNIST) or do not provide localization annotations (e.g., MNIST, CelebA).

8

Results & Discussions

In this chapter, we describe the main results of our proposed methods for the different embodiments, and we compare our work with state-of-the-art techniques (see section 7.2). We evaluate the visual explanation techniques for pneumonia and brain tumor detection problems in the first two sections. In the third section, we study the domain translation for these two medical image problems qualitatively and quantitatively. We also show how the method extends to other classification tasks on MNIST and CelebA. In the fourth section, we illustrate in different examples how our method can identify training biases.
We recall here :

1. The properties enforced in our different approaches and the generator's type we implemented (see Table 8.1).
2. The visual explanation technique and outputs of both our methods and the state-of-the-art approaches to be compared with (see Table 8.2)
3. The metrics used to assess the quality of the visual explanation (see Table 8.3).

Table 8.1: Enforced properties and generator type. For each method introduced in Chapters 4, 5 and 6, we sum up the enforced properties from the general formulation(Chapter 3). We point out what kind of generator is used, and the definition of the counterfactual (g_c) and the stable (g_s) generators on space χ (or sub-spaces χ_0 and χ_1).

	Relevance	Regularity	Realism	Ordered	Gen. type	g_c	g_s
AGen (4.1)	(✓)*				Single Gen.	$g_c(\chi)$	
Duo SAGen (4.2)	✓	✓			Two Gen.	$g_c(\chi)$	$g_s(\chi)$
Single SAGen (4.2)	✓	✓			Single Gen. w/ 2 heads	$g_c(\chi)$	$g_s(\chi)$
SSyGen SP** (5.1)	✓	✓	(✓)		Single Gen.	$g_c(\chi)$	${g_c}^2(\chi)$
SSyGen DP** (5.1)	✓	✓	✓		Single Gen.	$g_c(\chi)$	${g_c}^2(\chi)$
SyCE (5.2)	✓	✓	✓		Two Gen.	$g_0(\chi_0)$, $g_1(\chi_1)$	$g_0^2(\chi_0)$, $g_1^2(\chi_1)$
CyCE (5.3)	✓		✓		Two Gen.	$g_0(\chi_0)$, $g_1(\chi_1)$	
CyLatentCE (5.4.3)	✓	✓	✓		Single Cond.*** Gen.	$g_f(\chi, 1-c_f)$	$g_f(\chi, c_f)$
CyImageCE (5.4.4)	✓	✓	✓		Single Cond.*** Gen.	$g_f(\chi, 1-c_f)$	$g_f(\chi, c_f)$
SySCGen (5.5)	✓	✓	✓		Single Gen. w/ 2 heads	$g_c(\chi)$	$g_s(\chi)$
CySCGen (5.5)	✓	✓	✓		Two Gen. w/ 2 heads	${g_c}^0(\chi_0)$, ${g_c}^1(\chi_1)$	${g_s}^0(\chi_0)$, ${g_s}^1(\chi_1)$
Integ. Counterf. Gen. 6.1	✓	✓	✓	✓	Any**** Counterf. Gen.	Any g_c	Any g_s
IG w/ Counterf. Ref. 6.4	✓	✓	✓	✓	Any Counterf. Gen.	Any g_c	Any g_s

* (✓) means partially enforced.
** SP and DP stands for Single Path and Dual Path. It refers to the practical optimization (see Figures A.1 and 5.2 for SP and DP).
*** Cond. stands for "Conditioned". In CyLatentCE, the domain code is injected in the latent space of the generator, while it is passed at the image-level in CyImageCE.
**** For the two path-based methods introduced in Chapter 6, "Any" stands for "any generator structures used for counterfactual generation in Chapter 5"

Table 8.2: Visual explanation technique and outputs. For each method introduced in Chapters 4, 5, 6 and the state-of-the-art comparison techniques, we indicate if the method provides an attribution map and a counterfactual image. We also point out the outputs of each method and the general technique used.

	Attribution map	Counterfactual	Outputs	Technique
Gradient [Simonyan 14]	✓		Saliency	Gradient backprop.
Integ. Grad. [Sundararajan 17]	✓		Saliency	Path-based + Gradient backprop.
GradCAM [Selvaraju 17]	✓		Saliency	Gradient backprop.* + Upsampling
RISE [Petsiuk 18]	✓		Mask	Perturbation: Multi regions sampling
BBMP [Fong 17]	✓		Mask	Perturbation: Mask optim.
MGen [Dabkowski 17]	✓		Mask	Perturbation: Mask Gen. optim.
AGen	✓		Adv.** image	Adv. Gen. optim + Diff.***
SAGen	✓		St.** & Adv. images	St. Adv. Gen. optim + Diff.
Counterfactual Gen. (5)	✓	✓	St. & C.** images	St. C. Gen. optim + Diff.
Integ. Counterfactual Gen. (6)	✓	✓	St. & C. images + Saliency	St. C. Gen. optim + Path-based

* In GradCAM the output's gradient is backpropagated to the last convolutional layer.
** St. , Adv. and C. stand for Stable, Adversarial and Counterfactual.
*** Diff. means that the attribution map is defined as the difference between the stable (or the input for AGen and CyCE) and the counterfactual (or adversarial) images.

Table 8.3: Evaluation metrics. The different metrics used to evaluate the localization performance, the feature importance and the quality of the generated counterfactual.

Localization	Feature Importance	Counterfactual
Intersection over Union (IoU)	AOPC (2.3)	Accuracy (Acc)*
False Negative Rate (FNR)	Relevance score (R)	Fréchet distance on VAE μ (FD_μ)
Normalized Cross-Correlation (NCC)		Fréchet Inception distance (FID_{tr}**)
		Jenson-Shannon distance (JS)
		Bias detection***

* The accuracy measure if the counterfactual generation is at least classified in the targeted counterfactual domain.
** The index tr means the Inception is trained on the medical task at stake with the classifier's predictions (c_f) as ground truth.
*** Compared with previous metrics measuring the quality of the domain translation, the bias detection task is a qualitative experiment showing the benefit of using counterfactual images to explain a model's decision.

8.1 Localization performance

As described in Section 7.3, a common method to evaluate visual explanation techniques is correlating the feature attributions they produce with human annotations. For a competitive classifier, we expect the highlighted supporting regions of the input image to match human annotations.

Highlights:

- We evaluate the performance of the visual explanation (i.e., the attribution map) to localize annotated pathologies.
- Qualitative results: we display attribution maps for different pathological samples and compare them against the expert annotation. The most (resp. least) relevant regions found by the visual explanation method are shown in red (resp. blue) in the attribution map.
- Quantitative results: we provide IoU, FPR, and NCC metrics.
- Results are given for the pneumonia detection problem on chest X-rays and the brain tumor detection problem on MRI.

8.1.1 Evaluating embodiment 1: Adversarial explanation: AGen and SAGen

Tables 8.4a and 8.4b display the localization metrics (IoU, FPR and NCC) on the two medical datasets for the three adversarial implementations introduced in Chapter 4 then in Section 7.2. First, using a stable generation, the SAGen method outperforms the naive adversarial explanation (AGen) for the two problems and all metrics. Then, the single generator architecture (S) outperforms the double generators version (D) on the two classification problems. This is not surprising because SAGen S compelled the stable and the adversarial generator to capture the same information from the input image by sharing a common autoencoder structure.

Table 8.4: Localization results for AGen and SAGen. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. We highlight in red the best scores.

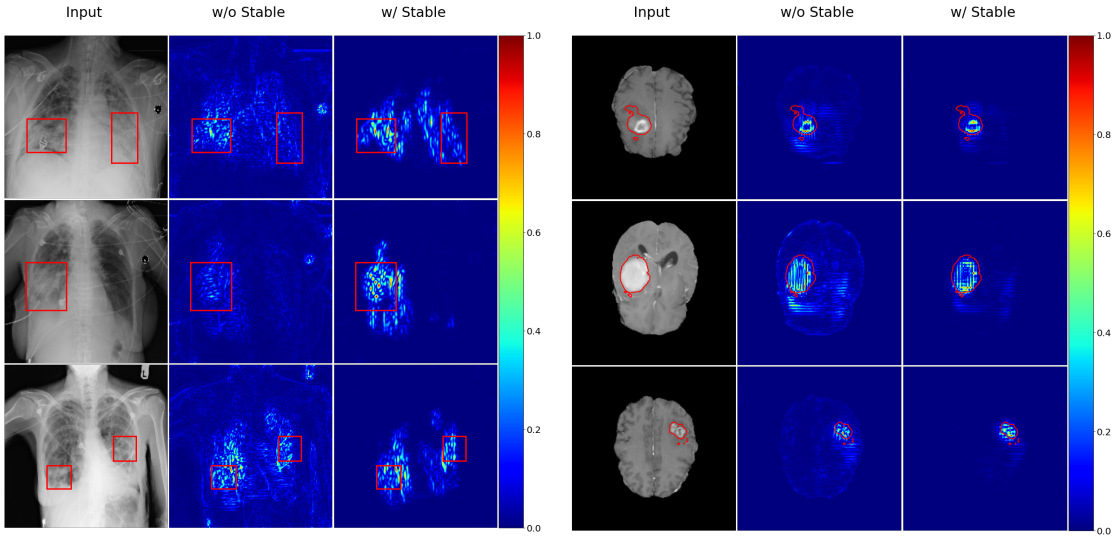
(a) Pneumonia detection

Metric	Perc.	AGen	SAGen		
			D	S	S w/o St.
IoU ↑	90	0.158	0.170	**0.232**	0.159
	95	0.118	0.132	**0.173**	0.122
	98	0.064	0.079	**0.097**	0.069
FPR ↓	90	0.758	0.686	**0.584**	0.691
	95	0.701	0.644	**0.535**	0.647
	98	0.665	0.614	**0.495**	0.626
NCC ↑	-	0.190	0.241	**0.325**	0.224

(b) Brain tumor detection

Metric	Perc.	AGen	SAGen		
			D	S	S w/o St.
IoU ↑	98	0.195	0.242	**0.330**	0.261
	99	0.146	0.224	**0.284**	0.241
FPR ↓	98	0.690	0.632	**0.525**	0.600
	99	0.551	0.491	**0.408**	0.468
NCC ↑	-	0.270	0.457	**0.515**	0.476

The generation process of the stable and the adversarial images is more related in SAGen, which reduces reconstruction errors when taking their difference (i.e., computing the explanation map). The Table C.6 in the appendix shows similarity metrics between input, stable and adversarial generations. It also supports these findings. The last column of



(a) Pneumonia detection

(b) Brain tumor detection

Figure 8.1: Comparison between adversarial explanation maps computed with and without the stable generation. (a) and (b) show examples of pneumonia and brain tumor detection problems, respectively. From left to right for each figure: the input image, the explanation map for SAGen computed without the stable generation displaying plenty of reconstruction errors, and the explanation map for SAGen computed with the stable generation (showing a significant decrease of reconstruction errors and irrelevant details).

each Tables (8.4a and 8.4b) emphasizes that the introduction of the stable generation significantly improves the localization capacity of the adversarial approach (e.g. ≥ 5 points of IoU at the 98th percentile). The Figures 8.1 illustrate how the stable generation allows to remove reconstruction errors from the attribution maps. We also investigate the impact of the weights penalization (L_g^w) and the total variation regularization (L_{reg}) in Equation (4.14). The results are shown in the appendix in Table C.7. It suggests that both terms improve the localization performance of the attribution map: the weight penalization reduces the distance and the reconstruction errors between stable and adversarial images. At the same time, the total variation smooths their differences.

8.1.2 Evaluating embodiment 2: Counterfactual explanation

Counterfactual generation results- Table 8.5 shows the results for the three metrics (IoU, FPR and NCC) on the pneumonia and the brain tumor detection problems.

Table 8.5: Localization results - Comparison between counterfactual techniques. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. For each implementation, we highlight in blue the best score between the explanation computed with and without stable generation. The best scores are shown in red.

(a) Pneumonia detection

Metric	Perc.	St.	SSyGen SP	SSyGen DP	CyCE	SyCE	CyLatentCE SP	CyLatentCE DP	CyImageCE	SySCGen	CySCGen	Ensemble
IoU ↑	90	w/o	0.211	0.291	0.221	0.294	0.271	0.308	0.308	0.292	0.273	**0.337**
		w/	0.230	0.292	-	0.299	0.276	**0.320**	0.310	0.293	0.275	
	95	w/o	0.168	0.227	0.191	0.238	0.212	0.247	**0.257**	0.224	0.228	**0.279**
		w/	0.183	0.229	-	0.244	0.217	0.256	**0.257**	0.224	0.230	
	98	w/o	0.095	0.130	0.116	0.142	0.122	0.140	0.154	0.123	0.132	**0.169**
		w/	0.107	0.132	-	0.151	0.127	0.146	**0.155**	0.123	0.133	
FPR ↓	90	w/o	0.605	0.504	0.596	0.495	0.524	0.488	0.488	0.501	0.524	**0.451**
		w/	0.582	0.502	-	0.492	0.517	**0.474**	0.485	0.500	0.522	
	95	w/o	0.541	0.428	0.494	0.403	0.446	0.405	0.388	0.429	0.422	**0.348**
		w/	0.510	0.425	-	0.399	0.435	0.392	**0.386**	0.428	0.421	
	98	w/o	0.468	0.369	0.430	0.329	0.388	0.351	0.310	0.383	0.367	**0.267**
		w/	0.422	0.364	-	0.319	0.342	0.338	**0.308**	0.383	0.365	
NCC ↑	-	w/o	0.309	0.485	0.337	0.498	0.439	0.500	0.503	0.500	0.414	**0.571**
		w/	0.363	0.490	-	0.506	0.451	**0.516**	0.511	0.500	0.418	

(b) Brain tumor detection

Metric	Perc.	St.	SSyGen SP	SSyGen DP	CyCE	SyCE	CyLatentCE SP	CyLatentCE DP	CyImageCE	SySCGen	CySCGen	Ensemble
IoU ↑	98	w/o	0.197	0.413	0.322	0.406	0.401	0.426	**0.437**	0.406	0.369	**0.468**
		w/	0.205	0.410	-	0.411	0.380	0.409	0.412	0.409	0.369	
	99	w/o	0.165	0.352	0.270	0.345	0.358	0.365	**0.381**	0.353	0.329	**0.399**
		w/	0.166	0.348	-	0.348	0.338	0.349	0.359	0.356	0.330	
FPR ↓	98	w/o	0.683	0.439	0.542	0.474	0.459	0.438	**0.426**	0.450	0.492	**0.396**
		w/	0.673	0.442	-	0.462	0.478	0.451	0.447	0.448	0.492	
	99	w/o	0.613	0.308	0.440	0.368	0.314	0.308	**0.288**	0.323	0.362	**0.263**
		w/	0.612	0.314	-	0.344	0.340	0.324	0.313	0.320	0.361	
NCC ↑	-	w/o	0.377	**0.639**	0.516	0.609	0.623	0.631	0.628	0.629	0.545	**0.685**
		w/	0.382	0.633	-	0.615	0.606	0.625	0.620	0.632	0.546	

We compare the localization performance of all the counterfactual implementations described in Chapter 5. As stated at the end of Section 5.1, the single path version (SP) of SSyGen produces poorer results than other methods. In this implementation, the generator is too constrained and does not produce satisfying generations. The difference between the stable and the counterfactual images contains generation errors. Domain transposition results presented in Section 8.3 support these findings. However, the dual path version (SSyGen DP) seems to solve this issue and even performs among the best localizers in the brain tumor detection problem (see Table 8.5b). Similarly, the single path version of CyLatentCE performs poorly compared to the dual path version (DP). In the single path, at each optimization step, a single generator has to generate a counterfactual and a stable image for a given input of any class, while the conditional discriminator has to learn to discriminate between classes in addition to discriminating between real and generated images. In contrast, in the dual path version, at each step, the generator first learns to translate a batch of inputs from χ_0, then a batch from χ_1. The two discriminators are dedicated to a specific class domain which simplifies their training. The intuition is that

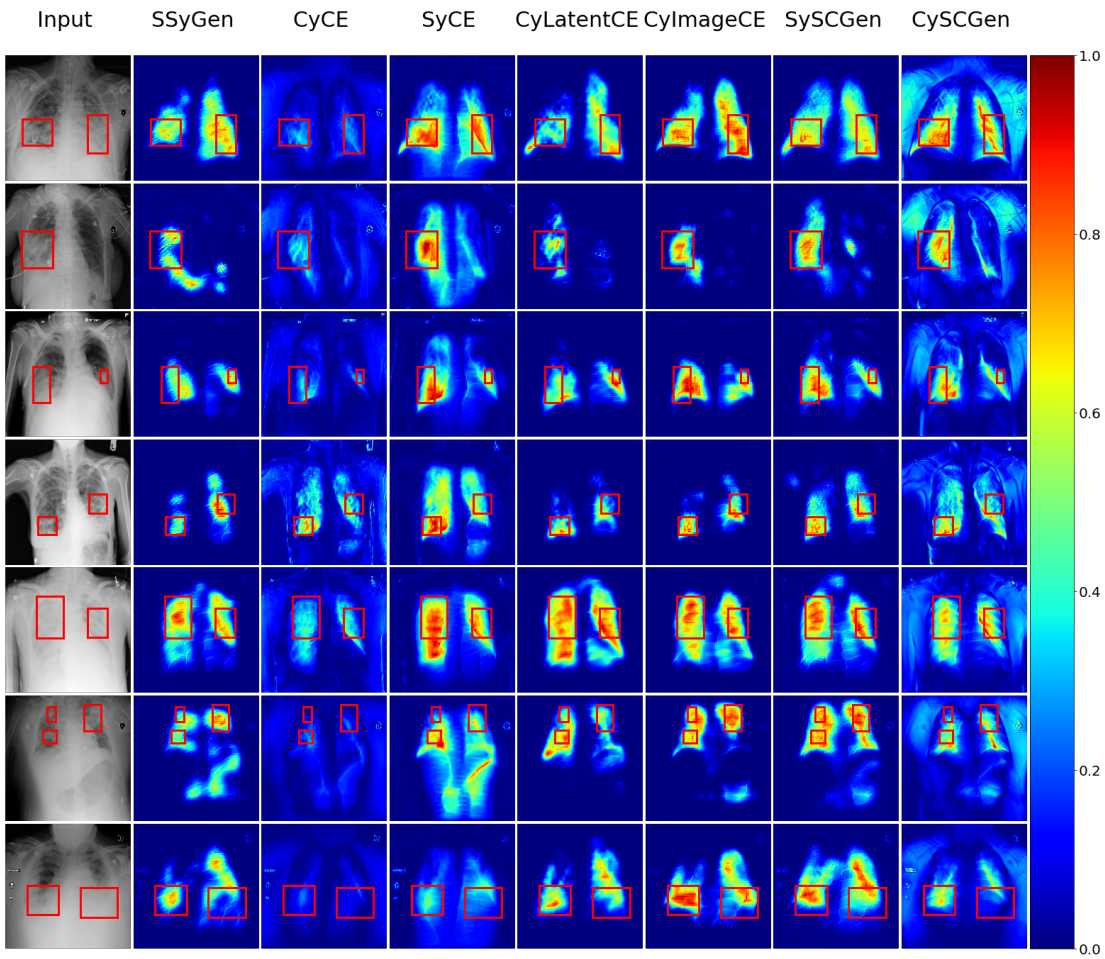


Figure 8.2: Pneumonia detection - Comparison between counterfactual attribution techniques and against ground truth annotations. Ground truth annotations are displayed with red contours. Dual path optimization is used for all the counterfactual methods (heatmaps shown in the different columns).

the dual path is more suited for training the generator model in the binary classification setting by relaxing the optimization. This especially applies to medical image problems where the images of the different classes are mostly similar and only differ in small and fine details.
Then, all the counterfactual techniques perform similarly (SSyGen DP, SyCE, CyLatentCE DP, CyImageCE, SySCGen, and CySCGen) in terms of localization. They outperform the relaxed implementation CyCE where we remove the stable generation and the symmetrical constraints while using two domain-specific generators. Figures 8.2 and 8.3 show some attribution maps for the different methods. CyCE produces more irrelevant attributions (see Figure 8.2) compared to other counterfactual approaches. It underlines the benefit of more constrained approaches. First, CySCGen introduces a stable generation. Irrelevant attributions are also found in CySCGen maps, but the most important differences (orange to red regions) match the ground truth annotations. Then, SyCE uses a symmetrical constraint that compels the generator model to perturb fewer regions than CyCE. These regions are the most discriminative for the classifier. Third, using a single

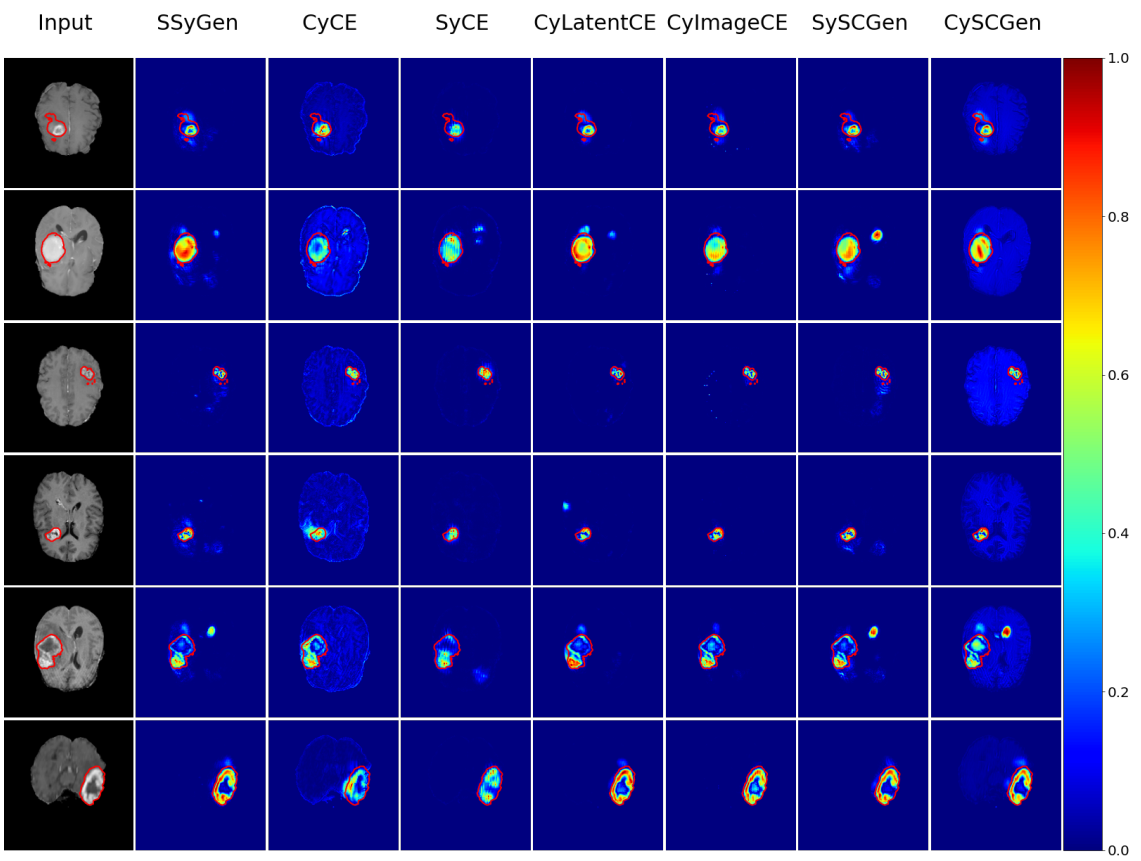


Figure 8.3: Brain tumor detection - Comparison between counterfactual attribution techniques and against ground truth annotations. Ground truth annotations are displayed with red contours. Dual path optimization is used for all the counterfactual methods (heatmaps shown in the different columns).

generator model, SSyGen, CyLatentCE, CyImageCE, and SySCGen apply the strongest constraint. In these maps, only focused regions are changed; the image background is left intact.
Moreover, the stable contribution is less important than in the adversarial embodiment (see previous Section 8.1.1). The realism constraint encourages the generator to produce a counterfactual image by changing relevant regions with realistic patterns of the opposite class. In these two medical problems, adding or removing an opacity (or a tumor) induces significant differences (in intensity) between the input and the counterfactual image. This mitigates the impact of reconstruction errors when comparing directly the counterfactual with the input (w/o St.). On the opposite, in adversarial generations (AGen, SAGen), the differences are not always visible. Yet, using the stable generation slightly improves the localization in most cases except for SSyGen, CyLatentCE, and CyImageCE in the brain tumor detection problem. In these cases, the counterfactual removes the tumor region for input predicted as pathological, but the stable generation sometimes adds a pathological pattern (a tumor) instead of reconstructing the input. We elaborate more about this issue in Section 8.3. Additional figures are provided in the appendices C.1, C.2, C.3 and C.4.

Ensemble approach- A common approach in machine learning systems to boost the performance and robustness of the model is to combine different models and their predictions. We test the same technique and propose an ensemble approach that combines all the constrained counterfactual methods, i.e., SSyGen (DP), SyCE, CyLatentCE (DP), CyImageCE (DP), SySCGen, and CySCGen. We compute the average of all the normalized explanation maps.

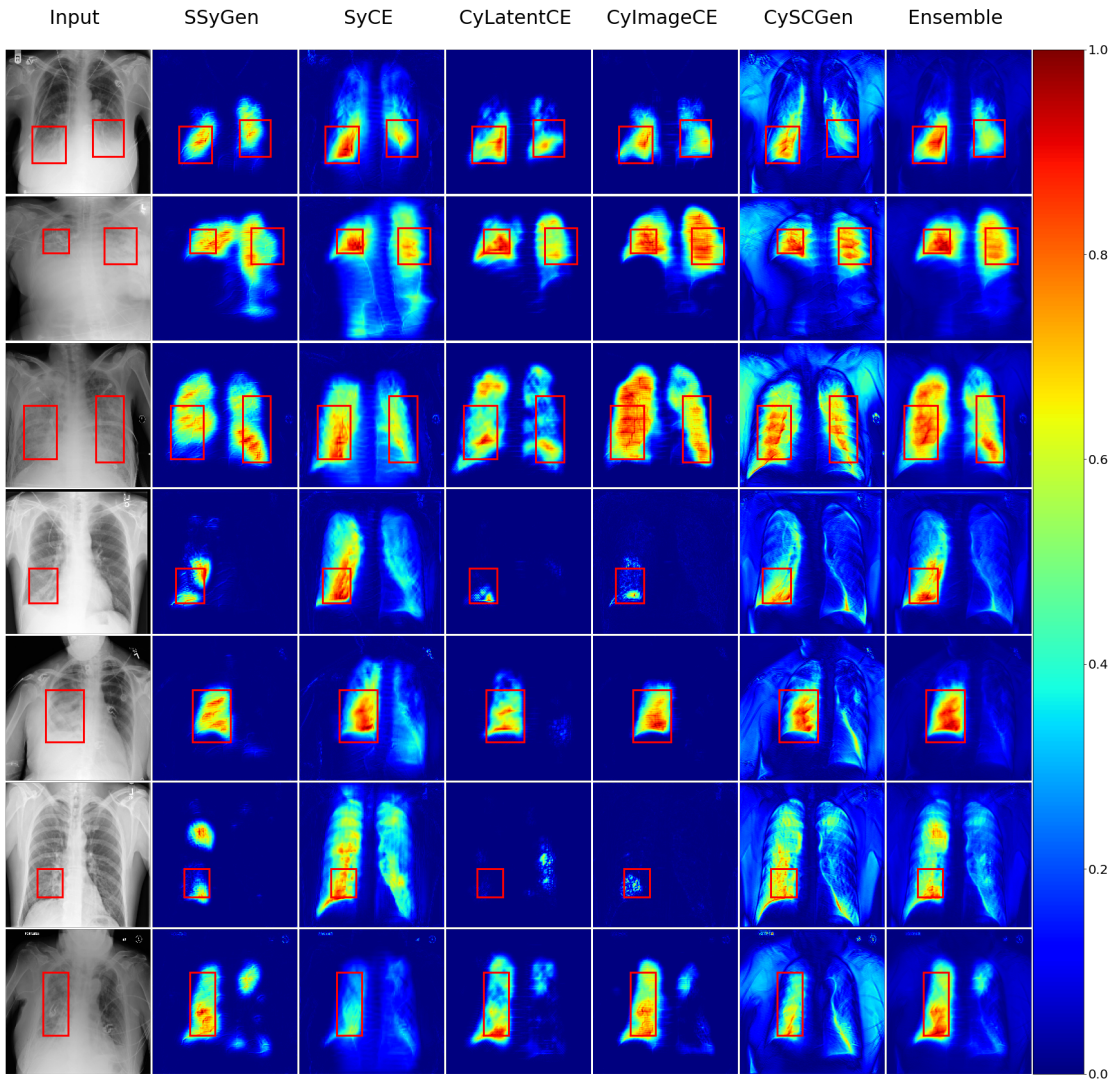


Figure 8.4: Pneumonia detection - Comparison between performing counterfactual techniques and ensemble approach. In the first column: the input image and the ground truth annotations; In columns 2 to 6: diverse counterfactual attributions; and in the last column, the ensemble approach combining the previous techniques.

The localization results are provided in the last column of Tables 8.5a and 8.5b. The ensemble approach outperforms all other techniques. We compare some attributions maps generated by the ensemble approach to other counterfactual methods in Figures 8.4 and 8.5. Although the ensemble maps contain more elements (irrelevant included), they mitigate some errors (e.g. SSyGen and CyLatentCE in the 6th row of Figure 8.4 or the errors made in the 2nd or 5th row of Figure 8.5); they better summarize and point out the most

relevant regions which also match the pathology annotations.

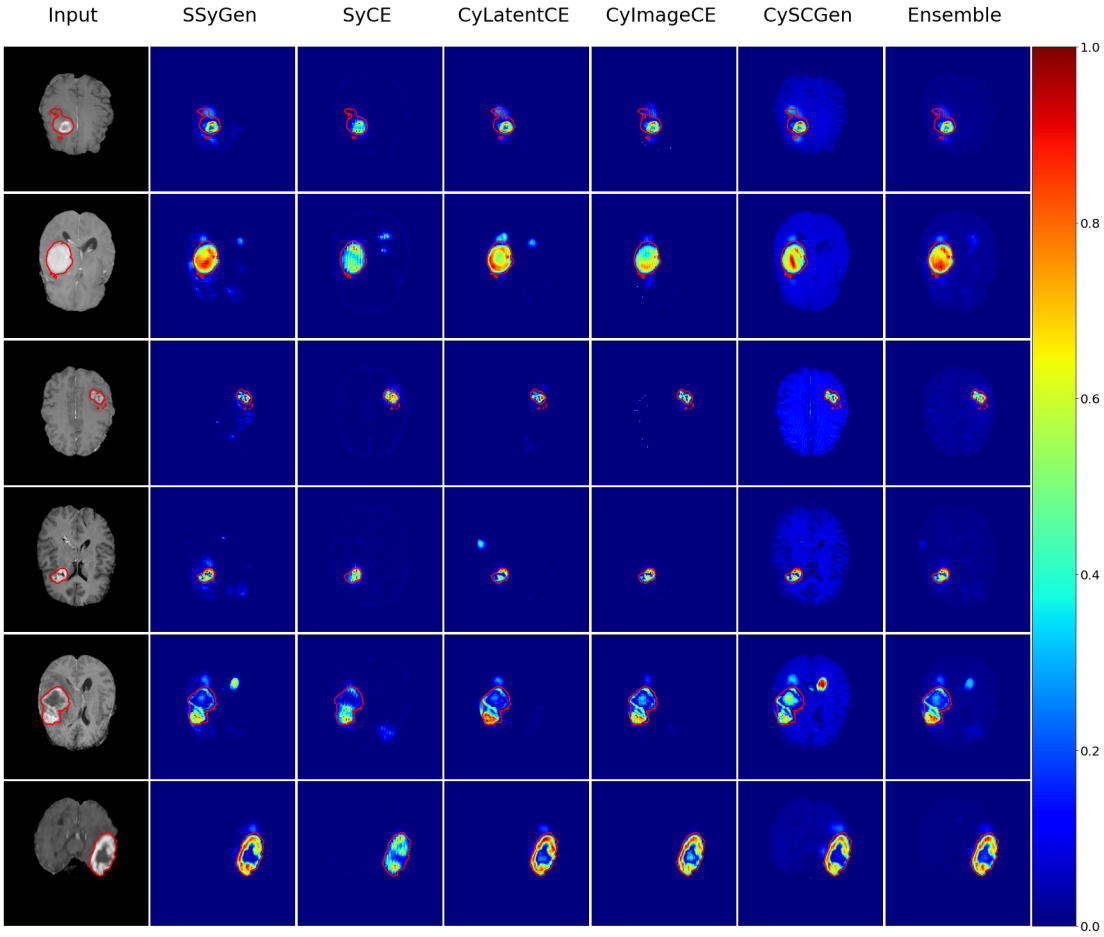


Figure 8.5: Brain tumor detection - Comparison between performing counterfactual techniques and ensemble approach. In the first column: the input image and the ground truth annotations; In columns 2 to 6: diverse counterfactual attributions; and in the last column, the ensemble approach combining the previous techniques.

Ablation Study- To demonstrate the efficiency of our proposed optimizations, we study the impact of the terms of the total cost function (see Equation (5.35)) in CyLatentCE. We recall here the loss function to minimize in CyLatentCE:

$$L_{Total} = \left[\begin{array}{l} \lambda_f^c L_f^c + \lambda_{GAN} L_{GAN} + \lambda_d^{st} L_d^{st} + \\ \lambda_f^{cy} L_f^{cy} + \lambda_d^{cy} L_d^{cy} + \lambda_{cont} L_{cont} \end{array} \right] \tag{8.1}$$

To do so, we performed new optimizations by removing either the stable generation (i.e. removing L_d^{st}), the cyclic constraints (i.e. L_d^{cy} and L_f^{cy}), the classification constraints that guide the transposition (i.e. $L_f^{c,cy}$ and $L_f^{c,cy}$), or all the GAN terms (and the discriminators D_0 and D_1) that enforce realism of the transformation (L_{GAN}).
Tables 8.6a and 8.6b show the localization performances resulting from each optimization, and Figures 8.6 and 8.7 display attributions examples.

Table 8.6: Localization results - Ablation study for CyLatentCE. We compare the impact on localization results of removing different optimization terms from CyLatentCE optimization. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. For each metric, we highlight in bold the best score between explanation computed with and without stable generation, in blue the best score between optimizations (for each line), and in red the best scores.

(a) Pneumonia detection

Metric	Perc.	St.	L_d^{st}	$L_{d,f}^{cy}$	$L_f^{c,cy}$	L_{GAN}	Ours
IoU ↑	90	w/o	0.083	0.217	0.040	0.180	0.308
		w/	-	**0.224**	**0.080**	**0.205**	**0.320**
	95	w/o	0.056	0.170	0.018	0.129	0.247
		w/	-	**0.176**	**0.055**	**0.143**	**0.256**
	98	w/o	0.023	0.094	0.004	0.070	0.140
		w/	-	**0.098**	**0.030**	**0.076**	**0.146**
FPR ↓	90	w/o	0.832	0.629	0.907	0.651	0.488
		w/	-	**0.622**	**0.822**	**0.615**	**0.474**
	95	w/o	0.838	0.587	0.939	0.615	0.405
		w/	-	**0.579**	**0.810**	**0.585**	**0.392**
	98	w/o	0.863	0.562	0.974	0.583	0.351
		w/	-	**0.553**	**0.799**	**0.557**	**0.338**
NCC ↑	-	w/o	0.095	0.384	0.018	0.193	0.500
		w/	-	**0.397**	**0.091**	**0.209**	**0.516**

(b) Brain tumor detection

Metric	Perc.	St.	L_d^{st}	$L_{d,f}^{cy}$	$L_f^{c,cy}$	L_{GAN}	Ours
IoU ↑	98	w/o	0.191	0.200	0.328	**0.238**	**0.426**
		w/	-	0.200	**0.340**	0.237	0.409
	99	w/o	0.148	0.175	0.298	**0.194**	**0.365**
		w/	-	0.175	**0.308**	0.158	0.349
FPR ↓	98	w/o	0.695	0.686	0.535	**0.630**	**0.438**
		w/	-	0.686	**0.522**	0.632	0.451
	99	w/o	0.651	0.603	0.401	**0.551**	**0.308**
		w/	-	0.603	**0.387**	0.618	0.324
NCC ↑	-	w/o	0.333	0.380	0.537	**0.270**	**0.631**
		w/	-	0.380	**0.546**	0.256	0.625

First, our proposed optimization outperforms all the ablated versions for both problems. Second, the impact of each term differs in the two problems.

- Removing the terms L_d^{st} or the cyclic constraints have similar effects as they reduce the implicit proximity constraint on the counterfactual generation. More regions can be changed in the counterfactual, shown in the first two columns of the figures. Yet, removing the stable term degrades the localization results, more importantly in the pneumonia problem. The explanation maps contain many irrelevant regions with the same intensity as the relevant ones.
- Removing the classification terms brings the generation process closer to common domain translation techniques (see Related works Section 2.2.1). However, in the case of medical images, the image of the different classes (or domain) can be very similar. This is the case for the pneumonia problem, where the differences between pathological and healthy cases are only diffuse white regions in the thorax. Here the common domain translation setting is not enough to produce counterfactual images (it fails to generate images of the other class). Poor localization performance and attribution map are reported. In contrast, a tumor region has a different intensity

and texture compared with the healthy tissues of the brain. The generator translates images (even without classification guidance) and replaces the tumor with healthy tissue (in that direction). Localization performances and attributions are poorer compared with our method but remain satisfying. Note that a tumor region is very located and often stands out from the rest of the brain (in intensity).

- Removing the realism property is equivalent to optimizing our adversarial method but with cyclic constraint instead of explicit term between the adversary and the input (as in AGen or SAGen). The attributions are noisy (imperceptible differences between the stable and the adversarial generation), but the localization performances remain correct (compared to other state-of-the-art approaches, see results in Section 8.1.4).

A similar study is performed for CyImageCE in the appendix (see Table C.2 and Figure C.7). It supports the findings reported above, except for the stable term that has a smaller impact on the localization performances.

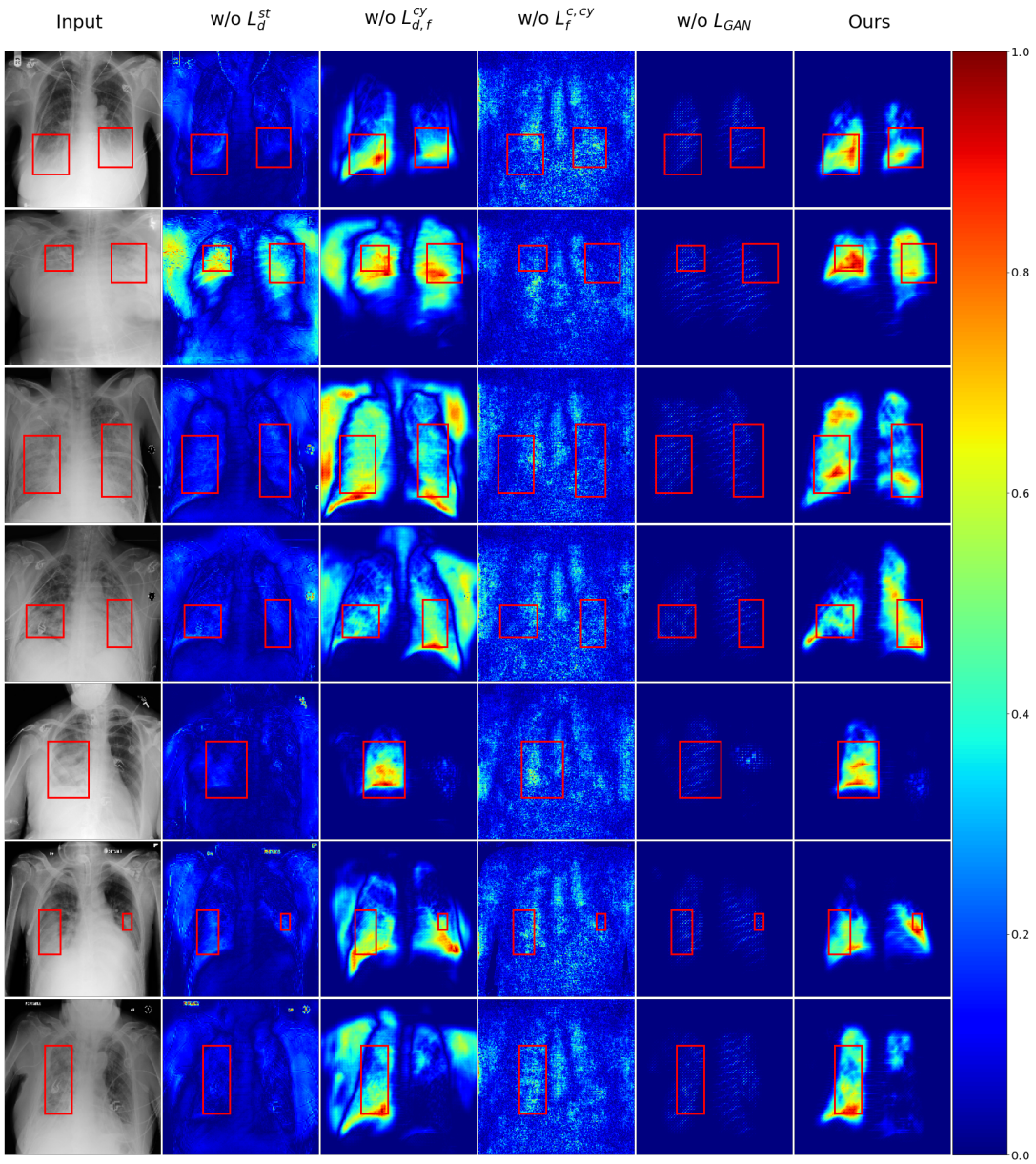


Figure 8.6: Pneumonia detection - Ablation study for CyLatentCE. From left to right: the input image; then the attribution maps from CyLatentCE optimized without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}; without the realism property (i.e. without the GAN term); and our CyLatentCE optimization proposed in Section 5.4.

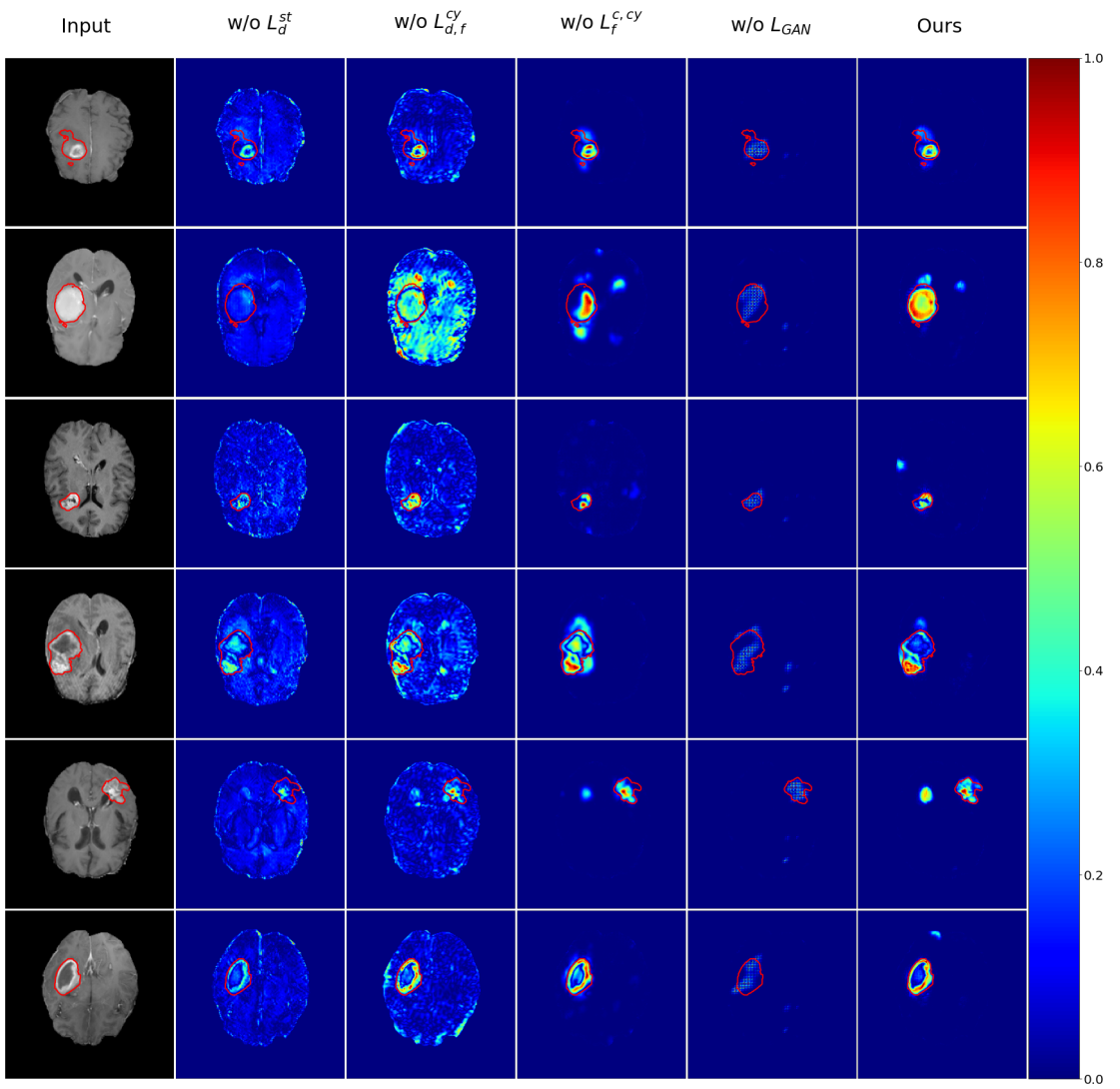


Figure 8.7: Brain tumor detection - Ablation study for CyLatentCE. From left to right: the input image; then the attribution maps from CyLatentCE optimized without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}; without the realism property (i.e. without the GAN term); and our CyImageCE optimization proposed in Section 5.4.

8.1.3 Evaluating embodiment 3: Integrated counterfactual explanation

Table 8.7 reports the localization scores for integrated versions of the main counterfactual generation baseline (we remove the single path versions).

Table 8.7: Localization results - Comparison between counterfactual and integrated methods. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. For each implementation, we highlight in bold the best score between attributions (i.e., baseline and the different integration versions); in blue, the integrated methods outperforming the baseline counterfactual attribution. For each metric, the best score are shown in red.

		Pneunmonia detection							Brain tumor detection				
Metric		IoU ↑			FPR ↓			NCC ↑	IoU ↑		FPR ↓		NCC ↑
Perc.		90	95	98	90	95	98		98	99	98	99	
SSyGen	$\mathcal{E}$	0.292	0.229	0.132	0.502	0.425	0.364	**0.490**	0.413	0.352	0.439	0.308	**0.639**
	$\mathcal{E}_{FI}^{v1}$	0.290	0.233	0.139	0.507	0.422	0.348	0.308	0.407	0.347	0.441	0.308	0.408
	$\mathcal{E}_{FI}^{v2}$	0.306	0.250	0.149	0.489	0.396	0.319	0.367	0.429	0.369	0.422	0.285	0.485
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.316**	**0.262**	**0.159**	**0.478**	**0.378**	**0.297**	0.428	**0.469**	**0.399**	**0.391**	**0.256**	0.619
CyCE	$\mathcal{E}$	0.221	0.191	0.116	0.596	0.494	0.430	0.337	0.322	0.270	0.542	0.440	0.516
	$\mathcal{E}_{FI}^{v1}$	0.269	0.232	0.149	0.532	0.421	0.321	0.337	0.337	0.288	0.518	0.401	0.407
	$\mathcal{E}_{FI}^{v2}$	0.300	0.259	0.163	0.494	0.379	0.285	0.385	0.376	0.322	0.481	0.358	0.483
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.321**	**0.278**	**0.175**	**0.471**	**0.352**	**0.257**	**0.450**	**0.428**	**0.363**	**0.433**	**0.311**	**0.609**
SyCE	$\mathcal{E}$	0.299	0.244	0.151	0.492	0.399	0.319	**0.506**	0.411	0.348	0.462	0.344	**0.615**
	$\mathcal{E}_{FI}^{v1}$	0.289	0.242	0.150	0.507	0.407	0.314	0.374	0.404	0.338	0.462	0.347	0.349
	$\mathcal{E}_{FI}^{v2}$	0.323	0.271	0.165	0.469	0.363	0.275	0.429	0.426	0.357	0.446	0.329	0.435
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.335**	**0.285**	**0.177**	**0.457**	**0.344**	**0.250**	0.483	**0.432**	**0.364**	**0.441**	**0.324**	0.555
CyLatentCE	$\mathcal{E}$	0.320	0.256	0.146	0.474	0.392	0.338	**0.516**	0.426	0.365	0.438	0.308	**0.631**
	$\mathcal{E}_{FI}^{v1}$	0.310	0.250	0.147	0.486	0.398	0.328	0.363	0.436	0.371	0.423	0.289	0.393
	$\mathcal{E}_{FI}^{v2}$	0.325	0.268	0.157	0.468	0.371	0.299	0.363	0.449	0.386	0.412	0.274	0.465
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.337**	**0.278**	**0.165**	**0.456**	**0.359**	**0.282**	0.421	**0.467**	**0.401**	**0.400**	**0.262**	0.590
CyImageCE	$\mathcal{E}$	0.310	0.257	0.155	0.485	0.386	0.308	**0.511**	0.437	0.381	0.426	0.288	**0.628**
	$\mathcal{E}_{FI}^{v1}$	0.308	0.256	0.155	0.489	0.391	0.310	0.315	0.438	0.376	0.422	0.286	0.367
	$\mathcal{E}_{FI}^{v2}$	0.325	0.273	0.165	0.470	0.365	0.283	0.368	0.451	0.390	0.411	0.273	0.435
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.338**	**0.286**	**0.175**	**0.479**	**0.372**	**0.284**	0.350	**0.463**	**0.401**	**0.404**	**0.264**	0.581
SySCGen	$\mathcal{E}$	0.293	0.224	0.123	0.500	0.428	0.383	**0.500**	0.409	0.356	0.448	0.320	**0.632**
	$\mathcal{E}_{FI}^{v1}$	0.292	0.232	0.137	0.502	0.418	0.344	0.339	0.409	0.356	0.442	0.304	0.398
	$\mathcal{E}_{FI}^{v2}$	0.312	0.252	0.147	0.481	0.390	0.320	0.395	0.427	0.374	0.427	0.286	0.470
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.321**	**0.263**	**0.154**	**0.471**	**0.375**	**0.300**	0.446	**0.464**	**0.401**	**0.399**	**0.263**	0.602
CySCGen	$\mathcal{E}$	0.275	0.230	0.133	0.522	0.421	0.365	0.418	0.369	0.330	0.492	0.361	0.546
	$\mathcal{E}_{FI}^{v1}$	0.287	0.248	0.155	0.510	0.396	0.303	0.356	0.370	0.329	0.486	0.345	0.392
	$\mathcal{E}_{FI}^{v2}$	0.319	0.257	0.169	0.472	0.357	0.270	0.416	0.400	0.355	0.459	0.316	0.461
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.336**	**0.291**	**0.181**	**0.454**	**0.335**	**0.244**	**0.476**	**0.445**	**0.383**	**0.422**	**0.289**	**0.587**

First, for all counterfactual techniques, the integrated methods $\mathcal{E}_{FI}^{v2}$ and $\mathcal{E}_{FI,k_\sigma}^{v2}$ improve the IoU and the FPR metrics compared to the baseline $\mathcal{E}$ (shown in blue or red in the table). The first integrated version $\mathcal{E}_{FI}^{v1}$ improves the localization performance in the most relaxed optimizations (i.e., CyCE and CySCGen) and is competitive with the baseline in the more constrained implementations. Except for CyCE and CySCGen, the NCC scores remain better in the baseline $\mathcal{E}$. In all counterfactual techniques, $\mathcal{E}_{FI}^{v2}$ outperforms $\mathcal{E}_{FI}^{v1}$, but the regularized version $\mathcal{E}_{FI,k_\sigma}^{v2}$ is the best localizer (in red). Qualitative examples of attribution maps are given in Figures 8.8, 8.9, 8.10 and 8.11. The figures compare visual explanations from the baseline against the different integration methods for CyCE and CyImageCE. Similar figures are proposed in the appendices for all other counterfactual methods (see appendix Section C.2). Visualizations suggest that the integration technique focuses only on important regions for the classifier, which also correlate with human annotations, and removes irrelevant details (or residual errors) remaining in the baseline. This is especially the case for relaxed optimization CyCE and CySCGen, where we observe the most important gain in localization. However, the non-regularized technique produces a noisier explanation map (a drawback of the gradient approach). This explains why the localization scores between $\mathcal{E}$, $\mathcal{E}_{FI}^{v1}$, and $\mathcal{E}_{FI}^{v2}$ are more comparable for the most constrained

counterfactual techniques (that have already removed most residual errors). In addition, the NCC score considers all values of the explanation map. In the integrated versions, irrelevant features are set at a zero value, which decreases the NCC score for non-zero values in the ground truth annotations. For CyImageCE, we also show attribution produced by the counterfactual integrated gradient variations (see Figures 8.8 and 8.10).

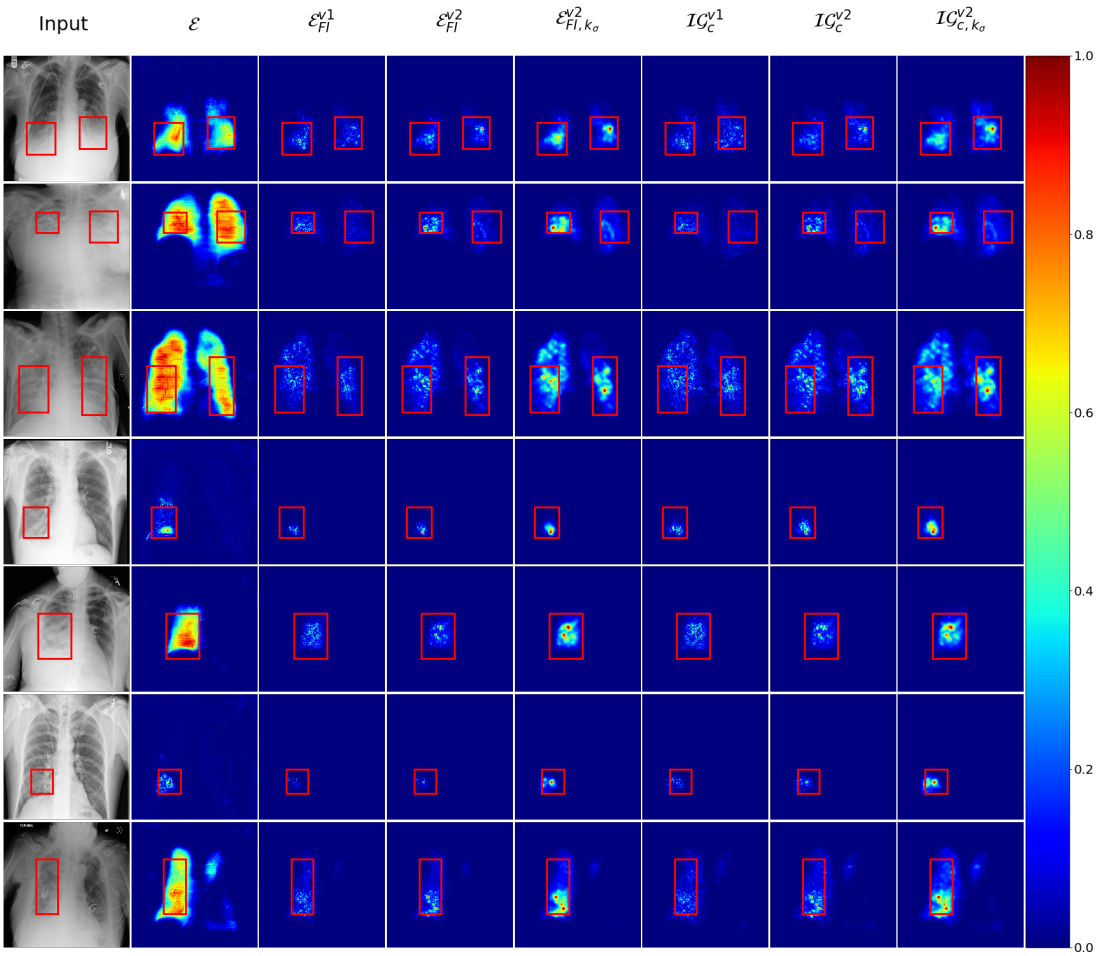


Figure 8.8: Pneumonia detection - Comparison between counterfactual baseline and path-based integration techniques for CyImageCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyImageCE) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

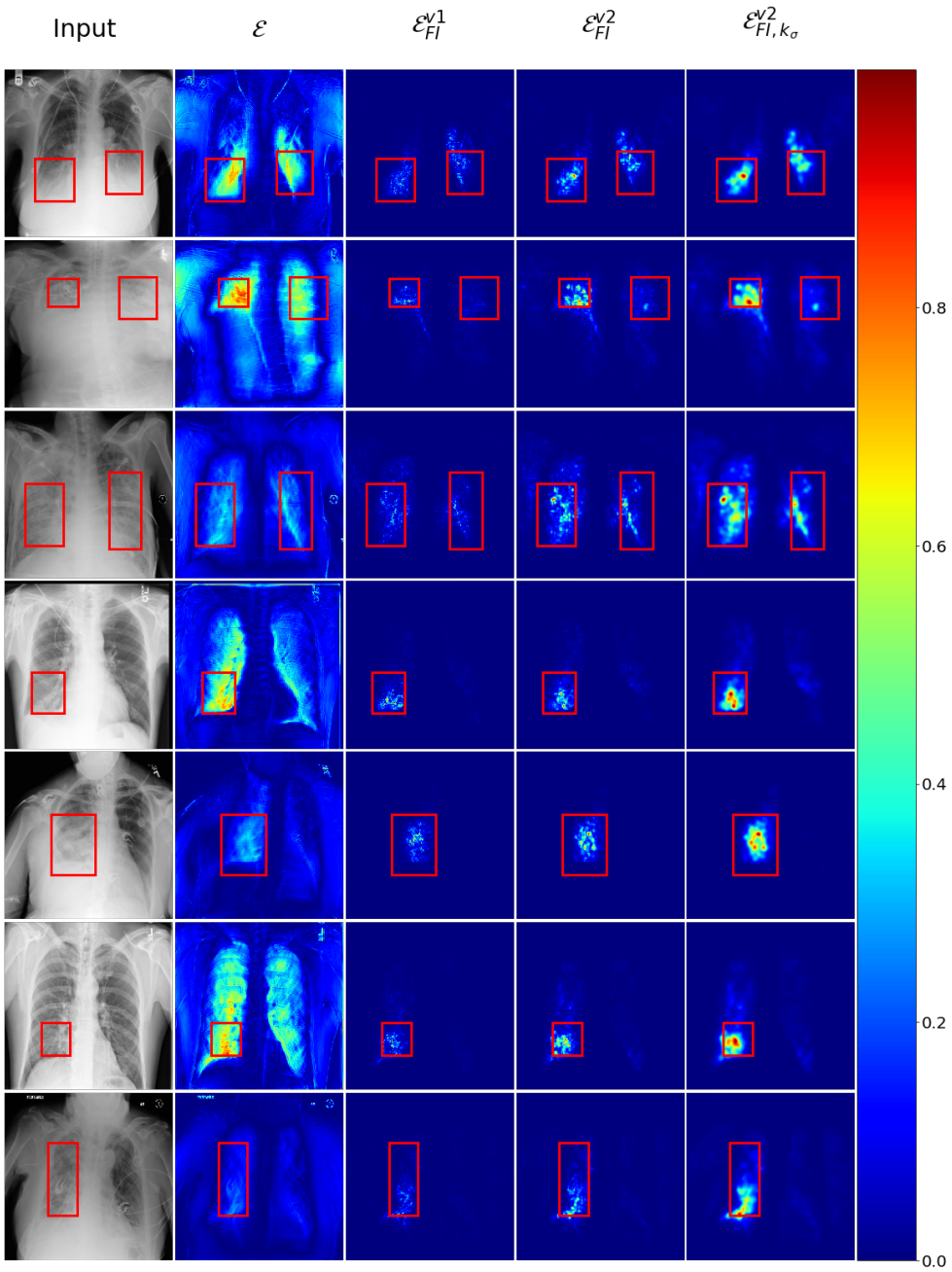


Figure 8.9: Pneumonia detection - Comparison between counterfactual baseline and path-based integration techniques for CyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

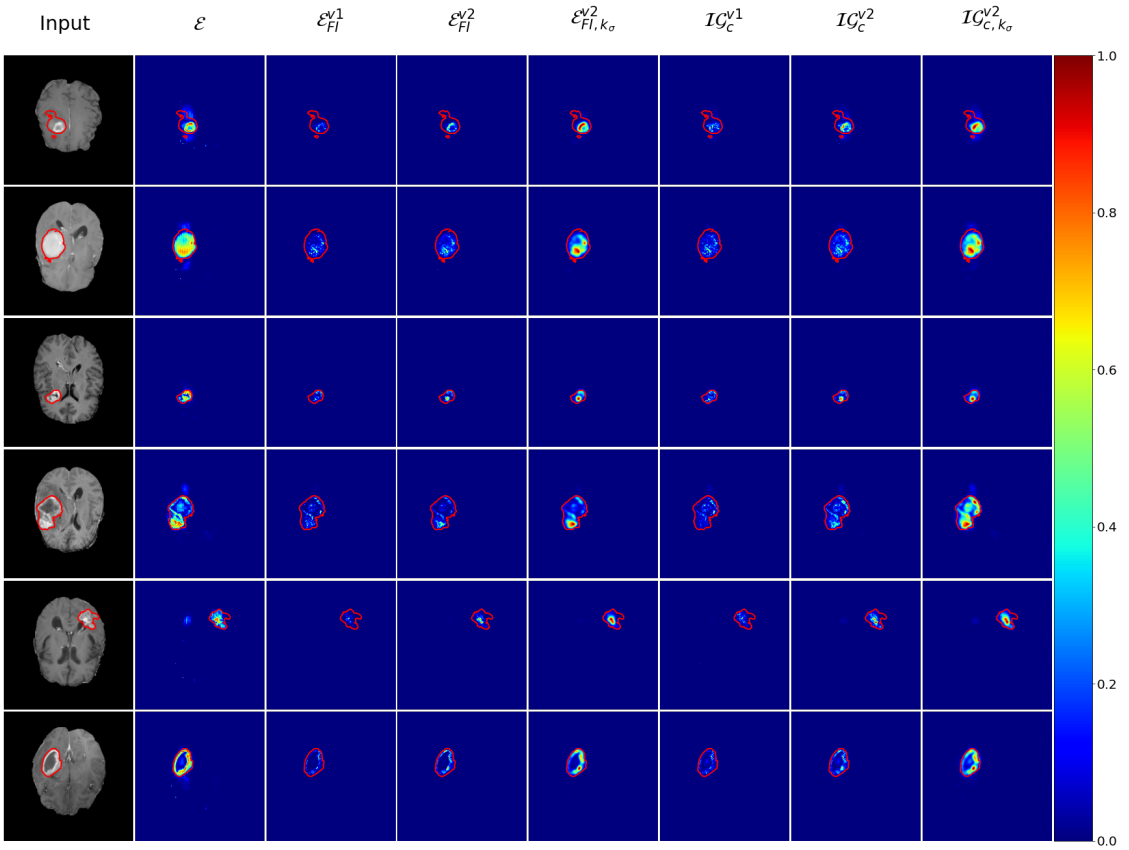


Figure 8.10: Brain tumor detection - - Comparison between counterfactual baseline and path-based integration techniques for CyImageCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyImageCE) computed against the input image; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

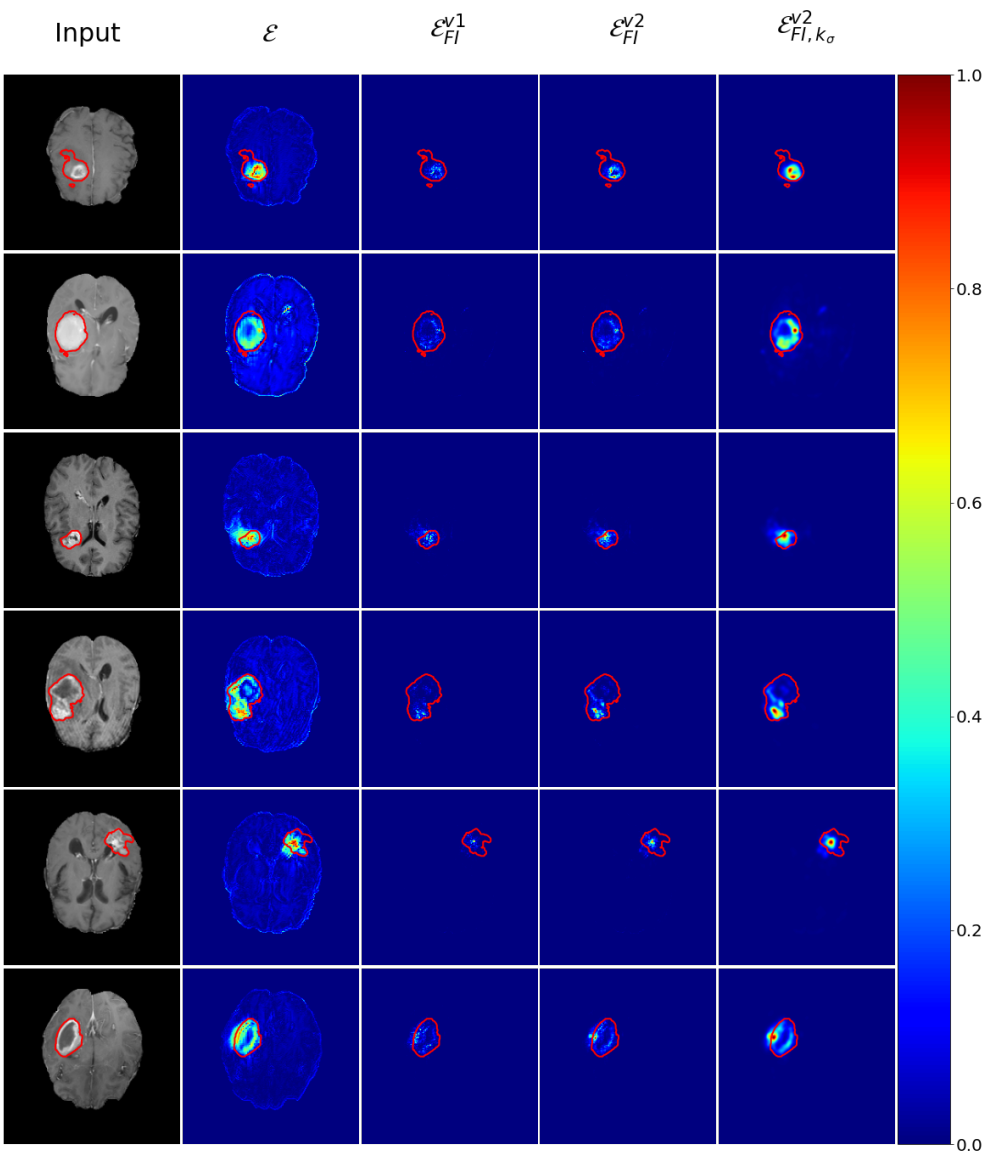


Figure 8.11: Brain tumor detection - Comparison between counterfactual baseline and path-based integration techniques for CyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

8.1.4 Comparison to state-of-the-art techniques

Tables 8.8 show the results for these three metrics, comparing our work against state-of-the-art approaches.

Table 8.8: Localization results - Comparison with state-of-the-art methods. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. We highlight in green the best scores for state-of-the-art techniques. For each of our techniques, we display in blue metrics outperforming state-of-the-art. For each metric, the best score is shown in red. Diff. stands for the counterfactual attribution computed as a "difference" between generations; Integ. stands for "integration".

		Pneunmonia detection							Brain tumor detection				
Metric		IoU ↑			FPR ↓			NCC ↑	IoU ↑		FPR ↓		NCC ↑
Perc.		90	95	98	90	95	98		98	99	98	99	
Gradient		0.187	0.152	0.097	0.639	0.584	0.508	0.312	0.154	0.131	0.744	0.687	0.330
IG		0.170	0.136	0.086	0.698	0.653	0.603	0.254	0.238	0.196	0.621	0.536	0.444
GradCAM		0.195	0.138	0.070	0.645	0.618	0.593	0.325	0.173	0.115	0.715	0.701	0.389
RISE		0.057	0.045	0.026				0.037	0.342	0.335			0.390
BBMP		0.204	0.154	0.087	0.623	0.576	0.537	0.348	0.290	0.263	0.580	0.451	0.409
MGen		0.208	0.169	0.103	0.620	0.542	0.461	0.325	0.318	0.274	0.534	0.413	0.448
SAGen		0.232	0.173	0.097	0.584	0.535	0.495	0.325	0.330	0.284	0.525	0.408	0.515
SSyGen	Diff.	0.292	0.229	0.132	0.502	0.425	0.364	0.490	0.413	0.352	0.439	0.308	**0.639**
	Integ.	0.316	0.262	0.159	0.478	0.378	0.297	0.428	**0.469**	0.399	**0.391**	**0.256**	0.619
CyCE	Diff.	0.221	0.191	0.116	0.596	0.494	0.430	0.337	0.322	0.270	0.542	0.440	0.516
	Integ.	0.321	0.278	0.175	0.471	0.352	0.257	0.450	0.428	0.363	0.433	0.311	0.609
SyCE	Diff.	0.299	0.244	0.151	0.492	0.399	0.319	0.506	0.411	0.348	0.462	0.344	0.615
	Integ.	0.335	0.285	0.177	0.457	0.344	0.250	0.483	0.432	0.364	0.441	0.324	0.555
CyLatentCE	Diff.	0.320	0.256	0.146	0.474	0.392	0.338	**0.516**	0.426	0.365	0.438	0.308	0.631
	Integ.	0.337	0.278	0.165	0.456	0.359	0.282	0.421	0.467	**0.401**	0.400	0.262	0.590
CyImageCE	Diff.	0.310	0.257	0.155	0.485	0.386	0.308	0.511	0.403	0.341	0.462	0.348	0.620
	Integ.	**0.338**	0.286	0.175	0.479	0.372	0.284	0.350	0.457	0.389	0.462	0.348	0.619
SySCGen	Diff.	0.293	0.224	0.123	0.500	0.428	0.383	0.500	0.409	0.356	0.448	0.320	0.632
	Integ.	0.321	0.263	0.154	0.471	0.375	0.300	0.446	0.464	**0.401**	0.399	0.263	0.602
CySCGen	Diff.	0.275	0.230	0.133	0.522	0.421	0.365	0.418	0.369	0.330	0.492	0.361	0.546
	Integ.	0.336	**0.291**	**0.181**	**0.454**	**0.335**	**0.244**	0.476	0.445	0.383	0.422	0.289	0.587
Ensemble	Diff.	0.337	0.279	0.169	0.451	0.348	0.267	**0.571**	0.468	0.399	0.396	0.263	**0.685**
	Integ.	**0.354**	**0.301**	**0.186**	**0.437**	**0.324**	**0.229**	0.506	**0.489**	**0.416**	**0.377**	**0.242**	0.651

SAGen (adversarial approach) is competitive with the best localizers from the state-of-the-art. Then, our counterfactual method (Diff.) and its integrated version (Integ.) outperform all others for the two problems. We found similar results for the Densenet-121 classifier (see Table C.5 in the appendix). The Figures 8.12 and 8.13 compare the visual explanations from the different techniques in the two problems. The counterfactual and integrated counterfactual methods produce explanation maps more attached to the image structures while pointing out different supporting regions that are human-understandable. In comparison, other techniques produce either noisy (Gradient, SAGen) or coarse maps (GradCAM, BBMP); and can even display artifact regions (MGen, SAGen). They often focus on a single region (Gradient, GradCAM) and sometimes highlight irrelevant regions (BBMP, MGen): e.g., the body structures, the spine on the X-rays, or diffuse regions of the brain (inside or even outside). However, the higher values of the attribution maps (in red) from perturbation methods (BBMP and especially MGen) correlate with expert annotations quite well. These last observations are supported by the localization metrics in Table 8.8. We also remind that only perturbation (RISE, BBMP, MGen), adversarial (SAGen), and counterfactual methods are model-agnostic. Gradient and our integrated counterfactual technique need to access the gradients of the classification model, while GradCAM should have access to the whole structure of the neural network. Then, all the

techniques are not suited for testing or validating models from partners that are delivered as black boxes.
Additional figures and results are provided in the appendices C.3.

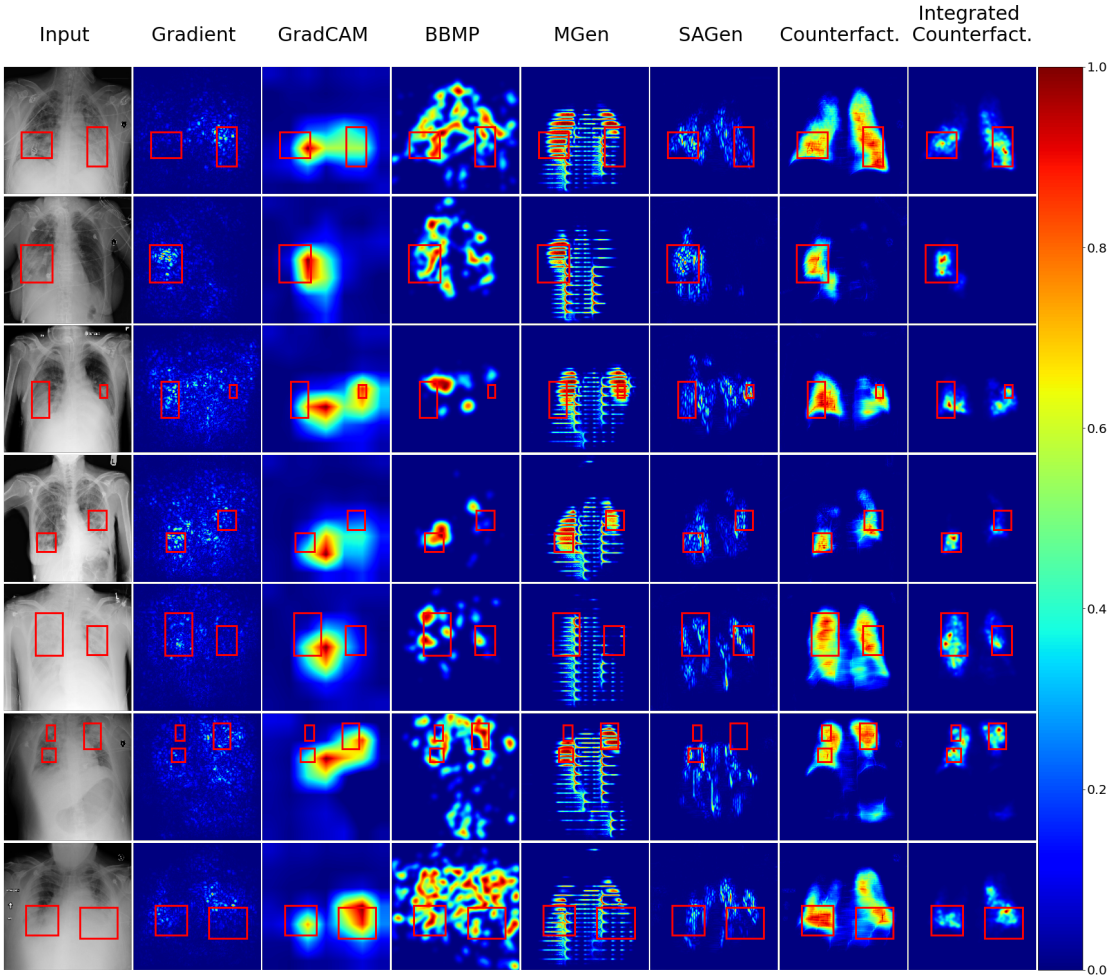


Figure 8.12: Pneumonia detection - Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then the attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen (Chap. 4), a counterfactual explanation (here CyImageCE in Chap. 5), and an integrated counterfactual explanation (Chap. 6) computed with CyImageCE counterfactual.

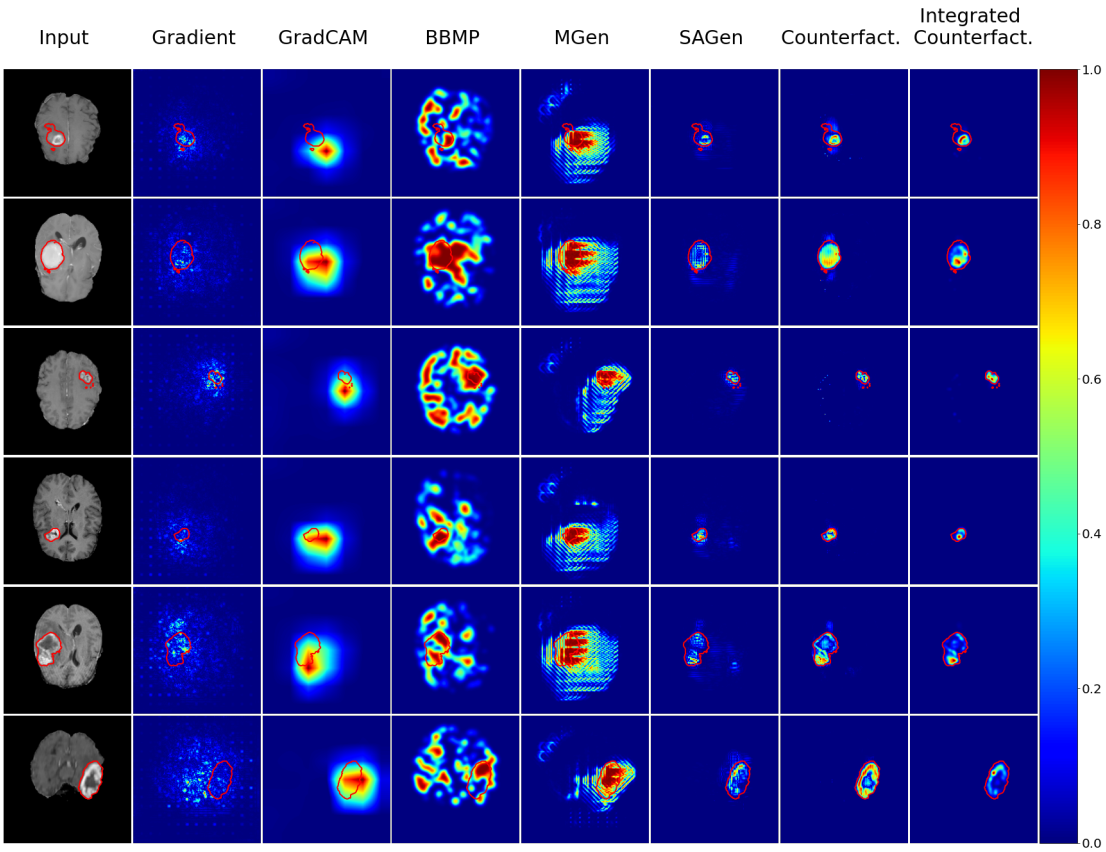


Figure 8.13: Brain tumor detection - Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then the attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen 4), a counterfactual explanation (here CyImageCE in Chap. 5), and an integrated counterfactual explanation (Chap. 6) computed with CyImageCE counterfactual.

8.2 Feature importance evaluation

While localization metrics enable human experts to assess the quality of the visual explanation, we have to measure the importance of the highlighted features (in the explanation map) for the classifier's prediction (via input degradation techniques). Indeed, high localization performance does not translate the capacity of the visual explanation to order regions of the input image with respect to their importance for the classifier's decision. As described in Section 7.3.2, we measure the AOPC score at different perturbation steps, and the feature relevance score R for each predicted class (χ_0 and χ_1) as well as all classes confound. Measuring feature importance for explanation maps of input predicted as pathological (χ_1) is suited for detection problems. Explaining why a classification model found something is more human-understandable than explaining why a model found nothing.

Highlights:

- We evaluate the performance of the visual explanation (i.e., the attribution map) to find relevant input regions for the classification model.
- Quantitative results:
  1. In the figures, we provide AOPC scores computed at different perturbation steps. We remind the AOPC score is defined as the area under the perturbation curve between the initial step (i.e., the input image) and a step L (i.e., the input to which the L most important regions have been perturbed). Satisfying attributions should identify the most relevant regions first (i.e., corresponding to a **rapid growth of the AOPC curve in the first steps**), then secondary regions and so on (i.e. corresponding to the **highest area under the AOPC curves**).
  2. In tables, we provide relevance score R computed for all inputs and inputs classified as pathological or healthy. R combines deletion and preservation measures. In deletion, the most relevant regions (w.r.t. the attribution map) are perturbed first. The model's output should rapidly decrease, and the area D under the deletion curve should be minimal (and $1-D$ maximal). In preservation, the least relevant regions are perturbed first. The model's output should not vary (or little) until it reaches the important regions at the end. The area P under the preservation curve should be maximal. Combining $1-D$ and P, R **is expected to be maximal**.
- Results are given for the pneumonia detection problem on chest X-rays and the brain tumor detection problem on MRI.

$\Diamond$

8.2.1 Comparing between the proposed counterfactual methods

Comparing counterfactual and integrated counterfactual explanations- Table 8.9 shows the feature relevance score R for specific (χ_0 or χ_1) and combined ("ALL") predicted classes. We compare the baseline counterfactual explanation *Expl* with the different integration methods for different counterfactual approaches. Then, Figures 8.14a and 8.14b show the evolution of the AOPC score on the two classification problems for the different visual explanation approaches relative to a random explanation map set as baseline.

Table 8.9: Feature Relevance Score R - Comparison between counterfactual and integration methods. Comparing the different counterfactual and integrated attributions methods on Pneumonia detection and Brain tumor problems. The score R is given for specific predicted classes 0 and 1 and the two combined (All).

		Pneunmonia detection			Brain tumor detection		
Metric		R Score ↑			R Score ↑		
Pred. Class		All	χ_0	χ_1	All	χ_0	χ_1
SSyGen	$\mathcal{E}$	0.621	0.662	0.535	0.775	0.735	0.811
	$\mathcal{E}_{FI}^{v1}$	0.609	0.709	0.422	0.790	0.758	**0.819**
	$\mathcal{E}_{FI}^{v2}$	0.645	**0.724**	0.499	0.794	0.766	**0.819**
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.670**	0.721	**0.585**	**0.803**	**0.786**	**0.819**
CyCE	$\mathcal{E}$	0.588	0.725	0.364	0.631	0.487	0.746
	$\mathcal{E}_{FI}^{v1}$	0.631	0.778	0.388	0.671	0.533	0.781
	$\mathcal{E}_{FI}^{v2}$	0.671	**0.812**	0.432	0.689	**0.563**	0.790
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.708**	0.808	**0.558**	**0.692**	0.542	**0.811**
SyCE	$\mathcal{E}$	0.681	0.783	0.524	0.708	0.611	0.789
	$\mathcal{E}_{FI}^{v1}$	0.641	0.784	0.407	0.735	0.649	0.806
	$\mathcal{E}_{FI}^{v2}$	0.689	**0.823**	0.465	0.737	0.654	0.806
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.726**	0.820	**0.589**	**0.743**	**0.660**	**0.813**
CyLatentCE	$\mathcal{E}$	0.678	0.763	0.553	0.803	0.787	0.818
	$\mathcal{E}_{FI}^{v1}$	0.633	0.716	0.496	0.819	0.808	**0.828**
	$\mathcal{E}_{FI}^{v2}$	0.665	0.741	0.541	0.812	0.814	0.826
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.700**	**0.782**	**0.574**	**0.822**	0.822	0.822
CyImageCE	$\mathcal{E}$	0.684	0.751	0.574	0.797	0.778	0.813
	$\mathcal{E}_{FI}^{v1}$	0.662	0.741	0.498	0.815	0.801	**0.827**
	$\mathcal{E}_{FI}^{v2}$	0.684	0.747	0.576	0.818	0.808	0.826
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.708**	**0.778**	**0.595**	**0.830**	**0.836**	0.823
SySCGen	$\mathcal{E}$	0.613	0.671	0.527	0.771	0.727	0.810
	$\mathcal{E}_{FI}^{v1}$	0.592	0.710	0.403	0.788	0.752	**0.820**
	$\mathcal{E}_{FI}^{v2}$	0.656	**0.768**	0.482	0.788	0.767	0.819
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.680**	0.761	**0.563**	**0.807**	**0.793**	**0.820**
CySCGen	$\mathcal{E}$	0.656	0.740	0.533	0.700	0.618	0.769
	$\mathcal{E}_{FI}^{v1}$	0.630	0.765	0.414	0.756	0.701	0.804
	$\mathcal{E}_{FI}^{v2}$	0.684	**0.806**	0.487	0.777	0.740	0.809
	$\mathcal{E}_{FI,k_\sigma}^{v2}$	**0.704**	0.796	**0.570**	**0.805**	**0.790**	**0.818**

1. On brain tumor detection, our proposed integrated methods improve the counterfactual generation baselines for both the relevance score on the two predicted classes (blue and red in Table 8.9) and the AOPC metric (red, green and yellow curves compared to the blue one in Figure 8.14b).
2. On pneumonia detection, only $\mathcal{E}_{FI}^{v2}$ and $\mathcal{E}_{FI,k_\sigma}^{v2}$ outperform the counterfactual baselines for both relevance score and the AOPC metric (red and green curves compared to the blue one in Figure 8.14a) in the majority of cases. $\mathcal{E}_{FI}^{v1}$ is either competitive or outperforms the baseline for the AOPC score (at least on the first perturbation steps) but produces a poorer feature relevance score (see Table 8.9).
3. For both problems, the regularized version $\mathcal{E}_{FI,k_\sigma}^{v2}$ (red curve) is at least competitive with $\mathcal{E}_{FI}^{v1}$ and $\mathcal{E}_{FI}^{v2}$ (or even outperforms them on the brain tumor problem).
4. We also compare $\mathcal{E}$ and $\mathcal{E}_{FI,k_\sigma}^{v2}$ for the ensemble approach (see the last AOPC curves

of each figure). The two attribution maps achieve similar feature importance results.

5. The AOPC scores from the first two columns of Figures 8.16a and 8.16b provide similar results as the feature relevance scores, when we compare $\mathcal{E}$ and $\mathcal{E}^{v2}_{FI,k_\sigma}$ for the different counterfactual approaches. Feature importance metrics are almost comparable for $\mathcal{E}^{v2}_{FI,k_\sigma}$ for all counterfactual approaches.

Results for the two problems slightly differ. Indeed, all the integration techniques outperform the counterfactual baseline for brain tumor detection. In this problem, the pathological region is very localized and has a different texture and intensity than the rest of the brain. The classification model seems to be sensitive to these types of input features. The integrated methods better focus the high attribution values in these regions of different texture (as reported in Section 8.1.3), while removing residual errors from $\mathcal{E}$. By using counterfactual inpainting, we essentially perturb the input in these tumor regions. The integrated methods thus achieve better feature importance performance (as they only highlight feature in regions with very different texture, i.e., the tumors). In contrast, the pathologies are much more diffuse in the pneumonia problem. Here the model is impacted by the presence of white and opaque regions. Despite improving the focus on relevant regions, the integrated attributions $\mathcal{E}^{v2}_{FI}$ and especially $\mathcal{E}^{v1}_{FI}$ tend to be noisy. In these cases, only top input features (e.g., pneumonia opacity) are removed (or added), which either leave parts of the pathological region (inpainting from χ_1 to χ_0) or add only traces of pathology (from χ_0 to χ_1). The baseline and the regularized integrated methods rather inpaint a global region, which seems to impact the classifier better.

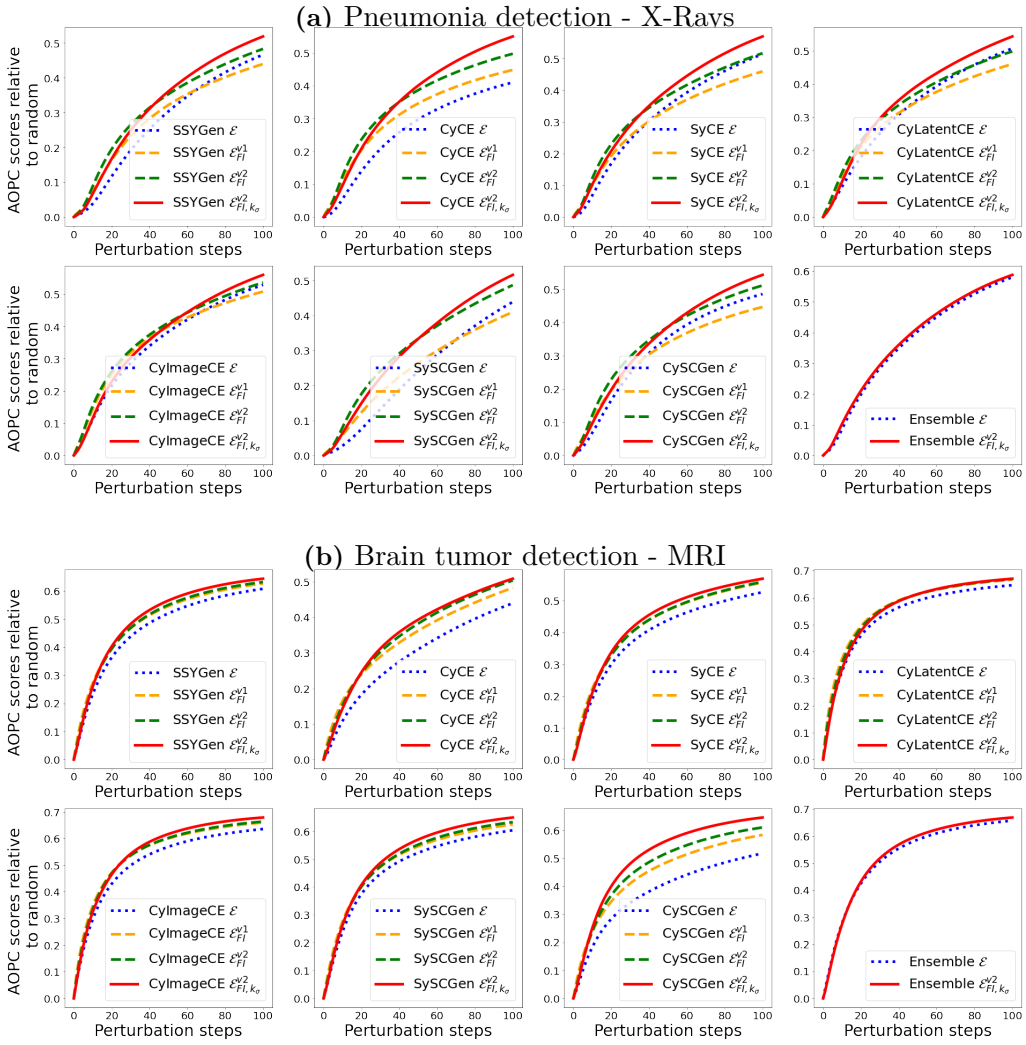


Figure 8.14: AOPC scores relative to random baseline - Comparison between counterfactual methods. (a) Results for the pneumonia detection problem and (b) the brain tumor detection problem. For each counterfactual method (except the ensemble approach at the bottom right of each figure): AOPC curves are displayed for the baseline counterfactual explanation $\mathcal{E}$ (blue), the two integrated techniques $\mathcal{E}_{FI}^{v1}$ (yellow) and $\mathcal{E}_{FI}^{v2}$ (green); and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$ (red). Note that only $\mathcal{E}$ and $\mathcal{E}_{FI,k_\sigma}^{v2}$ are shown for the ensemble approach. The AOPC score at each step is computed relative to the AOPC score of a random explanation map.

Ablation study- We assess the feature importance for the ablation study experiment described in the localization results of Section 8.1.2. We give the evolution of the AOPC scores for CyLatentCE on the two problems in Figures 8.15a and 8.15b. Our proposed optimization method significantly outperforms all the ablated variations. This confirms and supports the localization results reported in Table 8.6, and the observation made on attribution maps from Figures 8.6 and 8.7. The impact of each term also differs in the two problems as they do for localization:

- Removing the terms L_d^{st} or the cyclic constraints results in similar behaviors. More irrelevant features are found in the attribution maps (as the proximity constraints decrease), which is translated in the AOPC curves (blue and yellow). As for localization, the stable term has more impact on the pneumonia problem.
- As for localization, removing the classification terms in the pneumonia problem produces noisy maps containing plenty of irrelevant features. This results in the poorest AOPC scores (green curves). In contrast, tumor regions are modified even without classification guidance. In this case, the AOPC curve (green) outperforms the optimization without L_d^{st} or the cyclic constraints and is competitive with the adversarial ablation (without L_{GAN}).
- Removing the realism property (adversarial equivalent) produces a noisy but focused map. It highlights relevant features as suggests the AOPC scores in the two problems (purple curve).

Similarly, we report the same type of findings from the AOPC scores displayed in Figures D.2a and D.2b in the appendices.

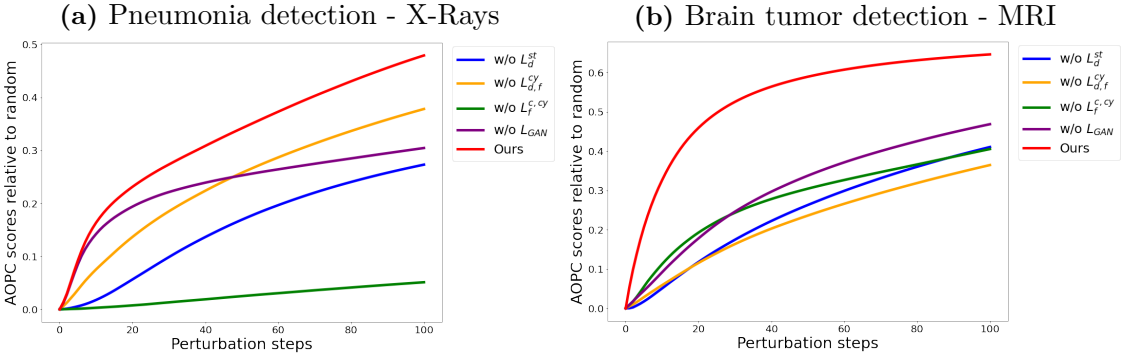


Figure 8.15: AOPC scores relative to random baseline - Ablation study for CyLatentCE. (a) Results for the pneumonia detection problem and (b) the brain tumor detection problem. Results for different CyLatentCE optimizations are compared: optimization without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}; without the realism property (i.e. without the GAN term); and our CyLatentCE optimization.

Alignment with localization findings- In each case of the ablation study, the results from feature importance evaluation align with the corresponding localization performance and tell us more about the classification model sensitivity. For instance, in pneumonia detection, the optimization without realism produces noisy explanation maps but found impacting features (purple curve competitive with the red curve on the first perturbation steps) that match with human expectation (see Table 8.6).
This also aligns with the integration techniques $\mathcal{E}_{FI}^{v1}$ and $\mathcal{E}_{FI}^{v2}$ that are competitive or even

outperform the baseline in the first perturbation steps, then have a decreasing influence. In the same spirit, we compare features importance metrics from the different counterfactual techniques (baseline and integration) in table 8.9 and in the first two columns of Figures 8.16a and 8.16b. We observe that the best (resp. the poorest) localizers (see Table 8.7) also achieve the best (resp. the poorest) feature importance results.

These comparable behaviors (between localization and feature importance) also suggest that the classification models especially focus on input regions that match expert annotations for these two medical problems with localized pathology. In this situation, and despite the lack of consensus, the localization evaluation is suited for assessing the quality of explanation maps in medical image problems.

8.2.2 Comparison against state-of-the-art techniques

The last columns of Figures 8.16a and 8.16b show the evolution of the AOPC score for the different visual explanation approaches. We compare state-of-the-art techniques with our proposed methods. In these figures, we only display the poorest and the best performer for both counterfactual and integrated counterfactual methods. Similarly, Table 8.10 shows the corresponding relevance scores for all these methods. First, our counterfactual methods do not explicitly enforce that high values of the explanation map correlate with high feature importance. However, we observe that both the poorest (SySCGen for pneumonia detection and CyCE for brain tumor detection) and the best performer (the ensemble approach) are competitive or even outperform state-of-the-art approaches in the two medical classification problems (see brown and gray dashed curves in the figures). Second, the regularized integrated method outperforms all state-of-the-art techniques. Then, our proposed adversarial explanation (SAGen) also achieves competitive feature importance results, especially in the brain tumor detection problem.

We provide additional results and figures in the Appendix D.

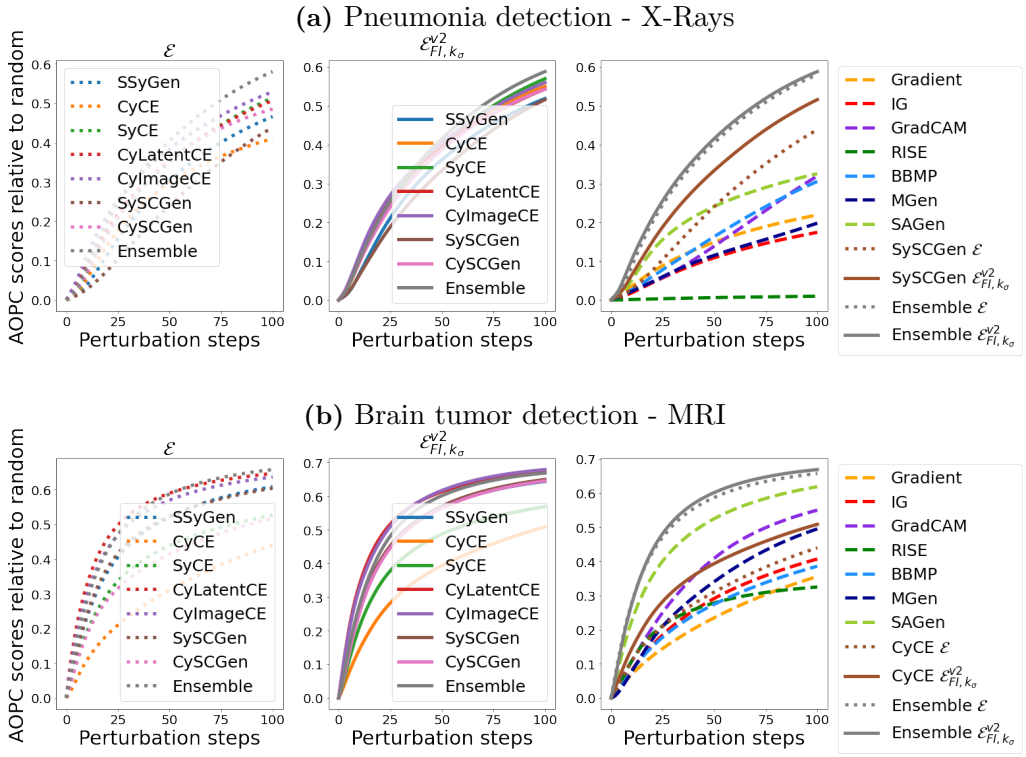


Figure 8.16: AOPC scores relative to random baseline - Comparison with state-of-the-art methods. (a) Results for the pneumonia detection problem and (b) the brain tumor detection problem. From left to right: the comparison of AOPC scores between baseline counterfactual methods $\mathcal{E}$; the comparison of AOPC scores between the regularized integrated methods $\mathcal{E}^{v2}_{FI,k_\sigma}$; and the comparison with state-of-the-art techniques as well as SAGen. In the last column, only the poorest and the best performers from the counterfactual methods are displayed. The AOPC scores are relative to random.

Table 8.10: Feature Relevance Score R - Comparison with state-of-the-art methods. Comparing the different attributions methods on Pneumonia detection and Brain tumor problems. The score R is given for specific predicted classes 0 and 1 and the two combined (All).

		Pneunmonia detection			Brain tumor detection		
Metric		R Score ↑			R Score ↑		
Pred. Class		All	χ_0	χ_1	All	χ_0	χ_1
Random		0.051	0.058	0.042	0.064	0.085	0.046
Gradient		0.381	0.523	0.122	0.551	0.491	0.603
IG		0.324	0.453	0.112	0.602	0.475	0.704
GradCAM		0.393	0.141	0.457	0.601	0.313	0.716
RISE		0.067	0.070	0.060	0.481	0.175	0.731
BBMP		0.467	0.521	0.386	0.577	0.374	0.732
MGen		0.352	0.226	0.483	0.681	0.562	0.758
SAGen		0.492	0.620	0.196	0.783	0.786	0.779
SSyGen	Diff.	0.621	0.662	0.535	0.775	0.735	0.811
	Integ.	0.670	0.721	0.585	0.803	0.786	0.819
CyCE	Diff.	0.588	0.725	0.364	0.631	0.487	0.746
	Integ.	0.708	0.808	0.558	0.692	0.542	0.811
SyCE	Diff.	0.681	0.783	0.524	0.708	0.611	0.789
	Integ.	0.726	0.820	0.589	0.743	0.660	0.813
CyLatentCE	Diff.	0.678	0.763	0.553	0.803	0.787	0.818
	Integ.	0.700	0.782	0.574	0.822	0.822	0.822
CyImageCE	Diff.	0.684	0.751	0.574	0.797	0.778	0.813
	Integ.	0.708	0.778	0.595	**0.830**	**0.836**	0.823
SySCGen	Diff.	0.613	0.671	0.527	0.771	0.727	0.810
	Integ.	0.680	0.761	0.563	0.807	0.793	0.820
CySCGen	Diff.	0.656	0.740	0.533	0.700	0.618	0.769
	Integ.	0.704	0.796	0.570	0.805	0.790	0.818
Ensemble	Diff.	0.735	0.814	0.623	0.813	0.801	0.823
	Integ.	**0.741**	**0.832**	**0.610**	0.822	0.818	**0.825**

8.3 Evaluation of domain translation

In this section, we aim to assess the quality of the domain translation imputed to our method. It differs from the two previous sections where we validated the quality of the features attributions produced by our visual explanations. Following the previous section, we first study pneumonia and brain tumor detection problems, then show that our method extends to other classification problems.

Highlights:

- We evaluate the quality of our counterfactual generations in terms of domain translation.
- Qualitative results:
  1. We display counterfactual generations (for different methods) against the input in the figures. Examples are given for the two translation directions (i.e., Pathological to Healthy and inversely).
  2. We show 2-dimensional PCA projections (of the VAE mean vector) in the figures. In blue (resp. in red), we display input images predicted healthy (resp. pathological). The counterfactual projections are plotted in green. We indicate in each plot the source and the target distribution. Better domain translations correspond to a high overlap between the green points and the target distribution. However, the best domain translations and counterfactuals for visual explanation are not necessary equivalents (as Realism is only one property to be satisfied).
- Quantitative results:
  1. In tables, We report classification accuracies of stable and counterfactual generations (against their targeted classification).
  2. In tables, we provide FD_{μ}, JS, and FID_{tr} metrics for domain translation assessment.
- Qualitative and quantitative results are given for the pneumonia detection problem on chest X-rays and the brain tumor detection problem on MRI (see Section 8.3.1).
- We show how the method extends to other image domains in Section 8.3.3. We mainly provide qualitative results.

$\Diamond$

8.3.1 Pneumonia and Brain tumor detection

Classification Accuracy- As introduced in the experiment Chapter (see Section 7.3.3), the adversarial or the counterfactual generations should be classified (by f) in the opposite class compared with the input's prediction ($c_f(x)$) i.e. $1 - c_f(x)$. Similarly, the stable generation should have the same prediction as the input. We enforce these conditions during the training of the generators g_s and g_c. The classification objective w.r.t to the classification model f is at least required to satisfy the domain translation (but not sufficient).

Thus, we measure the accuracy between the classifier's prediction of the input and the generated images to evaluate our methods capacity to produce stable and counterfactual images. Table 8.11 presents the classification accuracy of the classifier predictions on stable generation (Acc_s) and adversaries (Acc_c) for MGen, SAGen and different counterfactual implementations. We also provide another metric ($Acc_c^{\geq 0.2}$) that indicates if the adversary

impacts the classifier's output from at least 0.2 in the target direction (reminding that $f(x) \in [0, 1]$).

Table 8.11: Classification results. Accuracies (Acc_s, Acc_c) and "indicative" accuracies ($Acc_c^{\geq 0.2}$) computed between a target prediction and the model's prediction on the generated image. Acc_s is the accuracy between the model's decision on the input ($c_f(x)$ and on the stable generation. Acc_c is the accuracy between the opposite of the input prediction ($1 - c_f(x)$ and the model's prediction on the adversarial/counterfactual generation. $Acc_c^{\geq 0.2}$ is an accuracy that indicates if the counterfactual/adversarial image changed the model's prediction from at least 0.2 (in the target direction).

Method	Pneunmonia detection			Brain tumor detection		
	$Acc_s \uparrow$	$Acc_c \uparrow$	$Acc_c^{\geq 0.2} \uparrow$	$Acc_s \uparrow$	$Acc_c \uparrow$	$Acc_c^{\geq 0.2} \uparrow$
MGen	-	0.861	0.928	-	0.668	0.774
SAGen	0.981	0.897	0.933	0.957	0.757	0.828
SSyGen (SP)	0.968	0.911	0.939	0.941	0.785	0.847
SSyGen (DP)	0.974	0.890	0.921	0.952	0.734	0.830
CyCE w/o $L_f^{c,cy}$	-	0.261	0.426	-	0.034	0.216
CyCE	-	0.960	0.991	-	0.910	0.950
SyCE	0.977	0.972	0.992	0.966	0.904	0.929
CyLatentCE	0.998	0.961	0.971	0.997	0.935	0.941
CyImageCE	0.999	0.968	0.984	0.999	0.950	0.955
SySCGen	0.999	0.919	0.940	0.995	0.728	0.814
CySCGen	0.999	**0.997**	**0.999**	0.996	**0.953**	**0.963**

First, all the methods trained with a classification target generate adversaries (perturbed, adversarial, or counterfactual images) classified in the opposite class, especially in pneumonia detection. Classification translation is poorer ($\sim 0.67 - 0.79$) on brain tumor detection for several techniques (MGen, SAGen, SSyGen, and SySCGen), but the $Acc_c^{\geq 0.2}$ metric suggests that a larger amount of cases have a significant change of prediction. For a similar architecture and training configuration, we note that CyCE trained without a classification target (CyCE w/o $L_f^{c,cy}$), i.e., a common CycleGAN (as in [Narayanaswamy 20]), produces a much poorer classification results. We also observe that methods based on adversarial or counterfactual generation produce slightly better results than the perturbation mask approach (MGen) using Gaussian blur. Then, relaxed counterfactual techniques that use two specific generators (CyCE, SyCE, CySCGen) or class conditioning (CyLatentCE, CyImageCE) achieve better accuracy compared with more constrained structures (SSyGen, SySCGen). Stable generations achieve good accuracy for all the related techniques with slightly better performance for methods minimizing a stable distance (e.g., CyLatentCE, CyImageCE, CySCGen) rather than a symmetrical distance (e.g., SSyGen, SyCE, SySCGen).

Qualitative and quantitative assessment of domain translation- While the classification translation of counterfactual images is necessary, it is insufficient. The generated images should belong to the distribution of real images from the targeted class (χ_0 or χ_1).

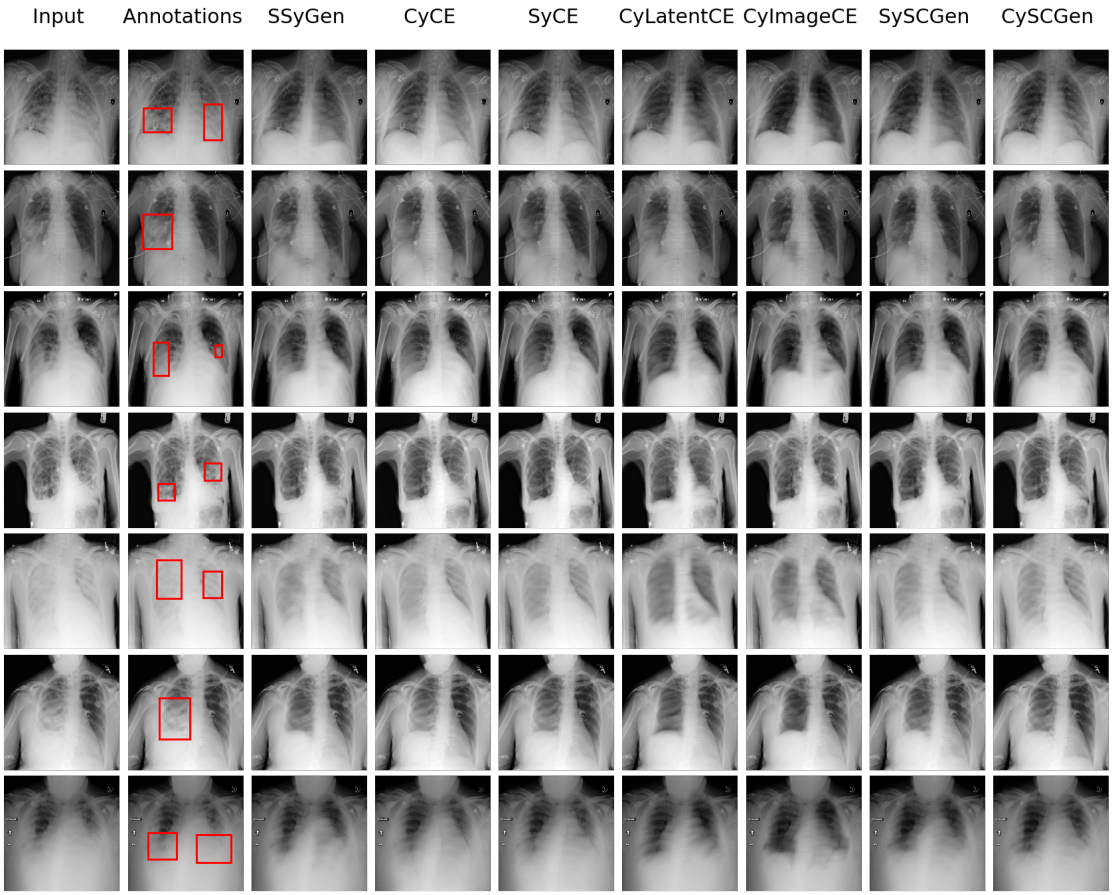


Figure 8.17: Pneumonia detection - Comparison between counterfactual generation techniques: From pathological to healthy image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in Chapter 5. Dual path optimization is used for all the counterfactual methods.

Figures 8.17 and 8.18 display some examples of counterfactual generated images for the different counterfactual approaches and for inputs predicted (correctly) as pathological on both problems. We observe that our approaches generate counterfactual images which are perceived as real images of the opposite domain, e.g., in MR, bright focal regions (tumors) are replaced by darker regions in the counterfactuals (healthy tissue). In contrast, white diffuse opacity regions (inside red bounding boxes) are removed in chest X-Rays.

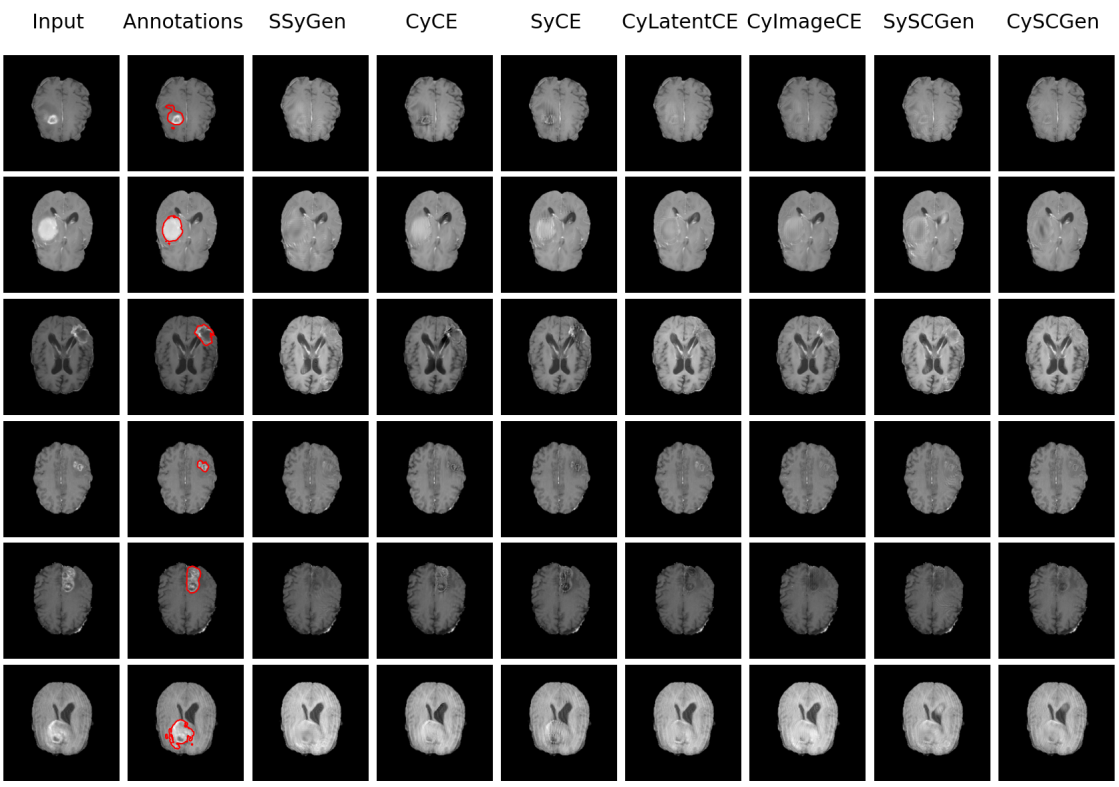


Figure 8.18: Brain tumor detection - Comparison between counterfactual generation techniques: From pathological to healthy image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in Chapter 5. Dual path optimization is used for all the counterfactual methods.

When mapping healthy to pathological images (see Figure 8.19), the counterfactual generated images seem plausible, especially for implementations using two generators (CyCE, SyCE and CySCGen). These techniques generate "pathological" counterfactuals by adding opacities in the pulmonary regions (white and diffuse patterns) or localized white tumors in the brain. The transformations are subtle for more constrained methods, especially in brain tumor detection. Additional illustrations (for both translation directions) are displayed in the appendices Section E.1.1.

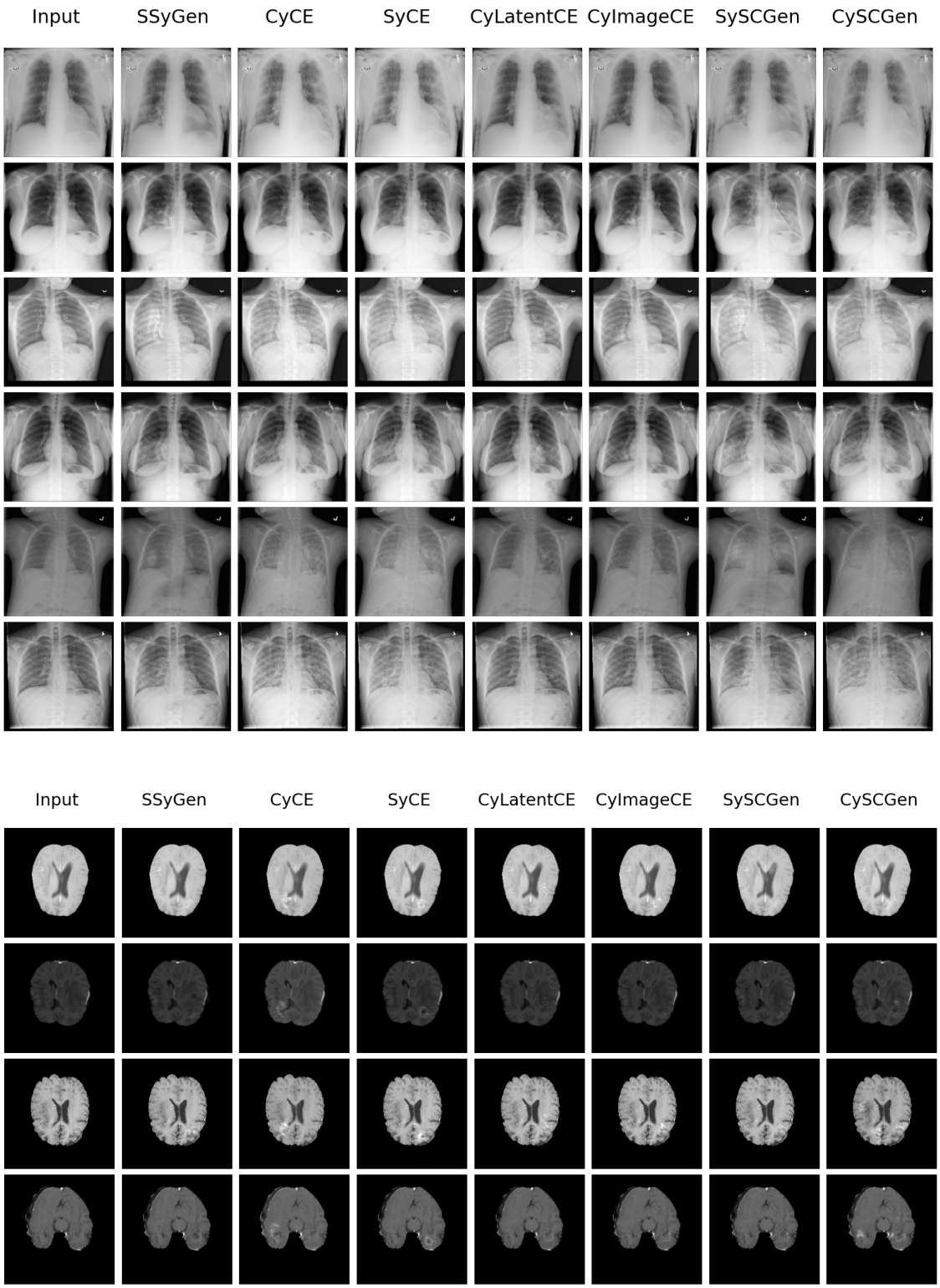


Figure 8.19: Comparison between counterfactual generation techniques: From healthy to pathological image. **Top**: Examples for pneumonia detection on X-rays. **Bottom**: Examples for brain tumor detection on MRI. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in Chapter 5. Dual path optimization is used for all the counterfactual methods.

The Figures 8.20 and 8.21 compare the perturbation-based technique MGen (using Gaussian blur) against our adversarial embodiment SAGen and a counterfactual embodiment (CyImageCE here) when translating pathological input to healthy ones. We provide additional Figures in the appendices Section E.2 (for both $\chi_0 \longrightarrow \chi_1$ and $\chi_0 \longrightarrow \chi_1$ transpositions). We point out several observations comparing MGen and SAGen with our counterfactual generations:

1. MGen generates adversaries that humans perceive as synthetic (especially on the X-rays). In addition, the perturbation is not attached to the image structures, which is a consequence of the heuristics regularization applied to the mask.
2. SAGen produces adversarial images similar to the original images (differences are almost imperceptible). There is no domain transfer for human eyes in the sense that the generated adversary does not look like a real image predicted in the other class. In some cases, we rather identify some artifacts in the region of interest.

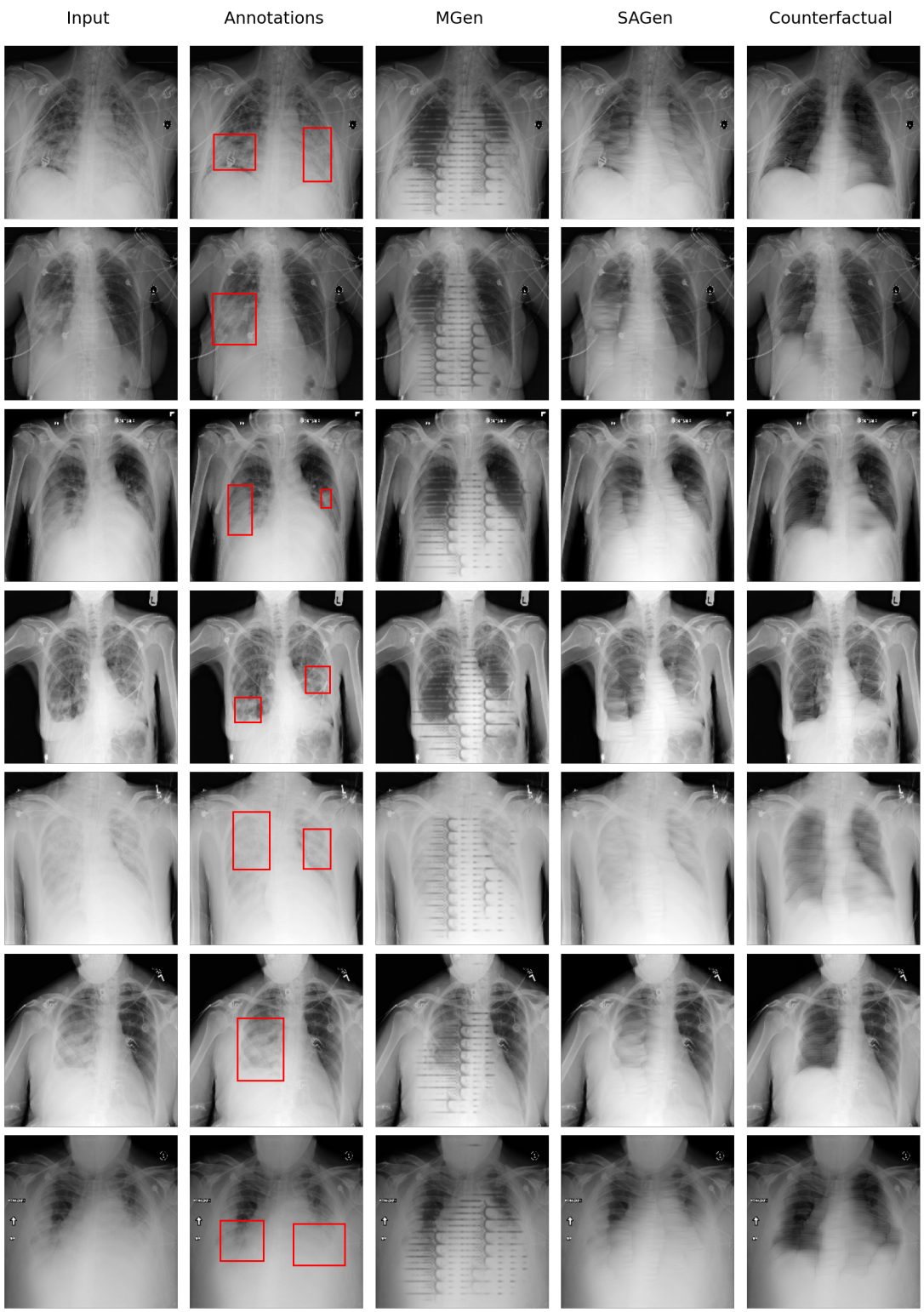


Figure 8.20: Pneumonia detection - Comparison with other generation / perturbation techniques: From pathological to healthy image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

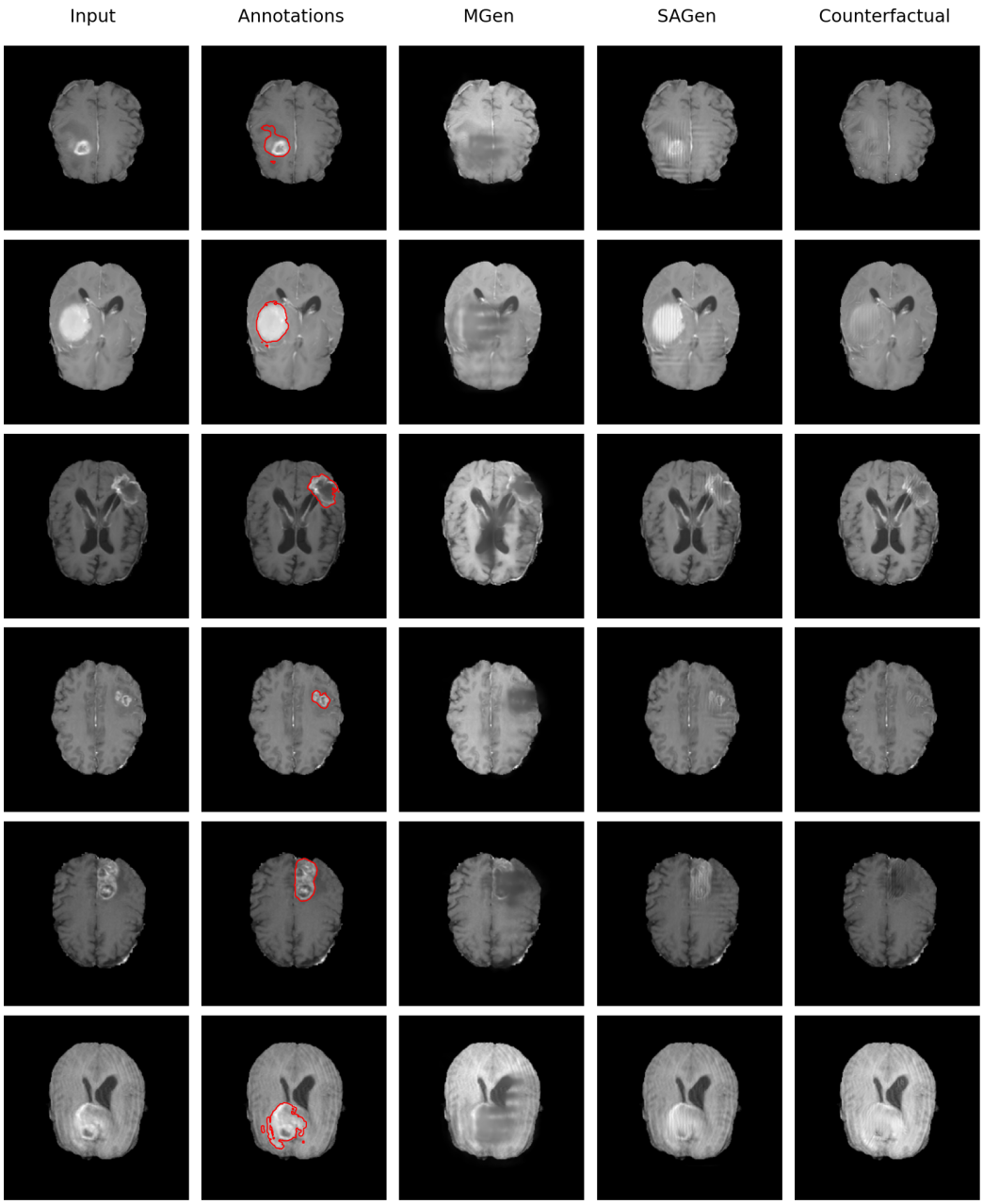


Figure 8.21: Brain tumor detection - Comparison with other generation / perturbation techniques: From pathological to healthy image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

We report in Table 8.12 the "embedding" metrics introduced in Section 7.3.3 comparing all the previous methods: Fréchet Distance on the mean encoded vector (FD_μ) of the VAE, the Jenson-Shannon distance (JS) on the estimated 2-dimensional distribution, and the FID computed on the Inception Network trained with the label predicted by f. We also show the PCA representations in Figures 8.22, 8.23, 8.24 and 8.25. Counterfactual techniques significantly outperform both MGen and SAGen in producing adversaries closer to the opposite image distribution (lower scores for three metrics).

Table 8.12: Domain Translation results - Comparison between perturbation, adversarial and counterfactual techniques. **(a)** and **(b)**: Fréchet Distance (FD_μ), Jenson-Shannon distances (JS) and Fréchet Inception Distance (FID_{tr}) –computed with an Inception network trained on the task– on the two medical problems. Here, we measure the distance between the perturbation/adversarial/counterfactual generations and the distribution of target images.

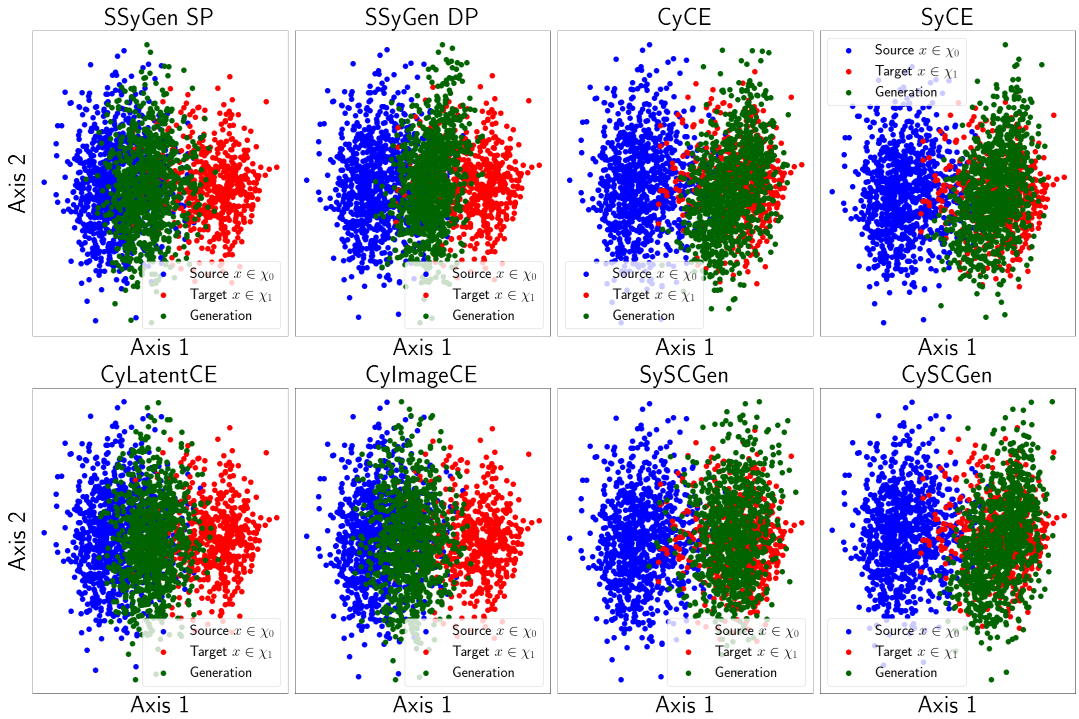
(a) $\chi_0 \longrightarrow \chi_1$

Method	Pneunmonia detection			Brain tumor detection		
	$FD_\mu\,(e-4)\downarrow$	$JS\downarrow$	$FID_{tr}\downarrow$	$FD_\mu\downarrow$	$JS\downarrow$	$FID_{tr}\downarrow$
MGen	58.07	0.82	1.54	0.32	0.50	1.88
SAGen	89.81	0.85	0.50	0.62	0.67	2.99
SSyGen (SP)	67.78	0.80	0.44	0.62	0.68	2.20
SSyGen (DP)	32.04	0.71	0.24	0.33	0.52	1.93
CyCE	1.92	0.32	0.12	**0.08**	0.39	**0.43**
SyCE	2.32	0.35	0.17	0.17	**0.37**	0.93
CyLatentCE	47.38	0.68	0.36	0.32	0.49	1.53
CyImageCE	48.71	0.69	0.38	0.29	0.44	1.56
SySCGen	1.78	0.29	0.26	0.31	0.46	1.79
CySCGen	**1.12**	**0.31**	**0.05**	0.24	0.42	1.38

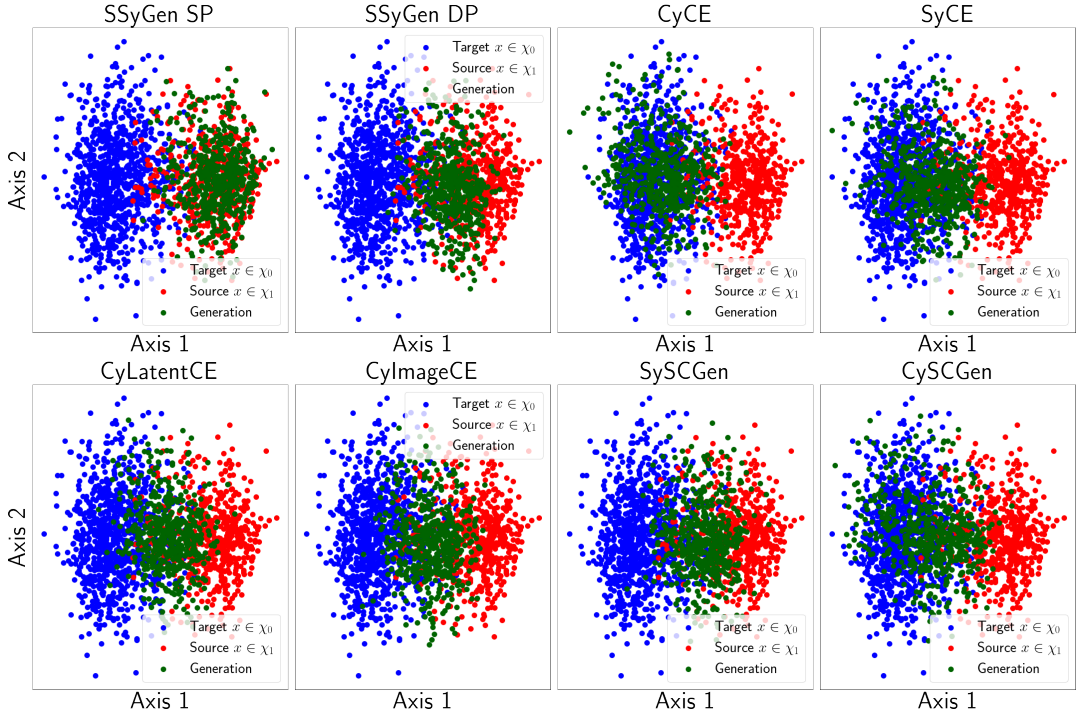
(b) $\chi_1 \longrightarrow \chi_0$

Method	Pneunmonia detection			Brain tumor detection		
	$FD_\mu\,(e-4)\downarrow$	$JS\downarrow$	$FID_{tr}\downarrow$	$FD_\mu\downarrow$	$JS\downarrow$	$FID_{tr}\downarrow$
MGen	63.11	0.78	0.85	0.51	0.61	1.27
SAGen	95.17	0.85	0.71	0.63	0.74	0.89
SSyGen (SP)	150.05	0.92	1.94	1.00	0.84	1.65
SSyGen (DP)	50.66	0.73	0.53	0.20	0.46	0.16
CyCE	**1.56**	**0.26**	0.29	0.15	0.27	0.40
SyCE	9.71	0.40	0.37	0.22	0.44	0.17
CyLatentCE	30.24	0.59	0.33	0.13	0.34	0.08
CyImageCE	29.87	0.58	0.36	0.13	0.30	0.06
SySCGen	42.94	0.70	0.54	0.19	0.45	0.18
CySCGen	10.31	0.41	**0.12**	**0.09**	**0.26**	**0.05**

Both the metrics and the PCA Figures 8.22 and 8.23 show that the least constrained implementations achieve better translation (see green points for CyCE, SyCE, and CySCGen compared to other counterfactual methods), especially for in the direction healthy to pathological. We note that accumulating constraints decreases the domain translation performance but better enforces the generated counterfactual images to be close to the input image.

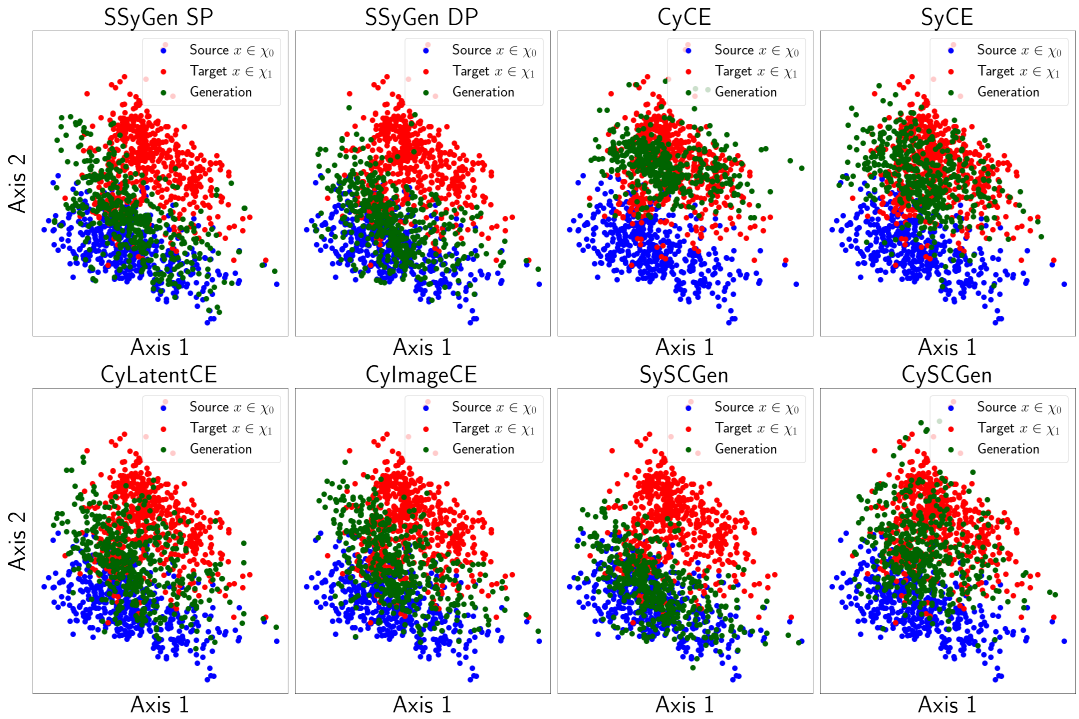


(a) $\chi_0 \longrightarrow \chi_1$

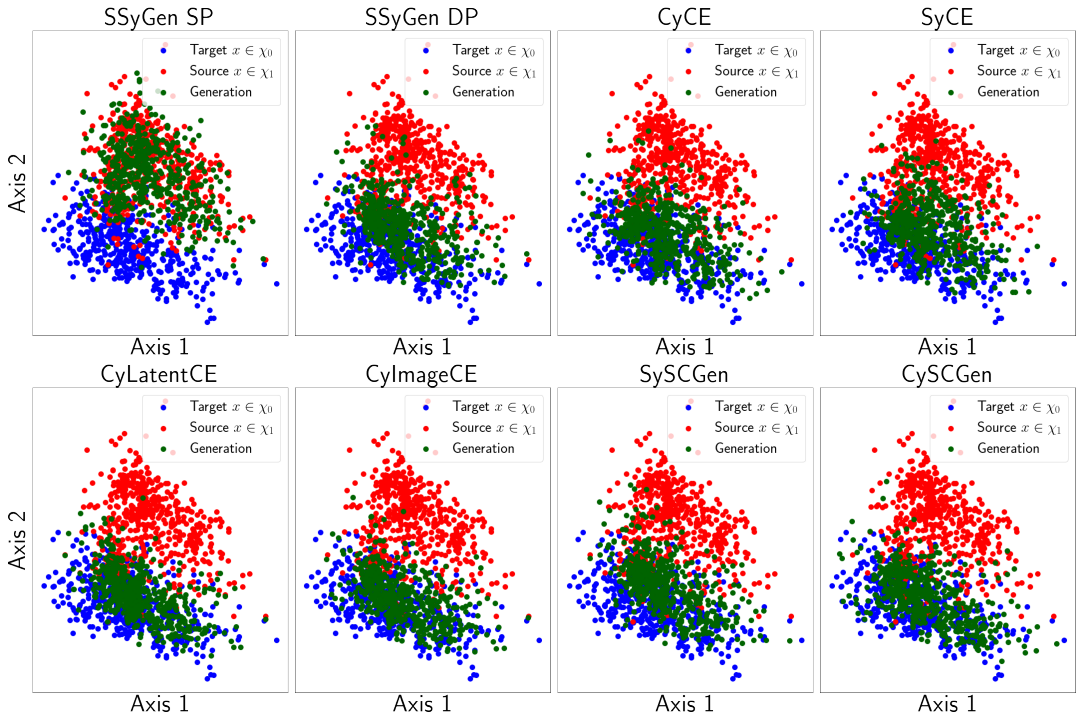


(b) $\chi_1 \longrightarrow \chi_0$

Figure 8.22: Pneumonia detection - Qualitative VAE Results: Comparison between counterfactual generation techniques. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.



(a) $\chi_0 \longrightarrow \chi_1$



(b) $\chi_1 \longrightarrow \chi_0$

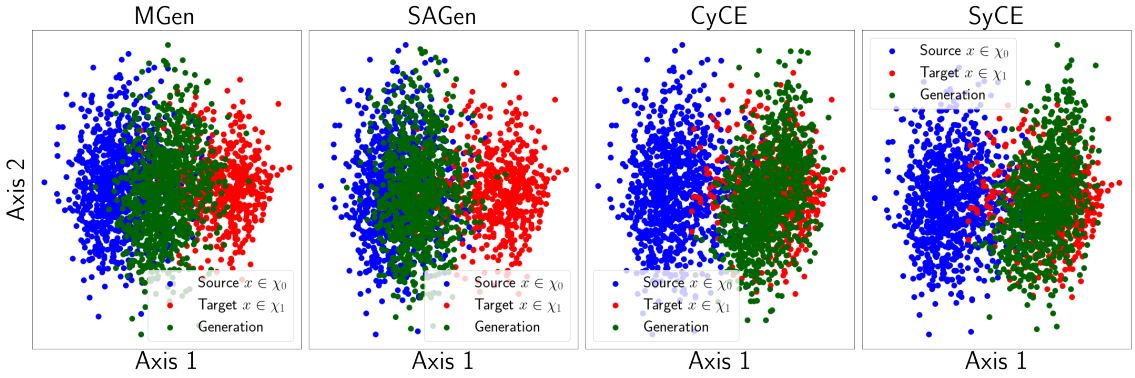
Figure 8.23: Brain tumor detection - Qualitative VAE Results: Comparison between counterfactual generation techniques. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

We summarize the impacting constraints of all these methods in Table 8.13. Using two generators seems to relax the constraints and improve the domain translation quality compared with using a single (and not conditioned) generator.

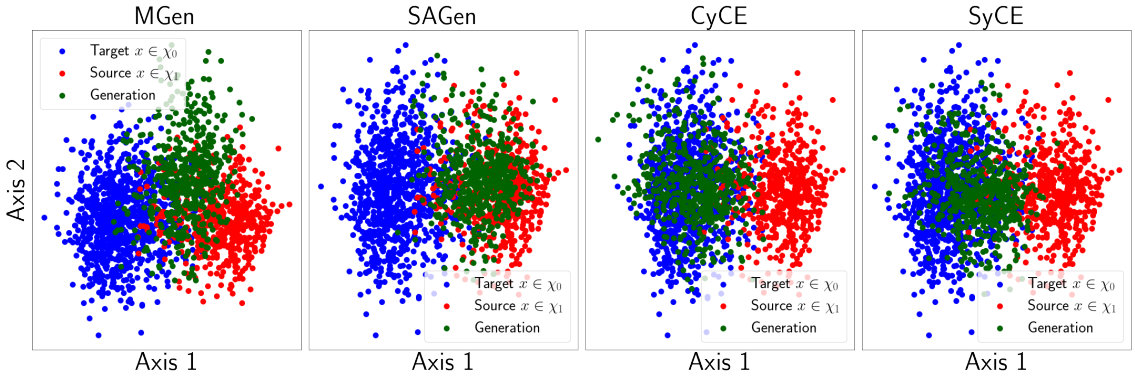
Table 8.13: Implementation constraints for the different counterfactual methods. The relative strength of constraints is displayed (sorted by order from the strongest to the more relaxed).

Method	Generator type	SP vs DP	L_d type
SSyGen (SP)	Single (1)	SP (1)	L_d^{sy} (3)
SSyGen (DP)	Single (1)	DP (2)	L_d^{sy} (3)
CyCE	Duo (4)	DP (2)	L_d^{cy} (5)
SyCE	Duo (4)	DP (2)	L_d^{sy} & L_d^{cy} (2)
CyLatentCE	Single Cond. (2)	DP (2)	L_d^{cy} & L_d^{st} (4)
CyImageCE	Single Cond. (2)	DP (2)	L_d^{cy} & L_d^{st} (4)
SySCGen	Single (2 heads) (3)	DP (2)	L_d^{sy} & L_d^{st} (1)
CySCGen	Duo (2 heads) (5)	DP (2)	L_d^{cy} & L_d^{st} (4)

We retrieve the findings from the localization Section 8.1.2, where CyCE, SyCE, and CySCGen explanation maps contained more details (not only restricted to the relevant regions), translating these relaxed constraints in the difference between the counterfactual and the input (or the stable) images. In addition, compared to the cyclic constraint (in CyCE or CySCGen), the symmetry (e.g., SyCE) is more restrictive and better enforces the proximity between the input and the counterfactual. The single and not conditioned generator versions (i.e., SSyGen and SySCGen) are the most constrained frameworks. They do not have access to the input's prediction (i.e., not used in these frameworks). Compared to other generation methods that produce counterfactual images for the classifier, these single generator versions sometimes do not follow the classifier's prediction and generate images increasing the sign of the input's prediction (i.e., translation in the wrong direction). These observations support the classification findings from Table 8.11. Hence, these frameworks do not perfectly translate what the classifier has learned, as the class chosen for the counterfactual sometimes differs.
SAGen has the lowest performance despite a higher perception of realism than MGen. We observe in Figures 8.24 and 8.25 that SAGen generates adversarial images that remain very close to the original distribution; while those produced by MGen can be shifted from both real distributions (see Figure 8.24b).

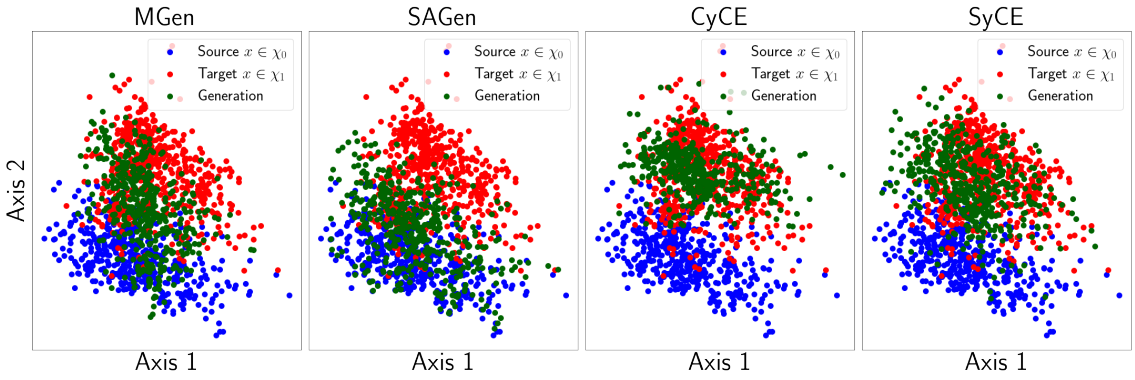


(a) $\chi_0 \longrightarrow \chi_1$

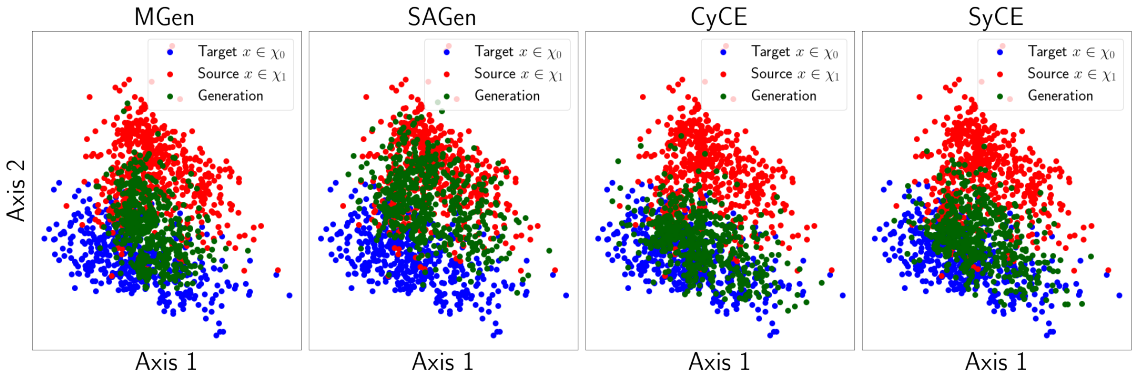


(b) $\chi_1 \longrightarrow \chi_0$

Figure 8.24: Pneumonia detection - Qualitative VAE Results: Comparison with other generation or perturbation techniques: MGen, SAGen, and counterfactual generations. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.



(a) $\chi_0 \longrightarrow \chi_1$



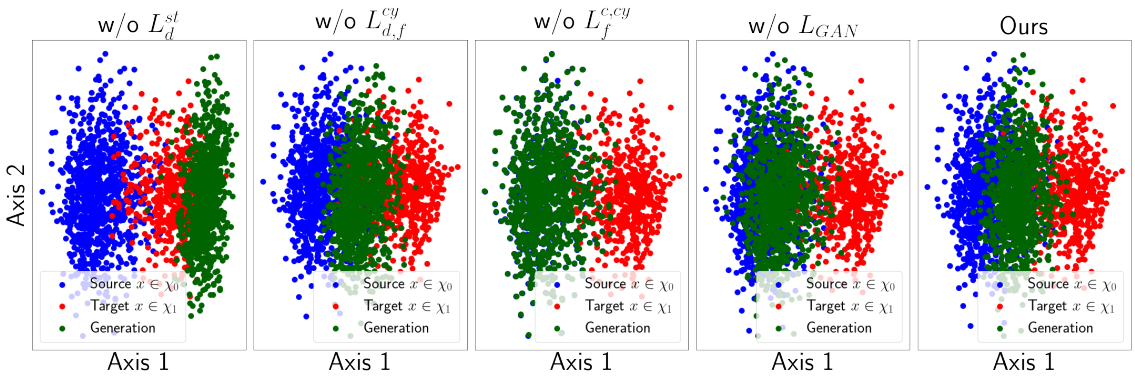
(b) $\chi_1 \longrightarrow \chi_0$

Figure 8.25: Brain tumor detection - Qualitative VAE Results: Comparison with other generation or perturbation techniques: MGen, SAGen, and counterfactual generations. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

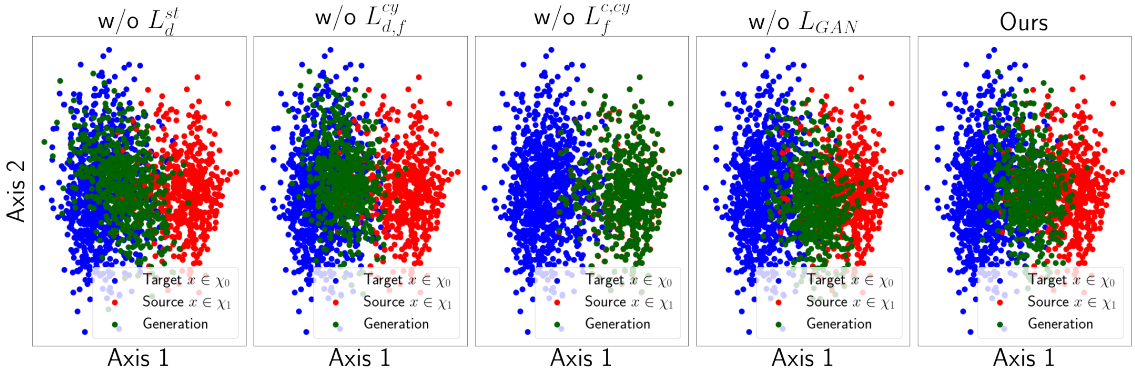
Ablation study- As for previous result sections, we try to assess the impact of the different optimization terms in our counterfactual methods. The Figures 8.26 and 8.27 show how the translation quality is impacted by the different terms for CyLatentCE on the pneumonia and the brain tumor problems. Similar figures for CyImageCE are found in the Appendix Section E.1.3. In all the cases:

- Removing L_d^{st} or the cyclic constraints improves the domain translation quality by reducing the proximity constraint to the input.
- Removing the classification constraints prevents translation in the pneumonia problem (generated and source points are confounded) and diminishes the translation capacity in the other problem.
- Removing the realism term (L_{GAN}) produces results similar to SAGen with images remaining close to the input (or source) distribution.

These findings correlate with classification accuracy shown in the appendix in Tables E.1 and E.2, and the Figures from E.1.3 that illustrate counterfactual images generated via these ablated optimizations. It also aligns with the discussions in the localization and feature importance results sections.

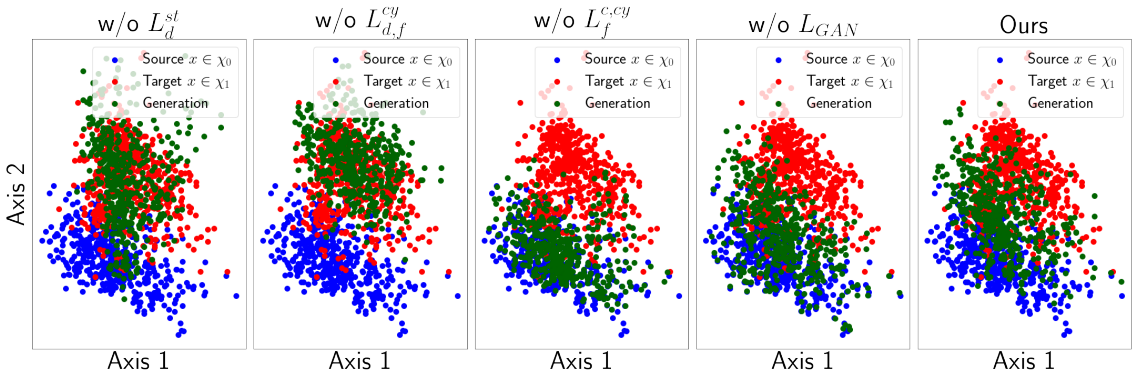


(a) $\chi_0 \longrightarrow \chi_1$

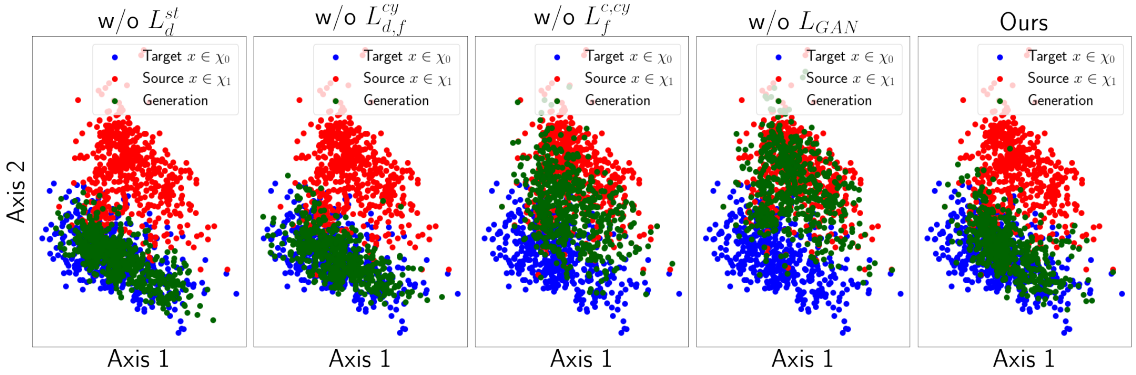


(b) $\chi_1 \longrightarrow \chi_0$

Figure 8.26: Pneumonia detection - Qualitative VAE Results: Ablation study for CyLatentCE. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.



(a) $\chi_0 \longrightarrow \chi_1$



(b) $\chi_1 \longrightarrow \chi_0$

Figure 8.27: Brain tumor detection - Qualitative VAE Results: Ablation study for CyLatentCE. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

8.3.2 Sum up and Limitations

The different counterfactual techniques perform well in capturing the different types and locations of impactful regions for the classifier, especially when acting on pathological images. This supports the results for localization and features importance evaluation and demonstrates which types of patterns are relevant for the classification model. In addition, coupling the methods for both attributions (ensemble approach) and counterfactual study improves the overall method by giving more or less advantage to the domain translation compared to the proximity to the input. When mapping healthy to pathological images, the generated images seem plausible compared to other perturbation approaches (SAGen or MGen). Indeed synthetic perturbations are added by MGen while the differences between the input and the SAGen's adversary are not visible (see Figures E.21, E.22 and E.23 in Appendix E.2). Yet, counterfactual techniques often perturb healthy images in similar locations (see Figure 8.19). This is a known drawback of symmetric and cyclic constraints forcing a one-to-one map between domains of different complexity [Bashkirova 19]. Guiding the domain translation with a classification loss also increases this behavior and tends to find the easiest way to generate a pathological pattern. For methods using a single generator, the generated tumor region sometimes looks like an

artifact pattern (see Figure 8.19). Although translating healthy to pathological images does not seem essential for explaining the decision of a classification model that detects abnormal regions, it would be interesting to study all the types of pathology the model has learned (through this translation direction). We elaborate more on this idea in the perspective Section 9.2.2.

8.3.3 Application to other classification problems

In this section, we validate our approach, designed for medical image classification tasks, to other binary or multi-classification problems. We show how the method extends to other image domains.

Identify 3 vs 8 digits- We first study how our proposed methods work on a simple task: identifying 3 from 8 white digits on a black background. We optimized different adversarial (SAGen) and counterfactual techniques (SSyGen, CyCE, SyCE, CySCGen) to explain the binary classifier. As described in the implementation Section 7.2, we adapt the generator and discriminator (except for SAGen) architectures to fit with 28 x 28 input images.

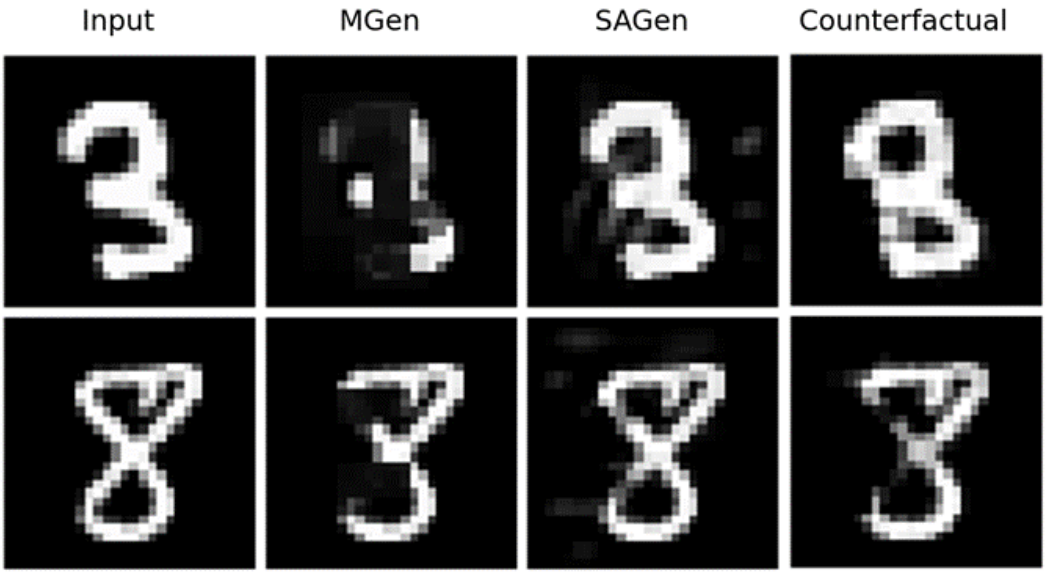


Figure 8.28: MNIST 3 vs 8 - Comparison with other generation / perturbation techniques. Top: From 3 to 8. Bottom From 8 to 3. From left to right: the input image, the annotated input image, the perturbated input through MGen, the adversarial generation from SAGen, and the counterfactual generation from SyCE.

Figure 8.28 compare the transposed image generated with MGen, SAGen and SyCE. Compared with the previous medical problems, the Gaussian blur perturbation used by MGen seems adapted to transpose an 8 into a 3, looking like a real 3. It is not the case in the other direction, where the perturbed 3 is not perceived as an 8 (synthetic image that does not belong to the distribution of real images). In this case, we can not conclude if the method points out relevant regions for the classifier because the model is not supposed to work on this type of image. SAGen produces similar yet poorer results than pneumonia or brain tumor problems. It generates adversarial examples that look like the input (with small changes) rather than an image from the opposite distribution. Note that the transformations are more perceptible than in the other problems. The method tries to attack either the empty loops of the 3 (adding a lighter color) or the filled loops of the 8 (darkening them). We also find artifacts in the background that do not highlight relevant details but are rather signatures of adversarial attacks. The counterfactual method produces satisfying transposition in the two cases while perturbing the minimum of pixels in

the relevant regions (i.e., the loops).

Table 8.14: Counterfactual classification results on Digits "3 vs 8". Accuracies (Acc_c) computed between the opposite of the input prediction ($1 - c_f(x)$) and the model's prediction on the adversarial/counterfactual generation.

Method	MGen	SAGen	SSyGen	CyCE w/o $L_f^{c,cy}$	CyCE	SyCE	CySCGen
$Acc_c \uparrow$	0.824	0.968	0.922	0.046	0.930	0.985	0.989

Table 8.14 provides the classification accuracy of the different visual explanation methods. As for pneumonia and brain tumor detection, the classification guidance is necessary to achieve classification transposition w.r.t. f ($Acc_c = 0.046$ for CyCE w/o $L_f^{c,cy}$). Similarly, adversarial and counterfactual techniques outperform MGen. Then, Table 8.15 shows the domain transposition metrics FD_μ and JS computed from a VAE representation. CyCE and SyCE produce comparable and the best transposition results except for the translation 8 to 3, where MGen is the best performer (as we pointed out in Figure 8.28). The PCA representations of the different counterfactual methods (Figure 8.29) and comparing the different perturbation or generation methods (Figure 8.30) are consistent with the domain transposition metrics and the visual observations. Compared with the medical problems, all the counterfactual methods achieve the transposition objective in the two directions (see Figure 8.29); the constraints applied on generators have smaller impacts on this simpler case.

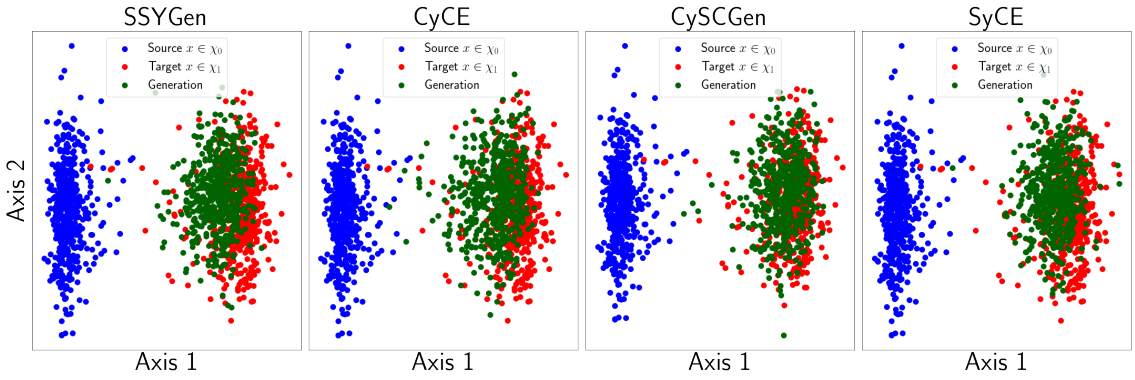
Table 8.15: Domain Translation results - Comparison with generation methods on Digits "3 vs 8". **(a)** and **(b)**: Fréchet Distance (FD_μ) and Jenson-Shannon distances (JS). Here, we measure the distance between the distribution of target images and the counterfactual / perturbated / adversarial generated images.

(a) $3 \longrightarrow 8$

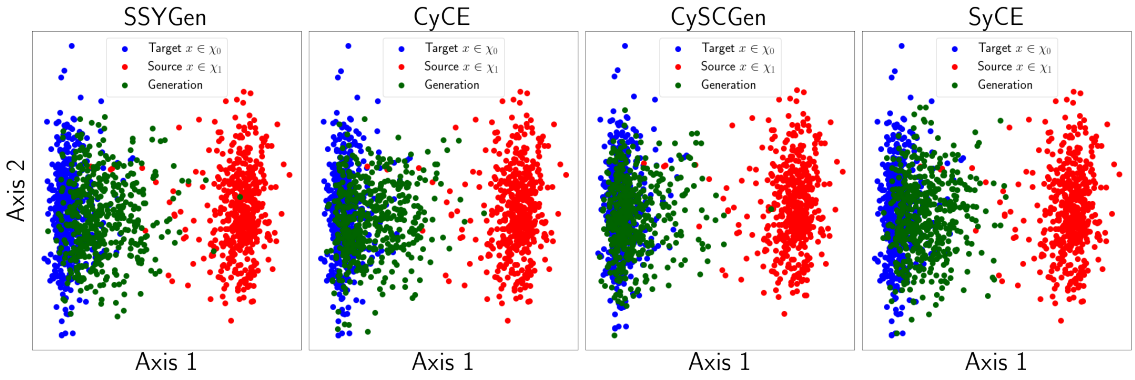
Method	MGen	SAGen	CyCE	SyCE
$FD_\mu \downarrow$	142.71	55.93	4.50	**2.84**
$JS \downarrow$	0.99	0.96	**0.55**	0.58

(b) $8 \longrightarrow 3$

Method	MGen	SAGen	CyCE	SyCE
$FD_\mu \downarrow$	**3.10**	12.04	9.37	8.61
$JS \downarrow$	**0.60**	0.99	0.75	0.71

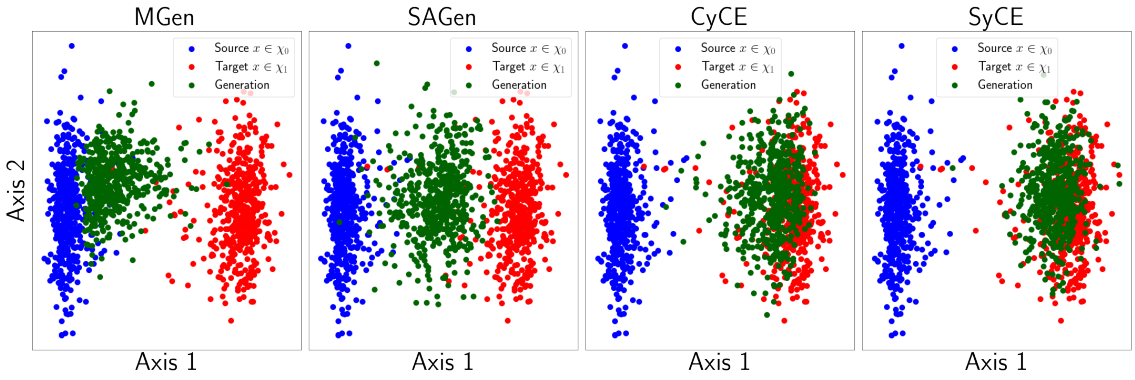


(a) $3 \longrightarrow 8$

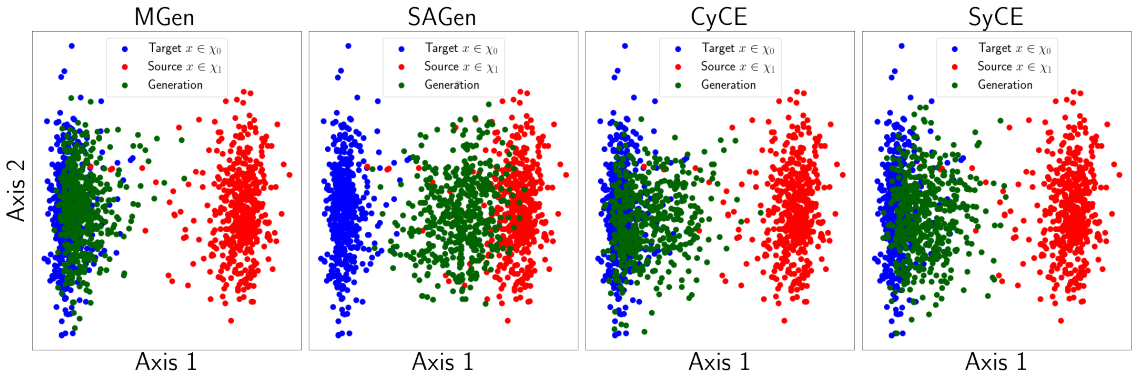


(b) $8 \longrightarrow 3$

Figure 8.29: MNIST 3 vs 8 - Qualitative VAE Results: Comparison between counterfactual generation techniques. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.



(a) $\chi_0 \longrightarrow \chi_1$



(b) $\chi_1 \longrightarrow \chi_0$

Figure 8.30: MNIST 3 vs. 8 - Qualitative VAE Results: Comparison between counterfactual generation techniques: MGen, SAGen, and counterfactual generations. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

Binary problems on RGB images- To demonstrate that our counterfactual approach extends to other complex tasks(compared to distinguishing digits "3 vs. 8"), we also conduct experiments on a colored images dataset: CelebA [Liu 15]. As presented in the experiment section, we apply our method to two tasks on celebrity faces: detecting the presence of a mustache and separating old from young people. The first task comes close to the previous problems of pathology detection as we aim to identify a kind of abnormality (i.e., mustache here) on a human face. In contrast, different signatures may be relevant for each class when distinguishing old against young people e.g., the skin texture, the glasses, the hairiness.
We train CyLatentCE generators on the two tasks using similar weighting parameters λ_i as in the medical cases. The only implementation difference, compared to the black and white setting, is that we do not concatenate 3 times the generated images to produce a suitable format for the ResNet50 (requesting RGB images). We directly generate RGB images (see the implementation Section 7.2.1).
Figures 8.35 (see Section 8.4) and 8.31 show some generations produced by CyLatentCE on the two tasks. First, the counterfactual generations seem quite realistic in the two cases. Second, only relevant attributes of the face are changed; the background is left intact. The expected input features to be identified in the two tasks are quite different: a mustache is precise and specifically located, while relevant features for identifying an old or a young person can be either diffuse or focused. In Figure 8.35, counterfactual generations emphasize that the classification model has learned what a mustache is. Indeed, the counterfactual adds (4th row) or removes (3rd row) the mustache. Although stable and counterfactual generations differ by localized details, they are not always as expected. In the 2nd and 3rd rows, the specific generator removes the mustache and the beard (or the shadow appearing as a beard in the 3rd row). Similarly, the other generator adds both a mustache and a small beard to input images of the 1st and 4th rows. Then, the last row shows a false positive case, where the generated counterfactual successfully translates the misleading attribute. We elaborate more on these findings in the next Section 8.4 on biases detection. In the "Young vs. Old" translations (see Figure 8.31), local and focused attributes (e.g., eye makeup, wrinkle) are changed, as well as more diffuse attributes (e.g., hair brightness, skin texture). See the Appendix Section E.2.4 for additional figures.

Counterfactual generations achieve transposition accuracies of 0.999 and 0.990 on mustache detection and old vs. young classification problems. For the two task, Figures 8.32a and 8.32b show the first 2 axis of the PCA computed from the VAE latent representation. Counterfactual images seem to achieve the transposition objective in the two directions and for the two problems (at least for this representation).

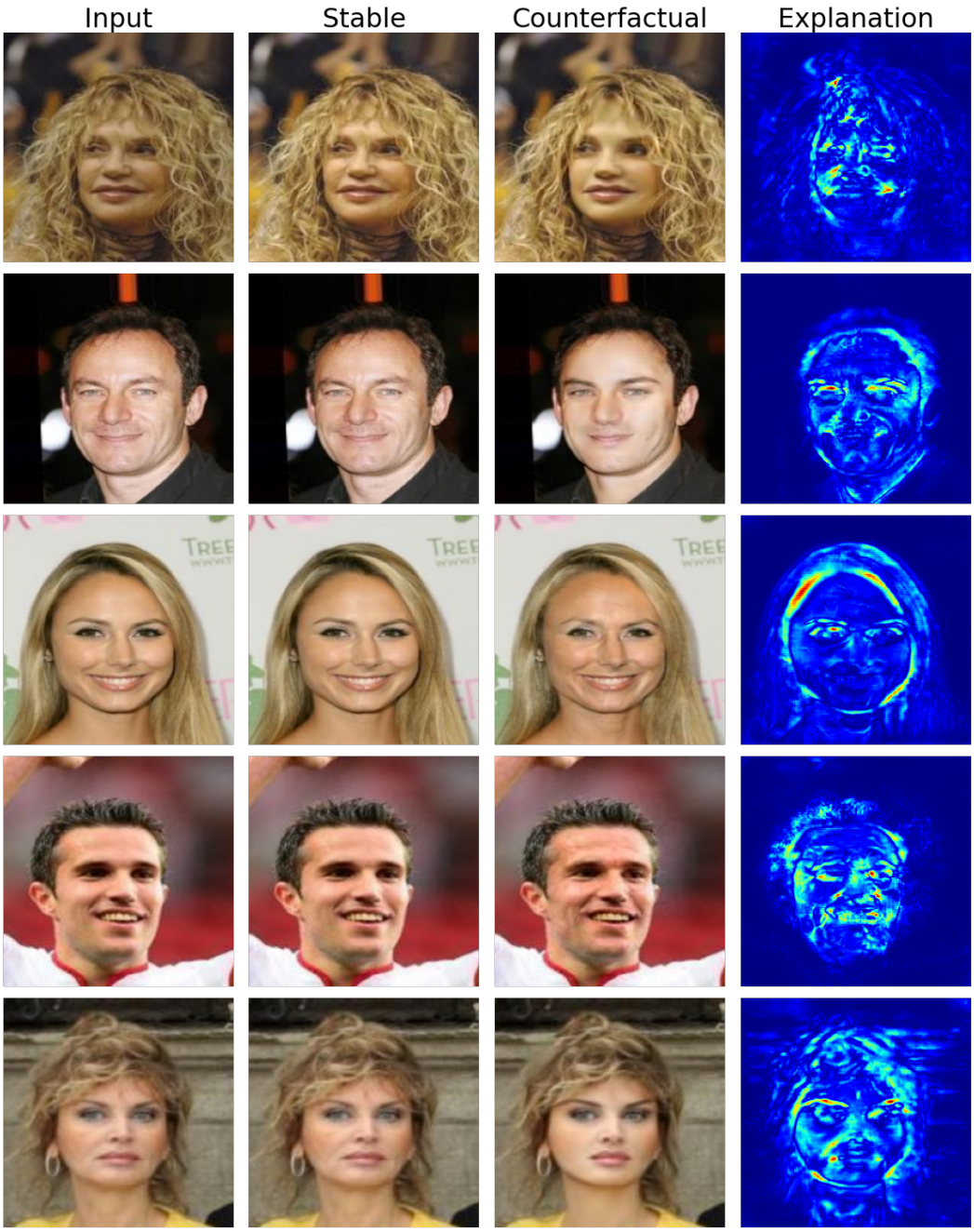
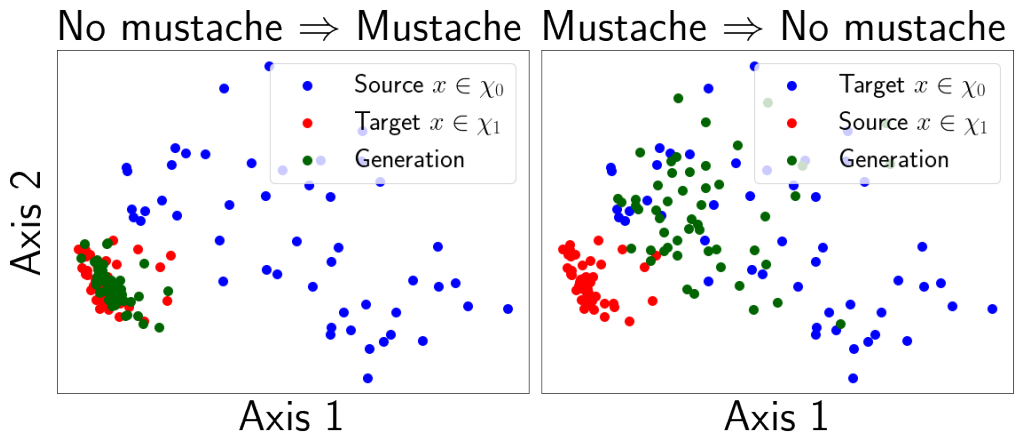
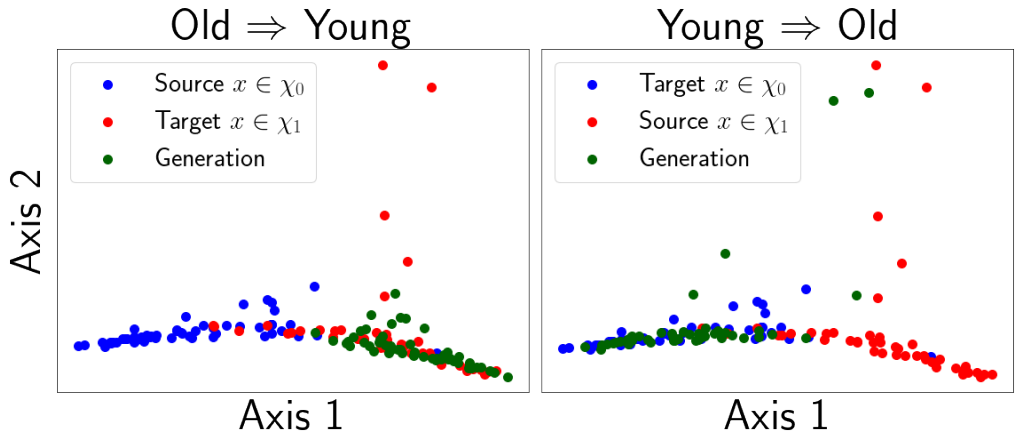


Figure 8.31: Young vs. Old - Generations and attributions. From left to right: the input image, the stable generation, the counterfactual generation, and the counterfactual explanation map. The results are produced with the CyLatentCE technique.



(a) Mustache detection



(b) Old vs. Young classification

Figure 8.32: CelebA - Qualitative VAE results for domain transposition using a counterfactual generation technique. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for a set of 100 images (real and generated) of the test set. **(a)**: Domain transpositions for the mustache detection problem. **(b)**: Domain transpositions for the classification problem "old" vs "young". Here, CyLatentCE performs the counterfactual generations

Multi-classification setting- In this paragraph, we propose preliminary experiments in the multi-classification setting on MNIST, where the classification model has been trained to identify the 10 different digits.
Figure 8.33 displays some generated stable and counterfactual images for the targeted optimization. The stable generation (2nd column) successfully preserves the input image. The generator also produces realistic counterfactual images for each class while minimally transforming the input image ("relevance" property). The original shape of the input image is preserved except when the differences between two digits are too important (e.g., 1 towards 0). We show in the Appendix Figure E.32 that the symmetric/cyclic constraint $||x - g_f(g_f(x, t_c), c_f(x)||$ is necessary to achieve the "relevance" objective (and not generate random digits as usual conditional GAN).

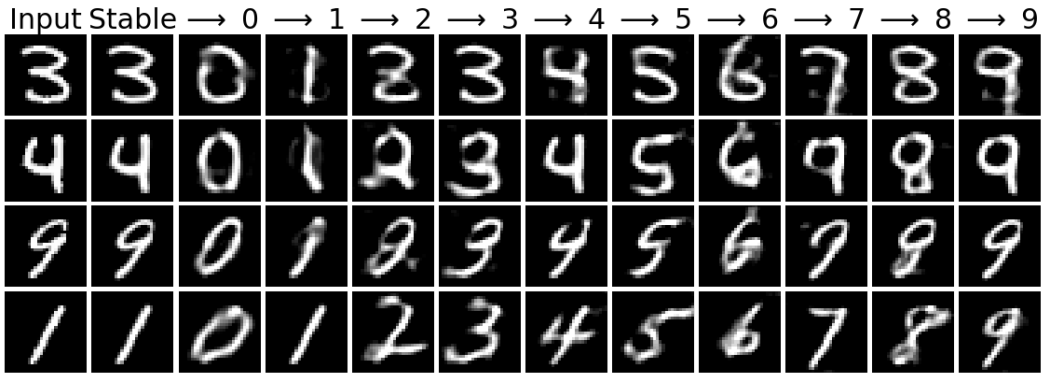


Figure 8.33: Targeted counterfactual generations for multi-classification model on MNIST. From left to right: the original image; the stable image; counterfactual image with respect to each class (columns 3 to 12).

Figure 8.34 shows some examples of stable and counterfactual generations in the untargeted setting, transforming the input image into the second choice of the classifier. In this case, the counterfactual generator performs the minimum possible edits on the input, e.g., closing the digit loops from 3 to 8, from 2 to 8, or 4 to 9, or removing them to produce a 5 from an 8. Additional figures are proposed in the Appendix Section E.2.5.

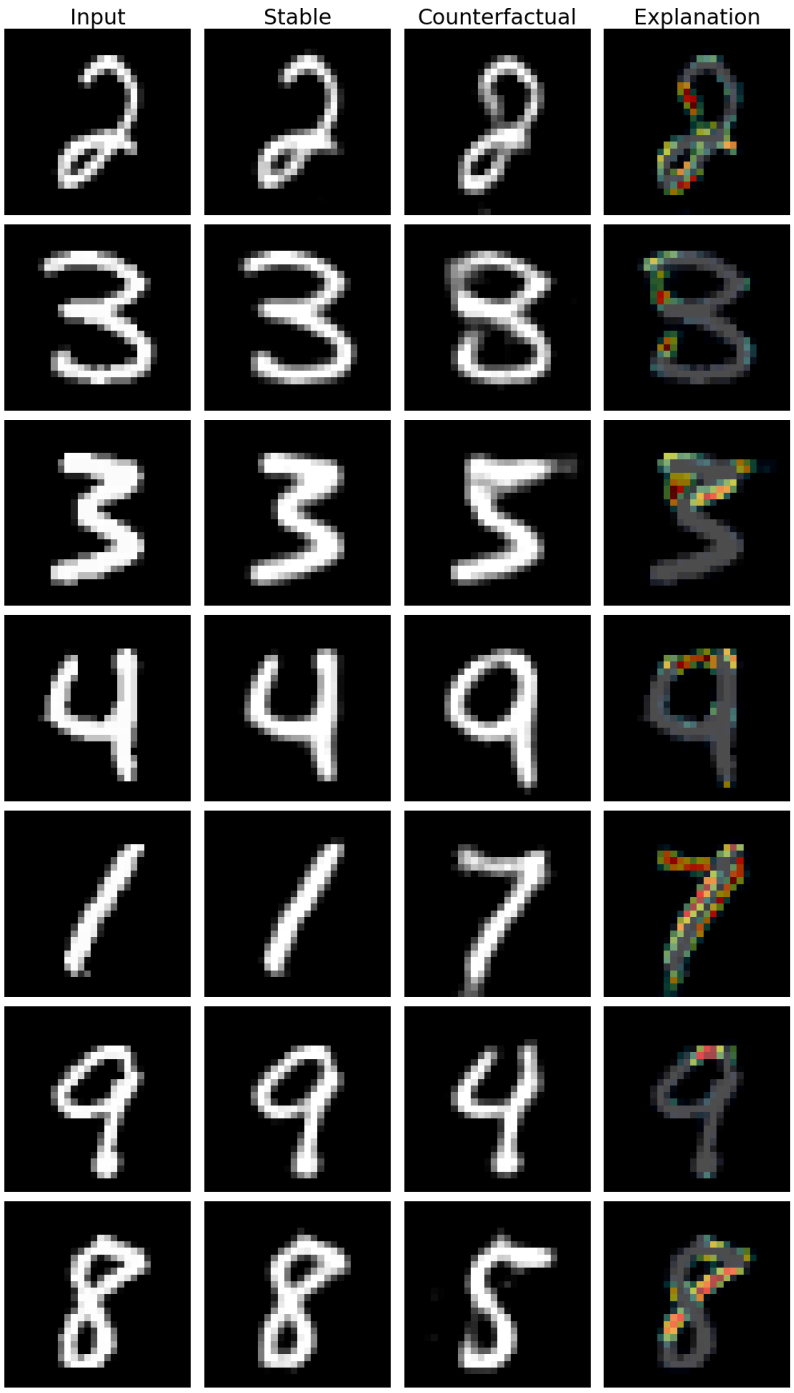


Figure 8.34: Untargeted counterfactual generations for multi-classification model on MNIST. From left to right: the original image, the stable image, the untargeted counterfactual image, and the visual explanation.

8.3.4 Usage and limitations of the counterfactual approach

Our visual explanation formulation (see Chapter 3) and its embodiments are specifically designed for medical image problems and especially well suited for binary classification tasks. In this context, the distributions of images (from the different classes) are generally close, often share similar content (e.g., background, body structures), and rather differ on a few local features (e.g., tumors, opacities). Compared to primary works that only localize relevant features, our technique also points out the minimal transformations to change the classifier's decision to another class. We qualitatively and quantitatively demonstrate that our techniques better localize relevant features for both the classifier and the clinicians. By combining counterfactual generations and the resulting attributions, we can detect some training biases and propose strategies for the model's improvement.
We qualitatively show in Section 8.3.3 that the counterfactual approach can be extended to multi-classification or colored image settings. However, it is not clear what a counterfactual generation should be when images of two (or more) classes are completely different (e.g., airplane vs. cats) both in the content of the images and in the type of object to be identified. In this scenario, it is plausible that our GAN-based approach cannot ensure the on-manifold generation, i.e., generated counterfactuals, would not belong to the same distribution of real images from the predicted differently. In this particular setting, too many changes should be applied to the input to translate it into the counterfactual domain. This limitation should increase with the number of classes (with very different images).

8.4 Identification of biases

Generating counterfactual examples for all inputs puts forward the structures of each class relevant to the classifier and goes further than just giving attributions. In parallel, the classification model may learn confounding biases found in the training data and make wrong associations. Our counterfactual method can discover such training biases by studying the generated attribution maps or the counterfactual transformations applied to the input. We offer and study different scenarios in the following.

Coupled attributes in the training data for mustache detection- As pointed out in the previous section, counterfactual generations in Figure 8.35 add or remove a mustache (depending on the input's prediction), but also modify unexpected details: removing the beard as well (2nd and 3rd rows) or adding hairs on the chin when creating a mustache (1st and 4th rows). This generator also increases "man-specific" attributes (e.g., eyebrow thickness in 1st row) in the counterfactual image (while the other generator rather increases woman attributes but lesser). All these observations are coherent with the biases found in the training dataset. Indeed, we do not find any woman with a mustache in the training set. So men's attributes can be necessary to identify a mustache case (for the classifier). Similarly, we found that only 2% of the mustache cases have no beard, while 89% do not have one in the other set. In both cases, the mustache attribute is coupled with other attributes in the training dataset. In the last row, the model fails to identify that the woman places her hair above her mouth and detects a mustache. Our counterfactual method erases this portion of hair, highlighting that the classifier especially focuses on the location of the mustache rather than the constitution, texture, or shape. Note that this particular case is an outlier of the training database. However, the model still has seen different types of mustache during the training, and this observation suggests that location is the most impacting feature.

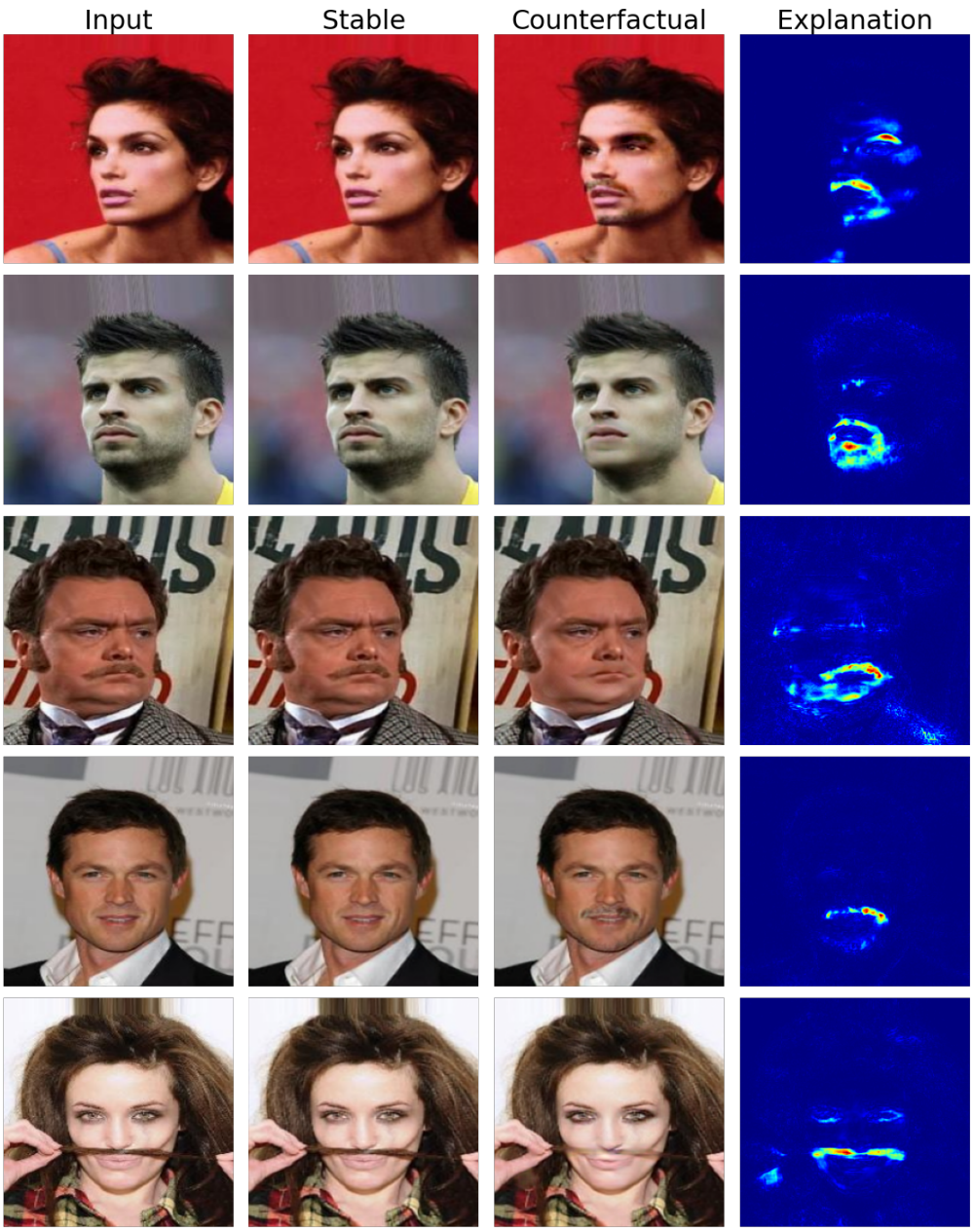


Figure 8.35: Mustache vs. No mustache - Generations and attributions. From left to right: the input image, the stable generation, the counterfactual generation, and the counterfactual explanation map. The results are produced with the CyLatentCE technique.

The non-negligible impact of geometric transformations- We trained on MNIST the same multi-classification model on the same training set but using random vertical flip augmentations. We show in Figure 8.36 that the model has learned biases related to this geometric transformation that is captured by our untargeted counterfactual generator (e.g., generating flipped digits 9, 2, 7 or 6).

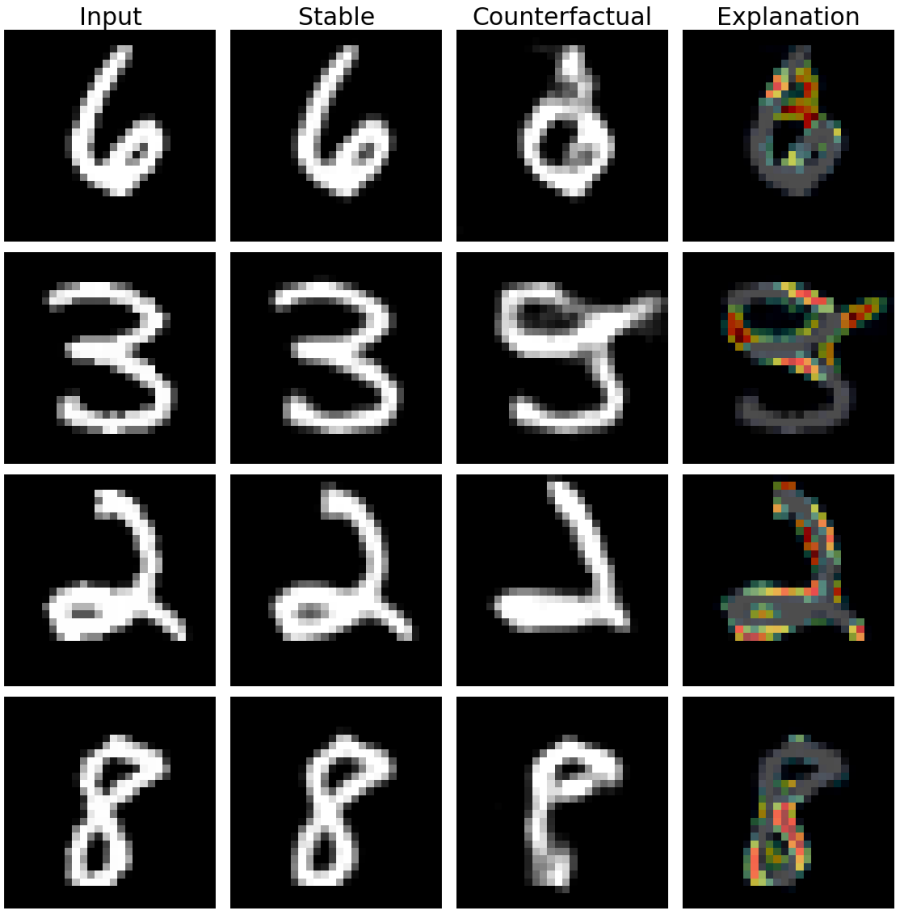


Figure 8.36: Untargeted counterfactual generations for biased multi-classification model on MNIST. From left to right: the original image, the stable image, the untargeted counterfactual image, and the visual explanation. Here the classification model is trained using random vertical flips.

Detecting and correcting a training bias on the synthetic dataset- While we studied biases using the counterfactual generation in the previous paragraphs, we directly use the attribution map in this case study on the synthetic dataset (see experiment Section 2.4.6). We recall that we have trained two classification models using a biased and an unbiased training set. Figure 8.37 shows the attribution maps (produced by CyLatentCE)for these two models (3rd and 5th columns) on different pathological or healthy cases. The attributions on L and R & L pathological cases highlight that the biased classifier only focuses on the right side. Similarly, the counterfactual generator only adds a pathological region on the right side for the biased classifier (5th row). If the model was given as a black box or the distribution of pathologies left unknown to the user, our explanation technique would have discovered a bias.

We propose a simple action to correct the bias using the same biased training database. We retrain the model using horizontal flip transformations. The new model achieves an accuracy of 0.993 (compared with 0.823 previously). Examples of attributions are shown in the 4th column, demonstrating that the action corrected the training bias. The new classification model behaves similarly to the model trained on an unbiased database.

This simple case study illustrates how visual attribution can drive the correction and the improvement of the classification model. We used the same strategy on Incepto's product Keros (see Section **??**).

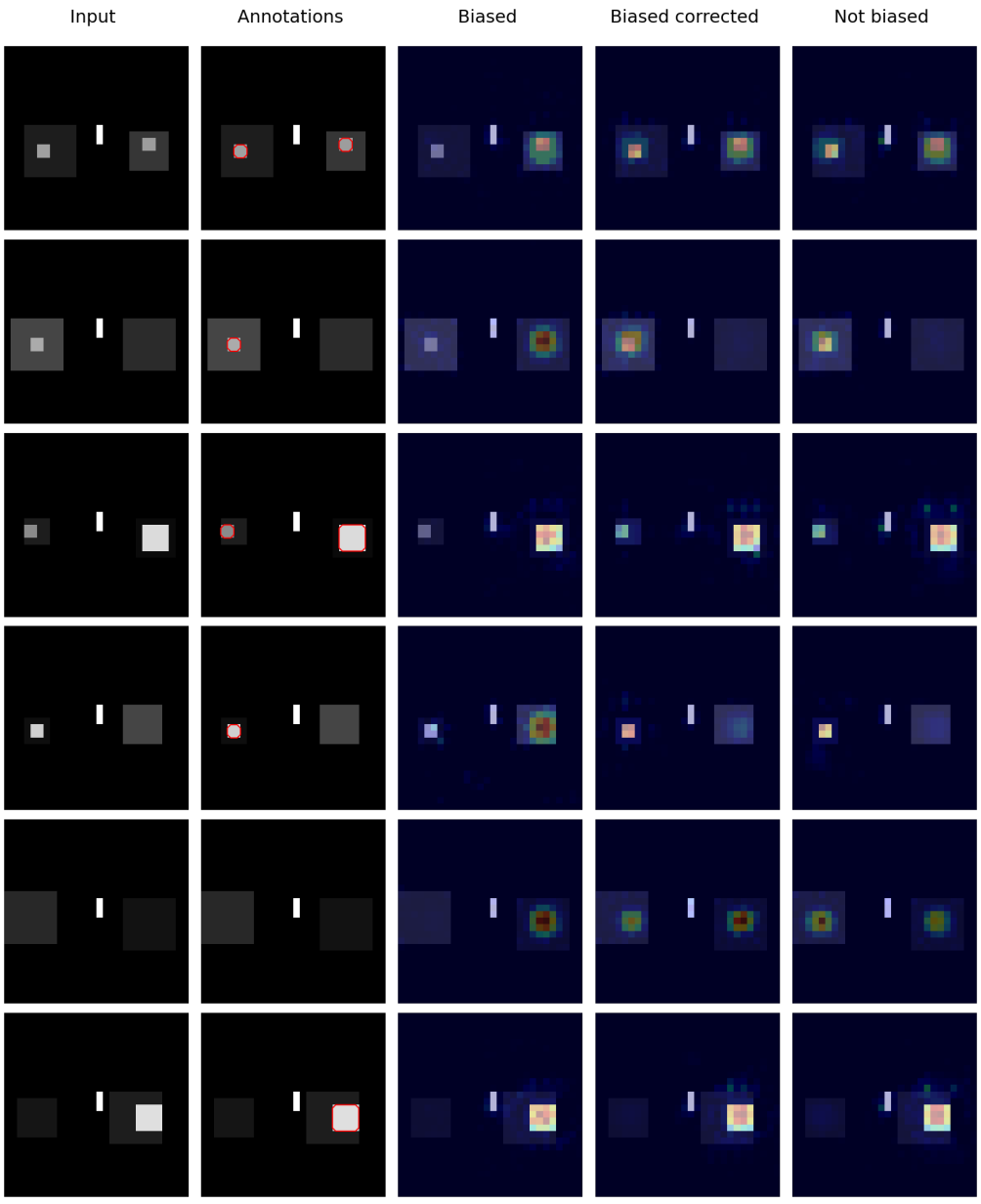


Figure 8.37: Biases detection and correction on synthetic dataset. From left to right: the input image; the annotated input image; the attribution map for the biased classifier; the attribution map for the biased classifier trained with random horizontal flips; and the attribution for an unbiased classifier. From top to bottom: a pathological case displaying the Right (R) and the Left (L) pathology types; a pathological case L where the biased model produces a false negative prediction; another pathological case R & L; a pathological case L where the biased model predicts well; a healthy case; and a pathological case R.

9
Conclusions & Perspectives

Deep learning models are increasingly used in medical imaging tasks, and are claimed to achieve performances competitive with clinicians. However, as humans, machine learning systems make mistakes. They can learn confounding biases inherited from practitioners who annotated the data or from the data distribution itself. These models often succeed in capturing the average cases but often fail when encountering specific or outlier cases. Compared to clinical practice, where radiologists justify their diagnoses, DL solutions only provide a decision without any argumentation. For localization or segmentation tasks, the algorithm produces a visualization pointing out one or multiple regions of interest. The practitioner is guided and can check whether the prediction is correct or not. In this situation, the user does not understand how the decision is made, but he can verify it. In contrast, for a classification task, the model predicts a category, and the user has no clue about what support the decision, whether it is right or wrong. In medical practice, classification annotations are simpler to obtain (than segmentation masks), making this task more widespread. They also mitigate (but do not eliminate) large annotations variability when the objects to locate is complex or diffused, and causes disagreement between clinicians. It is, for instance, the case in Keros: meniscus tears and cartilage lesions are not always well defined, and their localizations vary among annotators. Incepto's teams develop multiple deep classification models dedicated to the different knee's elements. It is thus imperative to provide explanations for the classification predictions, to validate what the model has learned and show the end-user what regions support its decision.
The notion of explaining model decisions is large and does not reach a perfect consensus among the XAI community. The nature and form of the explanation differs according to its objective, on which type of models it is applied, and for whom it is intended.

9.1 Summary of the contributions of the thesis

In this thesis, we leverage specificities of medical image problems (e.g., encountered at Incepto) to design visual explanations of deep classification model decisions with limited or no access to the model structures. Our work focuses on standard convolutional neural networks trained on medical image classification tasks. We do not seek to explain how the model works internally nor produce textual explanations of the decision. We assume that providing visual explanations to accompany the model's decision is a step forward for integrating AI solutions in medical practice.
Our method first targets the clinician by highlighting the relevant regions of the input image that support the model's decision. Then, data scientists assisted by clinicians can use it as a tool to validate the model globally on a database, ensuring that the model relies on the relevant structures and not on confounding bias.
Our work bridges the gap between feature attributions, localizing the regions of the input image relevant to the classifier, and counterfactual explanations highlighting image

patterns specific to each class. In a general formulation (see Chapter 3), we introduce four properties to design counterfactual generations from which visual explanations are produced: **Relevance**, **Regularity**, **Realism** and **Ordered by importance**. Following the chronology of the thesis, we propose embodiments (see Chapters 4, 5, 6) of the general formulation in the form of constrained optimization problems, each adding step by step an additional property. Taking into account

1. **Relevance**: we propose an adversarial generation approach AGen (See Section 4.1).
2. **Relevance and Regularity**: we enrich AGen by introducing a stable generation, and propose SAGen (see Section 4.2).
3. **Relevance, Regularity, and Realism**: we leverage domain translation techniques to generate in-distribution counterfactual images (see Chapter 5), instead of adversarial examples in the two previous embodiments.
4. **Relevance, Regularity, Realism, and Ordered by importance**: we unify counterfactual generation and path-based approaches (see Chapter 6).

We implement several variations of each embodiment through approximations, considering more or less constraints in the generator structure or the optimization. Our final method leverages domain translation techniques to generate counterfactual examples and path-based approaches to produce feature attributions.
We have assessed the quality of the visual explanations for the different embodiments on two medical tasks, performing pathology localization (see Section 8.1) and feature importance evaluation (see Section 8.2). Compared to previous work, our methods better localize discriminative regions for the classifier that are interpretable by clinicians. Revisiting variational autoencoders, we have validated that our techniques can either stabilize or translate an image to its original or counterfactual image distribution. At the same time, our approaches ensure proximity to the input image (see Section 8.3). In addition, we have pointed out that the constraints applied to the generation process set the balance between the localization performances, the proximity to the input, and the counterfactual translation quality. Our methods generalize to other image tasks when differences between classes are relatively localized and do not change the whole image's content (see Section 8.3.3). Then, we illustrated how the visual explanation could identify training bias and provide hints to improve the classification model (see Section 8.4).

9.2 Perspectives & Work in Progress

In chapter 3, we formulated a generation process to design a visual explanation of a classifier's decision for each image. The visual explanation is composed of two terms:

1. An attribution map highlighting where are the most important input features for the classifier (ordered by importance);
2. A counterfactual image showing how the highlighted input regions should be changed to produce a different model's outcome.

We demonstrate the efficiency of our approaches for localizing relevant pathological regions and identifying training bias. However, the current methods do not detect possible local errors on given inputs. They do not give the radiologists any insight or indicative measurement on whether the model made a mistake or the correct choice. In addition, a unique counterfactual image is produced for each input image, while diverse, realistic input modifications could impact the model's decision. We elaborate on these two directions in the following sections.

9.2.1 Detecting local classification errors

At the current state, the user can apply our method on a given input to reveal supporting regions or on a series of cases to identify discriminative patterns and spot possible bias. Our method visually translates what the model has learned. It is not designed to detect the model's mistakes or indicate confidence over its prediction. This additional information could raise the clinician's attention (especially for non-experts) on questionable cases. Many works [Kiureghian 09, Combalia 20, Gawlikowski 21, Mena 22] tackle this issue by studying the uncertainty of the model predictions. However, these techniques do not benefit from the information produced by explanation methods. In our case, the model's prediction is enhanced with an attribution map, a counterfactual example, and the prediction of this generated example. As visual explanations are not supposed to derive uncertainty measures, we would either add explanation information to standard uncertainty methods or use the ideas of uncertainty techniques on visual explanations instead.

Adding explanation information to uncertainty measurement- As many uncertainty techniques exist, we can add the generated explanation information in different contexts. It mainly depends on the type of the studied models (e.g., standard convolutional neural networks, bayesian models, an ensemble of models, or models predicting their uncertainty) and whether we have access and control to its internal structures. We invite the reader to refer to [Gawlikowski 21] for further details on uncertainty approaches. If the uncertainty measure targets the clinicians, its generation time should also fit with the clinical routine.

Applying uncertainty ideas to visual explanations- Our visual explanation cannot produce an uncertainty metric directly, as we do not enforce the method to capture the model's variability in the optimization. In parallel, the domain translation adding pathology to a healthy case does not give direct insights to detect misclassification. On the opposite, the other translation provides information whatever the initial prediction:

- True Positive: the attribution map localizes the relevant regions of the input while the comparison between the input and the counterfactual shows the pathological pattern.

- False Positive: the method shows the user what region looks pathological yet misleading to the model.
- False Negative: the method shows what region should be transformed (and how) if the model had predicted a pathology. Here the method may focus on the correct signs despite the model mistake.
- True negative: in the best scenario, the method should not modify the input. These examples work as control cases to set a baseline difference between the input and the generation for studying the other cases (especially the false negative samples).

We provide figures in Appendix F for the different model outcomes in pneumonia and brain tumor detection. We display the raw difference map for each case, i.e., $x - g_c^\star(x)$; $g_c^\star$ being the counterfactual generation for the translation "Predicted Pathological (χ_1) to Predicted Healthy (χ_0)" (except for SySCGen where we cannot control the translation). These raw maps offer other visualizations of relevant attributions and counterfactual generations. They point out the differences between the two images and how the counterfactual changes the input, e.g., change in texture or intensity. First, we notice that most false positives are small tumor regions (see Figure F.7). When preprocessing the images, we set to pathology all images having a tumor mask superior to 10 pixels (assuming the model would miss tumor frontier). However, the difference maps of all counterfactual methods highlight the small tumor regions. The only "real" false positive is shown in the fifth row. The model seems to focus on an area of higher intensity on the edge of the brain. In this case, the intensity and the location of the transformation vary between methods. Figure F.8 shows some false negative cases where the model did not catch a brain tumor. When compelling the counterfactual generator to produce a healthy generation, the different methods modify the relevant region in most cases, i.e., the tumor. In comparison, difference maps of the true negative cases (see Figure F.6) display a poor correlation when comparing the different counterfactual methods. In addition, as we mentioned in Section 8.3.1, SSyGen and SySCGen do not always follow the classifier's decision and sometimes produce the inverse translation (even if trained with the classification loss terms). In the false positive and negative examples above, SySCGen sometimes produces a "wrong direction" counterfactual w.r.t. the model or displays opposite transformations (some in one direction and some in the other). Partial or complete disagreements between the counterfactual generations should raise attention. Similar findings are found in the pneumonia detection problem (see Appendix F.1). The counterfactual difference maps point out confounding regions (often white and diffuse areas looking like an opacity) in false positive (see Figure F.3). The correlation between the difference maps (of diverse methods) in false negative samples highlights possible model mistakes (see Figure F.4).

In summary, to identify potential mistakes or doubtful cases in pathology detection problems:

- We should compare diverse counterfactual methods together
- We should use the "Pathological to Healthy" translation (when allowed)
- For pathological predictions, the visual explanations emphasize the relevant or confounding regions. The clinician should still pay attention and check for mistakes. Sometimes, we notice a lower correlation between the different techniques in false positive cases. Additional assessments are required to generalize this observation.
- For healthy prediction, enforcing "Pathological to Healthy" translations either generates heterogeneous transformations between methods (mostly the case for true negative), which are also of low intensity; or correlates with other methods towards similar regions, which is, in fact, the pathological region.
- If using a single generator without conditioning (e.g., SSyGen or SySCGen), we

should pay attention to opposite transformations or inverse translations.

These findings pave the way for future works on producing an uncertainty metric based on visual explanations. The observations gathered are qualitative and vary with the problem. We can also extract much more information from the various visual explanations. The way we should combine all these elements remains an open problem.

9.2.2 Generating Diverse Explanations

In the medical domain, the classifier can rely on several clinical signs to predict an image as pathological, e.g., edema, tissue texture, or intensity. Conversely, the classifier can be biased over a single sign (relevant or not for the clinician) and avoid all others. When transposing a (predicted) pathological input into a healthy image, we would like

1. To study the impact of different input patterns for the classifier (separately or not)
2. To reveal possible biases learned by the classifier

In the other direction (healthy $\longrightarrow$ pathological), our proposed method often produces the same pathology (with consistent texture) at a similar location (see limitation Section 8.3.2). Producing diverse counterfactuals could generate a mapping of pathologies the model has learned, e.g., showing the different textures, locations, sizes, types of pathology, or diverse clinical signs. It would define a global explanation translating the relevant patterns learned over the training database.
In the following, we update the formulation from Chapter 3, introducing primary attempts to produce diverse generations. Some other approaches consider the latent space of GANs, particularly the StyleGAN [Karras 19, Lang 21]. We mentioned them in the related work Section 2.2. These methods could be used alongside ours, as they provide complementary information.

9.2.2.1 A new property applied on the counterfactual generator

While the stable generation $g_s^\star(x)$ reconstructs the input image and satisfies the regularity condition, $g_c^\star(x)$ should capture the different type of image patterns relevant for f. If the classifier's decision relies on different factors (e.g., localization, texture, intensity, size, types of pathology), $g_c^\star(x)$ should translate this property of **Diversity**.

9.2.2.2 Updated optimization problem

Keeping the same notations as in Section 3.2, we update the general optimization of g_c and g_s for a binary classification task. Now summarizing Relevance, Regularity, Realism, and Diversity conditions, $g_s^\star$ and $g_c^\star$ may be searched as a solution couple to the optimization problem

$$(g_s^\star, g_c^\star) = \underset{g_s, g_c}{\operatorname{argmin}} \overbrace{\Big[\underbrace{r_g(g_s, g_c)}_{\textbf{Regularity}} + \mathbb{E}_x\left(d_s(x, g_s(x)) + d_c(x, g_c(x))\right)\Big]}^{\textbf{Relevance}} + \underbrace{\mathbb{E}_x\, v_{g_c}(g_c(x), \chi_0, \chi_1)}_{\textbf{Diversity}}$$
$$\underbrace{\text{s.t.} \left\{ \begin{array}{l} g_c(\chi_0) \subset \chi_1, g_c(\chi_1) \subset \chi_0 \\ g_s(\chi_0) \subset \chi_0, g_s(\chi_1) \subset \chi_1 \end{array} \right\}}_{\textbf{Realism}} \tag{9.1}$$

Where the addition function v_{g_c} encourages g_c to produce diverse counterfactual images mapping the different realistic patterns from the sub-spaces χ_0 and χ_1 learned by f.

9.2.2.3 Implementation attempts

This last property on **Diversity** is still a work in progress, and we did not achieve conclusive experimental results. We briefly describe some propositions and attempts in the following. Our propositions are mainly inspired from domain translation works enforcing multi-modality and therefore diversity.

Noise injection- Inspired by StyleGAN [Karras 19], we first propose to inject noise into our generator model. In their work, they add noise at different generator scales to stabilize the generation and increase the generation diversity. In practice, we sample a random vector η, passed to the generator's decoder part. Different η are sampled for the stable and the counterfactual image. Figure 9.1 (Top) illustrates the noise injection for the CyLatentCE framework. The idea is that different η should produce diverse generations for a given class domain. We tested different strategies for η:

- $\eta \sim \mathcal{N}(0, 1)$.
- $\eta = M(z, f(x))$, where M is a mapping model as in StyleGAN [Karras 19] or StarGAN2 [Choi 20]; $z \sim \mathcal{N}(0, 1)$; and M can be conditioned by the prediction (or the target) $f(x)$. In this case, we try to learn a non-linear mapping space that disentangles different image attributes.

When sampling $\eta \sim \mathcal{N}(0, 1)$, our approach resembles SMIT [Wang 19] and SDIT [Romero 19] presented in the related work Section 2.2.1. They sample a random code and pass it through a conditional normalization layer to obtain diverse attributes for each sampling. The random code is combined with a domain code that drives the image translation. In addition, they use attention mechanisms (e.g., image-level mask or latent attention module) to restrict the transformations towards the attributes of interest. These frameworks are dedicated to multi-modal translations and not to visual explanations. The domain optimization is not guided by a trained (and fixed) classification model (as in we do). Instead, they learn the class domain through a discriminator with an additional classification head.
Inspired by [Lee 18, He 19, Choi 20], we explicitly enforce diversity in the second framework (see Figure 9.1 Bottom), adding a loss function L_{div}. This term encourages that two counterfactual images differ when generated from two random codes η and η'. We either maximize a distance between the two generations (considering or not the distance between η and η') [Lee 18, Choi 20], or we train an additional encoder to predict the sampled random code by maximizing the mutual information [He 19].
In practice, we obtain some generation diversity in the two directions of translation. When removing a pathology, the generations mainly differ on the intensity level, i.e., to what extent healthy tissues replace the pathology. In the other direction, the size and intensity of the added pneumonia opacity or brain tumor vary (more visible on pneumonia). However, the location and the texture remain very similar throughout the generation, while variability exists in the training database.

Content and Attributes disentanglement- Some other works (introduced in Section 2.2.1) separate the content (e.g., the image pose, the global image structures) and the style (e.g., the colors, the fine-grained details) of images to perform domain translation.

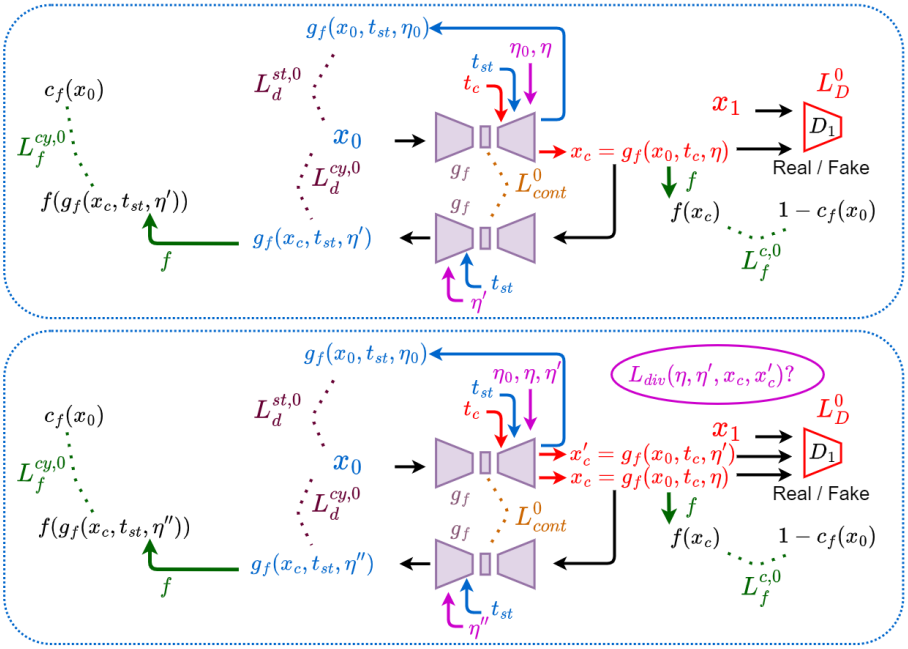


Figure 9.1: Overview of CyLatentCE with diversity frameworks. Top: A first optimization framework injecting noise to encourage diverse generations. Bottom: A second framework with explicit terms to enforce diversity. For the two figures, we illustrate a training step of g_f for an original image $x_0 \in \chi_0$. Similar procedures for the opposite translation. The terms L_i^0 are the loss parts L_i that act on $x_0 \in \chi_0$. **Top**: ***Counterfactual path***: an input image x_0 is given to the generator g_f (black arrow) and conditioned through its latent space by a random vector η and by $t_c = c_f(x_1) = 1 - c_f(x_0)$ in the binary case (or directly $f(x_1)$). It produces a counterfactual image $x_c = g_f(x_0, c_f(x_1), \eta)$. This generated image is enforced ($L_f^{c\,0}$) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D_1, trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). The injection of η adds some randomness to the generation and implicitly enforces diversity. ***Stable path***: the input image x_0 is given to the generator g_f and conditioned by a random vector η_0, and by the original prediction $t_{st} = c_f(x_0)$ (or $f(x_0)$). It produces a stable image $g_f(x_0, c_f(x_0), \eta_0)$ which is enforced to be pixel-wise close to x_0 by the term $L_d^{st,0}$. ***Cyclic path***: x_c is also mapped back to χ_0 through cycle consistency (black arrow bellow x_c). Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L_d^{cy,0}$) and ($L_f^{cy,0}$). **Top**: The procedure is similar, except that two random vectors η and η' are sampled at each step and produce two counterfactual generations x_c and x_c'. The counterfactual x_c' is given to the discriminator to compute a second L_D^0 term. An additional term L_{div} can be added to enforce generation diversity explicitly.

They first extract content and style information from the input using dedicated encoders. Then, they translate an image into a new domain by mixing its content with another encoded style. The content is generally encoded as feature maps while the style is compressed into a vector of low dimension, e.g., 8 to 512. Many ways of combining content and style information have been proposed [Lee 18, Yu 18c, Bass 20, Choi 20]. Except for using two specific encoders, these frameworks remain close to our work. They impose similar constraints, e.g., stable proximity, cycle consistency, domain discriminator for adversarial

training, and sometimes classification losses (applied to a single discriminator model with an additional classification head). They impose diversity by combining different styles and content extracted from input images at each step. In addition, they often enforce the style space to match a Gaussian normal distribution, i.e., $\mathcal{N}(0,1)$. [Lee 18, Bass 20] even use the reparametrization trick from variational autoencoder and perform domain translation by mixing content feature maps either with a style vector encoded from an input of another domain, or a random style vector sampled in $\mathcal{N}(0,1)$. In contrast, [Choi 20] encourage the style space to match a mapping space similar to StyleGAN [Karras 19]. Finally, they explicitly enforce generation diversity by maximizing the distance (L_{div}) between translated generations produced with different style codes.
In our case, pathological and healthy images only differ in pathological regions, and the content of the images remains similar throughout the database for a given task. We assume we can extract pathology information as image attributes or a style. We thus consider a similar disentanglement strategy. We illustrate in Figures 9.2 and G.1, optimization frameworks where the counterfactual attributes are either switched with attributes of the opposite class or randomly sampled (we also try to combine the two approaches). We have tried to combine the different ideas of the literature, but we have not yet achieved conclusive results. The generation diversity is generally very limited (poorer than the noise injection strategy), i.e., we obtain very similar generations for the two translations, respectively, and the attribute information does not seem to encode the pathology variability. This framework is also more complex, longer to train, and less stable. In addition, most literature review considers problems where the style affects the global image, e.g., picture to painting or summer to winter landscape translations. The style impacts the colors and structures of the whole image. It is not restricted to specific and localized regions, except in [Bass 20], where the style (or attribute) vector learns to separate healthy and pathological brain cases. We elaborate more in Appendix G.

Image-level disentanglement- Instead of disentangling content and attributes in the latent space, we consider a similar problem at the image level. However, we redefine the problem and state that for pathology detection tasks, the image attributes of interest only concern the pathology. In this formulation, the image content is set to the healthy version of an input. Let g_{cont} and g_{attr} be the content and attribute generator. We thus have:

$$\begin{cases} x_0 \approx g_{cont}(x_0) + g_{attr}(x_0) \approx g_{cont}(x_0), & x_0 \in \chi_0 \\ x_1 \approx g_{cont}(x_1) + g_{attr}(x_1), & x_1 \in \chi_1 \end{cases} \tag{9.2}$$

Where $g_{attr}(x_0) \approx 0$ as x_0 is predicted healthy and should not contain pathology; $g_{cont}(x_1)$ is the healthy counterfactual image of x_1; and $g_{attr}(x_1)$ is the extracted pathology from x_1. Then, we add the extracted pathology to the healthy image to produce a pathological counterfactual image. In this formulation, the domain translation is not symmetrical as the image content is biased towards the healthy prediction. This approach is close in spirit to [Andermatt 18, Vorontsov 20] introduced in the related work Section 2.2.1, but rather focuses on weak segmentation tasks. Compared to our counterfactual embodiment (see Chapter 5), this formulation implies that

$$g_c(x) = \begin{cases} g_{cont}(x) + g_{attr}(x_1) & \text{if } x \in \chi_0,\ x_1 \in \chi_1 \\ g_{cont}(x) & \text{if } x \in \chi_1 \end{cases} \tag{9.3}$$

and

$$g_s(x) = \begin{cases} g_{cont}(x) + g_{attr}(x) \approx g_{cont}(x) & \text{if } x \in \chi_0 \\ g_{cont}(x) + g_{attr}(x) & \text{if } x \in \chi_1 \end{cases} \tag{9.4}$$

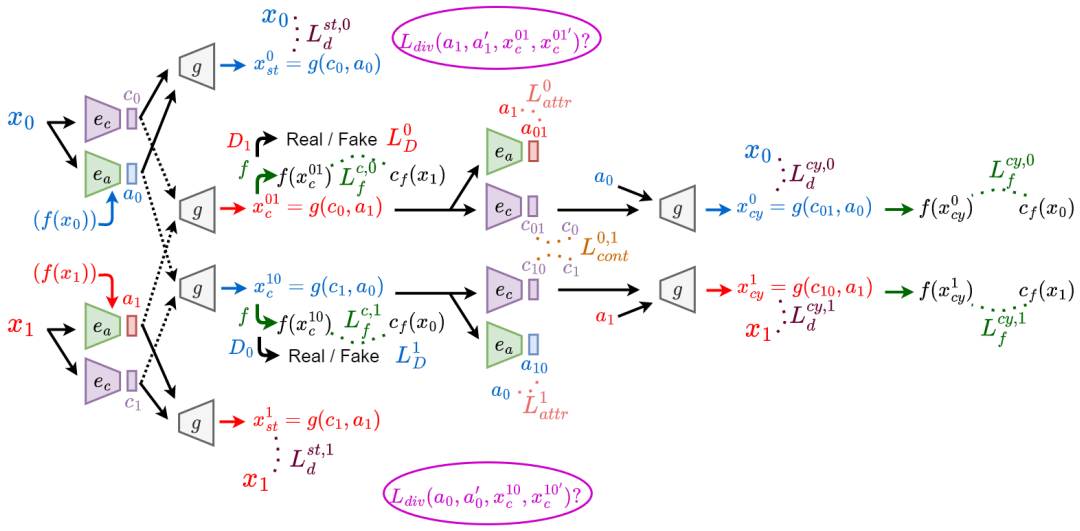


Figure 9.2: Overview of Content and Attributes disentanglement framework. The terms L_i^0 (resp. L_i^1) are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). ***Stable path***: At each step, input images x_0 and x_1 are given to the content e_c and attributes e_a encoders. The attribute encoding can also be conditioned by the input's prediction $f(x_0)$. This generates content and attributes encodings c_0 and a_0. Recombining the two encodings through a generator g (with a decoder structure) produces the stable image $g(c_0, a_0)$. It is enforced to be pixel-wise close to x_0 by the term $L_d^{st,0}$. ***Counterfactual path***: Counterfactual images are produced by switching the attributes encodings extracted from the two input images. We thus obtain $x_c^{01} = g(c_0, a_1)$ (resp. $x_c^{10} = g(c_1, a_0)$), the counterfactual image of x_0 (resp. x_1) in the domain χ_1 (resp. χ_0). This generated image is enforced ($L_f^{c,0,1}$) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D_1 (resp. D_0) that is trained to identify real from generated images in the distribution of images predicted in class 1 (resp 0), i.e., χ_1 (resp. χ_0). Diversity is implicitly encouraged as the attributes encoding should differ for each image of the counterfactual domain. An additional term L_{div} can be added to enforce generation diversity explicitly. ***Cyclic path***: x_c^{01} (resp. x_c^{10}) is also mapped back to χ_0 through cycle consistency. The counterfactual image is first encoded by e_c and e_a. A content loss enforces that the content remains the same across domains, i.e., $c_0 \approx c_{01}$ (for x_0). An attributes term enforces the attributes stability inside a domain (e.g., χ_0), i.e., $a_1 \approx a_{01}$. The cycle image x_{cy}^0 (resp. x_{cy}^0) is generated (through g) by combining the counterfactual encoded content c_{01} (resp. x_{cy}^0) and the initial encoded attributes a_0. Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L_d^{cy,0}$) and ($L_f^{cy,0}$).

We describe the optimization framework in the Appendix (see Figure G.2), and display some results in Figure 9.3.
We observe that extracted pathologies from the last column are well added to the healthy image (4th column). The addition retains the pathology location, size, and form as expected. In contrast, the relative intensity and texture of the pathology seem to vary (a little). However, this "additive" formulation is insufficient as the global input geometry is not equivalent to the counterfactual image from which we extract the pathology.

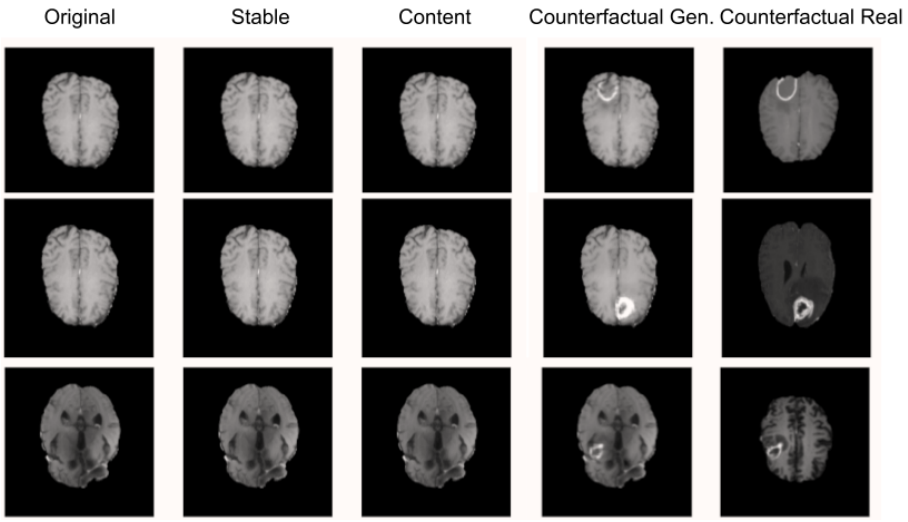


Figure 9.3: Examples of pathology translations - Image-level content and pathology disentanglement. From left to right: the healthy input image $x_0 \in \chi_0$, the stable generation $(g_{cont}(x_0) + g_{attr}(x_0))$, the content generation $(g_{cont}(x_0))$, the pathological counterfactual generation $(g_{cont}(x_0) + g_{attr}(x_1))$, and the counterfactual image $x_1 \in \chi_1$ used to extract the pathology $g_{attr}(x_1)$.

Figure 9.4 shows a case where the geometries between the healthy input and the pathological query differ. In that case, the tumor is added outside the brain, generating a non-realistic image.

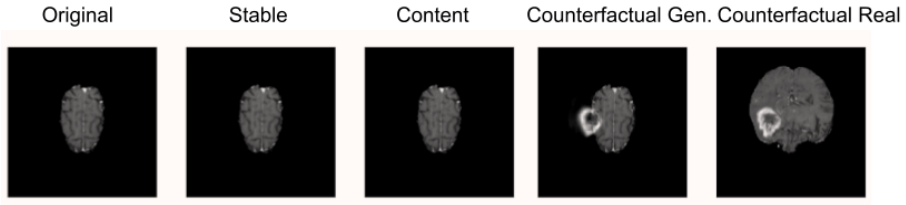


Figure 9.4: Example of pathology translations failure - Image-level content and pathology disentanglement. From left to right: the healthy input image $x_0 \in \chi_0$, the stable generation $(g_{cont}(x_0)+g_{attr}(x_0))$, the content generation $(g_{cont}(x_0))$, the pathological counterfactual generation $(g_{cont}(x_0) + g_{attr}(x_1))$ where the pathology is added outside the brain, and the counterfactual image $x_1 \in \chi_1$ used to extract the pathology $g_{attr}(x_1)$.

Future works focus on pathology transfer (ensuring generation realism) and how we can manipulate the extracted pathology during the addition, e.g., the size, the location, and the intensity.

We differ from the literature for all these approaches as we seek to explain a trained classification model kept fixed during the optimization. Indeed, we do not train a classification model or a segmentation model within our generation optimization. As we pointed out in Section 8.3.1, our generator often produces the same transformations at the exact location when adding pathology to a healthy image. The preliminary experiments on generation diversity suggest that the classification guidance may induce short cut in the healthy to pathological translation. As the classification model is fixed, a consistent

adversarial pattern or a sufficient pathology may satisfy the classification objective and limit the generation to these restricted examples.

9.2.3 White box visual explanations

In this thesis, we develop visual explanation techniques for classification with little or no access to the model structures –the integrated version has access to the model's gradient. However, we expect the counterfactual generations (especially their diversity) and visual explanations to improve when accessing the model's internal structures. It makes sense to adopt such a strategy for Incepto's products. In this situation, we can access the model's feature maps at different scales and provide more information about the generation process than a prediction score. From the generation diversity perspective, we also expect that we can trade the classification guidance (or at least reduce the impact) with direct access to the model's feature maps and enforce proximity between features maps of the input and the generated counterfactuals.

Bibliography

[Abdal 19] Rameen Abdal, Yipeng Qin & Peter Wonka. *Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?* In ICCV, 2019.

[Abdal 20] Rameen Abdal, Yipeng Qin & Peter Wonka. *Image2StyleGAN++: How to Edit the Embedded Images?* In CVPR, 2020.

[Adebayo 18] Julius Adebayo, J. Gilmer, M. Muelly, Ian J. Goodfellow, Moritz Hardt & Been Kim. *Sanity Checks for Saliency Maps.* In NeurIPS, 2018.

[Agarwal 20] Chirag Agarwal & Anh Nguyen. *Explaining image classifiers by removing input features using generative models.* In ACCV, 2020.

[Alaluf 21] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal & Amit H. Bermano. *HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing.* ArXiv, 2021.

[Alemi 17] Alexander A. Alemi, Ian S. Fischer, Joshua V. Dillon & Kevin P. Murphy. *Deep Variational Information Bottleneck.* ICLR, 2017.

[Alvarez-Melis 18] David Alvarez-Melis & T. Jaakkola. *Towards Robust Interpretability with Self-Explaining Neural Networks.* In NeurIPS, 2018.

[Ancona 18] Marco Ancona, Enea Ceolini, A. Cengiz Öztireli & Markus H. Gross. *A unified view of gradient-based attribution methods for Deep Neural Networks.* ICLR, 2018.

[Andermatt 18] Simon Andermatt, Antal Horváth, Simon Pezold & Philippe C. Cattin. *Pathology Segmentation using Distributional Differences to Images of Healthy Origin.* In MICCAI Brain lesion Workshop, 2018.

[Andrews 95] Robert Andrews, Joachim Diederich & Alan B. Tickle. *Survey and critique of techniques for extracting rules from trained artificial neural networks.* In Knowledge-based Systems, 1995.

[Arjovsky 17] Martín Arjovsky, Soumith Chintala & Léon Bottou. *Wasserstein GAN.* ArXiv, vol. abs/1701.07875, 2017.

[Arrieta 20] Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, A. Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila & Francisco Herrera. *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI.* In Information Fusion, 2020.

[Bach 15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller & Wojciech Samek. *On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation.* In PLoS ONE, 2015.

[Bakas 17] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S. Kirby, John B. Freymann, Keyvan Fara-

hani & Christos Davatzikos. *Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features.* In Scientific Data, 2017.

[Bakas 18] Spyridon Bakas, Mauricio Reyes, András Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, Marcel Prastawa, Esther Alberts, Jana Lipková, John B. Freymann, Justin S. Kirby, Michel Bilello, Hassan M. Fathallah-Shaykh, Roland Wiest, Jan Stefan Kirschke, Benedikt Wiestler, Rivka R. Colen, Aikaterini Kotrotsou, Pamela LaMontagne, Daniel S. Marcus, Mikhail Milchenko, Arash Nazeri, Marc-André Weber, Abhishek Mahajan, Ujjwal Baid, Dongjin Kwon, Manu Agarwal, Mahbubul Alam, Alberto Albiol, Antonio Albiol, Alex Varghese, Tran Anh Tuan, Tal Arbel, Aaron Avery, B. Pranjal, Subhashis Banerjee, Thomas Batchelder, Nematollah Batmanghelich, Enzo Battistella, Martin Bendszus, Eze Benson, José Bernal, George Biros, Mariano Cabezas, Siddhartha Chandra, Yi-Ju Chang & et al. *Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge.* ArXiv, 2018.

[Barnett 21] Alina Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y. Lo & Cynthia Rudin. *IAIA-BL: A Case-based Interpretable Deep Learning Model for Classification of Mass Lesions in Digital Mammography.* ArXiv, vol. abs/2103.12308, 2021.

[Bashkirova 19] Dina Bashkirova, Ben Usman & Kate Saenko. *Adversarial Self-Defense for Cycle-Consistent GANs.* In NeurIPS, 2019.

[Bass 20] Cher Bass, Mariana da Silva, C. Sudre, Petru-Daniel Tudosiu, S. Smith & E. Robinson. *ICAM: Interpretable Classification via Disentangled Representations and Feature Attribution Mapping.* In NeurIPS, 2020.

[Bau 17] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva & Antonio Torralba. *Network Dissection: Quantifying Interpretability of Deep Visual Representations.* In CVPR, 2017.

[Baumgartner 18] Christian F. Baumgartner, L. Koch, K. C. Tezcan, Jia Xi Ang & E. Konukoglu. *Visual Feature Attribution Using Wasserstein GANs.* In CVPR, 2018.

[Bermano 22] Amit H. Bermano, Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Oren Patashnik & Daniel Cohen-Or. *State-of-the-Art in the Architecture, Methods and Applications of StyleGAN.* Computer Graphics Forum, 2022.

[Bien 18a] N Bien, P Rajpurkar, RL Ball, J Irvin, A Park & E Jones. *Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation.* PLoS Med, 2018.

[Bien 18b] Nicholas Bien, Pranav Rajpurkar, Robyn Ball, Jeremy Irvin, Allison Park, Erik Jones, Michael Bereket, Bhavik Patel, Kristen Yeom, Katie Shpanskaya, Safwan Halabi, Evan Zucker, Gary Fanton, Derek Amanatullah, Christopher Beaulieu, Geoffrey Riley, Russell Stewart, Francis Blankenberg, David Larson & Matthew Lungren. *Deep-learning-assisted diagnosis for knee magnetic resonance*

imaging: Development and retrospective validation of MRNet. PLOS Medicine, vol. 15, 2018.

[Biffi 18] Carlo Biffi, O. Oktay, G. Tarroni, Wenjia Bai, A. S. M. D. Marvao, Georgia Doumou, Martin Rajchl, R. Bedair, S. K. Prasad, S. Cook, D. O'Regan & D. Rueckert. *Learning Interpretable Anatomical Features Through Deep Generative Models: Application to Cardiac Remodeling.* In MICCAI, 2018.

[Boehle 21] Moritz D Boehle, Mario Fritz & Bernt Schiele. *Convolutional Dynamic Alignment Networks for Interpretable Classifications.* In CVPR, 2021.

[Bojarski 18] Mariusz Bojarski, Anna Choromańska, Krzysztof Choromanski, Bernhard Firner, Larry J. Ackel, Urs Muller, Philip Yeres & Karol Zieba. *VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving.* In ICRA, 2018.

[Carlini 17] Nicholas Carlini & David A. Wagner. *Towards Evaluating the Robustness of Neural Networks.* In Symposium on Security and Privacy, 2017.

[Challen 07] J Challen, Y Tang, K Hazratwala & S Stuckey. *Accuracy of MRI diagnosis of internal derangement of the knee in a non-specialized tertiary level referral teaching hospital.* Australas Radiol., 2007.

[Chang 19] Chun-Hao Chang, Elliot Creager, Anna Goldenberg & David Kristjanson Duvenaud. *Explaining Image Classifiers by Counterfactual Generation.* In ICLR, 2019.

[Charachon 20] Martin Charachon, C. Hudelot, Paul-Henry Cournède, Camille Ruppli & R. Ardon. *Combining Similarity and Adversarial Learning to Generate Visual Explanation: Application to Medical Image Classification.* In ICPR, 2020.

[Charachon 21] Martin Charachon, Paul-Henry Cournède, Céline Hudelot & Roberto Ardon. *Visual Explanation by Unifying Adversarial Generation and Feature Importance Attributions.* In iMIMIC/TDA4MedicalData@MICCAI, 2021.

[Charachon 22] Martin Charachon, Paul-Henry Cournède, Céline Hudelot & Roberto Ardon. *Leveraging conditional generative models in a general explanation framework of classifier decisions.* In Future Generation Computer Systems, 2022.

[Chattopadhyay 18] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader & Vineeth N. Balasubramanian. *Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks.* In WACV, 2018.

[Chen 16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever & P. Abbeel. *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets.* In NIPS, 2016.

[Chen 18] Tian Qi Chen, Xuechen Li, Roger B. Grosse & David Kristjanson Duvenaud. *Isolating Sources of Disentanglement in Variational Autoencoders.* In NeurIPS, 2018.

[Chen 19] Chaofan Chen, Oscar Li, Alina Barnett, Jonathan K. Su & Cynthia Rudin. *This looks like that: deep learning for interpretable image recognition.* In NeurIPS, 2019.

[Cherepkov 21] A. V. Cherepkov, Andrey Voynov & Artem Babenko. *Navigating the GAN Parameter Space for Semantic Image Editing.* In CVPR, 2021.

[Chiou 20] Eleni Chiou, F. Giganti, S. Punwani, I. Kokkinos & E. Panagiotaki. *Harnessing Uncertainty in Domain Adaptation for MRI Prostate Lesion Segmentation.* In MICCAI, 2020.

[Choi 18] Yunjey Choi, Min-Je Choi, Mun Su Kim, Jung-Woo Ha, Sunghun Kim & Jaegul Choo. *StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation.* In CVPR, 2018.

[Choi 20] Yunjey Choi, Youngjung Uh, Jaejun Yoo & Jung-Woo Ha. *StarGAN v2: Diverse Image Synthesis for Multiple Domains.* In CVPR, 2020.

[Chong 21] Min Jin Chong, Wen-Sheng Chu, Abhishek Kumar & David Forsyth. *Retrieve in Style: Unsupervised Facial Feature Transfer and Retrieval.* In ICCV, 2021.

[Cohen 21] Joseph Paul Cohen, Rupert Brooks, Evan Zucker, Anuj Pareek, Matthew P. Lungren & Akshay Chaudhari. *Gifsplanation via Latent Shift: A Simple Autoencoder Approach to Counterfactual Generation for Chest X-rays.* In MIDL, 2021.

[Collins 20] Edo Collins, Raja Bala, Bob Price & Sabine Süsstrunk. *Editing in Style: Uncovering the Local Semantics of GANs.* In CVPR, 2020.

[Combalia 20] Marc Combalia, Ferran Hueto, Susana Puig, Josep Malvehy & Verónica Vilaplana. *Uncertainty Estimation in Deep Neural Networks for Dermoscopic Image Classification.* In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[Dabkowski 17] Piotr Dabkowski & Yarin Gal. *Real Time Image Saliency for Black Box Classifiers.* In NIPS, 2017.

[Deng 09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li & L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database.* In CVPR, 2009.

[Dhurandhar 18] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam & Payel Das. *Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives.* In NeurIPS, 2018.

[Doshi-Velez 17] Finale Doshi-Velez & Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning.* ArXiv, 2017.

[Du 18] Mengnan Du, Ninghao Liu, Qingquan Song & Xia Hu. *Towards Explanation of DNN-based Prediction with Guided Feature Inversion.* In ACM SIGKDD, 2018.

[Du 21] Yuanqi Du, Quan Quan, Hu Han & S. Kevin Zhou. *Where is the disease? Semi-supervised pseudo-normality synthesis from an abnormal image.* ArXiv, 2021.

[Egger 22] Jan Egger, Christina Gsaxner, Antonio Pepe, Kelsey L. Pomykala, Frederic Jonske, Manuel Kurz, Jianning Li & Jens Kleesiek. *Medical deep learning—A systematic meta-review.* Computer Methods and Programs in Biomedicine, 2022.

[Elliott 19] Andrew Elliott, Stephen Law & Chris Russell. *Adversarial Perturbations on the Perceptual Ball.* ArXiv, vol. abs/1912.09405, 2019.

[Endres 03] D. M. Endres & J. E. Schindelin. *A new metric for probability distributions.* IEEE Transactions on Information Theory, vol. 49, no. 7, 2003.

[Esser 20] Patrick Esser, Robin Rombach & Björn Ommer. *A Disentangling Invertible Interpretation Network for Explaining Latent Representations.* In CVPR, 2020.

[Esteva 17] Andre Esteva, Brett Kuprel, Roberto Novoa, Justin Ko, Susan Swetter, Helen Blau & Sebastian Thrun. *Dermatologist-level classification of skin cancer with deep neural networks.* In Nature, volume 542, 2017.

[Figueiredo 18] S Figueiredo, L Sa Castelo, AD Pereira, L Machado, JA Silva & A Sa. *Use of MRI by radiologists and orthopaedic surgeons to detect intra-articular injuries of the knee.* Revista Brasileira de Ortopedia, 2018.

[Fong 17] R. C. Fong & A. Vedaldi. *Interpretable Explanations of Black Boxes by Meaningful Perturbation.* In ICCV, 2017.

[Fong 19] Ruth Fong, Mandela Patrick & Andrea Vedaldi. *Understanding Deep Networks via Extremal Perturbations and Smooth Masks.* In ICCV, 2019.

[Fritz 20] B Fritz, G Marbach, F Civardi, SF Fucentese & CWA Pfirrmann. *Deep convolutional neural network-based detection of meniscus tears: comparison with radiologists and surgery as standard of reference.* Skeletal Radiology, 2020.

[Fu 19] Weijie Fu, Meng Wang, Mengnan Du, Ninghao Liu, Shijie Hao & Xia Hu. *Distribution-Guided Local Explanation for Black-Box Classifiers.* 2019.

[Gao 16] Yang Gao, Oscar Beijbom, Ning Zhang & Trevor Darrell. *Compact Bilinear Pooling.* In CVPR, 2016.

[Garreau 21] Damien Garreau & Dina Mardaoui. *What does LIME really see in images?* In ICML, 2021.

[Gawlikowski 21] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseo Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, M. Shahzad, Wen Yang, Richard Bamler & Xiaoxiang Zhu. *A Survey of Uncertainty in Deep Neural Networks.* ArXiv, 2021.

[Ghorbani 19] Amirata Ghorbani, James Wexler & Been Kim. *Automating Interpretability: Discovering and Testing Visual Concepts Learned by Neural Networks.* ArXiv, 2019.

[Gilpin 18] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter & Lalana Kagal. *Explaining Explanations: An Overview of Interpretability of Machine Learning.* In DSAA, 2018.

[Goetschalckx 19] Lore Goetschalckx, Alex Andonian, Aude Oliva & Phillip Isola. *GANalyze: Toward Visual Definitions of Cognitive Image Properties.* In ICCV, 2019.

[Goodfellow 14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville & Yoshua Bengio. *Generative Adversarial Nets.* In NIPS, 2014.

[Goodfellow 15] Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy. *Explaining and Harnessing Adversarial Examples.* In ICLR, 2015.

[Goodfellow 16] Ian J. Goodfellow, Yoshua Bengio & Aaron Courville. *Deep Learning.* In MIT Press, 2016.

[Goyal 19] Yash Goyal, Z. Wu, J. Ernst, Dhruv Batra, D. Parikh & Stefan Lee. *Counterfactual Visual Explanations.* In ICML, 2019.

[Guidotti 19] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi & Fosca Giannotti. *A Survey of Methods for Explaining Black Box Models.* In CSUR, 2019.

[Guidotti 22] Riccardo Guidotti. *Counterfactual explanations and how to find them: literature review and benchmarking.* In Data Mining and Knowledge Discovery, 2022.

[Gulrajani 17] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin & Aaron C. Courville. *Improved Training of Wasserstein GANs.* In NIPS, 2017.

[Härkönen 20] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen & Sylvain Paris. *GANSpace: Discovering Interpretable GAN Controls.* In NeurIPS, 2020.

[He 16] Kaiming He, Xiangyu Zhang, Shaoqing Ren & Jian Sun. *Identity Mappings in Deep Residual Networks.* In ECCV, 2016.

[He 19] Zhenliang He, Wangmeng Zuo, Meina Kan, S. Shan & Xilin Chen. *AttGAN: Facial Attribute Editing by Only Changing What You Want.* IEEE Transactions on Image Processing, 2019.

[Hendricks 16] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele & Trevor Darrell. *Generating Visual Explanations.* In ECCV, 2016.

[Heusel 17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler & S. Hochreiter. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.* In NIPS, 2017.

[Higgins 17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed & Alexander Lerchner. *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework.* In ICLR, 2017.

[Hochreiter 97] Sepp Hochreiter & Jürgen Schmidhuber. *Long Short-term Memory.* Neural computation, 1997.

[Hooker 19] Sara Hooker, D. Erhan, Pieter-Jan Kindermans & Been Kim. *A Benchmark for Interpretability Methods in Deep Neural Networks.* In NeurIPS, 2019.

[Hsieh 20] Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Ravikumar, Seungyeon Kim, Sanjiv Kumar & Cho-Jui Hsieh. *Evaluations and Methods for Explanation through Robustness Analysis.* ArXiv, vol. abs/2006.00442, 2020.

[Huang 17] Gao Huang, Zhuang Liu & Kilian Q. Weinberger. *Densely Connected Convolutional Networks.* In CVPR, 2017.

[Huang 18] X. Huang, Ming-Yu Liu, Serge J. Belongie & J. Kautz. *Multimodal Unsupervised Image-to-Image Translation.* In ECCV, 2018.

[Isola 17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou & Alexei A Efros. *Image-to-Image Translation with Conditional Adversarial Networks.* In CVPR, 2017.

[Jahanian 20] Ali Jahanian, Lucy Chai & Phillip Isola. *On the "steerability" of generative adversarial networks.* In ICLR, 2020.

[Jha 20] Anupama Jha, Joseph K. Aicher, Matthew R. Gazzara, Deependra Singh & Yoseph Barash. *Enhanced Integrated Gradients: improving interpretability of deep learning models using splicing codes as a case study.* In Genome Biology, 2020.

[Joshi 18] Shalmali Joshi, Oluwasanmi Koyejo, Been Kim & Joydeep Ghosh. *xGEMs: Generating Examplars to Explain Black-Box Models.* ArXiv, vol. abs/1806.08867, 2018.

[Jung 20] Dahuin Jung, Jonghyun Lee, Jihun Yi & Sungroh Yoon. *iCaps: An Interpretable Classifier via Disentangled Capsule Networks.* In ECCV, 2020.

[Kapishnikov 19] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Vi'egas & Michael Terry. *XRAI: Better Attributions Through Regions.* In ICCV, 2019.

[Karras 19] Tero Karras, Samuli Laine & Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks.* In CVPR, 2019.

[Karras 20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen & Timo Aila. *Analyzing and Improving the Image Quality of StyleGAN.* In CVPR, 2020.

[Karras 21] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen & Timo Aila. *Alias-Free Generative Adversarial Networks.* In NeurIPS, 2021.

[Khakzar 19] Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Seong Tae Kim & Nassir Navab. *Explaining Neural Networks via Perturbing Important Learned Features.* ArXiv, vol. abs/1911.11081, 2019.

[Khakzar 21] Ashkan Khakzar, Yang Zhang, Wejdène Mansour, Yuezhi Cai, Yawei Li, Yucheng Zhang, Seong Tae Kim & Nassir Navab. *Explaining COVID-19 and Thoracic Pathology Model Predictions by Identifying Informative Input Features.* In MICCAI, 2021.

[Kim 18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas & Rory Sayres. *Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV).* In ICML, 2018.

[Kim 21] Eunji Kim, Siwon Kim, Minji Seo & Sungroh Yoon. *XProtoNet: Diagnosis in Chest Radiography with Global and Local Explanations.* CVPR, 2021.

[Kindermans 19] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, D. Erhan & Been Kim. *The (Un)reliability of saliency methods.* In Explainable AI, 2019.

[Kingma 14] Diederik P. Kingma & Max Welling. *Auto-Encoding Variational Bayes.* In CoRR, 2014.

[Kingma 15] Diederik P. Kingma & Jimmy Ba. *Adam: A Method for Stochastic Optimization.* In ICLR, 2015.

[Kiureghian 09] Armen Der Kiureghian & Ove Ditlevsen. *Aleatory or epistemic? Does it matter?* Structural Safety, 2009.

[Kobayashi 21] Kazuma Kobayashi, Ryuichiro Hataya, Yusuke Kurose, Tatsuya Harada & Ryuji Hamamoto. *Decomposing Normal and Abnormal Features of Medical Images for Content-based Image Retrieval.* In Medical image analysis, 2021.

[Koh 17] Pang Wei Koh & Percy Liang. *Understanding Black-box Predictions via Influence Functions.* ICML, 2017.

[Krampla 09] W Krampla, M Roesel, K Svoboda, A Nachbagauer, M Gschwantler & W Hruby. *MRI of the knee: how do field strength and radiologist's experience influence diagnostic accuracy and interobserver correlation in assessing chondral and meniscal lesions and the integrity of the anterior cruciate ligament?* Eur Radiol., 2009.

[Krizhevsky 12] Alex Krizhevsky, Ilya Sutskever & Geoffrey E. Hinton. *ImageNet classification with deep convolutional neural networks.* In NIPS, 2012.

[Kurakin 17] Alexey Kurakin, Ian J. Goodfellow & Samy Bengio. *Adversarial examples in the physical world.* ArXiv, 2017.

[Lample 17] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer & Marc'Aurelio Ranzato. *Fader Networks: Manipulating Images by Sliding Attributes.* In NIPS, 2017.

[Lanfredi 19] Ricardo Bigolin Lanfredi, Joyce D. Schroeder, Clement Vachet & Tolga Tasdizen. *Adversarial regression training for visualizing the progression of chronic obstructive pulmonary disease with chest x-rays.* In MICCAI, 2019.

[Lang 21] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani & Inbar Mosseri. *Explaining in Style: Training a GAN to explain a classifier in StyleSpace.* In ICCV, 2021.

[Lecun 98] Y. Lecun, L. Bottou, Y. Bengio & P. Haffner. *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, vol. 86, no. 11, 1998.

[LeCun 10] Yann LeCun & Corinna Cortes. *MNIST handwritten digit database.* 2010.

[Lee 18] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh & Ming-Hsuan Yang. *Diverse Image-to-Image Translation via Disentangled Representations.* In ECCV, 2018.

[Lenis 20] D. Lenis, D. Major, M. Wimmer, A. Berg, Gert Sluiter & K. Bühler. *Domain aware medical image classifier interpretation by counterfactual impact analysis.* In MICCAI, 2020.

[Li 18] Oscar Li, Hao Liu, Chaofan Chen & Cynthia Rudin. *Deep Learning for Case-based Reasoning through Prototypes: A Neural Network that Explains its Predictions.* In AAAI, 2018.

[Lim 21] Dohun Lim, Hyeonseok Lee & Sungchan Kim. *Building Reliable Explanations of Unreliable Neural Networks: Locally Smoothing Perspective of Model Interpretation.* In CVPR, 2021.

[Liu 15] Ziwei Liu, Ping Luo, Xiaogang Wang & Xiaoou Tang. *Deep Learning Face Attributes in the Wild.* In ICCV, 2015.

[Liu 17] Ming-Yu Liu, Thomas Breuel & J. Kautz. *Unsupervised Image-to-Image Translation Networks.* In NIPS, 2017.

[Liu 18a] F Liu, Z Zhou, A Samsonov, D Blankenbaker, W Larison & A Kanarek. *Approach for Evaluating Knee MR Images: Achieving High Diagnostic Performance for Cartilage Lesion Detection.* Radiology, 2018.

[Liu 18b] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao & Bryan Catanzaro. *Image Inpainting for Irregular Holes Using Partial Convolutions.* In ECCV, 2018.

[Liu 19a] F Liu, B Guan, Z Zhou, A Samsonov, H Rosas & K Lian. *Fully Automated Diagnosis of Anterior Cruciate Ligament Tears on Knee MR Images by Using Deep Learning.* Radiology: Artificial Intelligence, 2019.

[Liu 19b] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo & Shilei Wen. *STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing.* In CVPR, 2019.

[Liu 19c] Shusen Liu, Bhavya Kailkhura, Donald Loveland & Yong Han. *Generative Counterfactual Introspection for Explainable Deep Learning.* In GlobalSIP, 2019.

[Liu 21] Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q. O'Neil & Sotirios A. Tsaftaris. *A Tutorial on Learning Disentangled Representations in the Imaging Domain.* ArXiv, 2021.

[Liu 22a] Kanglin Liu, Gaofeng Cao, Fei Zhou, Bozhi Liu, Jiang Duan & Guoping Qiu. *Towards Disentangling Latent Space for Unsupervised Semantic Face Editing.* IEEE Transactions on Image Processing, 2022.

[Liu 22b] Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q. O'Neil & Sotirios A.Tsaftaris. *Learning disentangled representations in the imaging domain.* Medical image analysis, 2022.

[López 21] Pau Rodríguez López, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam H. Laradji, Laurent Charlin & David Vázquez. *Beyond Trivial Counterfactual Explanations with Diverse Valuable Explanations.* In ICCV, 2021.

[Lundberg 17] Scott M. Lundberg & Su-In Lee. *A Unified Approach to Interpreting Model Predictions.* In NIPS, 2017.

[Madry 18] A. Madry, Aleksandar Makelov, Ludwig Schmidt, D. Tsipras & Adrian Vladu. *Towards Deep Learning Models Resistant to Adversarial Attacks.* In ICLR, 2018.

[Mahendran 15] Aravindh Mahendran & Andrea Vedaldi. *Understanding deep image representations by inverting them.* In CVPR, 2015.

[Major 20] David Major, Dimitrios Lenis, Maria Wimmer, Gert Sluiter, Astrid Berg & Katja Bühler. *Interpreting Medical Image Classifiers by Optimization Based Counterfactual Impact Analysis.* In ISBI, 2020.

[Mena 22] José Mena, Oriol Pujol & Jordi Vitrià. *A Survey on Uncertainty Estimation in Deep Learning Classification Systems from a Bayesian Perspective.* ACM Computing Surveys (CSUR), 2022.

[Menze 15] Bjoern H. Menze, András Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin S. Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth R. Gerstner, Marc-André Weber, Tal Arbel, Brian B. Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Çagatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andaç Hamamci, Khan M.

Iftekharuddin, Rajesh Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José Antonio Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen John Price, Tammy Riklin-Raviv, Syed M. S. Reza, Michael T. Ryan, Duygu Sarikaya, Lawrence H. Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos Alberto Silva, Nuno Sousa, Nagesh K. Subbanna, Gábor Székely, Thomas J. Taylor, Owen M. Thomas, N. Tustison, Gözde B. Ünal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes & Koenraad Van Leemput. *The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)*. In IEEE Transactions on Medical Imaging, 2015.

[Mirza 14] M. Mirza & Simon Osindero. *Conditional Generative Adversarial Nets*. ArXiv, vol. abs/1411.1784, 2014.

[Miyato 18] Takeru Miyato & Masanori Koyama. *cGANs with Projection Discriminator*. In ICLR, 2018.

[Montavon 18] Grégoire Montavon, Wojciech Samek & Klaus-Robert Müller. *Methods for interpreting and understanding deep neural networks*. In Digital Signal Processing, 2018.

[Mousavi 17] Ali Mousavi, Gautam Dasarathy & Richard Baraniuk. *DeepCodec: Adaptive sensing and recovery via deep convolutional neural networks*. In Allerton, 2017.

[Mrabet 12] Yassine Mrabet. *Human anatomy planes*, 2012.

[Murdoch 19] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl & Bin Yu. *Interpretable machine learning: definitions, methods, and applications*. ArXiv, 2019.

[Narayanaswamy 20] A. Narayanaswamy, Subhashini Venugopalan, D. R. Webster, Lily Peng, G. Corrado, Paisan Ruamviboonsuk, Pinal Bavishi, M. Brenner, P. Nelson & Avinash V. Varadarajan. *Scientific Discovery by Generating Counterfactuals using Image Translation*. In MICCAI, 2020.

[Nguyen 16] Anh M Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox & Jeff Clune. *Synthesizing the preferred inputs for neurons in neural networks via deep generator networks*. In NIPS, 2016.

[Oh 20] Kwanseok Oh, Jee Seok Yoon & Heung-Il Suk. *Born Identity Network: Multi-way Counterfactual Map Generation to Explain a Classifier's Decision*. ArXiv, 2020.

[Olah 18] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye & Alexander Mordvintsev. *The Building Blocks of Interpretability*. In Distill, 2018. https://distill.pub/2018/building-blocks.

[Pan 21] Deng Pan, Xin Li & D. Zhu. *Explaining Deep Neural Network Models with Adversarial Gradient Integration*. In IJCAI, 2021.

[Park 18] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell & Marcus Rohrbach. *Multimodal Explanations: Justifying Decisions and Pointing to the Evidence*. CVPR, 2018.

[Patr'icio 22] Cristiano Patr'icio, João C. Neves & Luís Filipe Teixeira. *Explainable Deep Learning Methods in Medical Diagnosis: A Survey*. ArXiv, 2022.

[Petsiuk 18] Vitali Petsiuk, Abir Das & Kate Saenko. *RISE: Randomized Input Sampling for Explanation of Black-box Models.* In BMVC, 2018.

[Pidhorskyi 20] Stanislav Pidhorskyi, Donald A. Adjeroh & Gianfranco Doretto. *Adversarial Latent Autoencoders.* In CVPR, 2020.

[Plumerault 20] Antoine Plumerault, Hervé Le Borgne & Céline Hudelot. *Controlling generative models with continuous factors of variations.* In ICLR, 2020.

[Qi 19] Zhongang Qi, S. Khorram & Fuxin Li. *Visualizing Deep Networks by Optimizing with Integrated Gradients.* In CVPR Workshops, 2019.

[Radford 16] Alec Radford, Luke Metz & Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.* In ICLR, 2016.

[Rajpurkar 17] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Yi Ding, Aarti Bagul, Curtis P. Langlotz, Katie S. Shpanskaya, Matthew P. Lungren & Andrew Y. Ng. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning.* ArXiv, 2017.

[Ribeiro 16] Marco Tulio Ribeiro, Sameer Singh & Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.* In ACM SIGKDD, 2016.

[Richardson 21] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro & Daniel Cohen-Or. *Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation.* In CVPR, 2021.

[Romero 19] Andrés Romero, Pablo Arbeláez, Luc Van Gool & Radu Timofte. *SMIT: Stochastic Multi-Label Image-to-Image Translation.* In ICCV Workshop, 2019.

[Ronneberger 15] Olaf Ronneberger, Philipp Fischer & Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* In MICCAI, 2015.

[Sabour 17] Sara Sabour, Nicholas Frosst & Geoffrey E. Hinton. *Dynamic Routing Between Capsules.* In NIPS, 2017.

[Samangouei 18] Pouya Samangouei, Ardavan Saeedi, Liam Nakagawa & Nathan Silberman. *ExplainGAN: Model Explanation via Decision Boundary Crossing Transformations.* In ECCV, 2018.

[Samek 17] W. Samek, Alexander Binder, Grégoire Montavon, S. Lapuschkin & K. Müller. *Evaluating the Visualization of What a Deep Neural Network Has Learned.* In IEEE Transactions on Neural Networks and Learning Systems, 2017.

[Schmitz 99] Gregor P. J. Schmitz, Chris Aldrich & F. S. Gouws. *ANN-DT: an algorithm for extraction of decision trees from artificial neural networks.* In IEEE transactions on neural networks, 1999.

[Schulz 20] Karl Schulz, Leon Sixt, Federico Tombari & Tim Landgraf. *Restricting the Flow: Information Bottlenecks for Attribution.* In ICLR, 2020.

[Schutte 21] Kathryn Schutte, Olivier Moindrot, Paul Hérent, Jean-Baptiste Schiratti & Simon Jégou. *Using StyleGAN for Visual Interpretability of Deep Learning Models on Medical Images.* In Neurips Workshop, 2021.

[Scott 92] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons, 1992.

[Seah 19] Jarrel C. Y. Seah, Jennifer S. N. Tang, Andy Kitchen, Frank Gaillard & Andrew F. Dixon. *Chest Radiographs in Congestive Heart Failure: Visualizing Neural Network Learning.* In Radiology, 2019.

[Selvaraju 17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh & Dhruv Batra. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization.* In ICCV, 2017.

[Seo 20] Dasom Seo, Kanghan Oh & Il-Seok Oh. *Regional Multi-Scale Approach for Visually Pleasing Explanations of Deep Neural Networks.* In IEEE Access, 2020.

[Shen 20a] Yujun Shen, Jinjin Gu, Xiaoou Tang & Bolei Zhou. *Interpreting the Latent Space of GANs for Semantic Face Editing.* In CVPR, 2020.

[Shen 20b] Zengming Shen, Yifan Chen, Thomas S. Huang, S.Kevin Zhou, Bogdan Georgescu & Xuqi Liu. *One-to-one Mapping for Unpaired Image-to-image Translation.* In WACV, 2020.

[Shitole 21] Vivswan Shitole, Li Fuxin, Minsuk Kahng, Prasad Tadepalli & Alan Fern. *One Explanation is Not Enough: Structured Attention Graphs for Image Classification.* In NeurIPS, 2021.

[Shrikumar 16] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina & Anshul Kundaje. *Not Just a Black Box: Learning Important Features Through Propagating Activation Differences.* ArXiv, 2016.

[Shrikumar 17] Avanti Shrikumar, Peyton Greenside & Anshul Kundaje. *Learning Important Features Through Propagating Activation Differences.* In ICML, 2017.

[Siddiquee 19] M. R. Siddiquee, Zongwei Zhou, Nima Tajbakhsh, Ruibin Feng, M. Gotway, Yoshua Bengio & Jianming Liang. *Learning Fixed Points in Generative Adversarial Networks: From Image-to-Image Translation to Disease Detection and Localization.* In ICCV, 2019.

[Simonyan 14] Karen Simonyan, Andrea Vedaldi & Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.* In ICLR, 2014.

[Simpson 19] A. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Ginneken, A. Kopp-Schneider, B. Landman, G. Litjens, B. Menze, O. Ronneberger, R. Summers, Patrick Bilic, P. Christ, R. Do, M. Gollub, Jennifer Golia-Pernicka, S. Heckers, W. Jarnagin, M. McHugo, S. Napel, E. Vorontsov, L. Maier-Hein & M. Jorge Cardoso. *A large annotated medical image dataset for the development and evaluation of segmentation algorithms.* ArXiv, vol. abs/1902.09063, 2019.

[Singla 20] Sumedha Singla, B. Pollack, Junxiang Chen & K. Batmanghelich. *Explanation by Progressive Exaggeration.* In ICLR, 2020.

[Singla 21] Sumedha Singla, Stephen Wallace, Sofia Triantafillou & K. Batmanghelich. *Using Causal Analysis for Conceptual Deep Learning Explanation.* Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention, 2021.

[Slack 21] Dylan Slack, Sophie Hilgard, Sameer Singh & Himabindu Lakkaraju. *Reliable Post hoc Explanations: Modeling Uncertainty in Explainability.* In NeurIPS, 2021.

[Smilkov 17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas & Martin Wattenberg. *SmoothGrad: removing noise by adding noise.* ArXiv, vol. abs/1706.03825, 2017.

[Springenberg 15] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox & Martin A. Riedmiller. *Striving for Simplicity: The All Convolutional Net.* In ICLR, volume abs/1412.6806, 2015.

[Sundararajan 17] Mukund Sundararajan, Ankur Taly & Qiqi Yan. *Axiomatic Attribution for Deep Networks.* In ICML, 2017.

[Szegedy 15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, D. Erhan, Vincent Vanhoucke & Andrew Rabinovich. *Going deeper with convolutions.* In CVPR, 2015.

[Tardy 21] Mickael Tardy & Diana Mateus. *Looking for Abnormalities in Mammograms With Self- and Weakly Supervised Reconstruction.* In IEEE Transactions on Medical Imaging, 2021.

[Thakur 21] Shailja Thakur & Sebastian Fischmeister. *A generalizable saliency map-based interpretation of model outcome.* In ICPR, 2021.

[Tov 21] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik & Daniel Cohen-Or. *Designing an encoder for StyleGAN image manipulation.* ACM Transactions on Graphics (TOG), 2021.

[Uzunova 19] Hristina Uzunova, Jan Ehrhardt, Timo Kepp & Heinz Handels. *Interpretable explanations of black box classifiers applied on medical images by meaningful perturbations using variational autoencoders.* In Medical Imaging: Image Processing, 2019.

[Vaswani 17] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. *Attention is All you Need.* In NIPS, 2017.

[Verma 20] Sahil Verma, John P. Dickerson & Keegan E. Hines. *Counterfactual Explanations for Machine Learning: A Review.* In Workshop NeurIPS, 2020.

[Vorontsov 20] Eugene Vorontsov, Pavlo Molchanov, Christopher Beckham, Jan Kautz & Samuel Kadoury. *Towards annotation-efficient segmentation via image-to-image translation.* In IEEE Transactions on Medical Imaging, 2020.

[Voynov 20] Andrey Voynov & Artem Babenko. *Unsupervised Discovery of Interpretable Directions in the GAN Latent Space.* In ICML, 2020.

[Wagner 19] Jörg Wagner, Jan Mathias Köhler, Tobias Gindele, Leon Hetzel, Jakob Thaddäus Wiedemer & Sven Behnke. *Interpretable and Fine-Grained Visual Explanations for Convolutional Neural Networks.* In CVPR, 2019.

[Wang 17a] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang & Xiaoou Tang. *Residual Attention Network for Image Classification.* In CVPR, 2017.

[Wang 17b] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri & Ronald M. Summers. *ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases.* In CVPR, 2017.

[Wang 19] Yaxing Wang, Abel Gonzalez-Garcia, Joost van de Weijer & Luis Herranz. *SDIT: Scalable and Diverse Cross-domain Image Translation.* In ACM MM, 2019.

[Wang 20] P. Wang & N. Vasconcelos. *SCOUT: Self-Aware Discriminant Counterfactual Explanations.* In CVPR, 2020.

[Wasserstein 69] Leonid N Wasserstein. *Markov processes over denumerable products of spaces describing large systems of automata.* Problems of Information Transmission, vol. 5, no. 3, 1969.

[Wei 18] Yi Wei, Ming-Ching Chang, Yiming Ying, Ser-Nam Lim & Siwei Lyu. *Explain Black-box Image Classifications Using Superpixel-based Interpretation.* In ICPR, 2018.

[Wolleb 20] Julia Wolleb, Robin Sandkühler & P. Cattin. *DeScarGAN: Disease-Specific Anomaly Detection with Weak Supervision.* In MICCAI, 2020.

[Woods 19] Walt Woods, Jack Chen & Christof Teuscher. *Adversarial explanations for understanding image classification decisions and improved neural network robustness.* In Nature Machine Intelligence, volume 1, 2019.

[Wu 20] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R. Lyu & Yu-Wing Tai. *Towards Global Explanations of Convolutional Neural Networks With Concept Attribution.* In CVPR, 2020.

[Wu 21] Zongze Wu, D. Lischinski & Eli Shechtman. *StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation.* In CVPR, 2021.

[Xia 19] Tian Xia, Agisilaos Chartsias & Sotirios A. Tsaftaris. *Adversarial Pseudo Healthy Synthesis Needs Pathology Factorization.* In MIDL, 2019.

[Xiao 18] Taihong Xiao, Jiapeng Hong & Jinwen Ma. *ELEGANT: Exchanging Latent Encodings with GAN for Transferring Multiple Face Attributes.* In ECCV, 2018.

[Xie 22] Ning Xie, Gabrielle Ras, Marcel van Gerven & Derek Doran. *Explainable Deep Learning: A Field Guide for the Uninitiated.* In Journal of Artificial Intelligence Research, 2022.

[Xu 20a] Shawn Xu, Subashini Venugopalan & Mukund Sundararajan. *Attribution in Scale and Space.* In CVPR, 2020.

[Xu 20b] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele & Zeynep Akata. *Attribute Prototype Network for Zero-Shot Learning.* In Advances in Neural Information Processing Systems, 2020.

[Yang 21a] F. Yang, Ninghao Liu, Mengnan Du & Xia Ben Hu. *Generative Counterfactuals for Neural Networks via Attribute-Informed Perturbation.* In ACM SIGKDD Explorations Newsletter, 2021.

[Yang 21b] Qing Yang, Xia Zhu, Yun Ye, Jong-Kae Fwu, Ganmei You & Yuan Zhu. *MFPP: Morphological Fragmental Perturbation Pyramid for Black-Box Model Explanations.* In ICPR, 2021.

[Yao 21] Xu Yao, Alasdair Newson, Yann Gousseau & Pierre Hellier. *A Latent Transformer for Disentangled Face Editing in Images and Videos.* In ICCV, 2021.

[Yeh 20] Chih-Kuan Yeh, Been Kim, Sercan Ö. Arik, Chun-Liang Li, Tomas Pfister & Pradeep Ravikumar. *On Completeness-aware Concept-Based Explanations in Deep Neural Networks.* In NeurIPS, 2020.

[Yosinski 14] Jason Yosinski, Jeff Clune, Yoshua Bengio & Hod Lipson. *How transferable are features in deep neural networks?* In NIPS, 2014.

[Yu 18a] Jiahui Yu, Zhe L. Lin, Jimei Yang, Xiaohui Shen, Xin Lu & Thomas S. Huang. *Generative Image Inpainting with Contextual Attention.* CVPR, 2018.

[Yu 18b] Xiaoming Yu, Xing Cai, Zhenqiang Ying, Thomas H. Li & Ge Li. *SingleGAN: Image-to-Image Translation by a Single-Generator Network using Multiple Generative Adversarial Learning.* In ACCV, 2018.

[Yu 18c] Xiaoming Yu, Zhenqiang Ying & Ge Li. *Multi-Mapping Image-to-Image Translation with Central Biasing Normalization.* ArXiv, 2018.

[Yu 19] Xiaoming Yu, Yuanqi Chen, Thomas H. Li, Shan Liu & Ge Li. *Multi-mapping Image-to-Image Translation via Learning Disentanglement.* In NeurIPS, 2019.

[Zech 18] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano & Eric Karl Oermann. *Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study.* PLOS Medicine, 2018.

[Zeiler 14] Matthew D. Zeiler & Rob Fergus. *Visualizing and Understanding Convolutional Networks.* In ECCV, 2014.

[Zhang 18a] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu & Song-Chun Zhu. *Interpreting CNN knowledge via an Explanatory Graph.* In AAAI, 2018.

[Zhang 18b] Quanshi Zhang, Ying Nian Wu & Song-Chun Zhu. *Interpretable Convolutional Neural Networks.* In CVPR, 2018.

[Zhang 18c] Quanshi Zhang & Song-Chun Zhu. *Visual interpretability for deep learning: a survey.* In Frontiers of Information Technology & Electronic Engineering, 2018.

[Zhang 19a] Quanshi Zhang, Yu Yang, Ying Nian Wu & Song-Chun Zhu. *Interpreting CNNs via Decision Trees.* In CVPR, 2019.

[Zhang 19b] Weijia Zhang. *Generating Adversarial Examples in One Shot With Image-to-Image Translation GAN.* In IEEE Access, volume 7, 2019.

[Zhang 21] Yang Zhang, Ashkan Khakzar, Yawei Li, Azade Farshad, Seong Tae Kim & Nassir Navab. *Fine-Grained Neural Network Explanation by Identifying Input Features with Predictive Information.* In NeurIPS, 2021.

[Zhang 22] Yunlong Zhang, Xin Lin, Yihong Zhuang, LiyanSun, Yue Huang, Xinghao Ding, Guisheng Wang, L. Yang & Yizhou Yu. *Harmonizing Pathological and Normal Pixels for Pseudo-healthy Synthesis.* In IEEE transactions on medical imaging, 2022.

[Zhou 14] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba & Aude Oliva. *Learning Deep Features for Scene Recognition using Places Database.* In NIPS, 2014.

[Zhou 15] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva & Antonio Torralba. *Object Detectors Emerge in Deep Scene CNNs.* In ICLR, 2015.

[Zhou 16a] B. Zhou, A. Khosla, Àgata Lapedriza, A. Oliva & A. Torralba. *Learning Deep Features for Discriminative Localization.* In CVPR, 2016.

[Zhou 16b] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva & Antonio Torralba. *Learning Deep Features for Discriminative Localization.* In CVPR, 2016.

[Zhu 17] Jun-Yan Zhu, Taesung Park, Phillip Isola & Alexei A Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.* In ICCV, 2017.

[Zhu 21] Xinqi Zhu, Chang Xu & Dacheng Tao. *Where and What? Examining Interpretable Disentangled Representations.* In CVPR, 2021.

[Zilke 16] Jan Ruben Zilke, Eneldo Loza Mencía & Frederik Janssen. *DeepRED - Rule Extraction from Deep Neural Networks.* In DS, 2016.

[Zintgraf 17] Luisa M. Zintgraf, Taco Cohen, Tameem Adel & Max Welling. *Visualizing Deep Neural Network Decisions: Prediction Difference Analysis.* In ICLR, 2017.

A

Appendix: Additional optimization frameworks

We illustrate additional optimization frameworks that follow different embodiments of Chapter 5:

- Figures A.1, A.2 and A.3 show frameworks using a single path optimization for SSyGen, CyLatentCE and CyImageCE respectively.
- Figure A.4 shows CySCGen optimization.

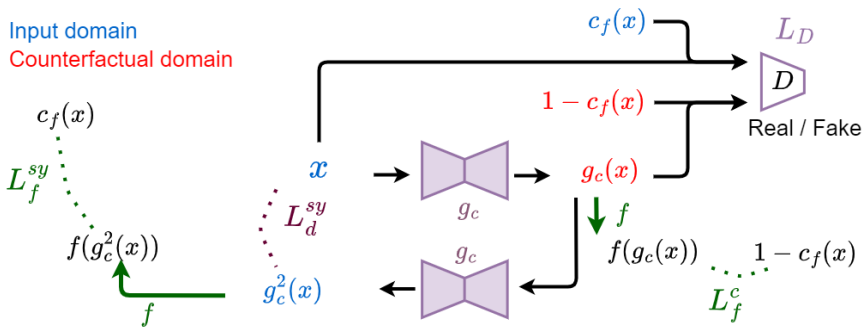


Figure A.1: Overview of SSyGen optimization framework with a single path. Training step of g_c for any input image $x \in \chi$. ***Counterfactual path***: an input image x is given to the generator g_c (black arrow) which produces a counterfactual image $g_c(x)$. This generated image is enforced (L_f^c) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the generated image $g_c(x)$ to fool the discriminator D that is trained to identify real ($x \in \chi$) from generated images. D is conditioned by the classification target of its input: either $c_f(x)$ for the real image x or $1-c_f(x)$ the generated counterfactual $g_c(x)$. ***Symmetrical path***: $g_c(x)$ is then mapped back to its initial domain through g_c (black arrow starting above $g_c(x)$). The resulting symmetrical image ${g_c}^2(x)$ is enforced to be pixel-wise close to x (L_d^{sy}) and classified the same way (L_f^{sy}).

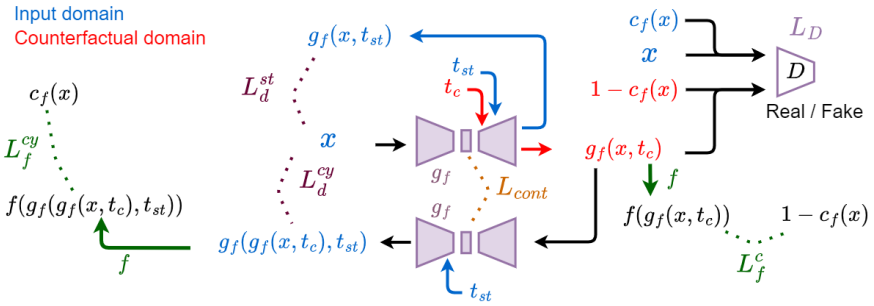


Figure A.2: Overview of CyLatentCE optimization framework with a single path. Training step of g_c for any input image $x \in \chi$. ***Counterfactual path***: an input image x is given to the generator g_f (black arrow) and conditioned through its latent space by $t_c = 1 - c_f(x)$ in the binary case (or directly $1 - f(x)$). It produces a counterfactual image $g_f(x, 1 - c_f(x))$. This generated image is enforced (L_f^c) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D that is trained to identify real (x) from generated images. D is conditioned by the classification target of its input: either $c_f(x)$ for the real image x or $1 - c_f(x)$ the generated counterfactual $g_c(x)$. ***Stable path***: the input image x is given to the generator g_f and conditioned by the original prediction $t_{st} = c_f(x)$ (or $f(x)$). It produces a stable image $g_f(x, c_f(x))$ which is enforced to be pixel-wise to x by the term L_d^{st}. ***Cyclic path***: $g_f(x, t_c)$ is also mapped back to its original domain through cycle consistency (black arrow bellow $g_f(x, t_c)$). Pixel-wise proximity and classification consistency to x are encouraged by the constraints (L_d^{cy}) and (L_f^{cy}).

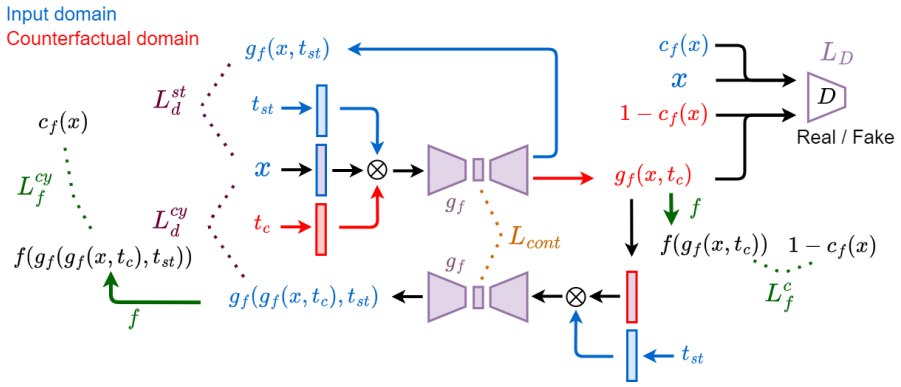


Figure A.3: Overview of CyImageCE optimization framework with a single path. Training step of g_c for any input image $x \in \chi$. ***Counterfactual path***: an input image x is given to the generator g_f (black arrow) and conditioned at image level by $t_c = 1 - c_f(x)$ in the binary case (or directly $1 - f(x)$). It produces a counterfactual image $g_f(x, 1 - c_f(x))$. This generated image is enforced (L_f^c) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D that is trained to identify real (x) from generated images. D is conditioned by the classification target of its input: either $c_f(x)$ for the real image x or $1 - c_f(x)$ the generated counterfactual $g_c(x)$. ***Stable path***: the input image x is given to the generator g_f and conditioned by the original prediction $t_{st} = c_f(x)$ (or $f(x)$). It produces a stable image $g_f(x, c_f(x))$ which is enforced to be pixel-wise to x by the term L_d^{st}. ***Cyclic path***: $g_f(x, t_c)$ is also mapped back to its original domain through cycle consistency (black arrow bellow $g_f(x, t_c)$). Pixel-wise proximity and classification consistency to x are encouraged by the constraints (L_d^{cy}) and (L_f^{cy}).

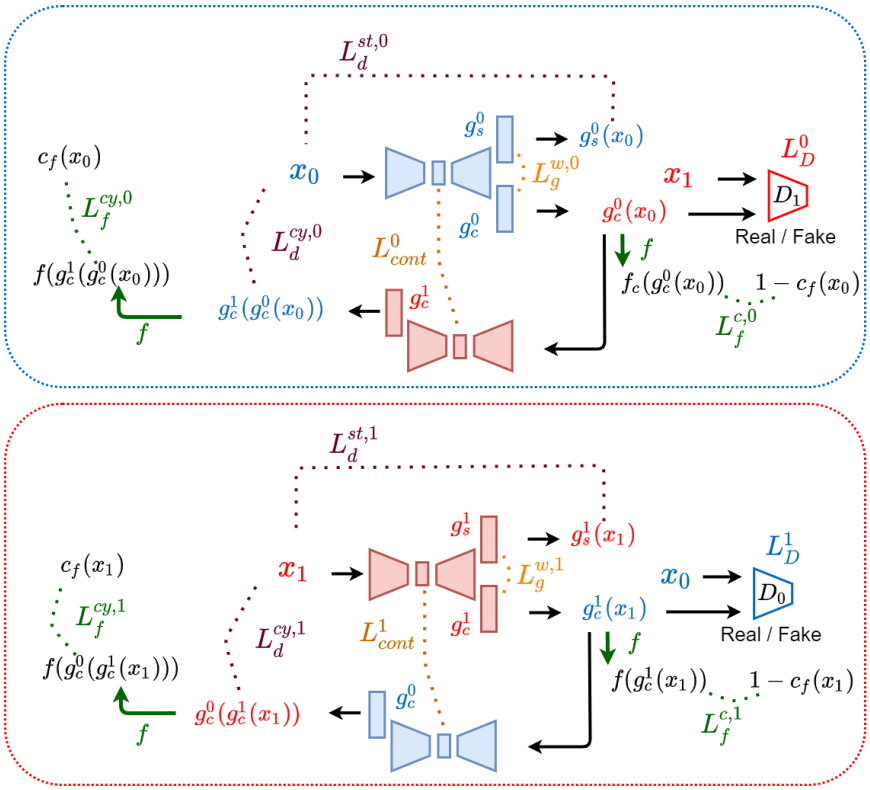


Figure A.4: Overview of CySCGen optimization framework. Top: training step of $g_s{}^0$, $g_c{}^0$ and $g_c{}^1$ for an original image $x_0 \in \chi_0$. Bottom: training step of $g_s{}^1$, $g_c{}^1$ and $g_c{}^0$ for an image $x_1 \in \chi_1$. The terms L_i^0 (resp. L_i^1) are the loss terms acting on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). **Top**: ***Counterfactual path***: an input image x_0 is given to the generator $g_c{}^0$ which produces a counterfactual image $g_c{}^0(x_0)$. This generated image is enforced $(L_f^{c\,0})$ to be classified in the class opposite to x_0. We also enforce the generated image $g_c{}^0(x_0)$ to fool the discriminator D_1 that is trained to identify real (x_1) from generated images in the distribution of images predicted in class 1 (χ_1). ***Stable path***: given x_0, $g_s{}^0$ generated a stable generation which is enforced to be pixel-wise close to x_0 by the term $L_d^{st,0}$. ***Cyclic path***: the counterfactual image $g_c{}^0(x)$ is mapped back to χ_0 through cycle constraint by applying $g_c{}^1$. Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints $(L_d^{cy,0})$ and $(L_f^{cy,0})$. Similar procedures for the other transposition (**Bottom**).

B

Appendix: Visual explanation implementation

B.1 Counterfactual Integrated Gradient

As introduced in Section 6.4, we can design a counterfactual integrated gradient version, where all derivatives are taken into account ($\mathcal{IG}_c^{v2}$), as well as a regularized version ($\mathcal{IG}_{c,k_\sigma}^{v2}$):

$$\begin{aligned} \mathcal{IG}_c^{v2}(x) &= |\mathbf{x_c} - \mathbf{x}| \int_0^1 \left| \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)) \right| du \\ \mathcal{IG}_{c,k_\sigma}^{v2}(x) &= \int_0^1 \left[|\mathbf{x_c} - \mathbf{x}| \left| \frac{\partial f}{\partial \mathbf{x}}(\boldsymbol{\gamma}(u)) \right| \right] * k_\sigma du \end{aligned} \tag{B.1}$$

B.2 Generator architecture

First, Table B.1 describes the generator architecture (e.g., used in CyCE, SyCE, or SAGen) adapted to the classification of digits in MNIST where images are found at resolution 28x28.

Table B.1: UNet-like generator architecture. Illustration for MNIST input images at scale 1x28x28

(a) Encoder path

Layer	Resample	Norm	Output shape
Input			1x28x28
Conv3x3	-	BN / -	16x28x28
ResBlock	MaxPool	BN / -	32x14x14
ResBlock	MaxPool	BN / -	64x7x7

(b) Decoder path

Layer	Resample	Norm	Output Shape
Enc. Fmaps			64x7x7
ResBlockSkipConcat	Upsampling	BN / -	32x14x14
ResBlockSkipConcat	Upsampling	BN / -	16x28x28

Then, we provide further details on the different blocks stated in the model architectures from Section 7.2.

Description of the different blocks:

1. Conv: a convolutional block composed of
   - A convolutional layer with kernel 3x3 or 4x4
   - A activation: ReLU or LeakyReLU
   - A normalization: BatchNorm (BN), Instance Norm (IN), a conditional normalization (cBN or AdaIN), or nothing.

2. ResBlock: residual blocks are composed of two successive convolutional blocks. The resulting layer is added to the input of the block (i.e., the residual). If the residual and the output of the second convolution block have dissimilar channels, the residual passes through a convolutional layer (kernel 1x1) to reach the target channel number. If there is a resampling operation, it can either follow the residual block or be inserted inside (after the first convolutional block). In the second case, the residual is also resampled.
3. DownBlock: a downsampling block is either composed of:
   - A convolution block with stride 2, following [Yu 18c, Bass 20]
   - A residual block with an average pooling operation inserted inside the block.
4. ResBlockSkipConcat: a residual block with skip connection concatenation following or inserted within the block.
5. Upsampling: bilinear or nearest neighbors upsampling operation.
6. UpBlock: an upsampling block is either composed of:
   - A transposed convolution block with stride 2, following [Yu 18c, Bass 20]. The transposed convolutional block replaces the convolutional operation with transposed convolution in Conv.
   - A residual block with an upsampling operation inserted inside the block.
7. AdaIN: Adaptive Instance Normalization used in [Yu 18c, Karras 19]. It normalizes (IN) the input layer and rescales the statistics given style vectors.
8. ModResBlock: a residual block with modulated/demodulated convolution. Following the work of [Karras 20], we replace conditioning with a conditional normalization layer (e.g., AdaIN) by modulated/demodulated convolution. No normalization is thus used.
9. ModUpBlock: idem as 8. with UpBlock.
10. Enc. Fmaps stands for encoded feature maps.
11. Y is the conditioning label or score.
12. ConditionBlock: this block conditions the model with the class label or score. We use either:
    - Tiling operation at image-level [Choi 18] or in the latent space [Lee 18]. The resulting feature maps are concatenated with the input or the latent encoded feature maps.
    - A dense layer, followed by a reshaping operation and a residual block as in [Bass 20]
    - one or several dense layer(s) to embed the class condition at a specific shape to compute the conditional normalization or the modulated/demodulated convolution.

In Table B.2, we give a range of values (tested) for each parameter of the different optimization function that produce satisfying results. We noticed that multiplying all parameter values of the generator model by 10 to 100 (while keeping the ratio between parameters fixed) generated similar results. In the chronology of the thesis, we first started with values shown in Tables B.3a, B.3b and B.3c; these parameters values are used in our work [Charachon 22]. We then validate that our methods produce similar results when setting parameters similar to other domain translation techniques (more similar to Table B.2)

Table B.2: Training parameters range. A range of values is given for the different visual explanation approaches. Similar values are used for the different problems and classification models.

Method	λ_d^{st}	λ_d^{sy}	λ_d^{cy}	λ_d^c	λ_f^c	λ_f^{st}	λ_f^{sy}	λ_f^{cy}	λ_{reg}	λ_g^w	λ_D	λ_D^d	λ_{gp}^d
AGen	-	-	-	50-200	0.25-1.0	-	-	-	1-20	-	-	-	-
SAGen	50-100	-	-	50-200	0.25-1.0	0-1	-	-	1-20	10-20	-	-	-
SSyGen	-	100-200	-	-	0.25-1.0	-	0.01-0.1	-	-	-	0.25-1.0	1.0	5-10
CyCE	-	-	100-200	-	0.25-1.0	-	-	0.01-0.1	-	-	0.25-1.0	1.0	5-10
SyCE	-	100-200	5-50	-	0.25-1.0	-	0.01-0.1	0-0.1	-	-	0.25-1.0	1.0	5-10
CyLatentCE	100-200	-	50-100	-	0.25-1.0	0	-	0.001-0.05	-	-	0.25-1.0	1.0	5-10
CyImageCE	100-200	-	50-100	-	0.25-1.0	0	-	0.001-0.05	-	-	0.25-1.0	1.0	5-10
SySCGen	100-200	50-100	-	-	0.25-1.0	0	0.001-0.05	-	-	10-20	0.25-1.0	1.0	5-10
cySCGen	100-200	-	50-100	-	0.25-1.0	0	-	0.001-0.01	-	10-20	0.25-1.0	1.0	5-10

Table B.3: Training parameters used in [Charachon 22]

(a) CyCE

Problem	Classifier	λ_d^{cy}	λ_f^c	λ_f^{cy}	λ_D	λ_D^d	λ_{gp}^d
Digits	LeNet	10.0	0.2	0.005	0.25	1.0	1.0
Pneumonia detect.	ResNet50	10.0	0.05	0.01	0.025	1.0	1.0
	DenseNet121	10.0	0.05	0.01	0.05	1.0	1.0
Brain tumor loc	ResNet50	20.0	0.1	0.01	0.05	1.0	1.0
	DenseNet121	20.0	0.2	0.01	0.05	1.0	1.0

(b) SyCE

Problem	Classifier	λ_d^s	λ_d^{cy}	λ_f^c	λ_f^s	λ_f^{cy}	λ_D	λ_D^d	λ_{gp}^d
Digits	LeNet	10.0	2.0	0.2	0.01	0.005	0.25	1.0	1.0
Pneumonia detect.	ResNet50	10.0	2.0	0.05	0.01	0.001	0.05	1.0	1.0
	DenseNet121	10.0	2.0	0.05	0.01	0.01	0.05	1.0	1.0
Brain tumor loc	ResNet50	20.0	1.0	0.2	0.05	0.01	0.025	1.0	1.0
	DenseNet121	20.0	1.0	0.2	0.05	0.001	0.025	1.0	1.0
Mustache/No Must.	ResNet50	10.0	1.0	0.025	0.005	0.0	0.025	1.0	1.0
Young/Old	ResNet50	10.0	1.0	0.025	0.005	0.0	0.025	1.0	1.0

(c) CyLatentCE

Problem	Classifier	λ_d^s	λ_d^{cy}	λ_f^c	λ_f^{cy}	λ_D	λ_D^d	λ_{gp}^d
Digits (**Mul.Cls.**)	LeNet	10.0	5.0	0.1	0.001	0.1	1.0	1.0
Pneumonia detect.(**Bin.Cls.**)	ResNet50	20.0	10.0	0.05	0.0025	0.05	1.0	5.0
Brain tumor loc.(**Bin.Cls.**)	ResNet50	20.0	10.0	0.05	0.0025	0.05	1.0	5.0

C

Appendix: Localization results

C.1 Counterfactual techniques

C.1.1 Comparison between Counterfactual techniques

We provide additional figures that compare attribution maps of the different counterfactual approaches for:

- Pneumonia detection on X-rays: in Figures C.1 and C.2
- Brain tumor detection on MRI slices: in Figures C.3 and C.4

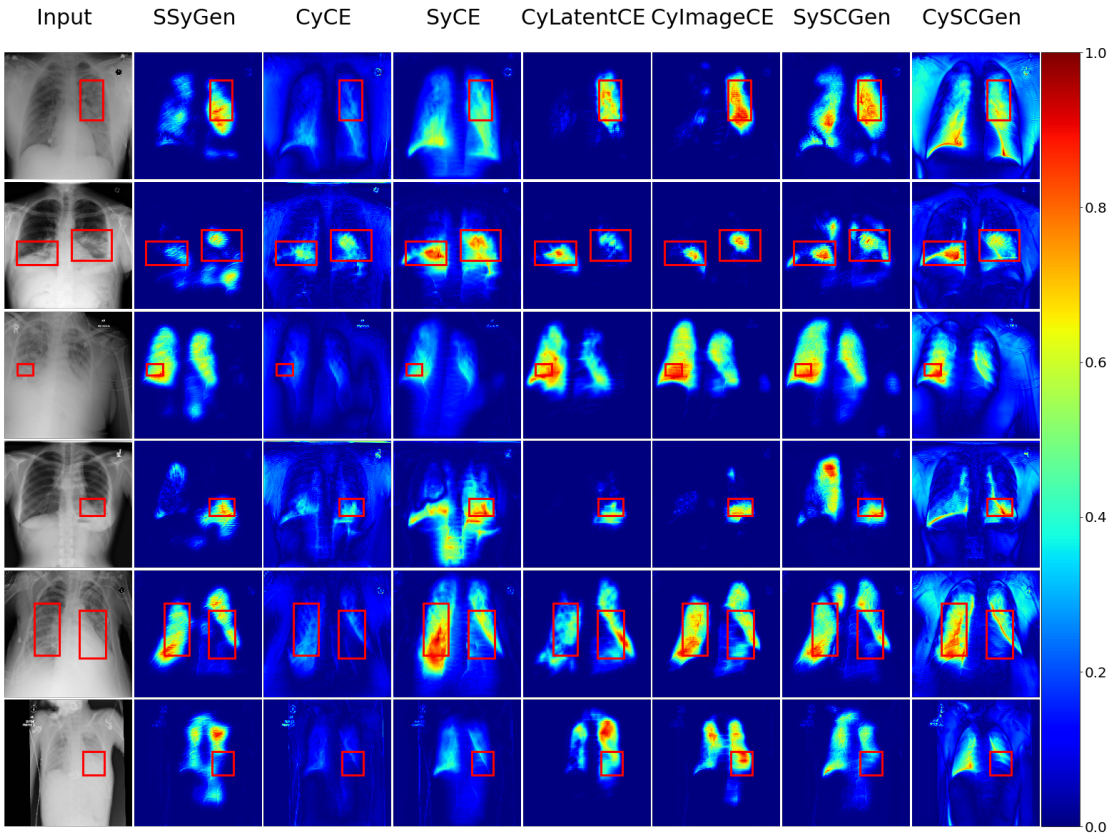


Figure C.1: Pneumonia detection (1) - Comparison between counterfactual attribution techniques and against ground truth annotations. Ground truth annotations are displayed with red contours. Dual path optimization is used for all the counterfactual methods (heatmaps shown in the different columns).

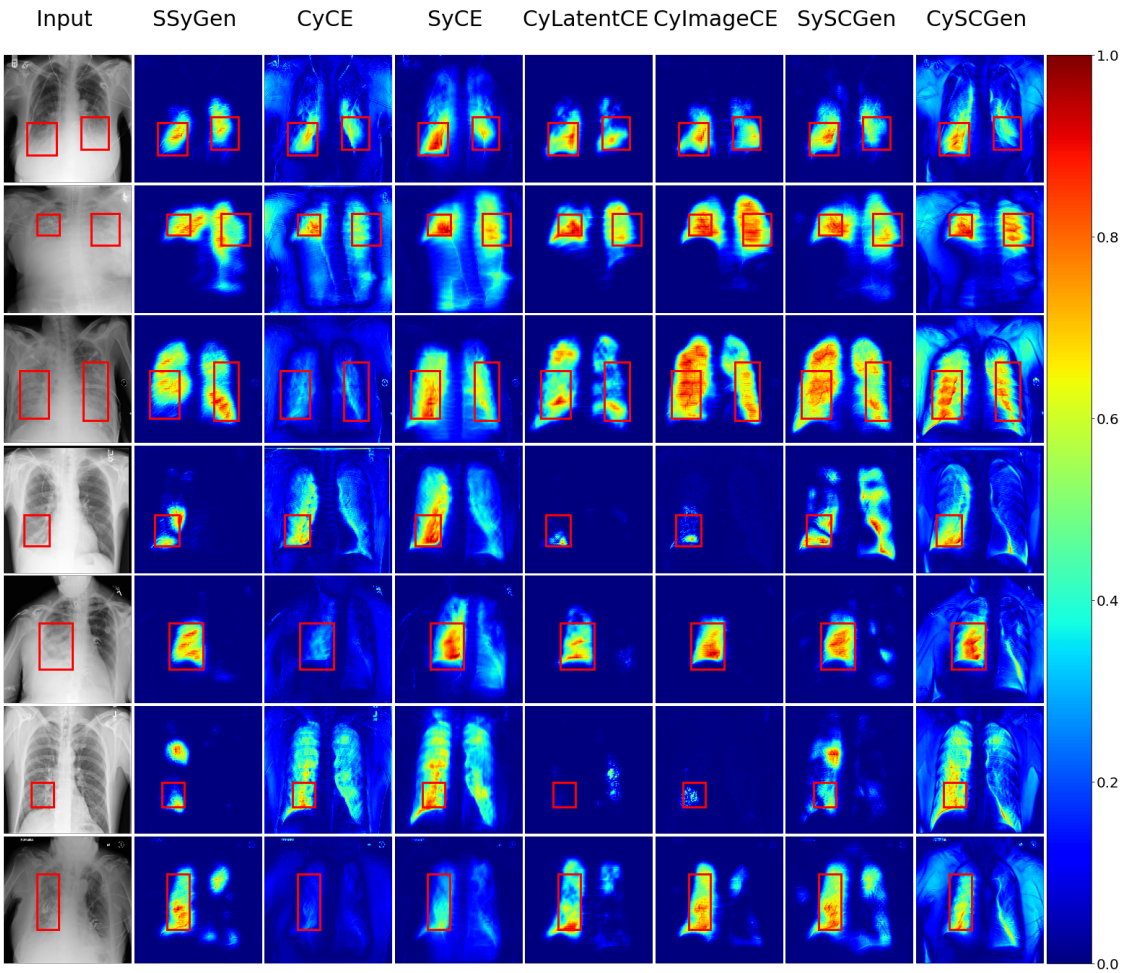


Figure C.2: Pneumonia detection (2) - Comparison between counterfactual attribution techniques and against ground truth annotations. Ground truth annotations are displayed with red contours. Dual path optimization is used for all the counterfactual methods (heatmaps shown in the different columns).

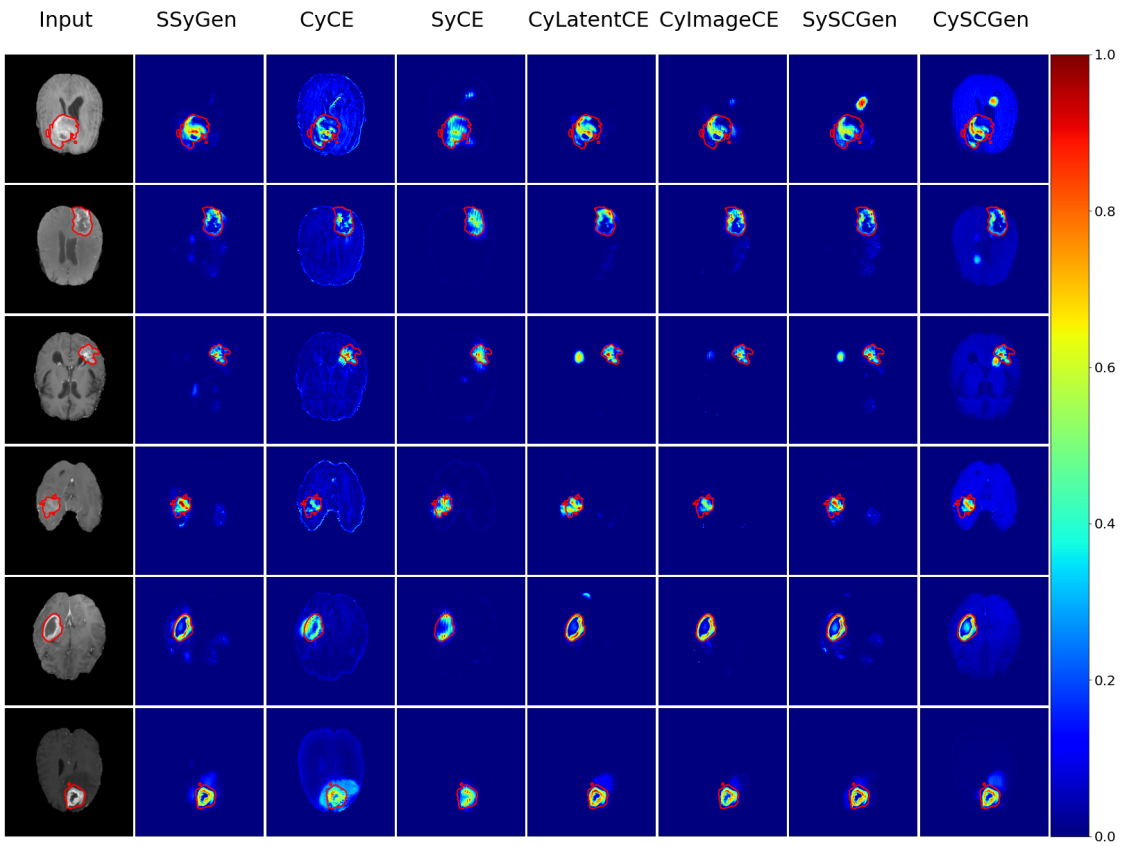


Figure C.3: Brain tumor detection (1) - Comparison between counterfactual attribution techniques and against ground truth annotations. Ground truth annotations are displayed with red contours. Dual path optimization is used for all the counterfactual methods (heatmaps shown in the different columns).

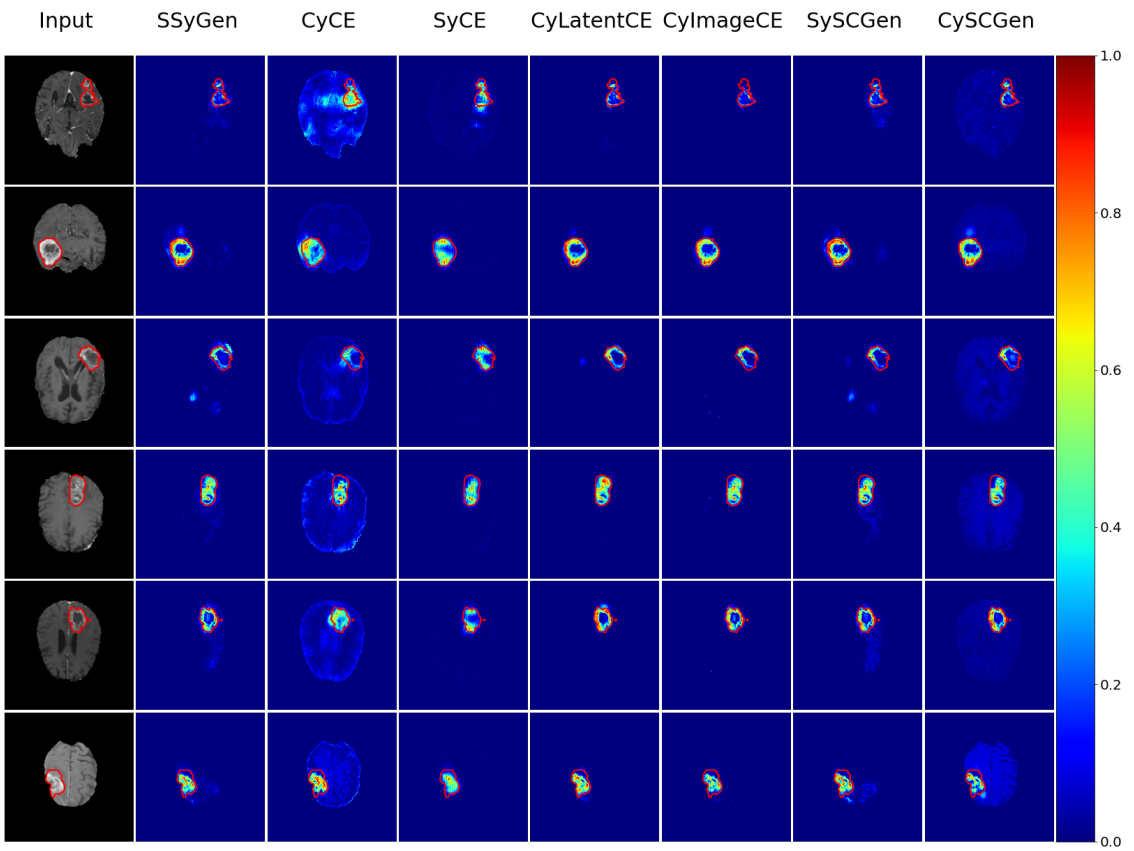


Figure C.4: Brain tumor detection (2) - Comparison between counterfactual attribution techniques and against ground truth annotations. Ground truth annotations are displayed with red contours. Dual path optimization is used for all the counterfactual methods (heatmaps shown in the different columns).

C.1.2 Comparison of generator architectures

In this section, we show the impact on localization performance of the generator architecture used for CyLatentCE:

- Table C.1 displays the localization metrics on both pneumonia and brain tumor detection. Except for the DRIT-like architecture (poorer results), the localization performance remains comparable.
- Figures C.5 and C.6 provides attribution maps for the different architectures on pneumonia and brain tumor detection respectively. These qualitative observations support the localization metrics obtained.

Table C.1: Localization results - Comparison between architectures for CyLatentCE. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. We highlight the best score between the explanation computed with and without stable generation for each metric in blue. The best scores are displayed in red the best scores.

				Pneunmonia detection				Brain tumor detection		
Metric				IoU ↑			NCC ↑	IoU ↑		NCC ↑
Architecture	Skip	Noise		90	95	98		98	99	
ICAM-like	✓	✓	w/o St.	0.308	0.247	0.140	0.500	0.397	0.350	0.612
			w/ St.	**0.320**	**0.256**	**0.146**	**0.516**	0.380	0.330	0.597
ICAM-like	✗	✓	w/o St	0.273	0.220	0.128	0.431	0.400	0.351	0.610
			w/ St.	0.290	0.232	0.134	0.454	0.386	0.335	0.602
Res-ICAM-like	✓	✓	w/o St	0.271	0.225	0.133	0.436	**0.426**	**0.365**	**0.631**
			w/ St.	0.294	0.241	0.143	0.471	0.408	0.348	0.625
DRIT-like	✗	✓	w/o St	0.151	0.118	0.073	0.268	0.296	0.275	0.500
			w/ St.	0.158	0.125	0.077	0.284	0.299	0.277	0.506
StyleGAN2-like	✗	✓	w/o St	0.267	0.211	0.119	0.417	0.382	0.344	0.589
			w/ St.	0.292	0.228	0.129	0.461	0.391	0.342	0.600
StyleGAN2-like	✗	✗	w/o St	0.291	0.233	0.133	0.469	0.392	0.355	0.589
			w/ St.	0.306	0.244	0.140	0.496	0.363	0.318	0.563

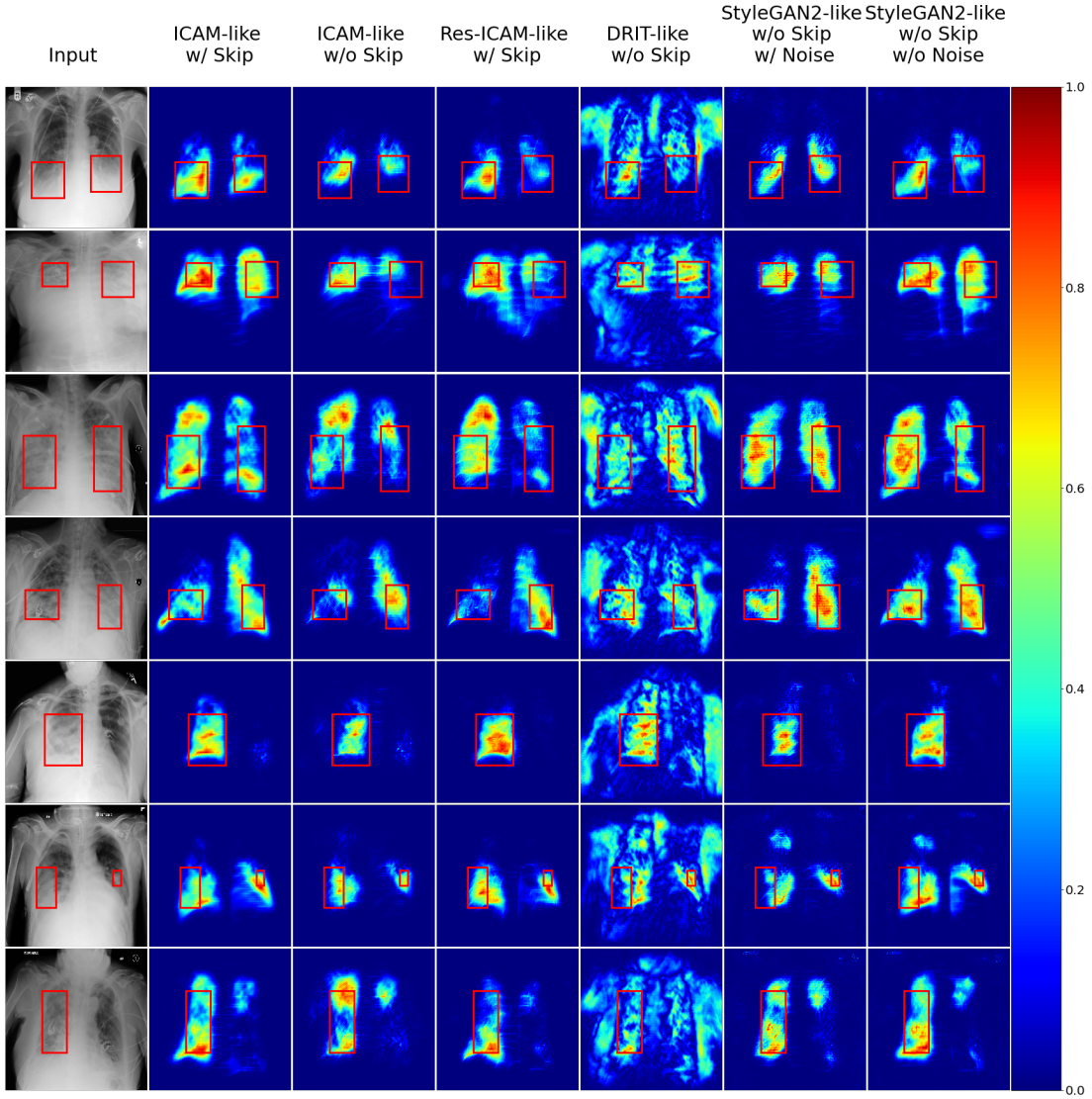


Figure C.5: Pneumonia detection - Comparison between different generator architectures for CyLatentCE. From left to right: the input image; then the attribution maps from CyLatentCE for different encoder-decoder architectures: Conditioning (ICAM, DRIT, or StyleGAN2-like); with or without skip connections; with residual blocks in downsampling and upsampling block (Res-) and with or without additive noise in the decoder path.

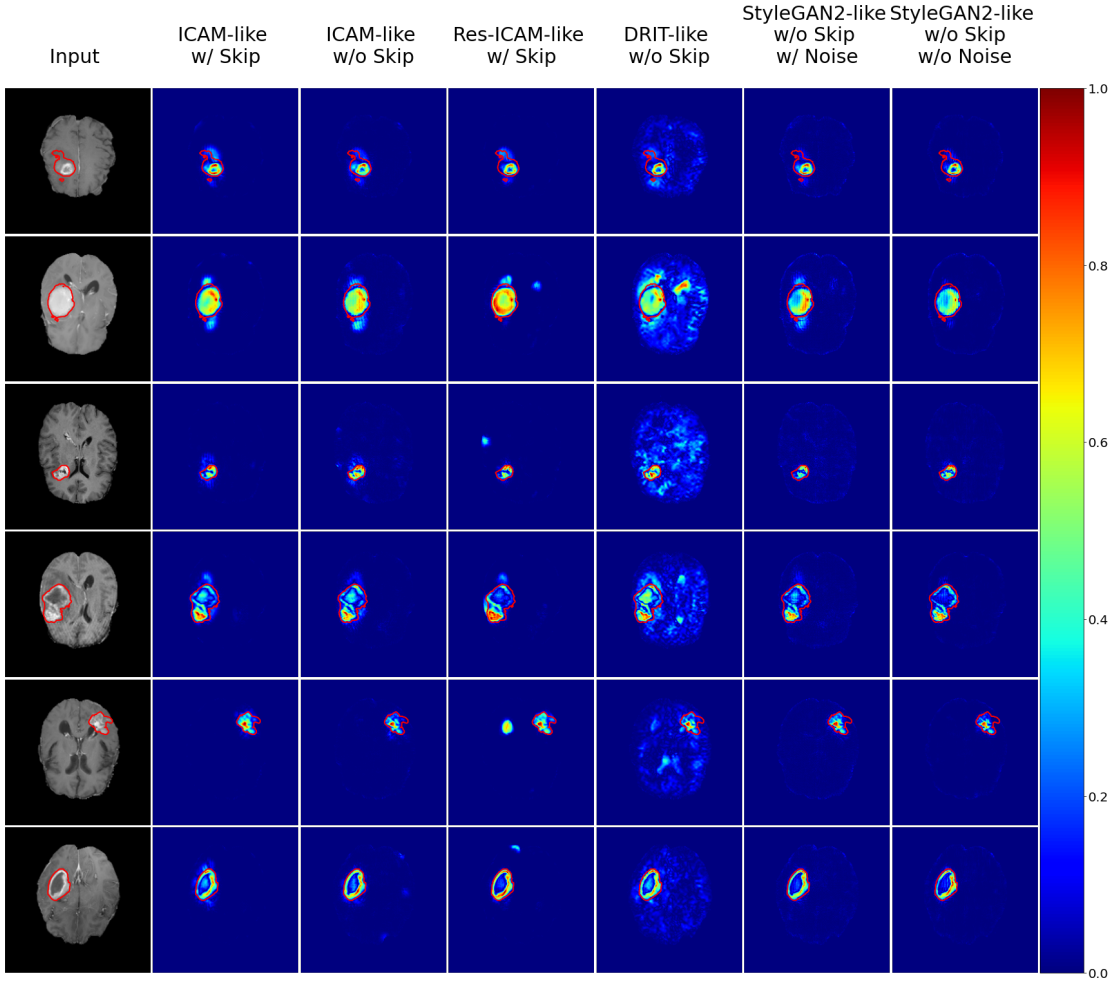


Figure C.6: Brain tumor detection - Comparison between different generator architectures for CyLatentCE. From left to right: the input image; then the attribution maps from CyLatentCE for different encoder-decoder architectures: Conditioning (ICAM, DRIT, or StyleGAN2-like); with or without skip connections; with residual blocks in downsampling and upsampling block (Res-) and with or without additive noise in the decoder path.

C.1.3 Ablation study

Here we provide additional qualitative and quantitative localization results on the ablation study introduced in Section 8.1.2.

- Table C.2 displays the localization metrics on both pneumonia and brain tumor detection for CyImageCE when removing the different terms of loss. Compared with CyLatentCE, removing L_d^{st} retains satisfying localization results.
- Figure C.7 illustrates the different cases and supports the quantitative findings.
- Figures C.8 and C.9 provides attribution maps computed with and without the stable generations. As pointed out in the manuscript, visualizations are similar. For pneumonia detection, the stable image slightly improves the importance affected to relevant regions while reducing reconstruction errors (see writings or body structures on SyCE). Yet, the impact is smaller than in the adversarial approach. In contrast, we observe artifacts when we use the stable image in the brain tumor problem.

Table C.2: Localization results - Ablation study for CyImageCE. We compare the impact on localization results of removing different optimization terms from CyLatentCE optimization. IoU (higher is better), FPR (lower is better) and NCC (higher is better) scores are given on (a) pneumonia detection and (b) brain tumor problems. For each problem, representative percentile values are displayed. For each metric, we highlight in bold the best score between explanation computed with and without stable generation, in blue the best score between optimizations (for each line), and in red the best scores.

(a) Pneumonia detection

Metric	Perc.	St.	L_d^{st}	$L_{d,f}^{cy}$	$L_f^{a,cy}$	Ours
IoU ↑	90	w/o	0.246	0.055	**0.072**	0.308
		w/	-	0.055	0.057	**0.310**
	95	w/	0.185	0.041	**0.043**	**0.257**
		w/	-	0.041	0.033	**0.257**
	98	w/o	0.096	0.024	**0.016**	0.154
		w/	-	0.024	0.013	**0.155**
FPR ↓	90	w/o	0.559	0.884	**0.840**	0.488
		w/	-	0.884	0.877	**0.486**
	95	w/o	0.505	0.875	**0.854**	0.388
		w/	-	0.875	0.897	**0.386**
	98	w/o	0.508	0.865	**0.894**	0.310
		w/		0.865	0.930	**0.308**
NCC ↑	-	w/o	0.361	0.027	**0.033**	0.503
		w/	-	0.027	0.016	**0.511**

(b) Brain tumor detection

Metric	Perc.	St.	L_d^{st}	$L_{d,f}^{cy}$	$L_f^{a,cy}$	Ours
IoU ↑	98	w/o	0.419	0.155	0.360	**0.437**
		w/	-	**0.156**	**0.371**	0.412
	99	w/o	0.368	0.119	0.303	**0.381**
		w/	-	0.119	**0.309**	0.359
FPR ↓	98	w/o	0.430	0.746	0.499	**0.426**
		w/	-	0.746	**0.489**	0.447
	99	w/o	0.289	0.711	0.396	**0.288**
		w/	-	0.711	**0.389**	0.313
NCC ↑	-	w/o	0.636	0.306	0.585	**0.628**
		w/	-	0.306	**0.591**	0.620

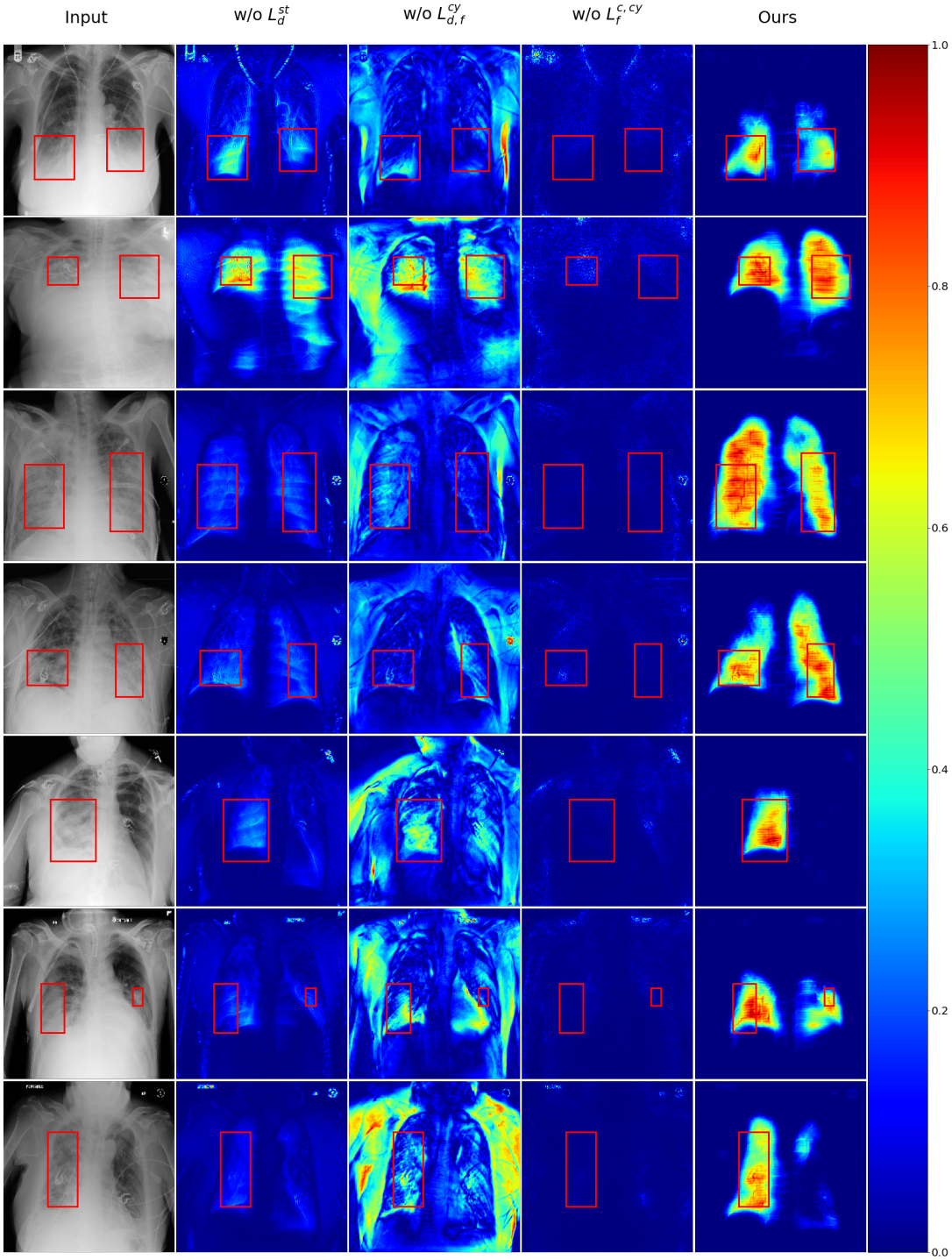


Figure C.7: Pneumonia detection - Ablation study for CyImageCE. From left to right: the input image; then the attribution maps from CyImageCE optimized without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}; and our CyImageCE optimization proposed in Section 5.4.

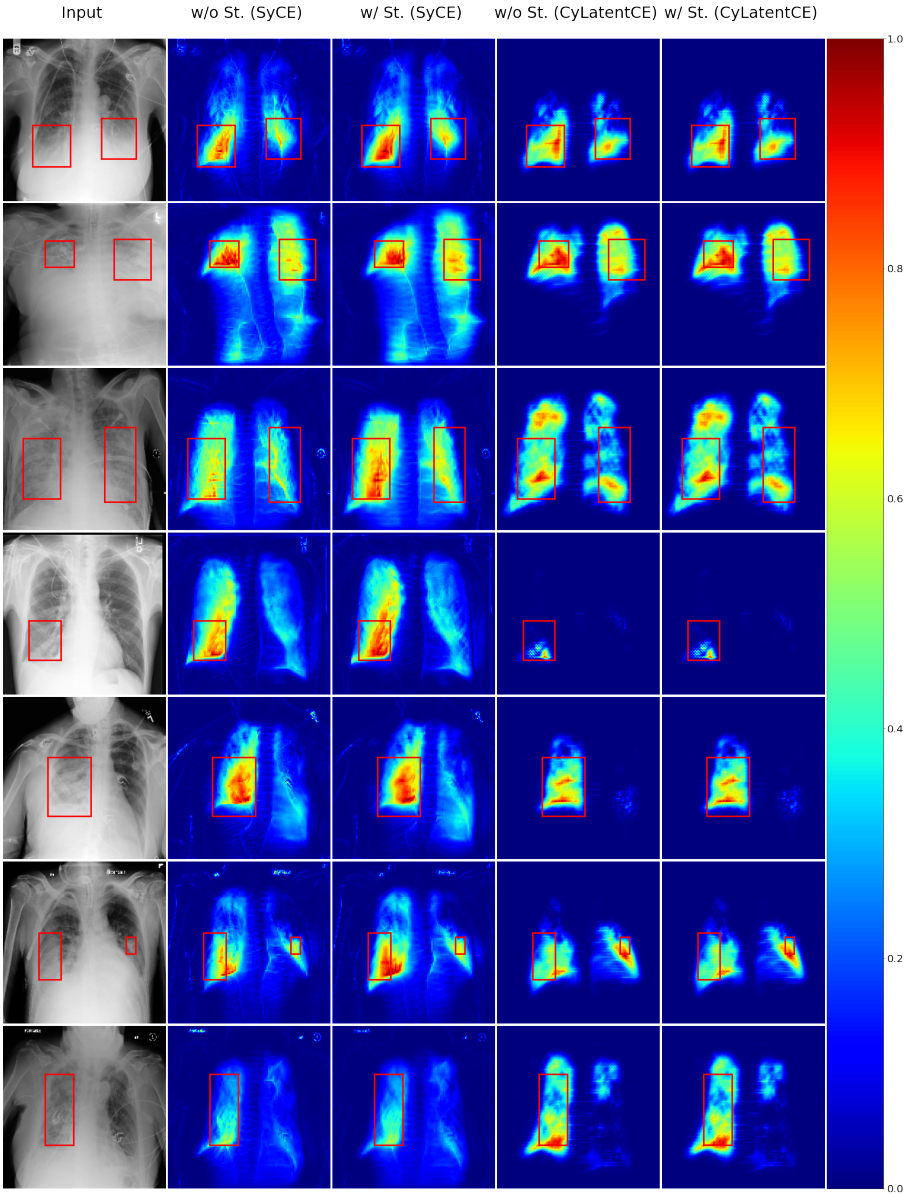


Figure C.8: Pneumonia detection - Explanation map with or without stable generation. From left to right: the input image; then the attribution maps from SyCE and CyLatentCE with or without the stable generation.

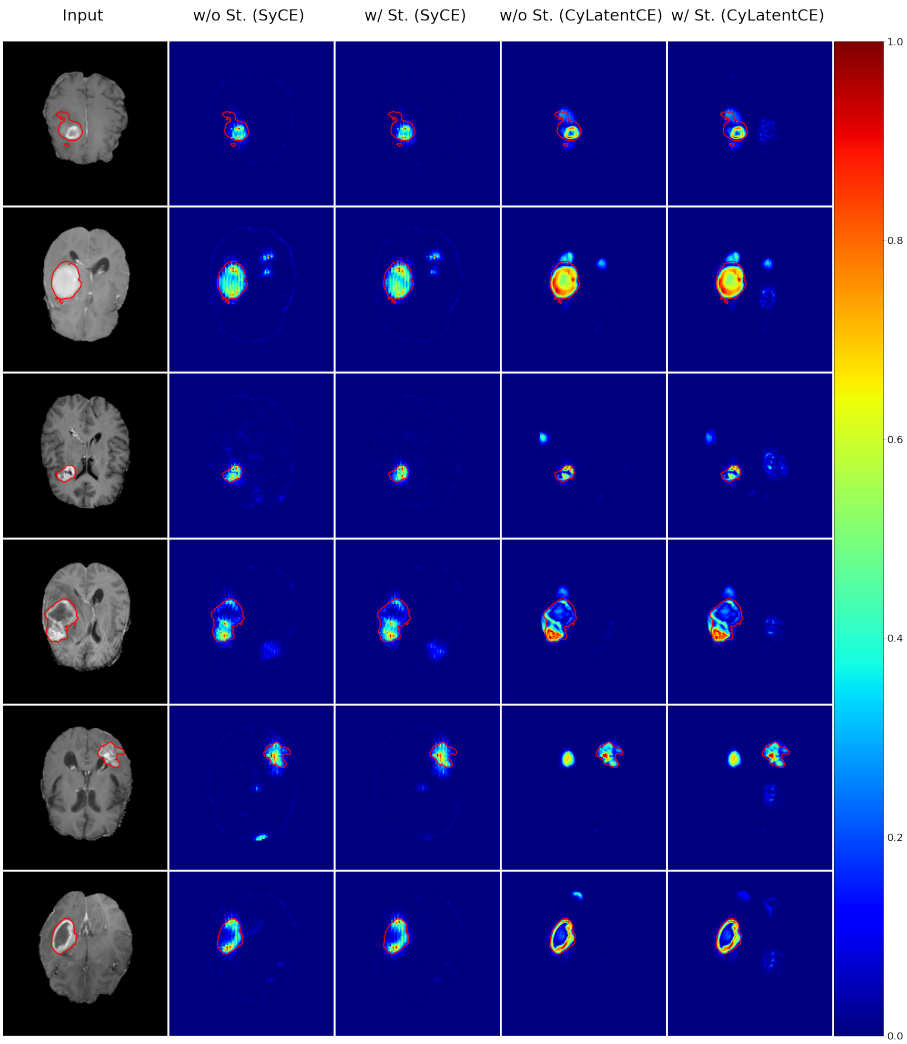


Figure C.9: Brain tumor detection - Explanation map with or without stable generation. From left to right: the input image; then the attribution maps from SyCE and CyLatentCE with or without the stable generation.

C.2 Integrated Counterfactual explanation

C.2.1 Pneumonia detection on X-Rays

Here, we provide visualizations of the diverse integrated counterfactual approach on the pneumonia detection problem. We compare the baseline attribution map $\mathcal{E}$ against the different integrated versions for all our counterfactual methods:

- SSyGen: Figures C.10 and C.11
- CyCE: Figure C.12
- SyCE: FiguresC.13 and C.14
- CyLatentCE: Figures C.15 and C.16
- CyImageCE: Figure C.17
- SySCGen: Figures C.18 and C.19
- CySCGen C.20 and C.21

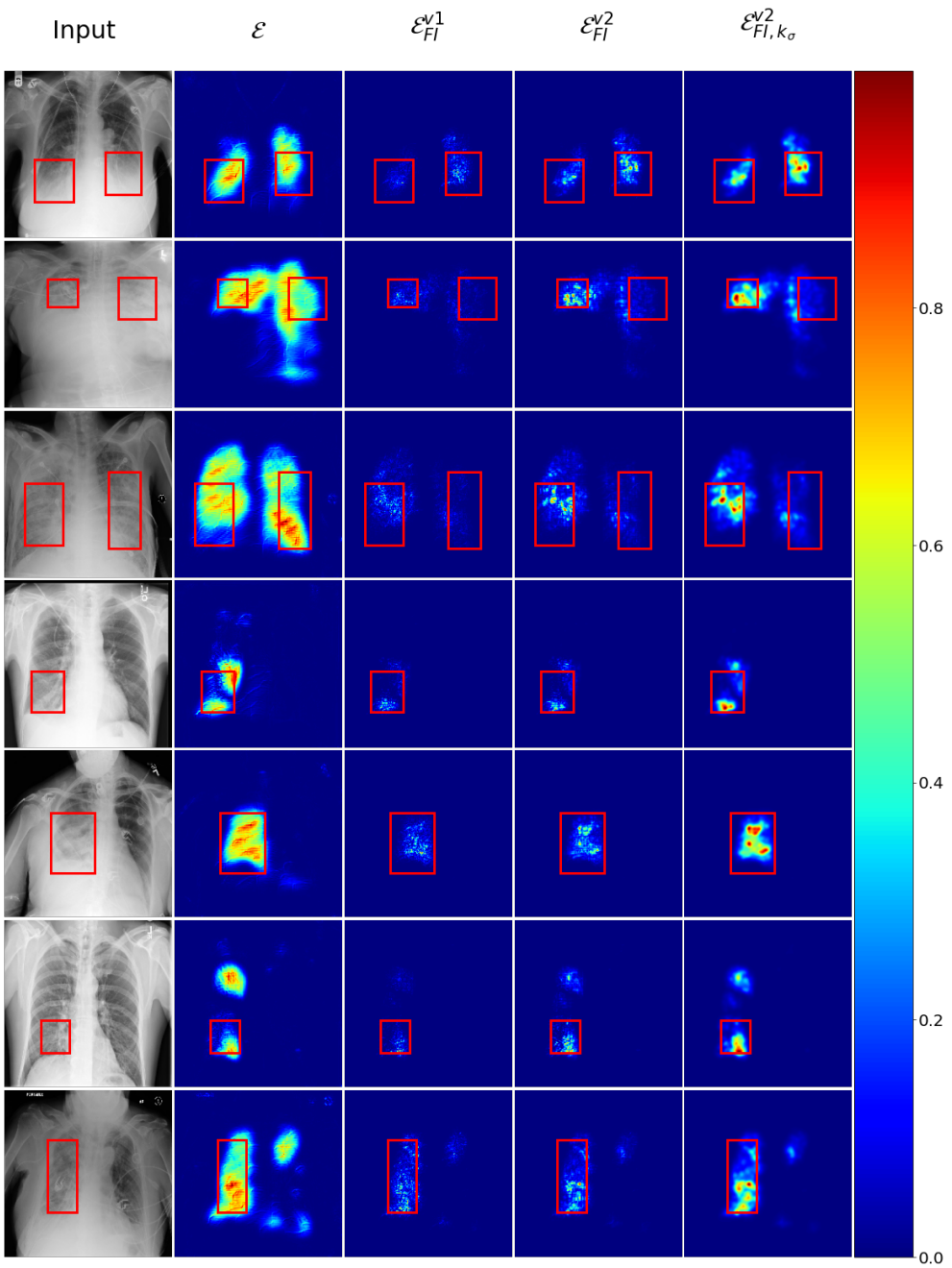


Figure C.10: Pneumonia detection (1) - Comparison between counterfactual baseline and path-based integration techniques for SSyGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SSyGen); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

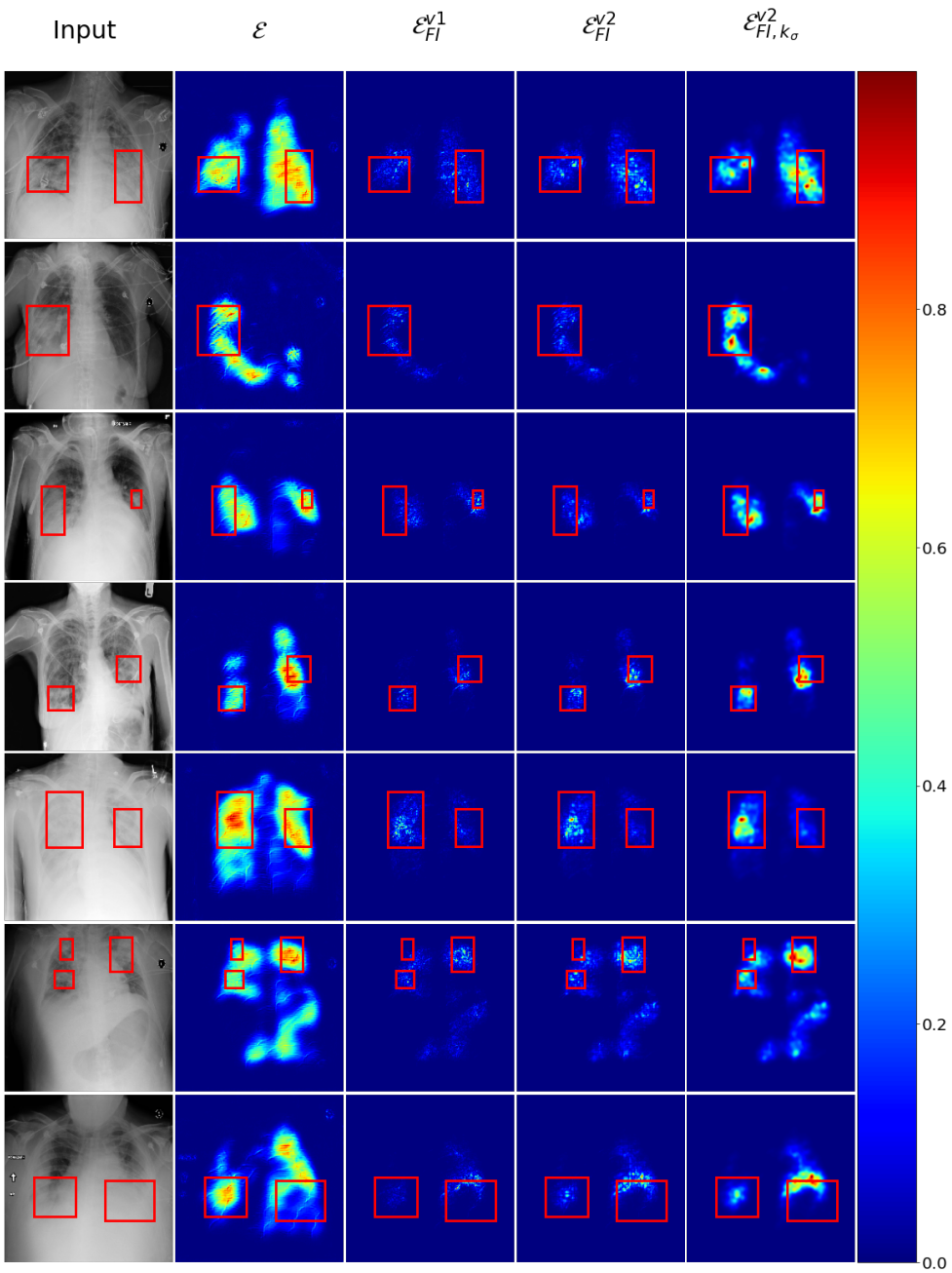


Figure C.11: Pneumonia detection (2) - Comparison between counterfactual baseline and path-based integration techniques for SSyGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SSyGen); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

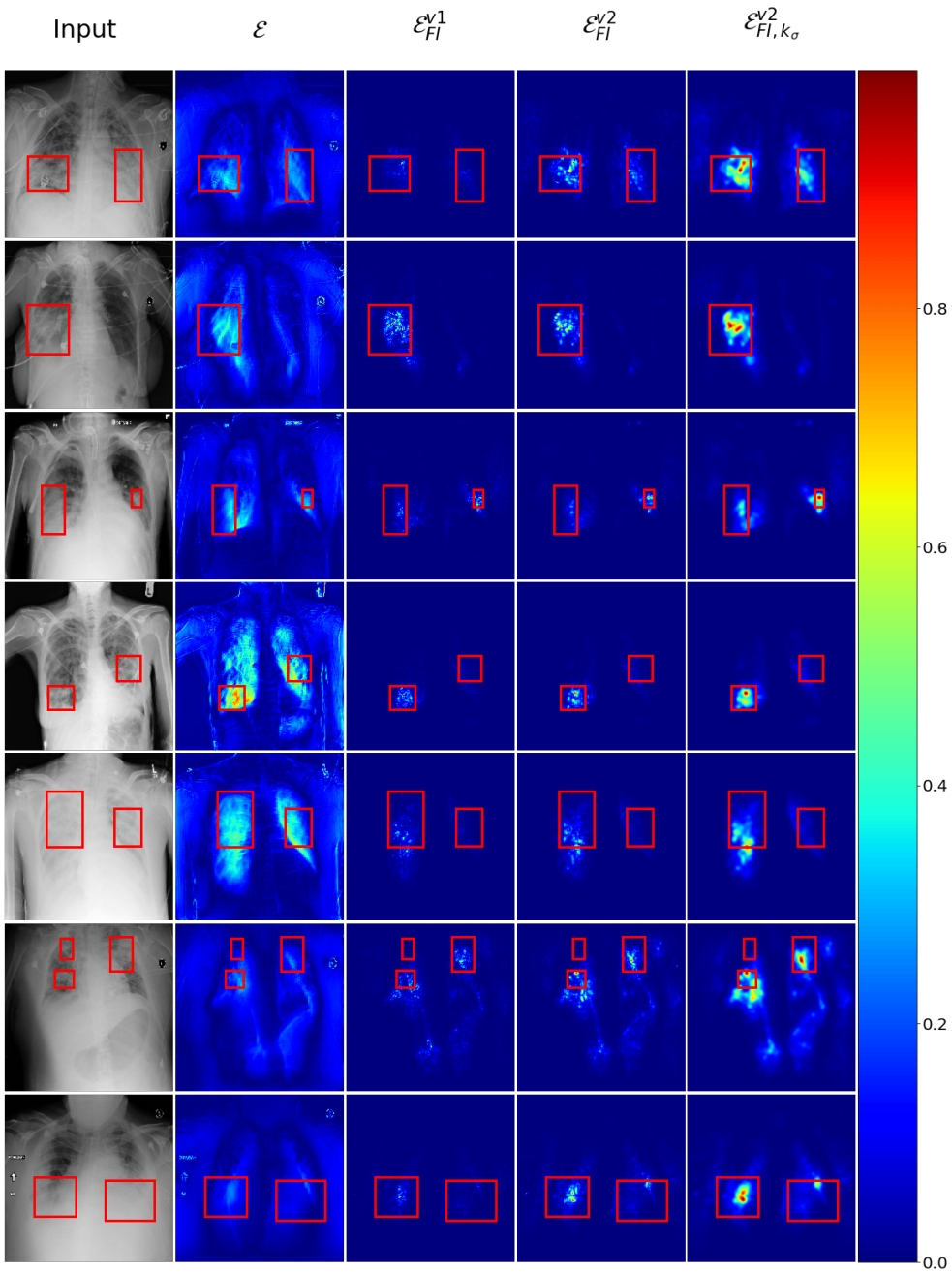


Figure C.12: Pneumonia detection - Comparison between counterfactual baseline and path-based integration techniques for CyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyCE); the integrated method v1 $\mathcal{E}^{v1}_{FI}$, the integrated method v2 $\mathcal{E}^{v2}_{FI}$; and the regularized integrated method $\mathcal{E}^{v2}_{FI,k_\sigma}$.

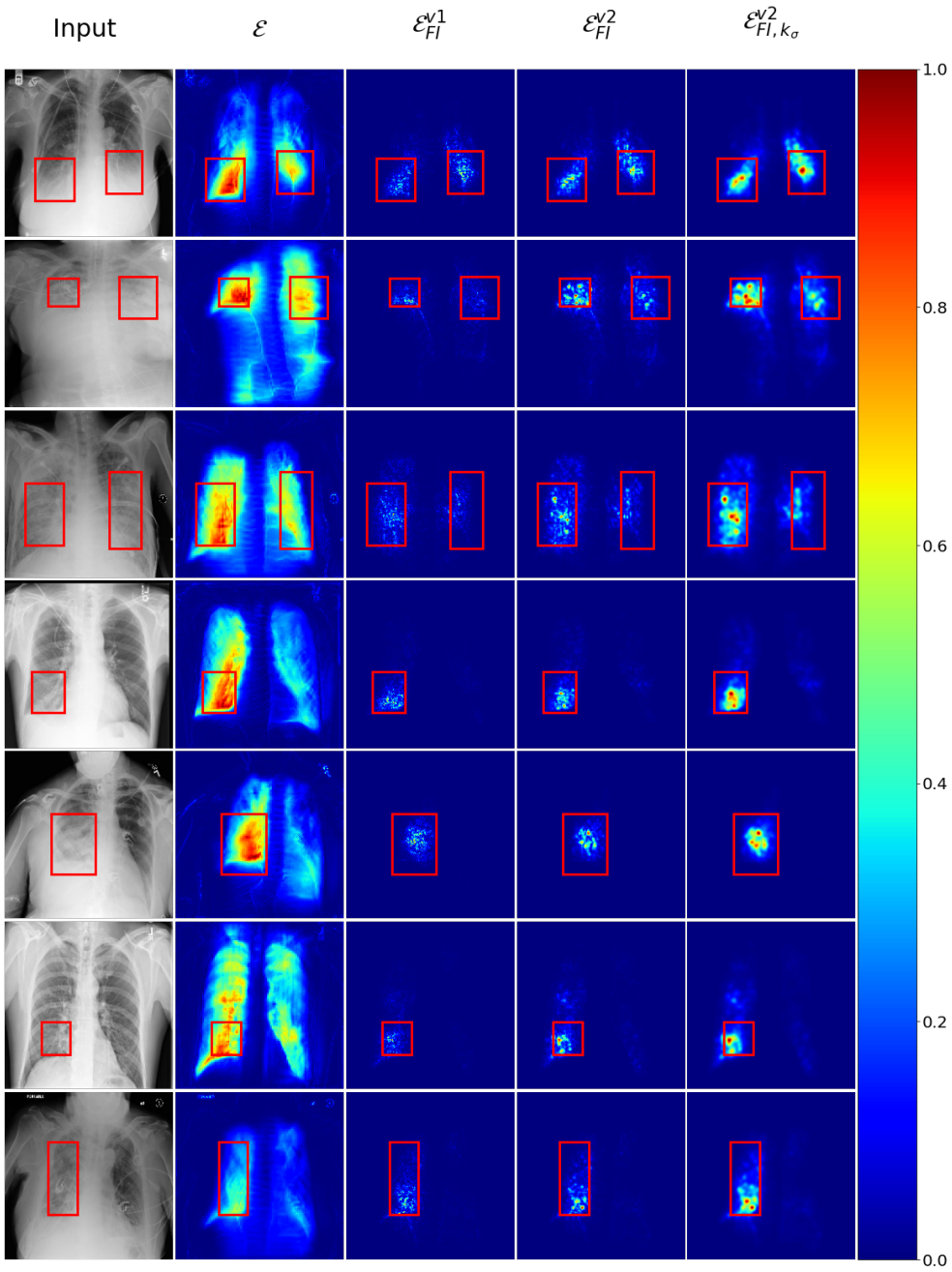


Figure C.13: Pneumonia detection (1) - Comparison between counterfactual baseline and path-based integration techniques for SyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

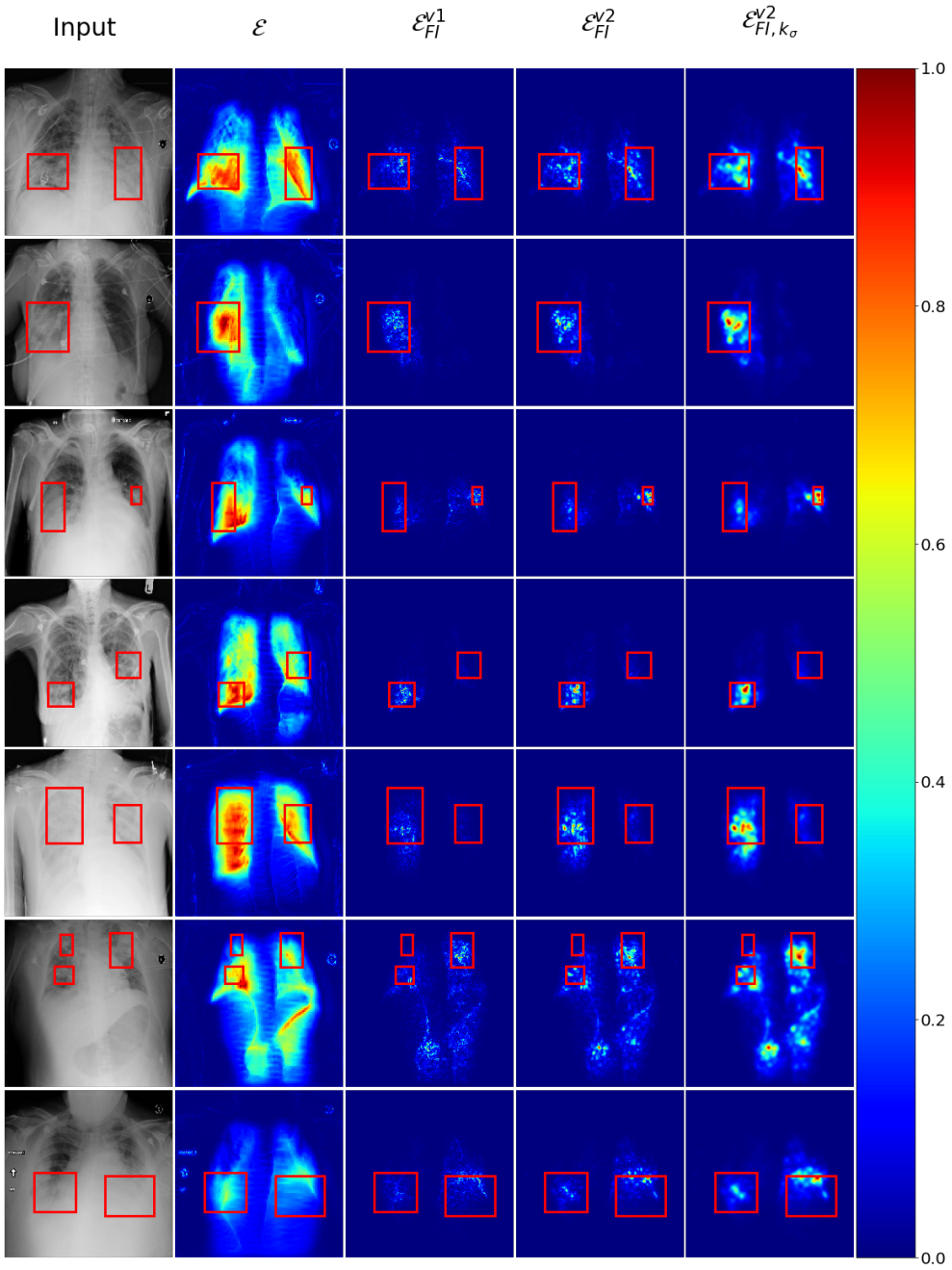


Figure C.14: Pneumonia detection (2) - Comparison between counterfactual baseline and path-based integration techniques for SyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

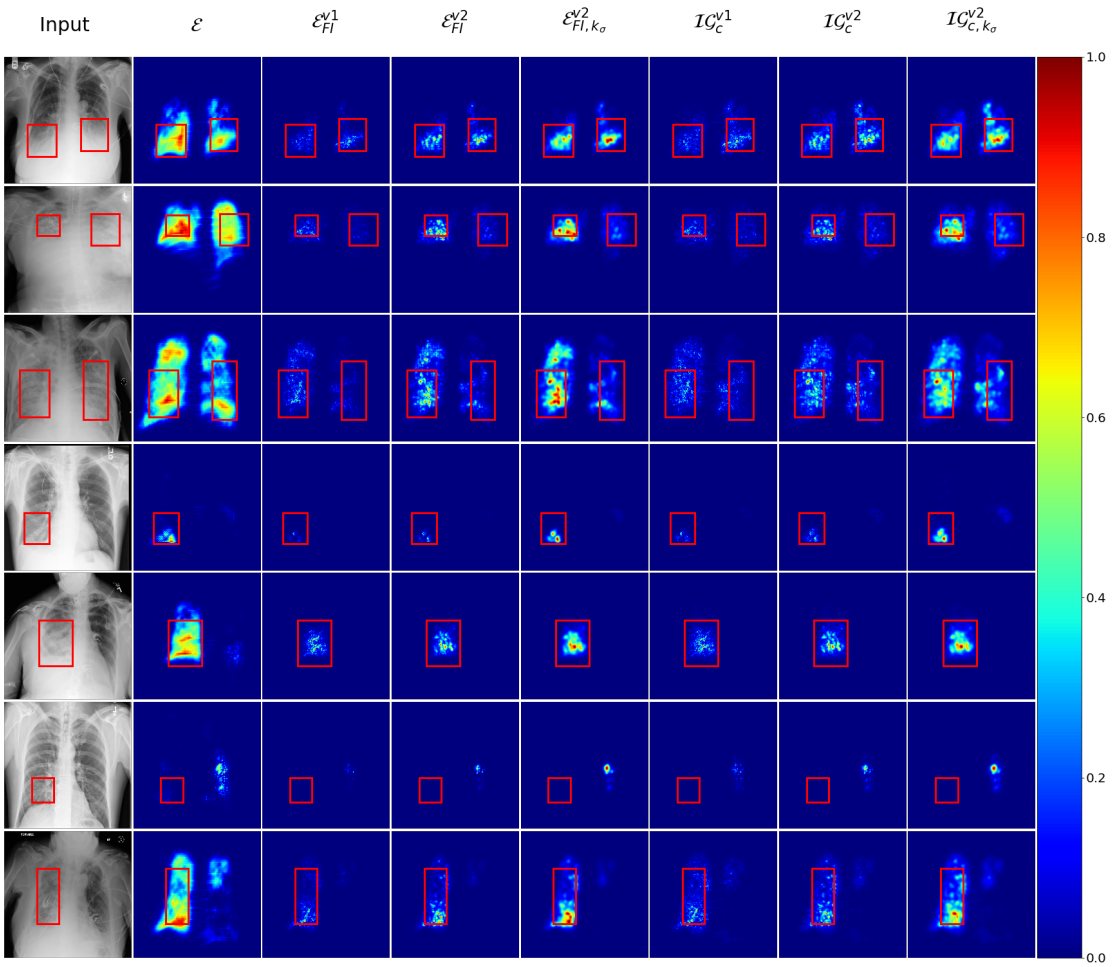


Figure C.15: Pneumonia detection (1) - Comparison between counterfactual baseline and path-based integration techniques for CyLatentCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyLatentCE) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

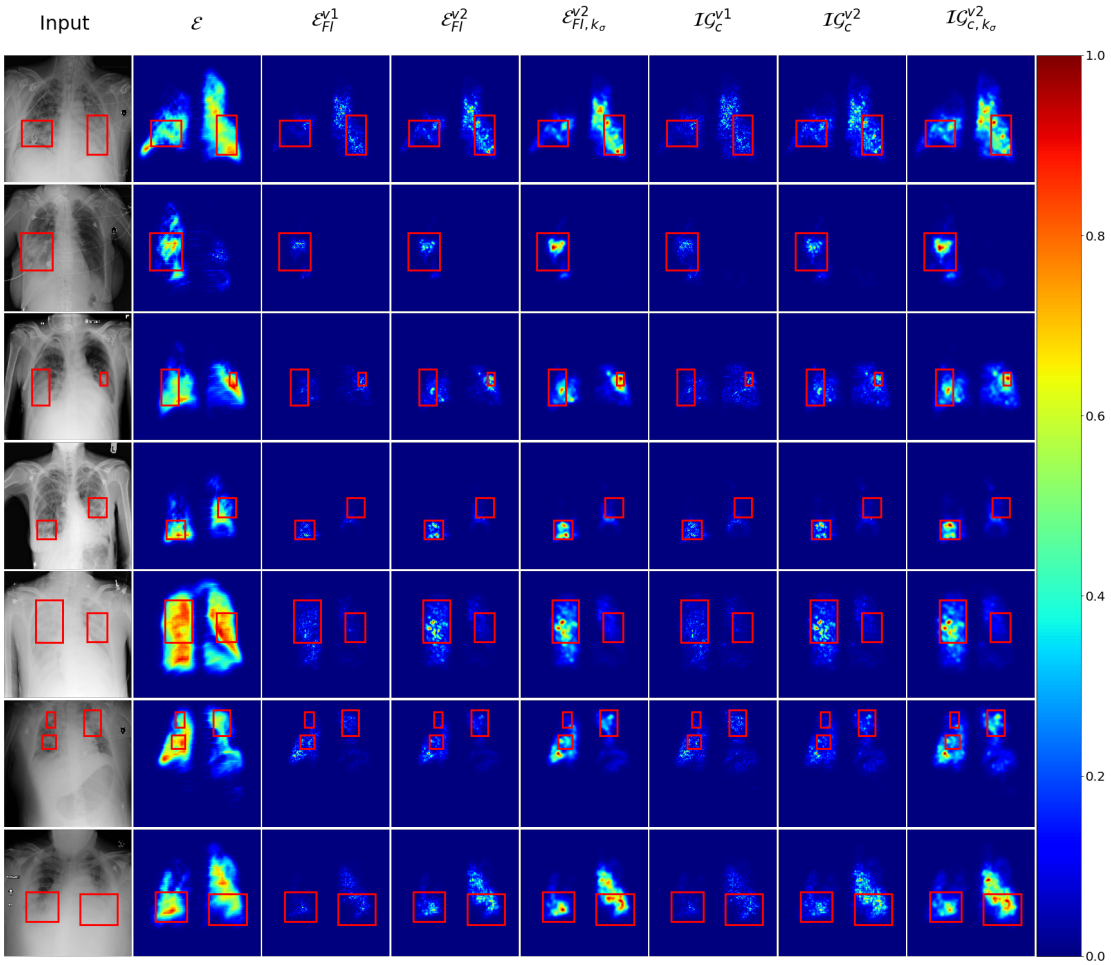


Figure C.16: Pneumonia detection (2) - Comparison between counterfactual baseline and path-based integration techniques for CyLatentCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyLatentCE) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

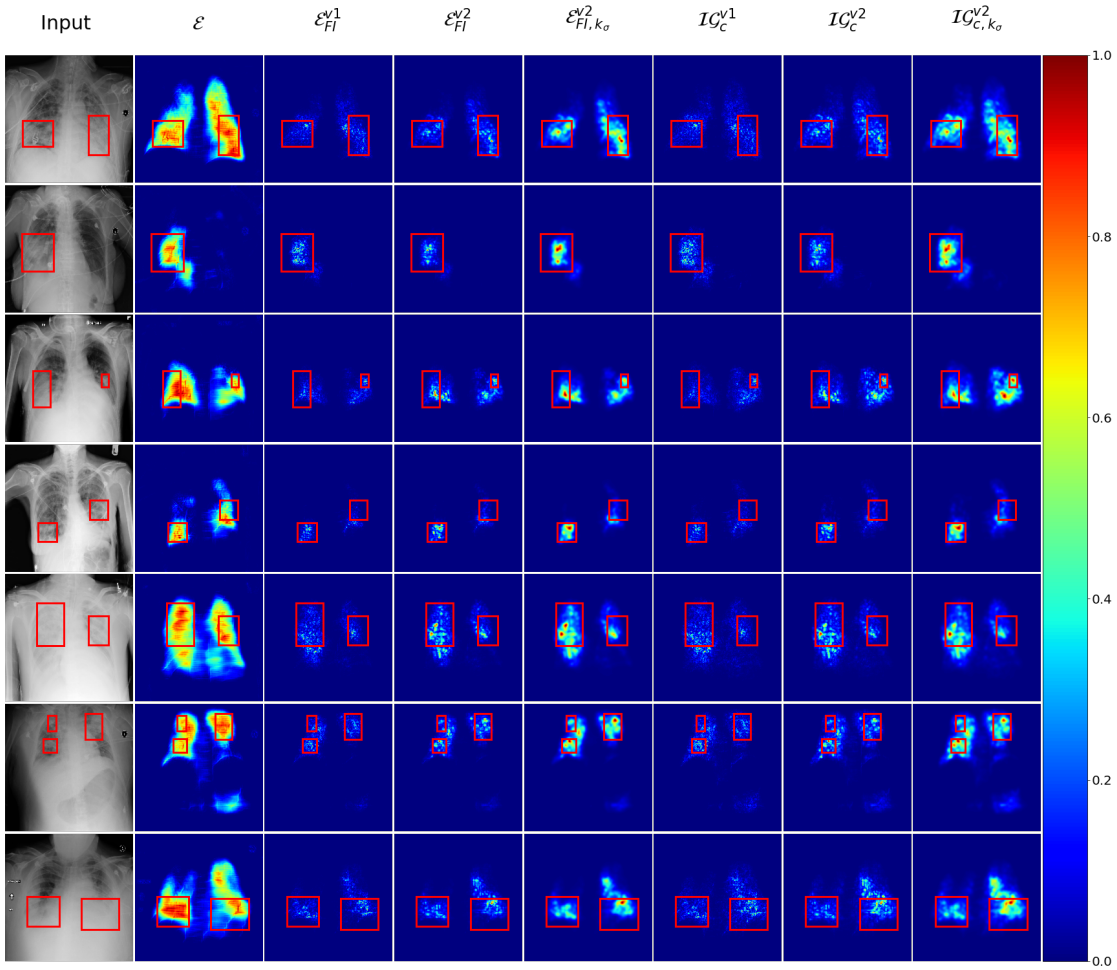


Figure C.17: Pneumonia detection - Comparison between counterfactual baseline and path-based integration techniques for CyImageCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyImageCE) computed against the stable generation; the integrated method v1 $\mathcal{E}^{v1}_{FI}$; the integrated method v2 $\mathcal{E}^{v2}_{FI}$; the regularized integrated method $\mathcal{E}^{v2}_{FI,k_\sigma}$; the counterfactual integrated gradient v1 $\mathcal{IG}^{v1}_{c}$; the counterfactual integrated gradient v2 $\mathcal{IG}^{v2}_{c}$; and the counterfactual integrated gradient regularized $\mathcal{IG}^{v2}_{c,k_\sigma}$.

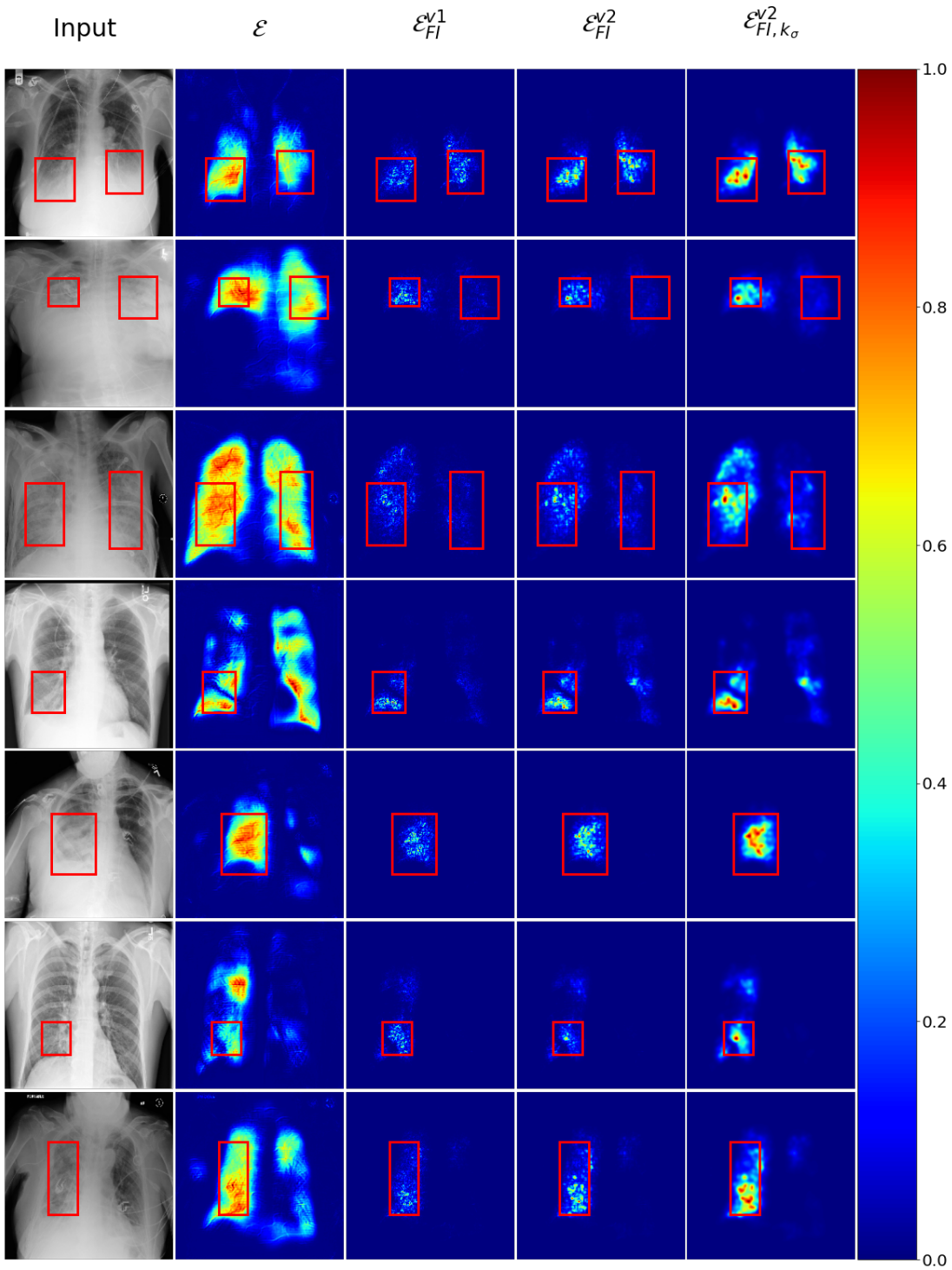


Figure C.18: Pneumonia detection (1) - Comparison between counterfactual baseline and path-based integration techniques for SySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

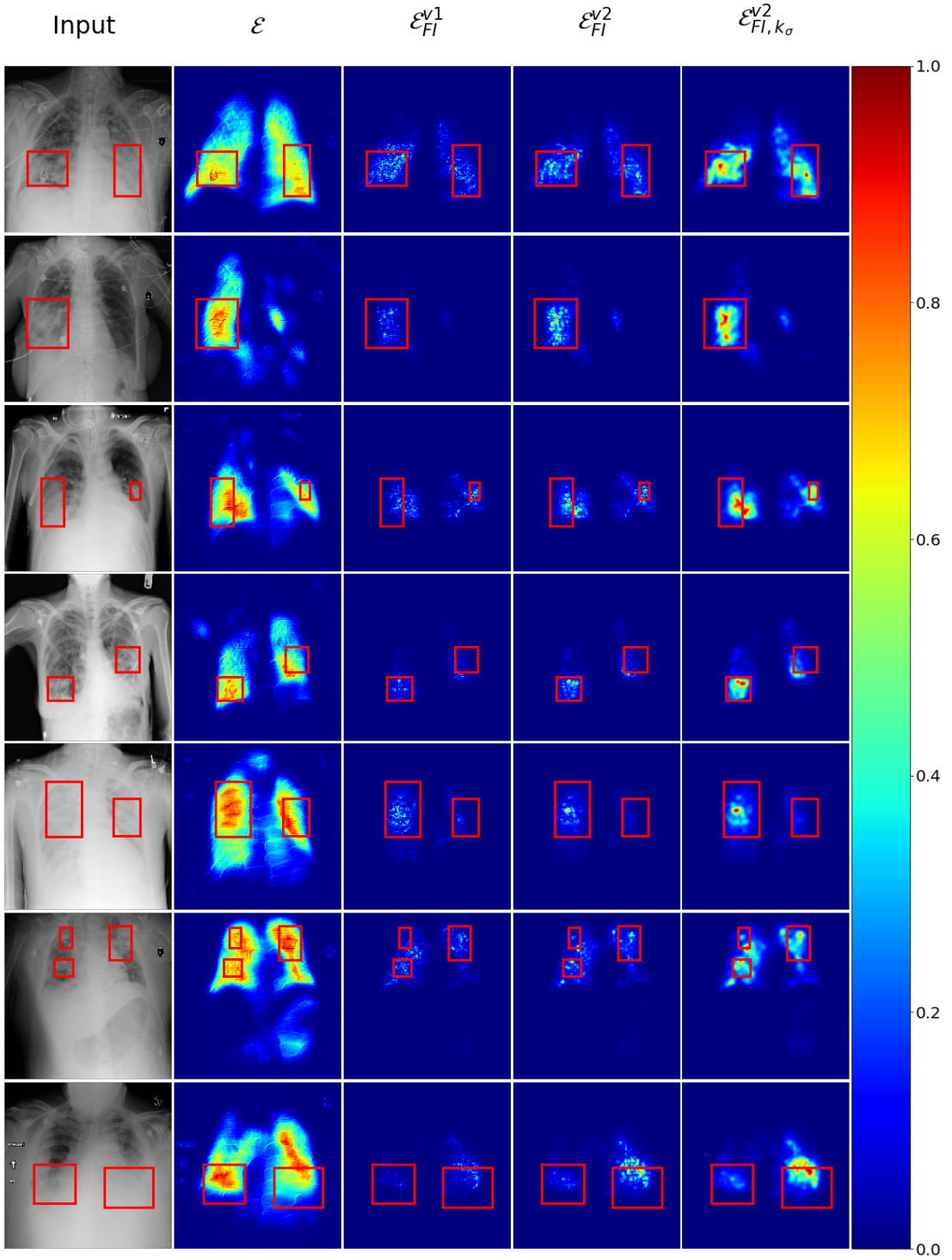


Figure C.19: Pneumonia detection (2) - Comparison between counterfactual baseline and path-based integration techniques for SySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}^{v1}_{FI}$; the integrated method v2 $\mathcal{E}^{v2}_{FI}$; and the regularized integrated method $\mathcal{E}^{v2}_{FI,k_\sigma}$.

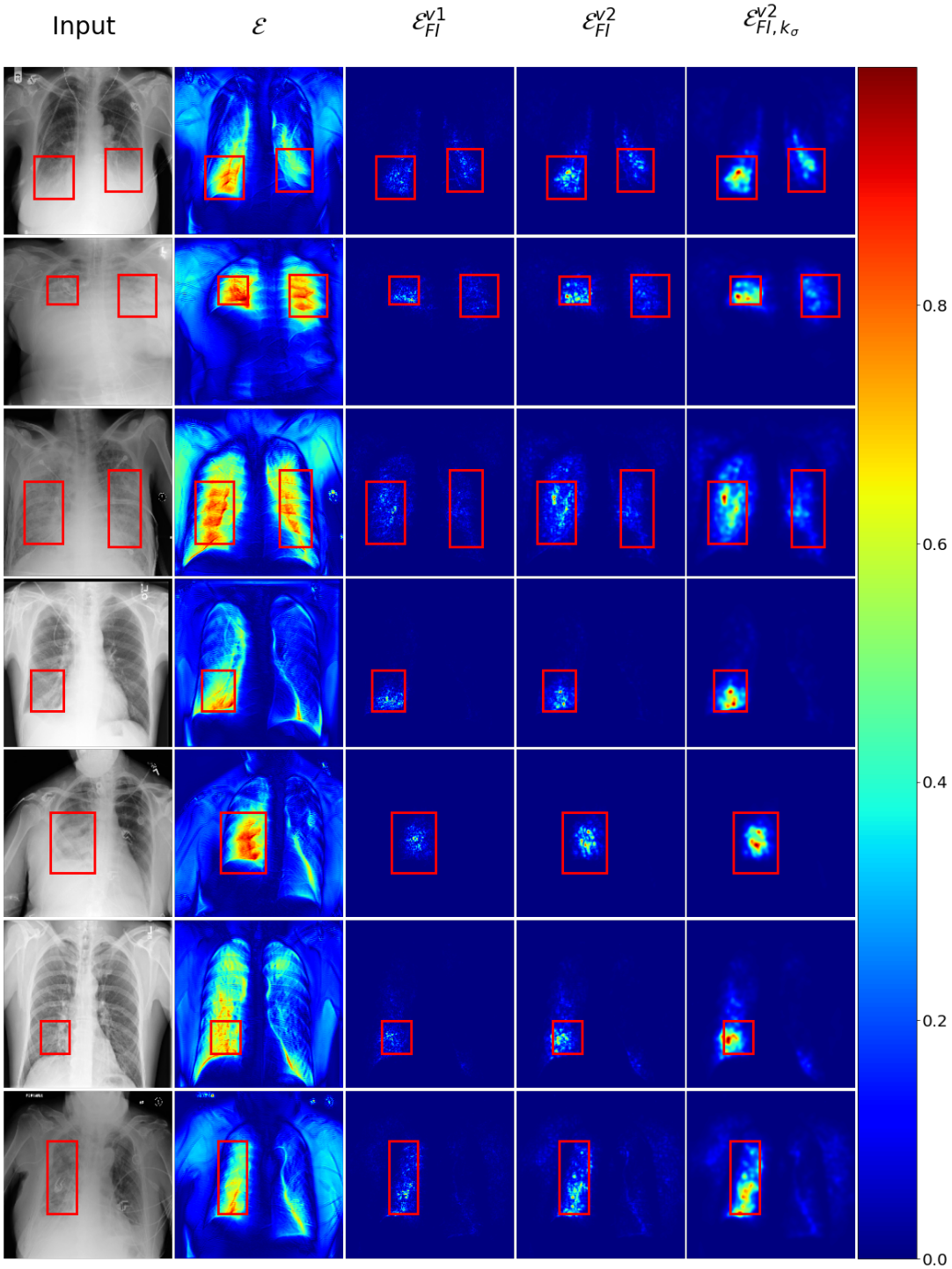


Figure C.20: Pneumonia detection (1) - Comparison between counterfactual baseline and path-based integration techniques for CySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

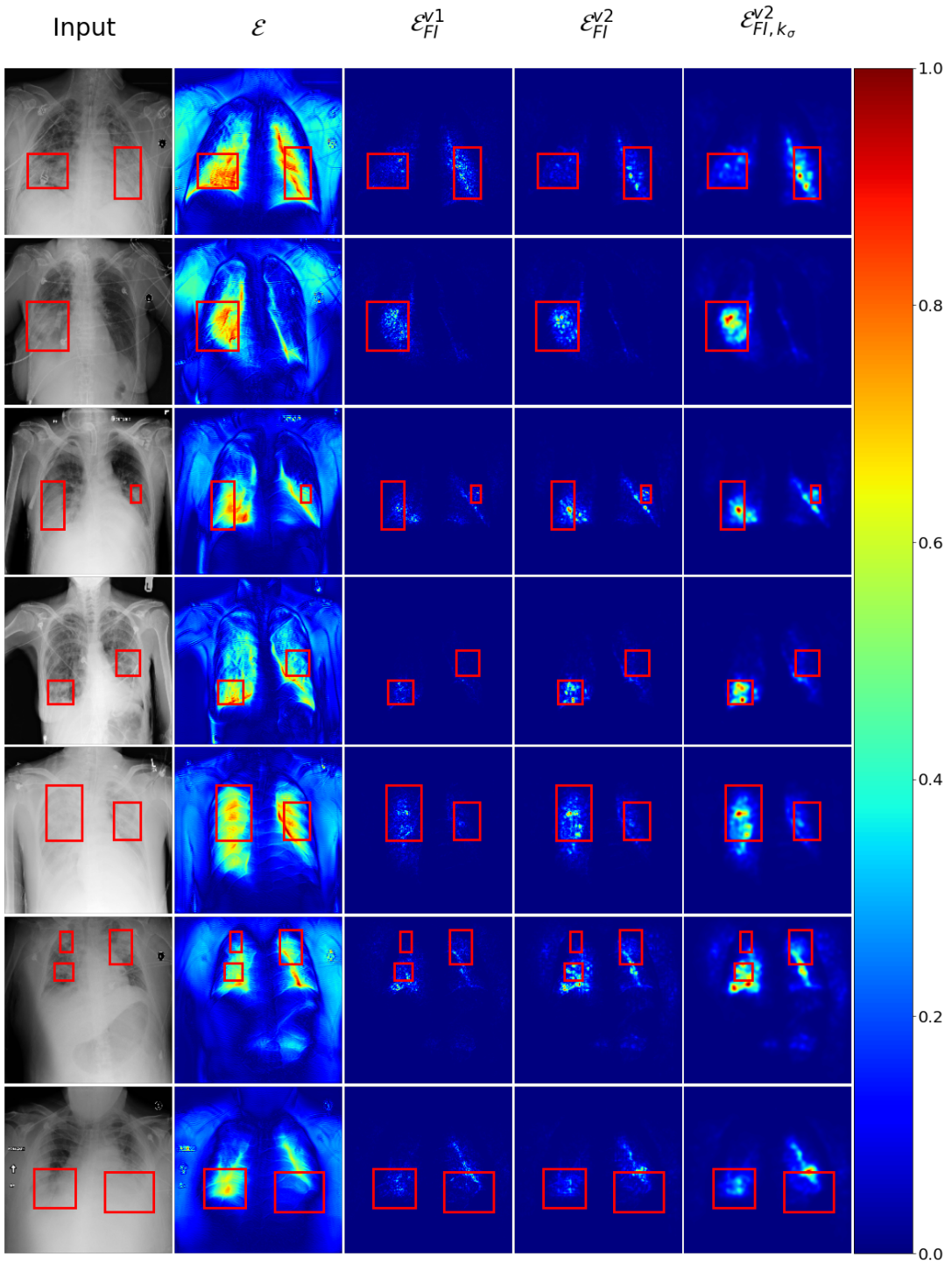


Figure C.21: Pneumonia detection (2) - Comparison between counterfactual baseline and path-based integration techniques for CySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

C.2.2 Brain tumor detection on MRI

We also provide similar visualizations of the diverse integrated counterfactual approach to the brain tumor detection problem:

- SSyGen: Figures C.22 and C.23
- CyCE: Figure C.24
- SyCE: FiguresC.25 and C.26
- CyLatentCE: Figures C.27 and C.28
- CyImageCE: Figure C.29
- SySCGen: Figures C.30 and C.31
- CySCGen: Figures C.32 and C.33

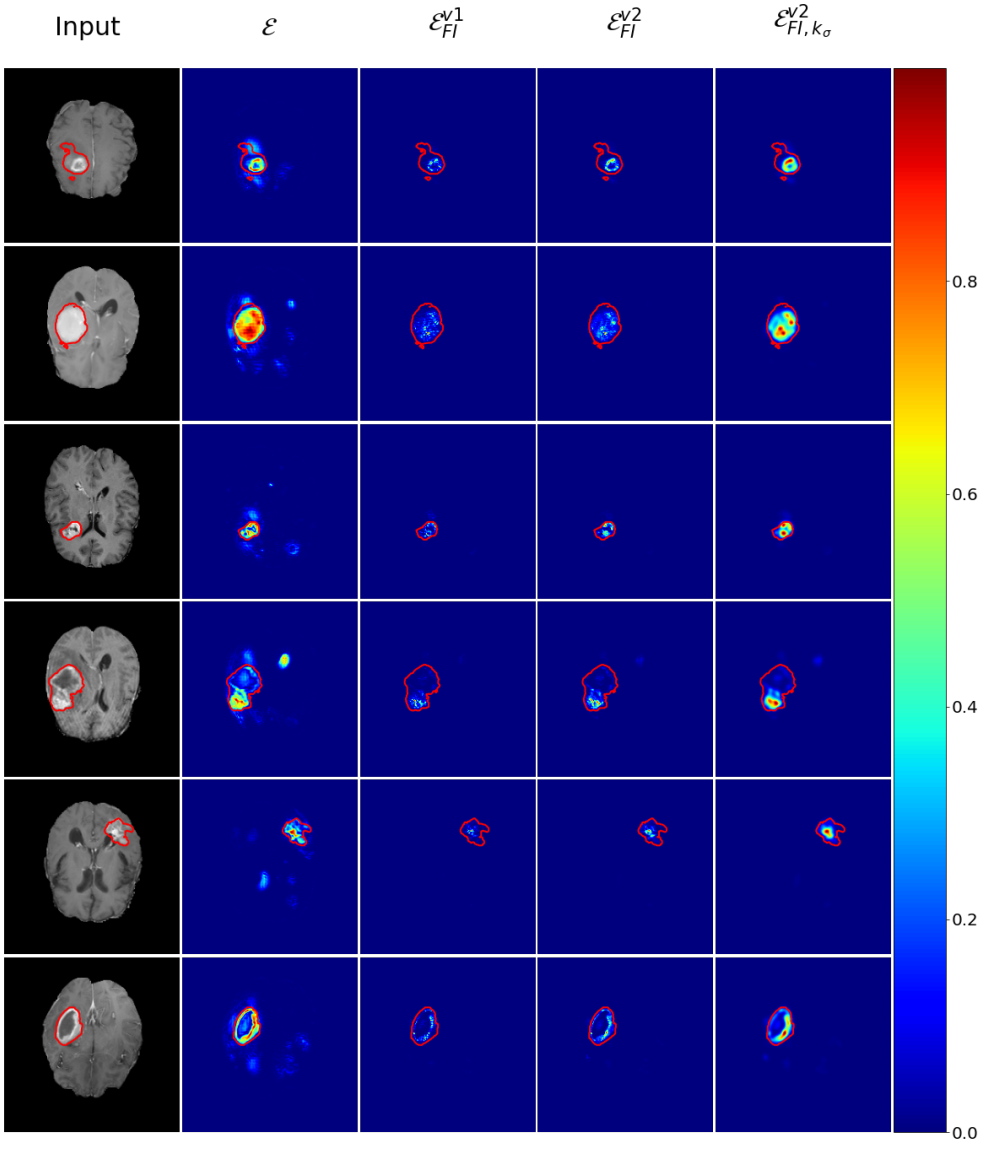


Figure C.22: Brain tumor detection (1) - Comparison between counterfactual baseline and path-based integration techniques for SSyGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SSyGen); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

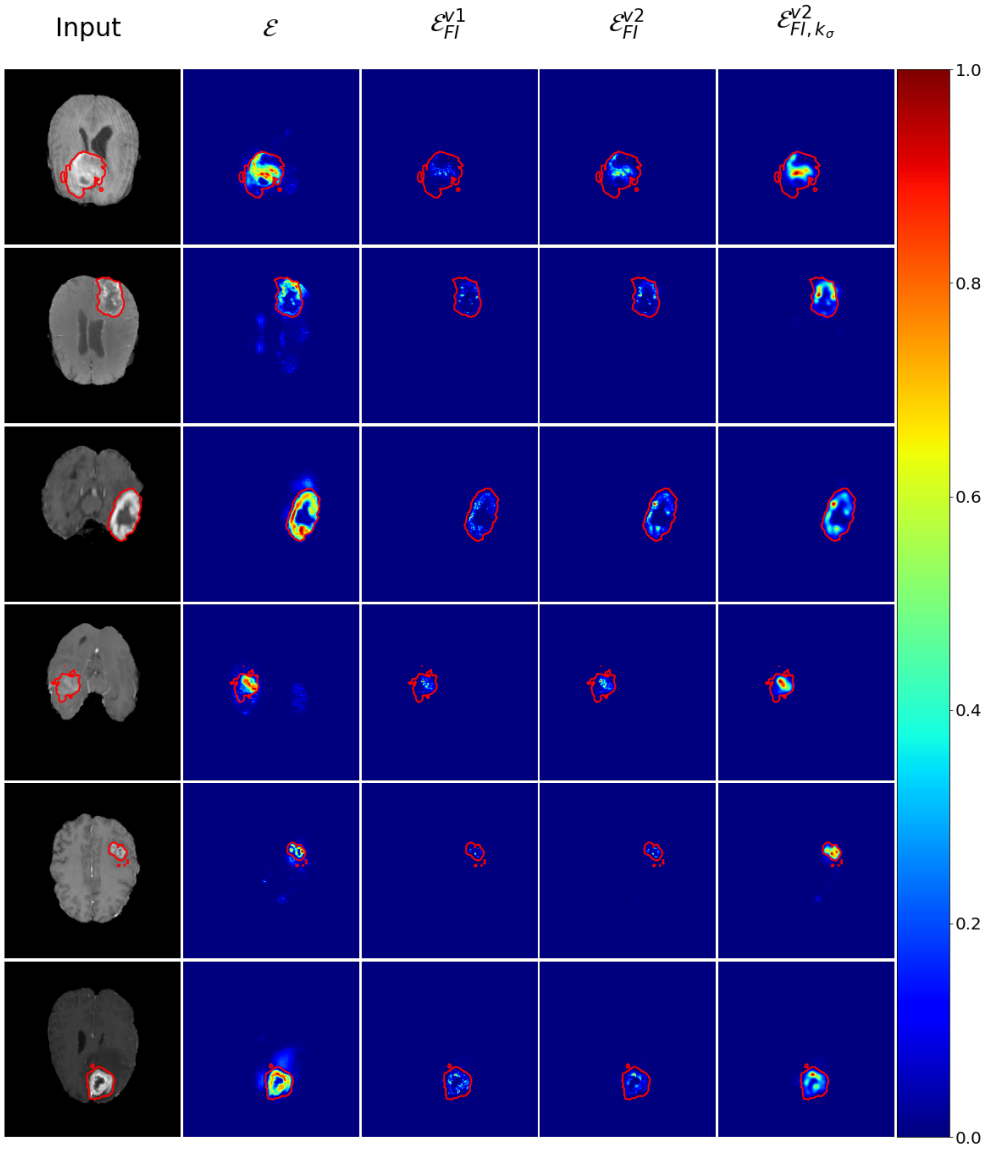


Figure C.23: Brain tumor detection (2) - Comparison between counterfactual baseline and path-based integration techniques for SSyGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SSyGen); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

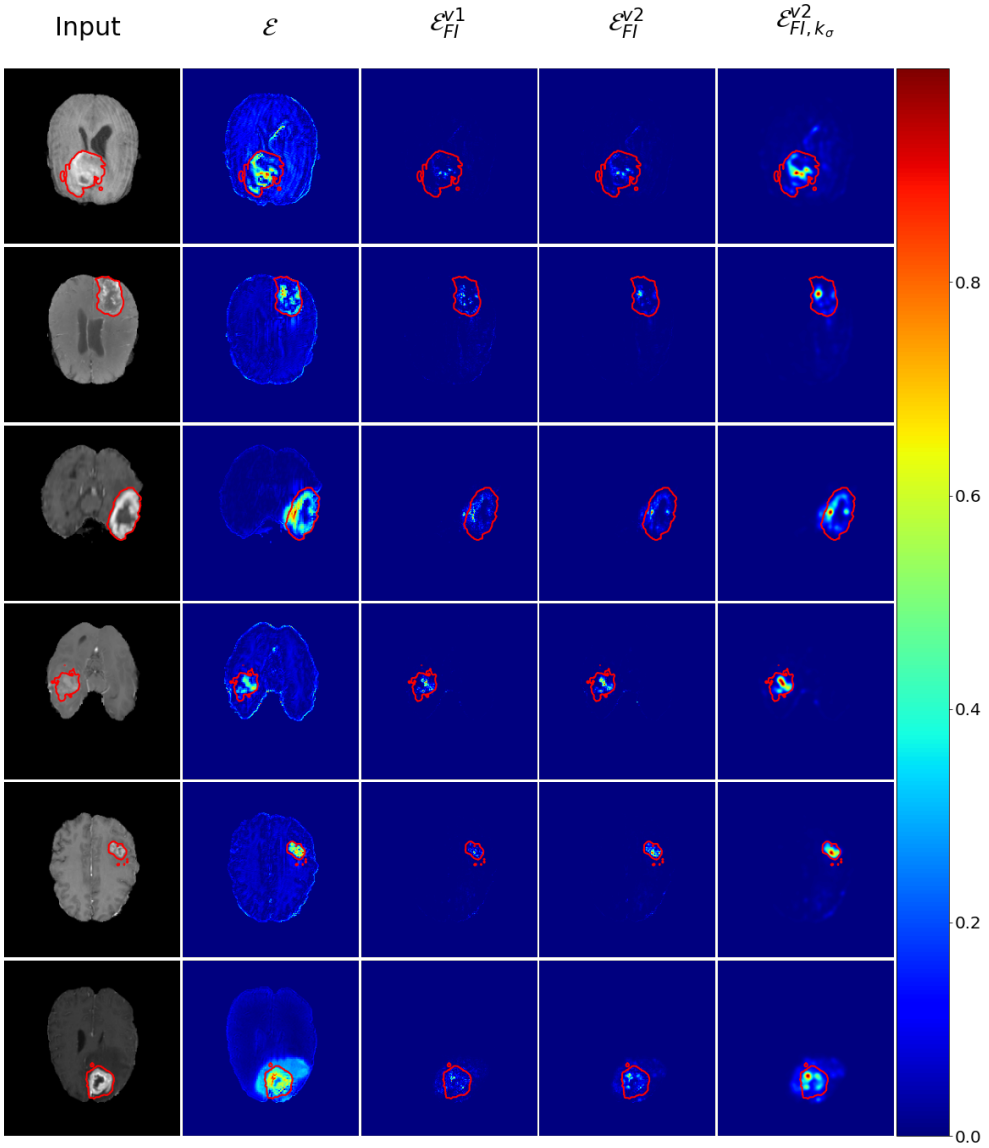


Figure C.24: Brain tumor detection - Comparison between counterfactual baseline and path-based integration techniques for CyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

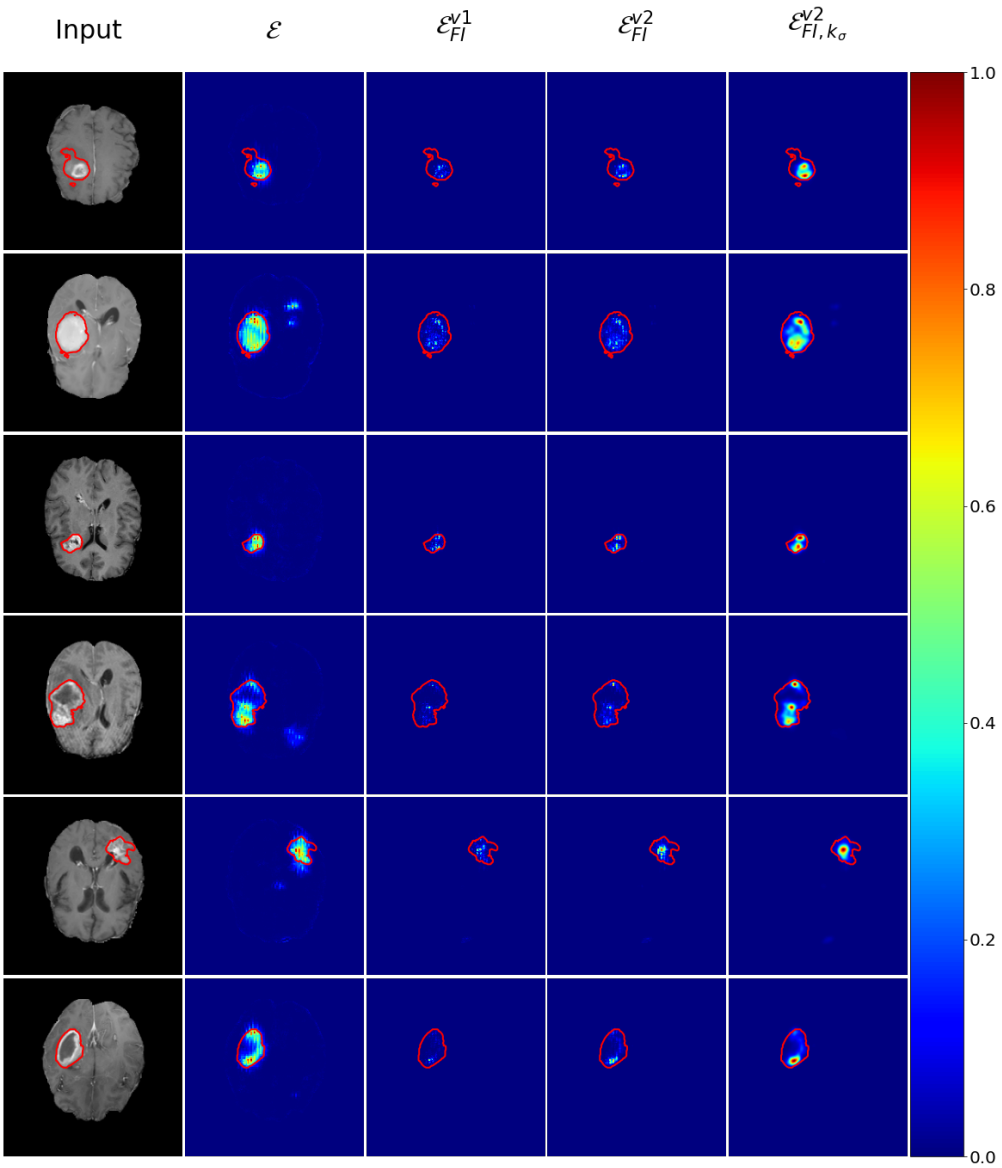


Figure C.25: Brain tumor detection (1) - Comparison between counterfactual baseline and path-based integration techniques for SyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

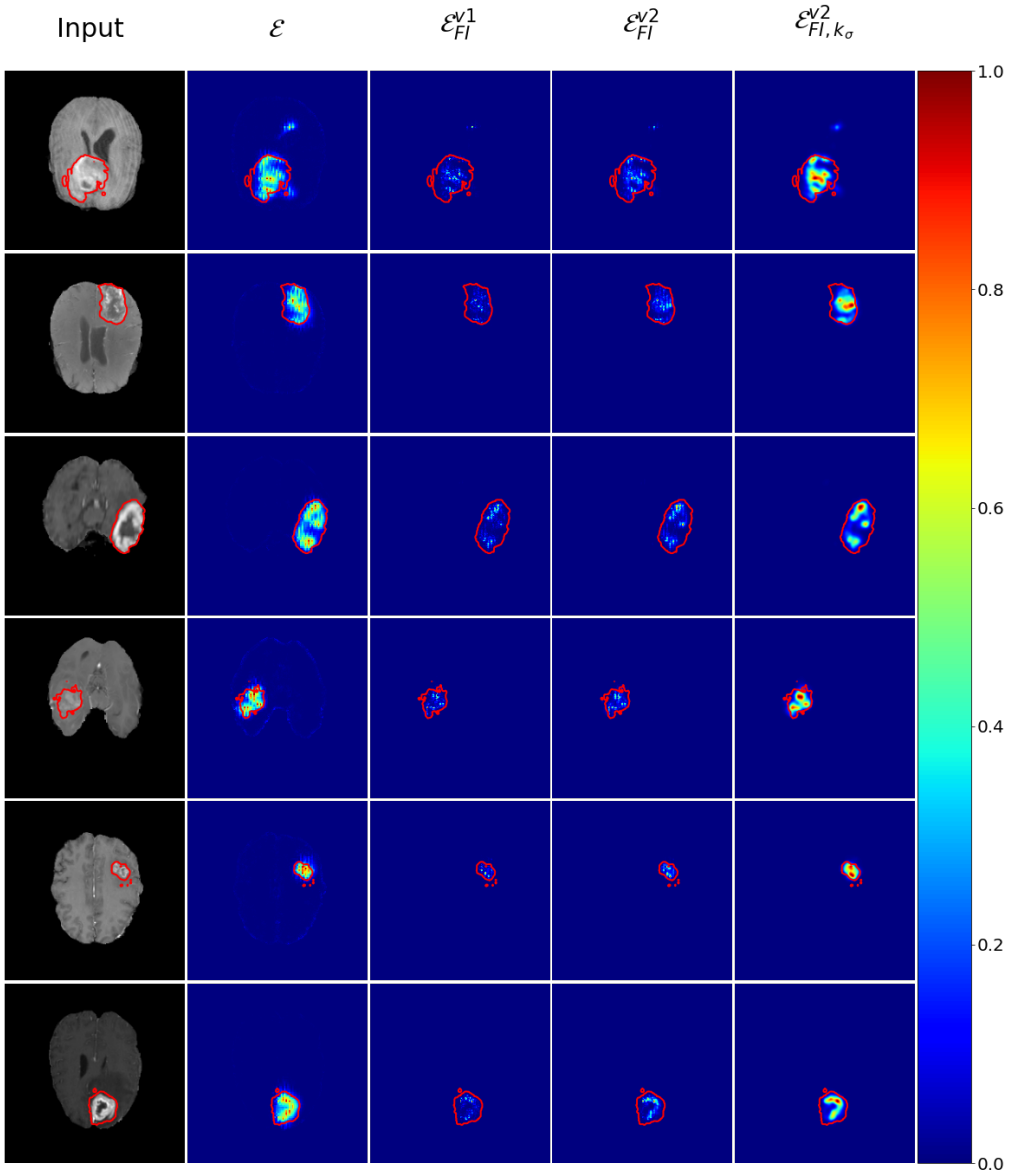


Figure C.26: Brain tumor detection (2) - Comparison between counterfactual baseline and path-based integration techniques for SyCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SyCE); the integrated method v1 $\mathcal{E}_{FI}^{v1}$, the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

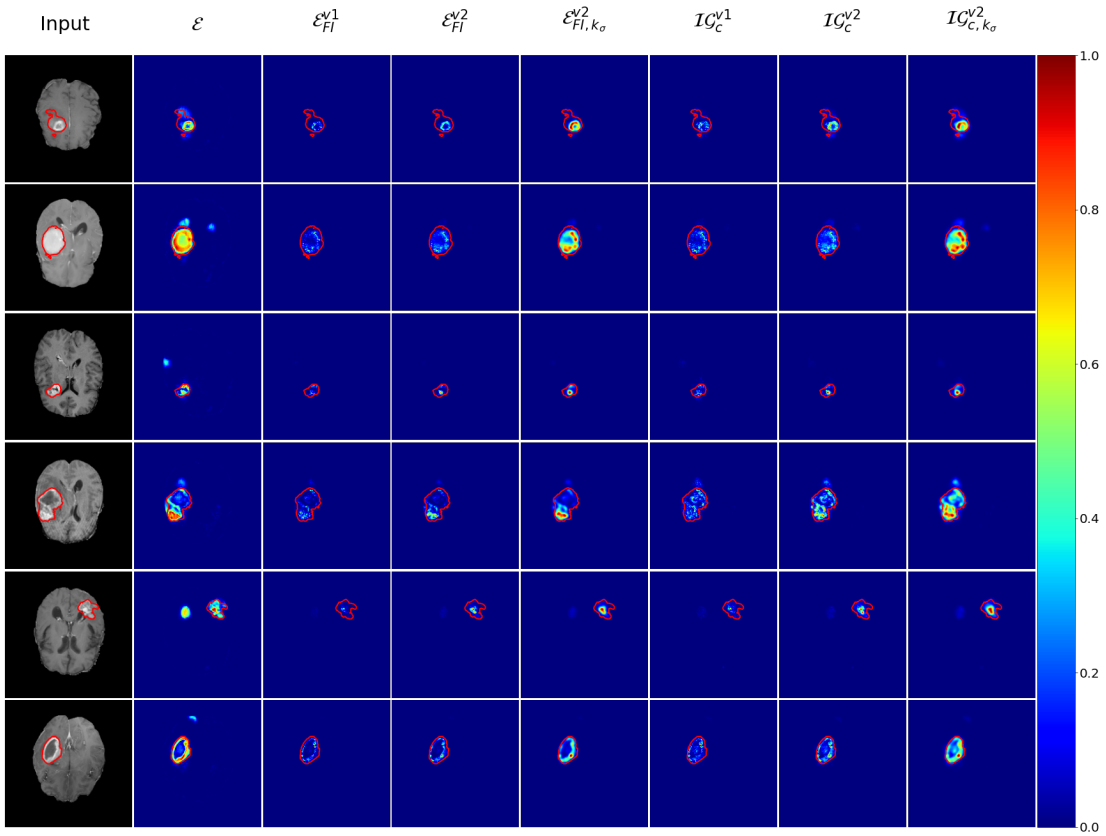


Figure C.27: Brain tumor detection (1) - Comparison between counterfactual baseline and path-based integration techniques for CyLatentCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyLatentCE) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

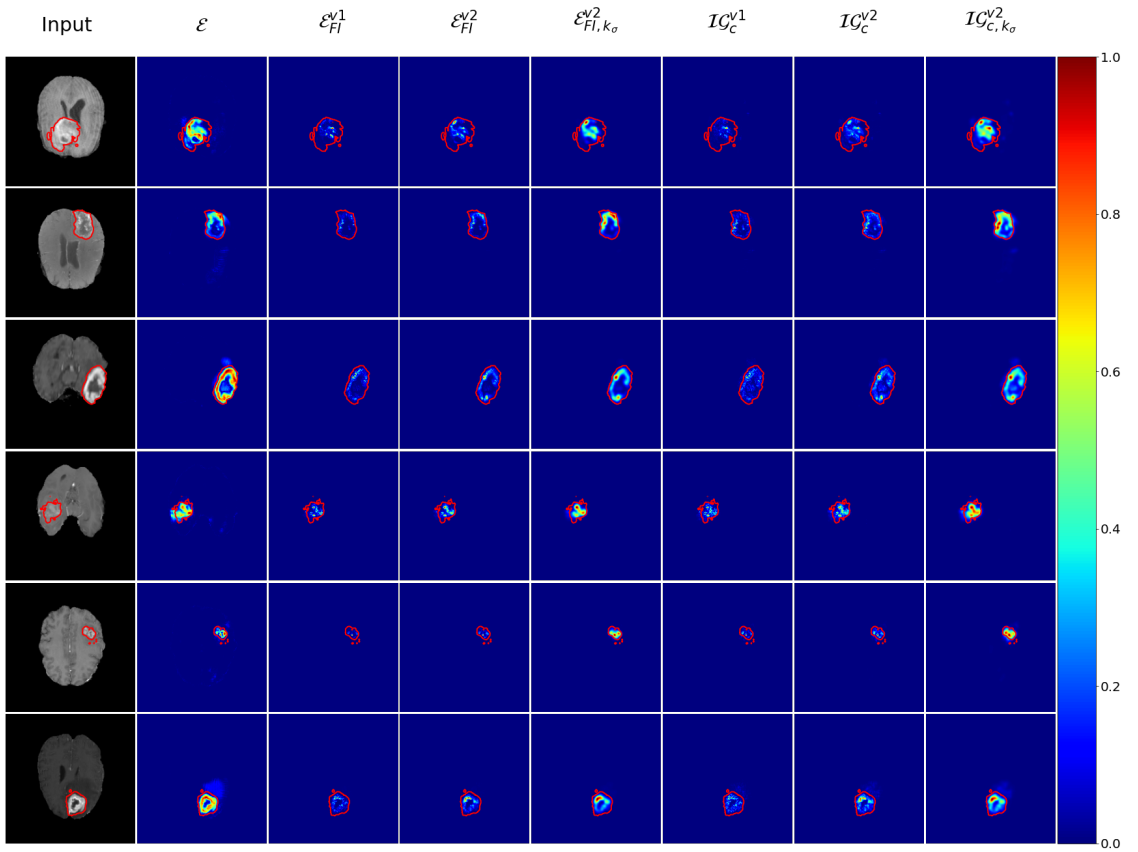


Figure C.28: Brain tumor detection (2) - Comparison between counterfactual baseline and path-based integration techniques for CyLatentCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyLatentCE) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

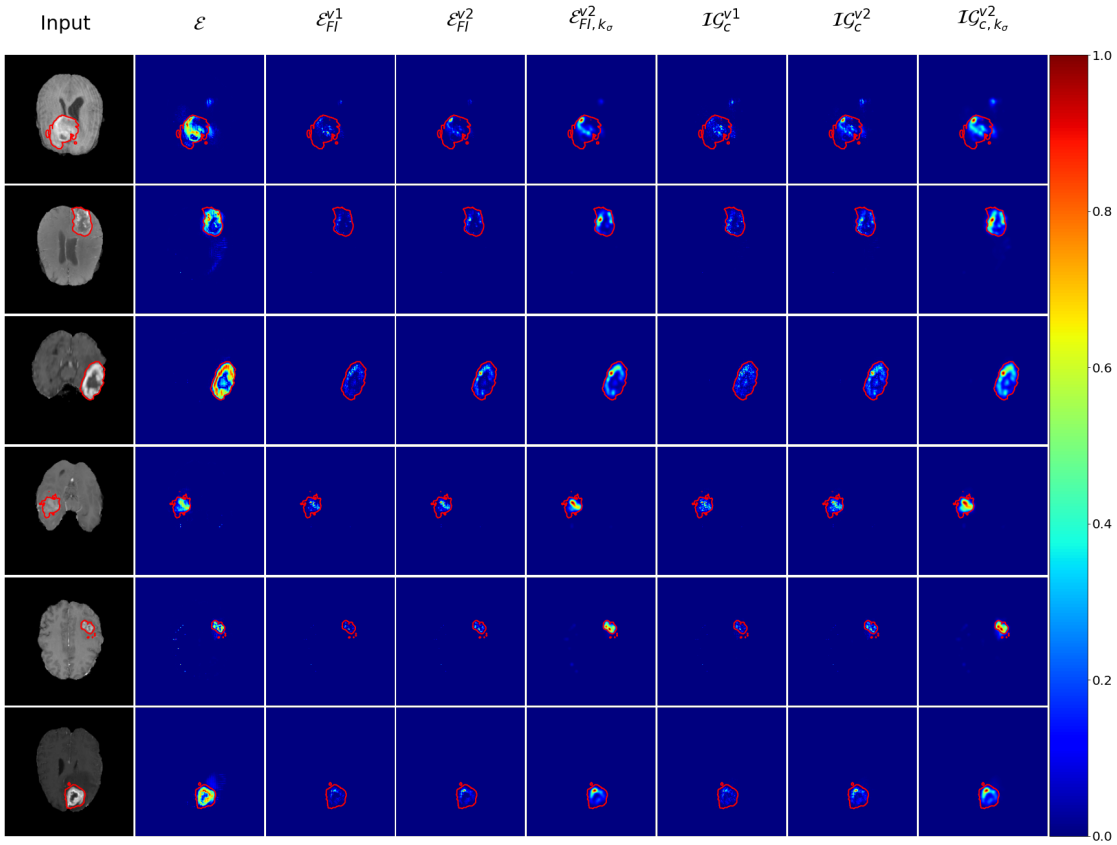


Figure C.29: Brain tumor detection - Comparison between counterfactual baseline and path-based integration techniques for CyImageCE. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CyImageCE) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$; the counterfactual integrated gradient v1 $\mathcal{IG}_c^{v1}$; the counterfactual integrated gradient v2 $\mathcal{IG}_c^{v2}$; and the counterfactual integrated gradient regularized $\mathcal{IG}_{c,k_\sigma}^{v2}$.

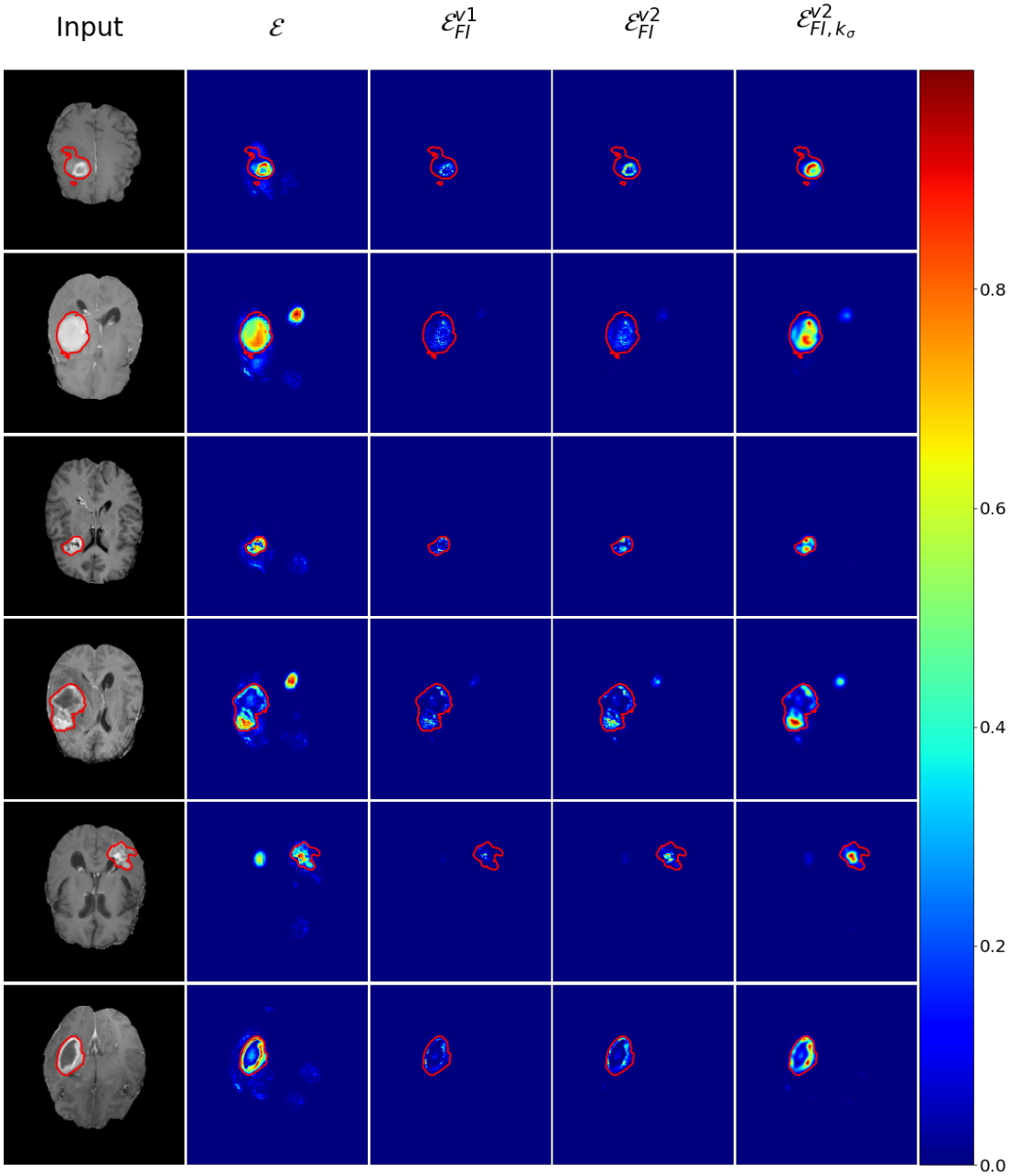


Figure C.30: Brain tumor detection (1) - Comparison between counterfactual baseline and path-based integration techniques for SySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

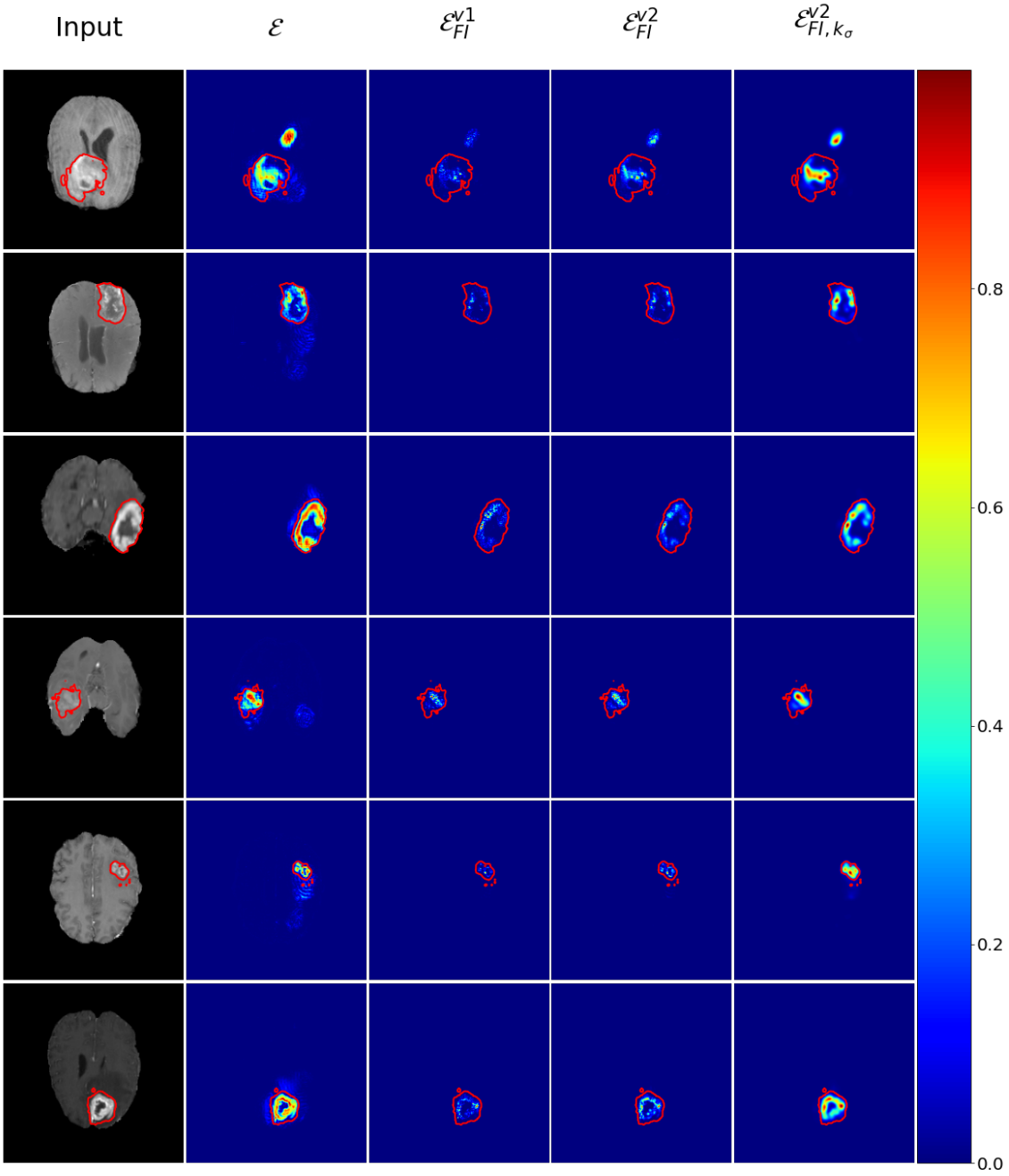


Figure C.31: Brain tumor detection (2) - Comparison between counterfactual baseline and path-based integration techniques for SySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (SySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

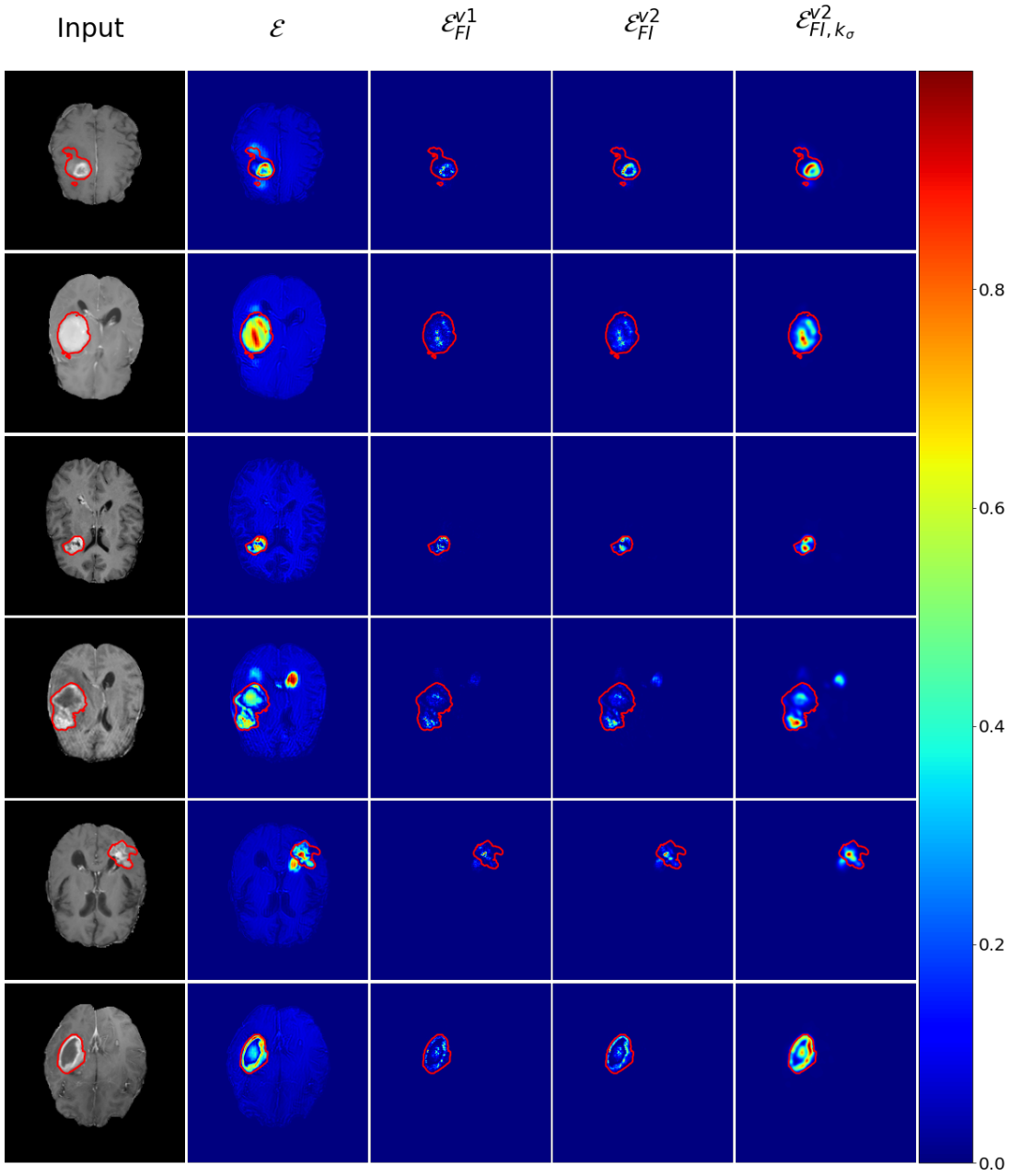


Figure C.32: Brain tumor detection (1) - Comparison between counterfactual baseline and path-based integration techniques for CySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

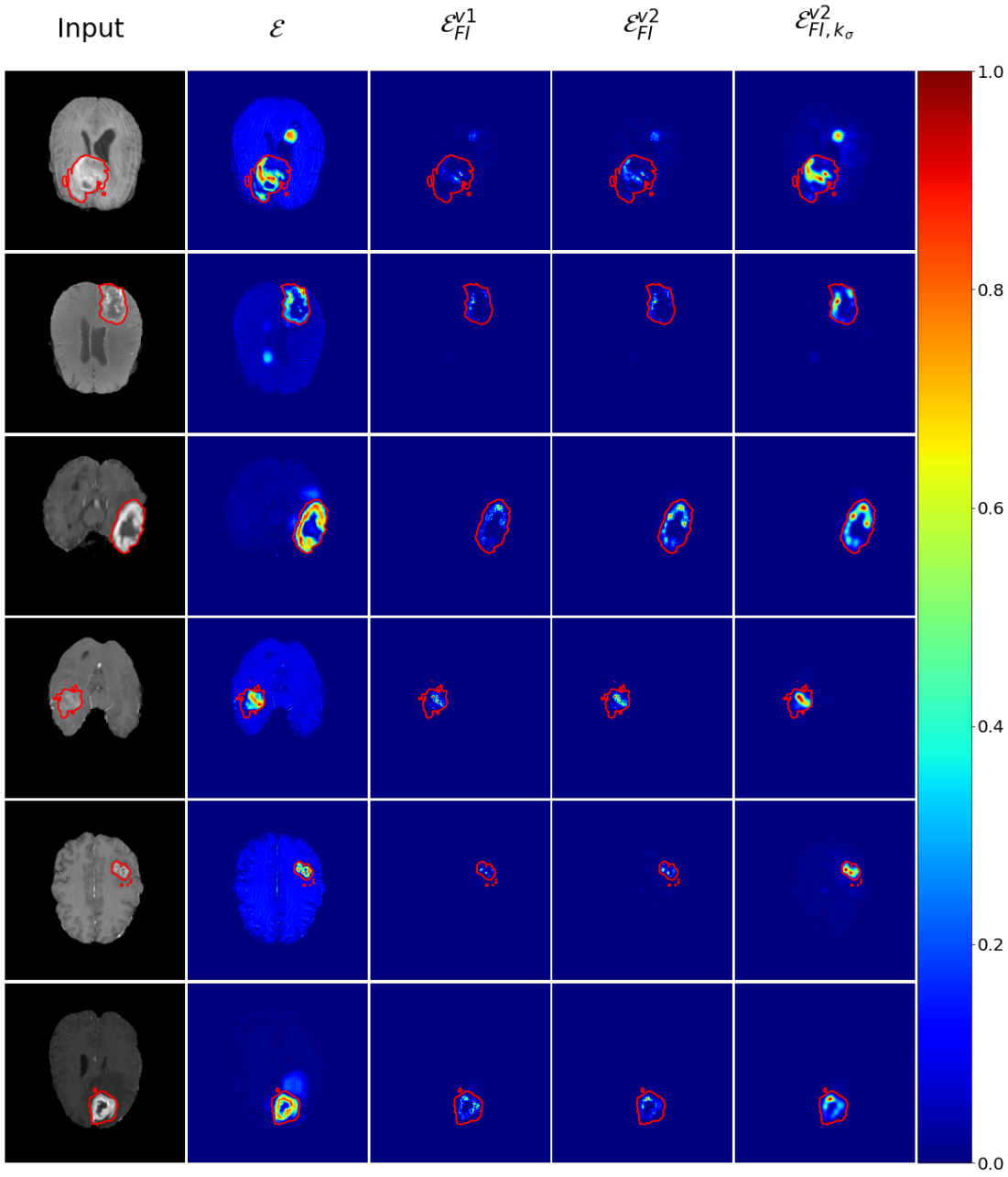


Figure C.33: Brain tumor detection (2) - Comparison between counterfactual baseline and path-based integration techniques for CySCGen. From left to right: the input image with ground truth annotations; the counterfactual attribution baseline $\mathcal{E}$ (CySCGen) computed against the stable generation; the integrated method v1 $\mathcal{E}_{FI}^{v1}$; the integrated method v2 $\mathcal{E}_{FI}^{v2}$; and the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$.

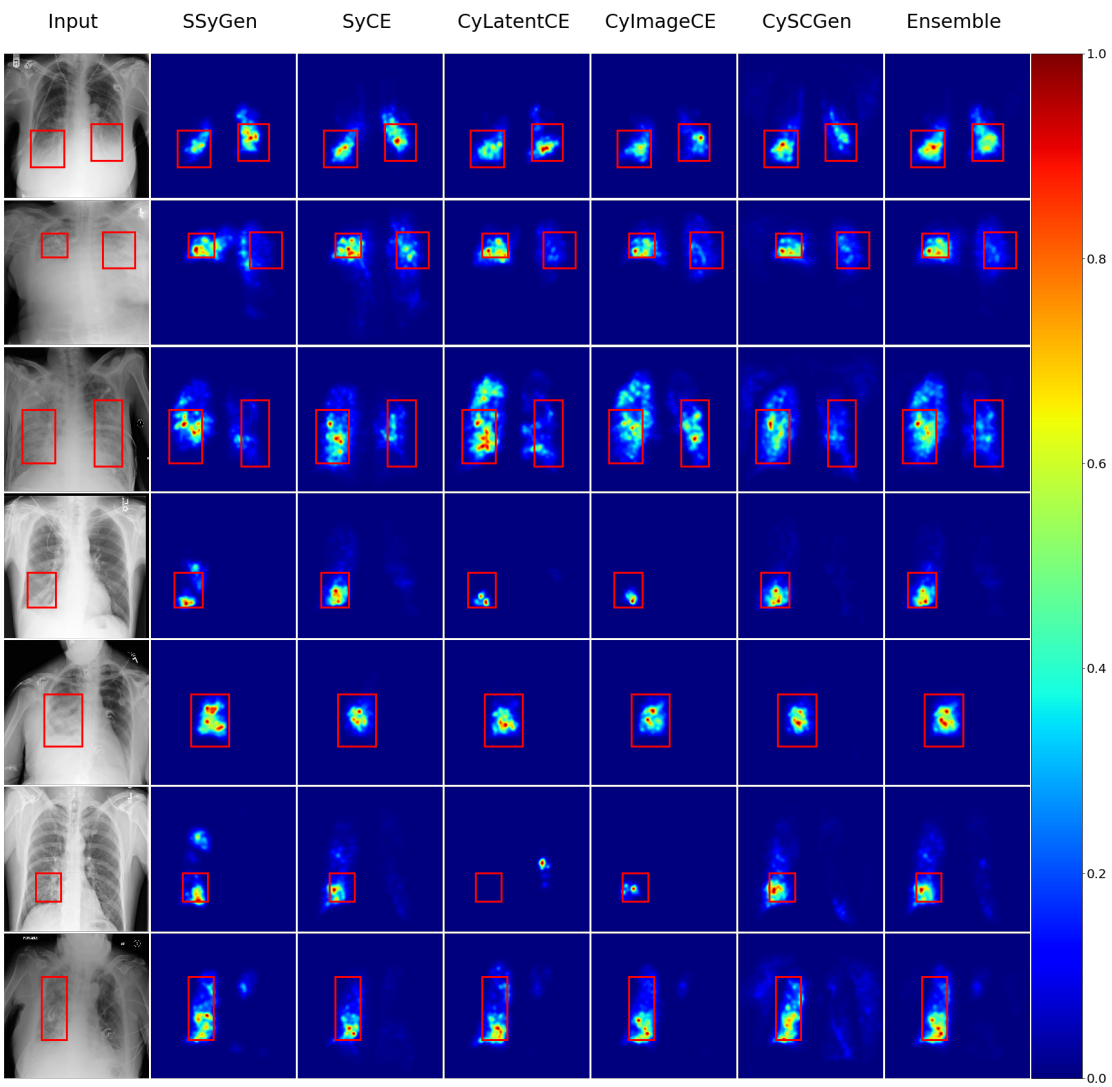


Figure C.34: Pneumonia detection - Comparison between performing counterfactual techniques and ensemble approach. In the first column: the input image and the ground truth annotations; In columns 2 to 6: diverse integrated counterfactual attributions; and in the last column, the ensemble approach combining the previous integrated techniques.

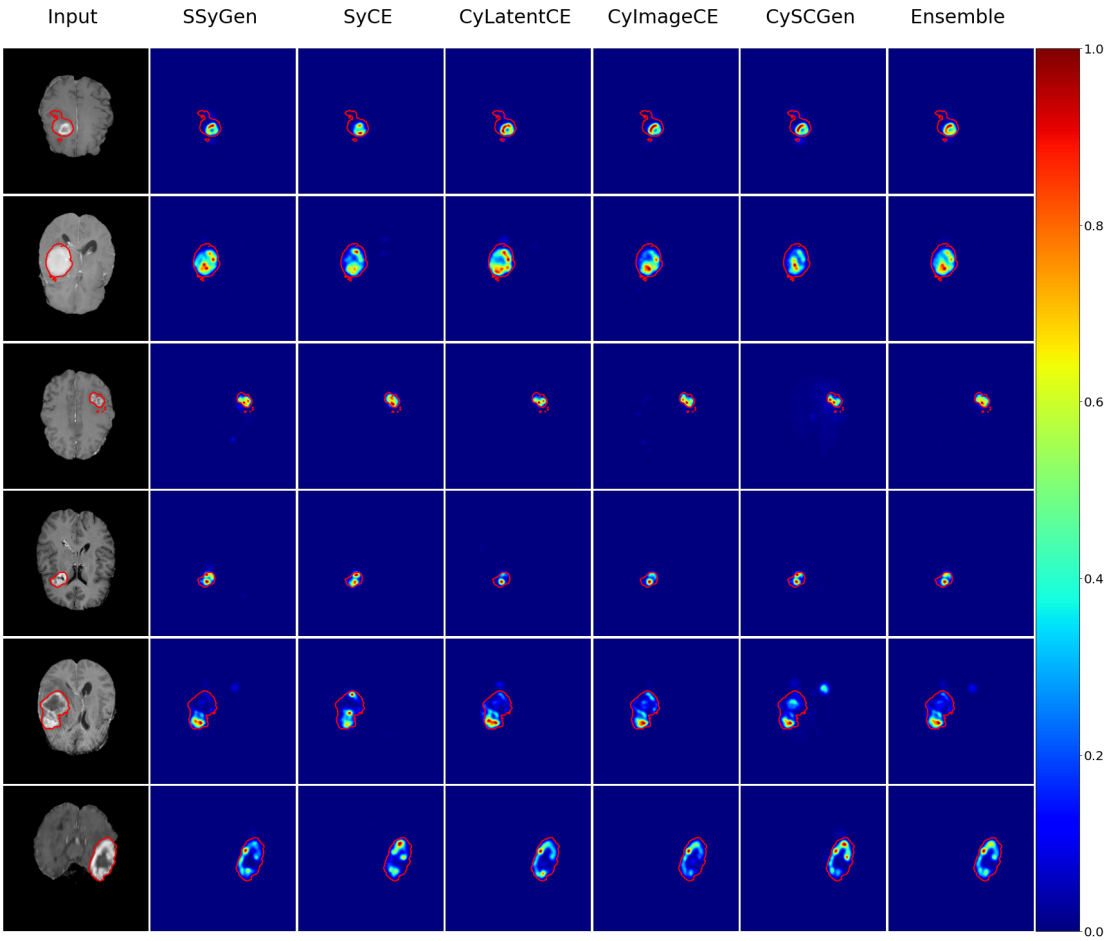


Figure C.35: Brain tumor detection - Comparison between performing counterfactual techniques and ensemble approach. In the first column: the input image and the ground truth annotations; In columns 2 to 6: diverse integrated counterfactual attributions; and in the last column, the ensemble approach combining the previous integrated techniques.

C.2.3 Variation of the localization performance with the number of integration steps

We provide the localization metrics (IoU and NCC) of CyLatentCE for different number of integration steps in Table

Table C.3: Computation time and localization performance given integration steps. We study the impact of the number of integration steps on the computation time and the localization performance for (a) pneumonia and (b) brain tumor detection.

(a) Pneumonia detection

Method	Metric	$\mathcal{E}$	$\mathcal{E}^{v2}_{FI,k_\sigma}$			
			50 steps	10 steps	5 steps	2 steps
CyLatentCE	Comput. time*	0.048	0.275	0.244	0.240	0.238
	IoU**	0.2551	0.2785	0.2784	0.2781	0.2752
	NCC	0.5160	0.4217	0.4208	0.4193	0.4146

(b) Brain tumor detection

Method	Metric	$\mathcal{E}$	$\mathcal{E}^{v2}_{FI,k_\sigma}$			
			50 steps	10 steps	5 steps	2 steps
CyLatentCE	Comput. time*	0.049	0.280	0.247	0.243	0.242
	IoU**	0.4259	0.4671	0.4671	0.4665	0.4665
	NCC	0.6309	0.5906	0.5895	0.5882	0.5878

* For 2D problems, all the steps are computed in one batch.
** IoU and FPR are given at the 95th percentile for pneumonia detection and the 98th for brain tumor detection.

C.3 Comparision against state-of-the-art

C.3.1 Raw heatmaps

In additional figures, we display attribution maps and compare counterfactual and integrated counterfactual methods against state-of-the-art techniques:

- On pneumonia detection: Figures C.36 and C.37
- On brain tumor detection: Figures C.38 and C.39

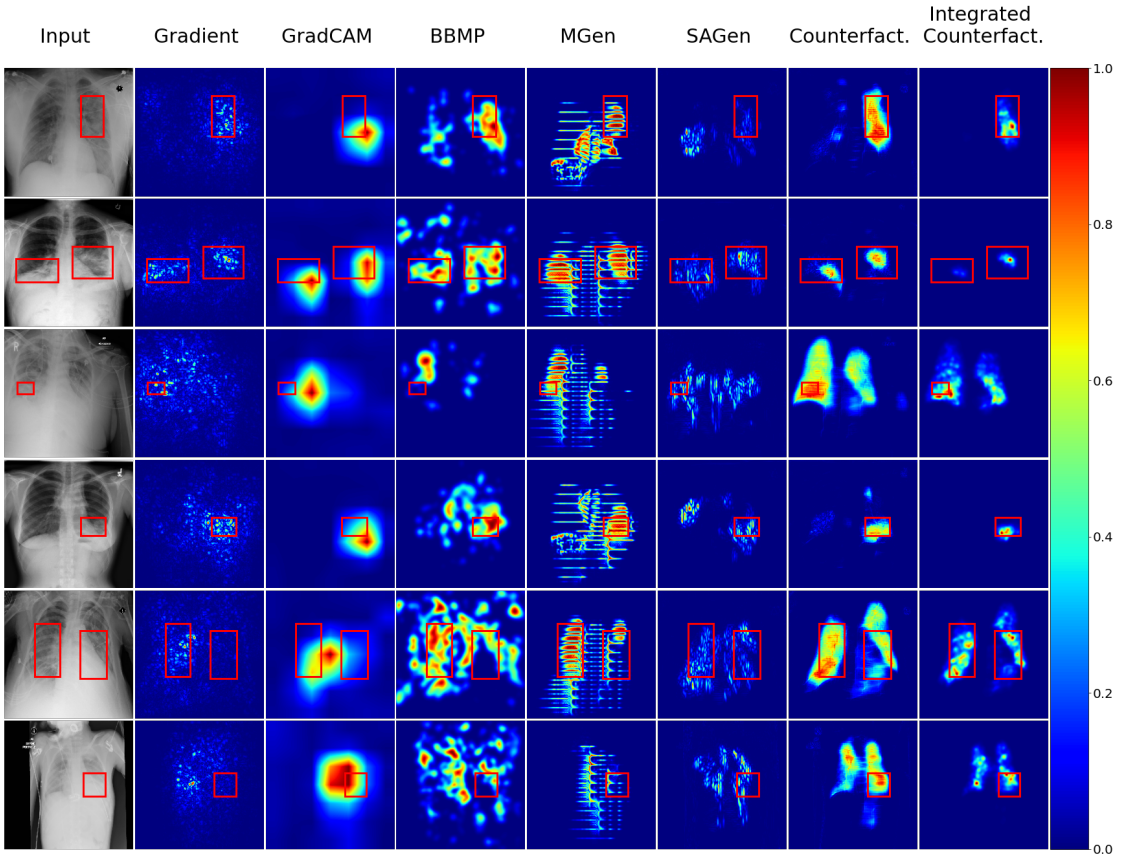


Figure C.36: Pneumonia detection (1) - Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then the attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen 4, a counterfactual explanation (here CyImageCE) 5, and an integrated counterfactual explanation (here from CyImageCE) 6.

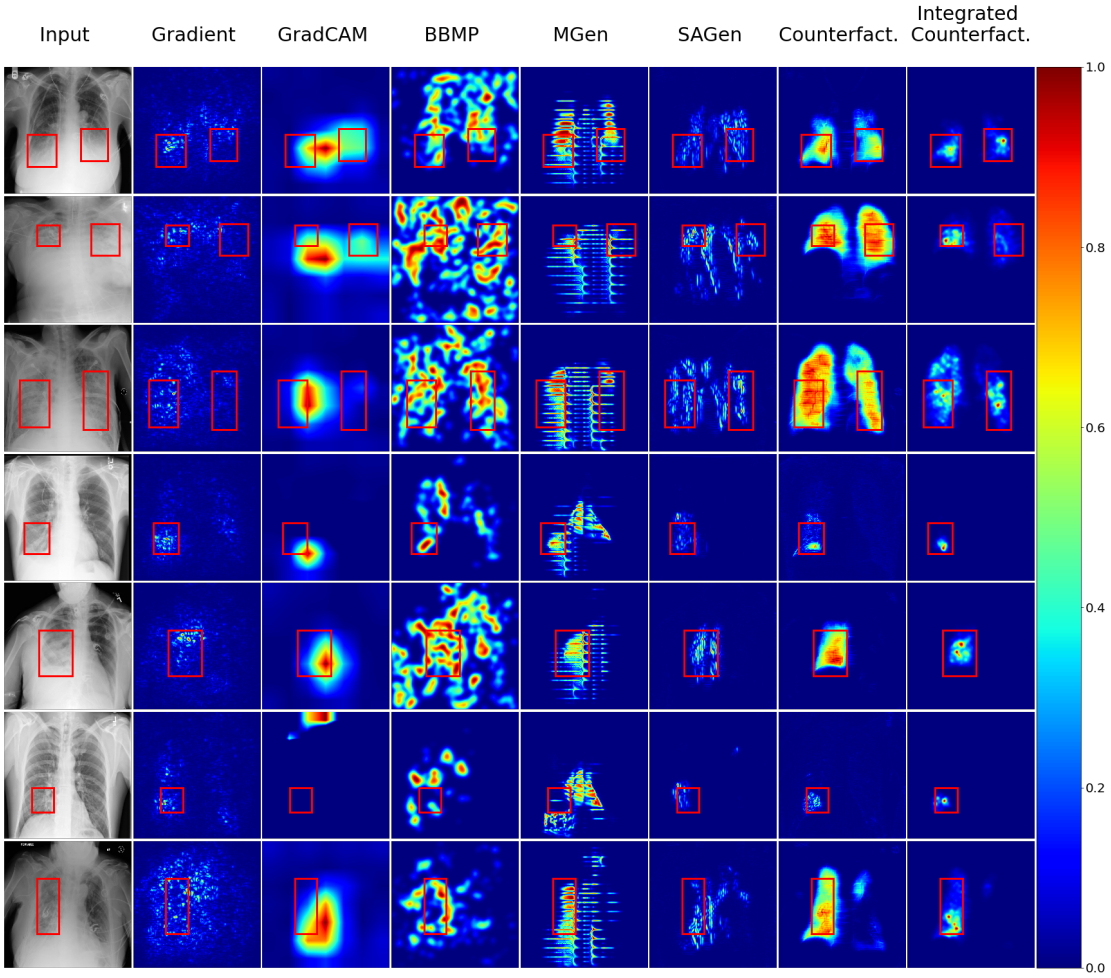


Figure C.37: Pneumonia detection (2) - Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then the attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen 4, a counterfactual explanation (here CyImageCE) 5, and an integrated counterfactual explanation (here from CyImageCE) 6.

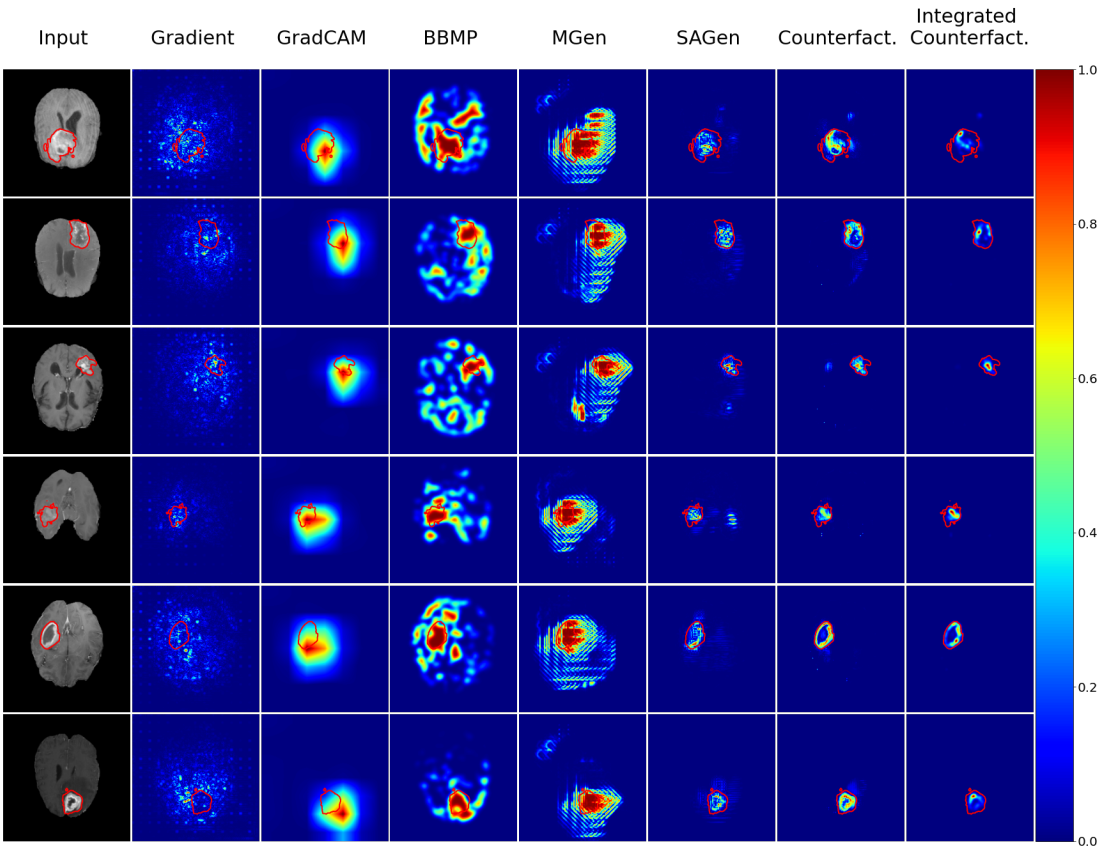


Figure C.38: Brain tumor detection (1) - Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then the attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen 4, a counterfactual explanation (here CyImageCE) 5, and an integrated counterfactual explanation (here from CyImageCE) 6.

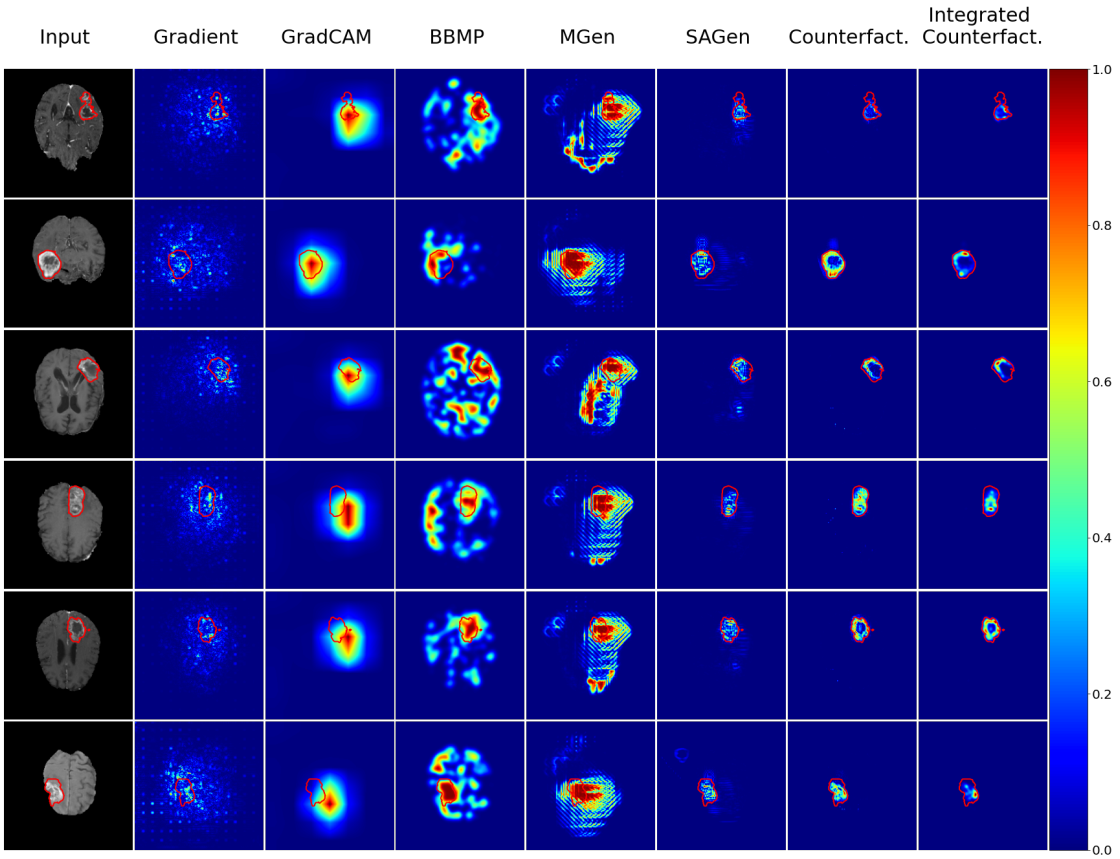


Figure C.39: Brain tumor detection (2) - Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then the attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen 4, a counterfactual explanation (here CyImageCE) 5, and an integrated counterfactual explanation (here from CyImageCE) 6.

C.3.2 Thresholded binary maps

Here, we compare binary explanation maps computed via a thresholding strategy.

- On pneumonia detection the attribution maps are thresholded at the 95th percentile: Figures C.40 and C.41
- On brain tumor detection the attribution maps are thresholded at the 98th percentile: Figures C.42 and C.43

These visualizations support both previous quantitative and qualitative findings.

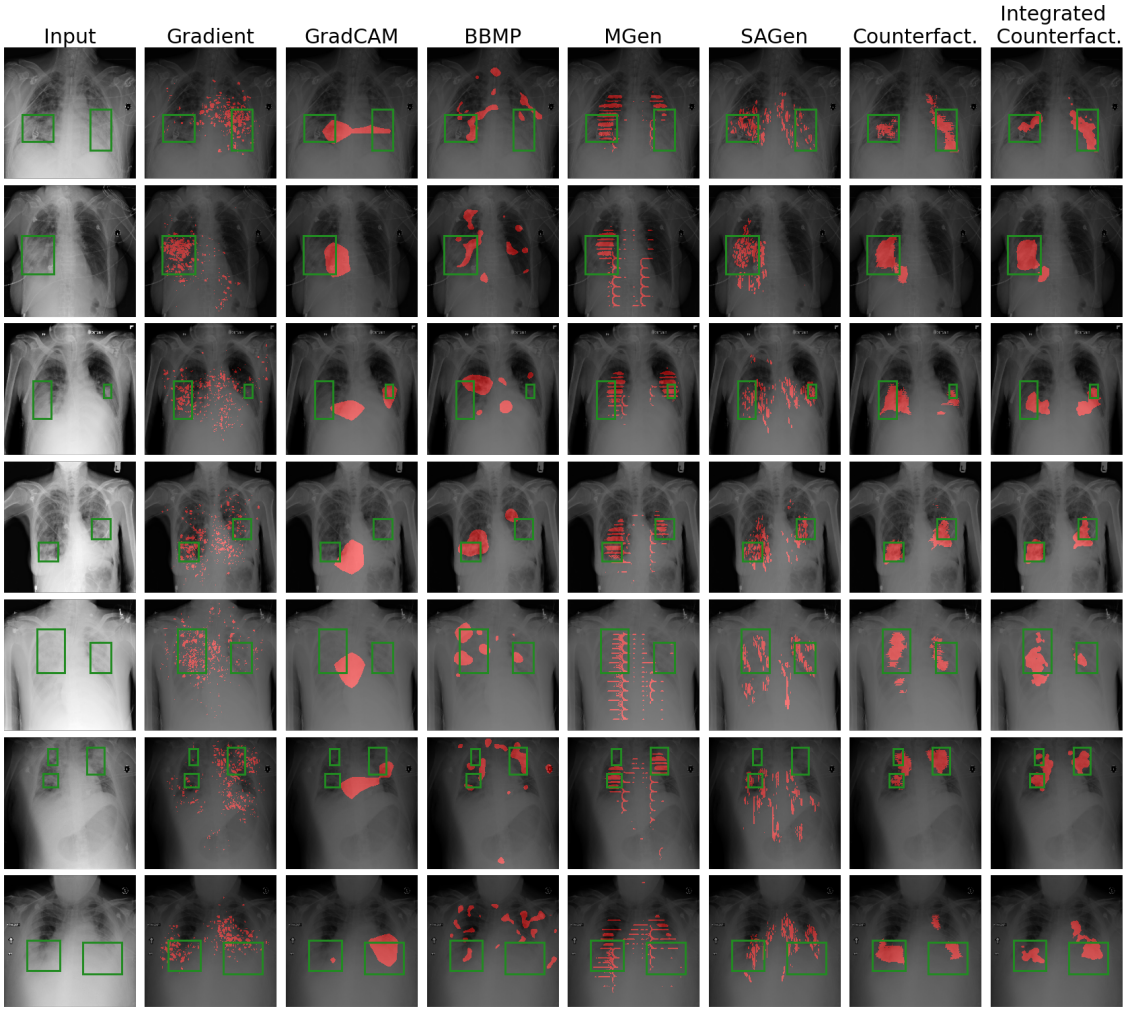


Figure C.40: Pneumonia detection (1) - Binary explanation maps: Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then binary attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen, a counterfactual explanation (here CyImageCE), and its integrated version. All explanation maps are thresholded at the 95th percentile and binarized.

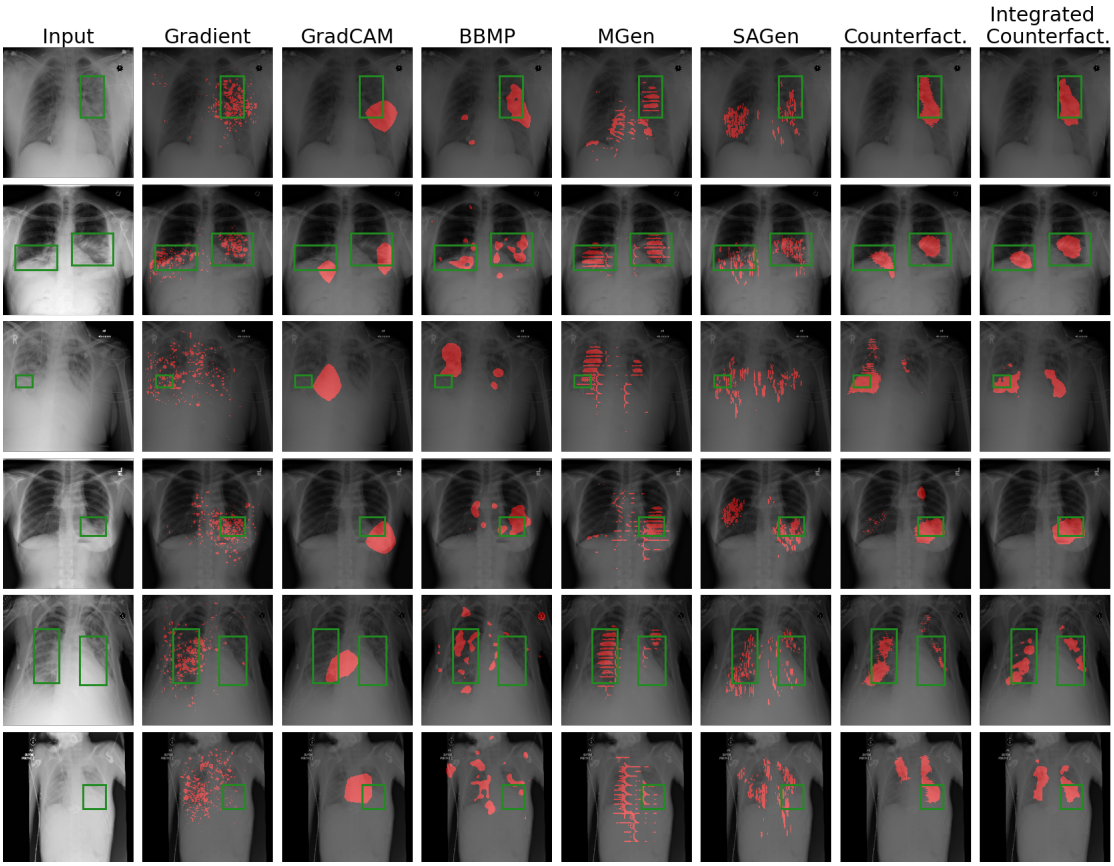


Figure C.41: Pneumonia detection (2) - Binary explanation maps: Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then binary attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen, a counterfactual explanation (here CyImageCE), and its integrated version. All explanation maps are thresholded at the 95th percentile and binarized.

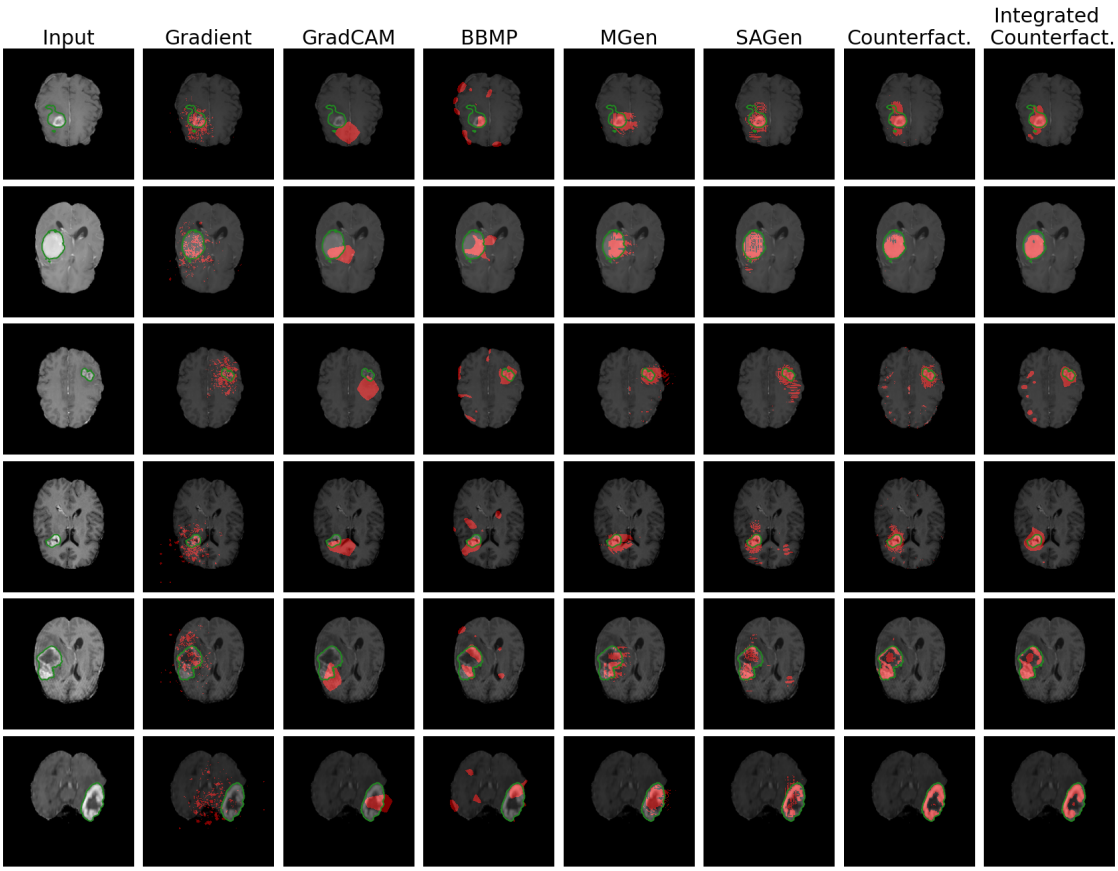


Figure C.42: Brain tumor detection (1) - Binary explanation maps: Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then binary attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen, a counterfactual explanation (here CyImageCE), and its integrated version. All explanation maps are thresholded at the 98th percentile and binarized.

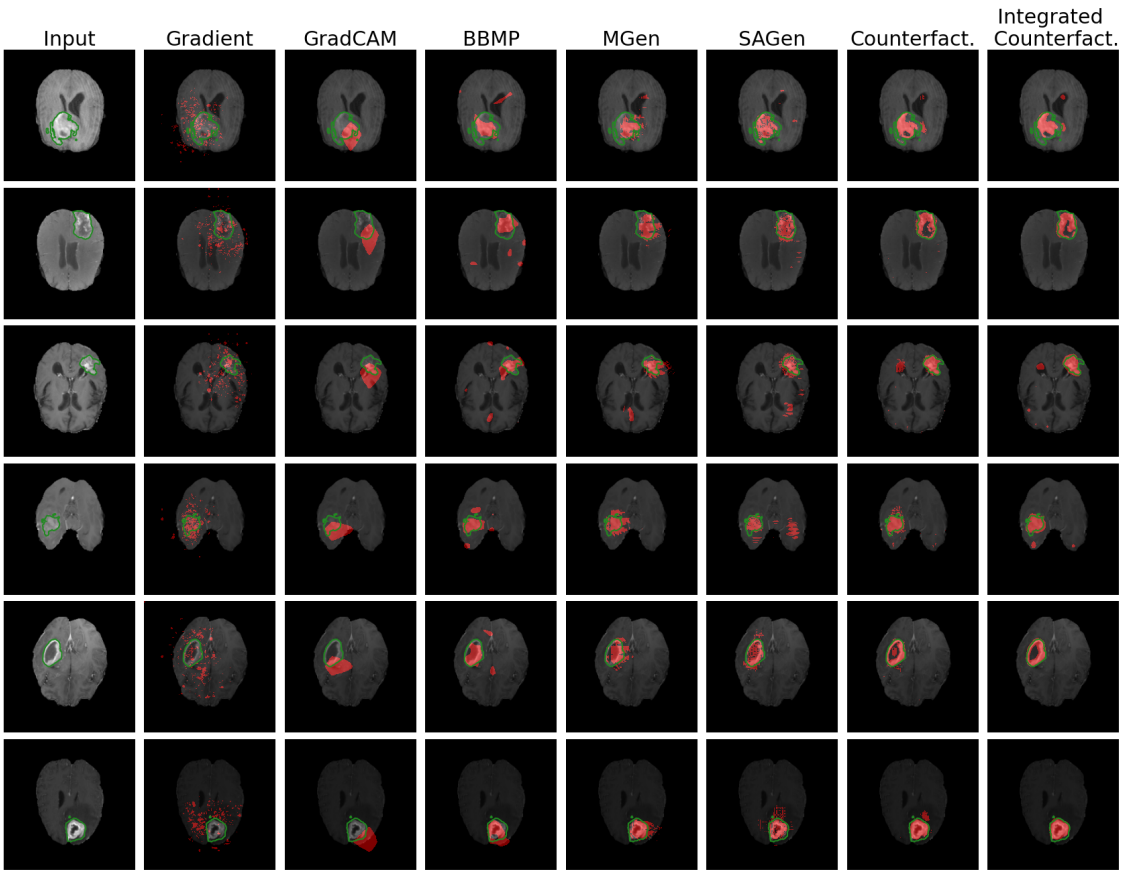


Figure C.43: Brain tumor detection (2) - Binary explanation maps: Comparison with state-of-the-art attribution techniques and against ground truth annotations. From left to right: the input image, then binary attribution maps computed from Gradient, Integrated Gradient (with black reference), GradCAM, RISE, BBMP, MGen, SAGen, a counterfactual explanation (here CyImageCE), and its integrated version. All explanation maps are thresholded at the 98th percentile and binarized.

C.3.3 Localization results for a DenseNet-121 classifier

Table C.5 provides localization results produced by different counterfactual methods (and SAGen) for the DenseNet-121 classifier. Results are comparable to ResNet-50. Our methods outperform state-of-the-art techniques. In addition, without realism (in SAGen) or regularity (in CyCE), localization performances decrease.

Table C.5: Localization results - DenseNet-121. Different attribution methods on Pneumonia detection and Brain tumor problems for the DenseNet121 classifier. IoU (the higher, the better) scores are given at representative percentile values for each problem.

		Pneunmonia detection				Brain tumor detection		
Perc.		90	95	98		98	99	
Gradient		0.159	0.127	0.081	0.267	0.128	0.099	0.261
IG		0.123	0.095	0.074	0.181	0.206	0.168	0.397
GradCAM		0.223	0.174	0.085	0.344	0.220	0.111	0.347
MGen		0.264	0.202	0.105	0.338	0.333	0.284	0.519
SAGen		0.255	0.191	0.107	0.337	0.264	0.222	0.440
SSyGen (SP)	w/o St.	0.191	0.139	0.070	0.284	0.196	0.166	0.364
	w/ St.	0.223	0.168	0.093	0.378	0.181	0.157	0.333
CyCE	w/o St.	0.251	0.205	0.111	0.393	0.264	0.236	0.424
SyCE	w/o St.	0.271	0.221	0.130	0.428	0.350	0.310	0.558
	w/ St.	**0.284**	**0.235**	**0.144**	**0.460**	**0.345**	**0.329**	**0.582**
SySCGen	w/ o St.	0.261	0.203	0.116	0.445	0.347	0.320	0.569
	w/ St.	0.263	0.205	0.117	0.451	0.347	0.320	0.569
CySCGen	w/ o St.	0.267	0.222	0.135	0.414	0.346	0.295	0.540
	w/ St.	0.272	0.228	0.141	0.461	0.348	0.298	0.543

C.4 Adversarial explanation and test time augmentations

In our paper [Charachon 20], we proposed the adversarial explanation approach SAGen. Localization results reveal that SAGen is competitive with the best performers from the state-of-the-art. However, the method sometimes adds adversarial patterns to the input that do not necessarily highlight important regions for the model. As the visual explanation is defined as the difference between two generated images, we suggest regularizing the output of our explanation method by averaging all outputs on random geometrical transformations of the input image. Thus, discriminative regions against reconstruction errors are further enforced, and the attack better focuses on the most relevant regions. This average reads:

$$\overline{E}_f(x) = \frac{1}{N+1}\left[E_f(x) + \sum_{i=1}^{N} \psi_i^{-1}\left(E_f(\psi_i(x))\right)\right] \tag{C.1}$$

Where ψ_i are random geometric transformations such as rotations, translations, zoom, or axis flip. This particular regularization can be applied to all other visual explanation techniques.

First, Tables C.6 and C.7 respectively show similarity and localization metrics for different adversarial generation optimization. We notice that using a shared architecture (Single SAGen) allows generating a stable image closer to the adversarial one, i.e., reducing reconstruction errors when taking their differences. Then, using both L_{reg} (i.e. total variation term) and L_g^w (i.e. penalization of the distance between generators weights) improve localization performances. Tables C.7 (bottom) and C.8 demonstrate the impact of using geometric augmentations at test time: improving localization results for all attribution methods.

Table C.6: Similarity metrics between generated and input images. We show SSIM and PSNR metrics for different architectures and regularization on AGen and SAGen. The column Conv. Out indicates the number of convolution layers used before the output convolution in each Single SAGen heads (see Table 7.3c). Results are provided for pneumonia detection.

Method				$x \leftrightarrow x_s$		$x \leftrightarrow x_a$		$x_s \leftrightarrow x_a$	
Architecture	Conv. Out	L_{reg}	L_g^w	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
AGen	-	✓	-	-	-	0.994	41.92	-	-
Duo SAGen	-	✓	-	0.996	44.07	0.987	39.47	0.994	43.89
	-	✓	✓	0.995	41.99	0.987	39.08	0.995	44.26
Single SAGen	1	✓	-	**0.997**	**44.57**	0.989	40.67	0.996	45.25
	1	-	✓	0.994	42.73	0.993	41.85	0.999	52.59
	1	✓	✓	0.992	41.79	0.991	41.35	**0.999**	**54.55**
	2	✓	-	0.995	43.61	0.994	42.42	0.999	52.26
	2	✓	✓	0.995	43.88	**0.994**	**42.63**	0.999	51.93

Table C.7: Localization performance - Comparison between SAGen. IoU (the higher, the better) scores are given at representative percentile values. Comparison across the different generator architectures, using regularization losses or not, and with or without augmentation at test time. The column Conv. Out indicates the number of convolution layers used before the output convolution in each Single SAGen heads (see Table 7.3c). Results are provided for pneumonia detection.

Method				IoU		
Architecture	Conv. Out	L_{reg}	L_g^w	90	95	98
AGen	-	✓	-	0.158	0.118	0.064
Duo SAGen	-	✓	-	0.164	0.122	0.070
	-	✓	✓	0.170	0.132	0.079
Single SAGen	1	✓	-	0.166	0.127	0.075
	1	-	✓	0.204	0.157	0.090
	1	✓	✓	0.220	0.171	***0.099***
	2	✓	-	0.229	0.172	0.095
	2	✓	✓	***0.232***	***0.173***	0.097
With Augmentations						
AGen	-	✓	-	0.208	0.156	0.087
Duo SAGen	-	✓	-	0.206	0.156	0.085
	-	✓	✓	0.227	0.166	0.093
Single SAGen	1	✓	-	0.218	0.156	0.086
	1	-	✓	0.233	0.181	0.105
	1	✓	✓	0.240	0.188	0.112
	2	✓	-	0.268	0.204	**0.115**
	2	✓	✓	**0.272**	**0.206**	**0.115**

Table C.8: Localization results - Augmentations at test time - Comparison with state-of-the-art. IoU (the higher, the better) scores are given at representative percentile values. Comparison between methods without (**Top**) and with (**Bottom**) augmentations. Results are provided for pneumonia detection.

Method	IoU				
Percentile	80	85	90	95	98
Gradient	0.203	0.199	0.187	0.152	0.097
	0.256	***0.252***	***0.236***	***0.190***	***0.117***
GradCAM	0.237	0.225	0.195	0.138	0.070
	0.271	***0.263***	***0.244***	***0.190***	***0.105***
Mask Generator	0.222	0.219	0.208	0.169	0.103
	0.259	***0.264***	***0.259***	**0.221**	**0.137**
Adv. *AE* (TV)	0.177	0.173	0.158	0.118	0.064
	0.239	***0.230***	***0.208***	***0.156***	***0.087***
Single AE_2 ***(W, TV)***	0.248	0.250	0.232	0.173	0.097
	0.292	**0.292**	**0.272**	***0.206***	***0.115***

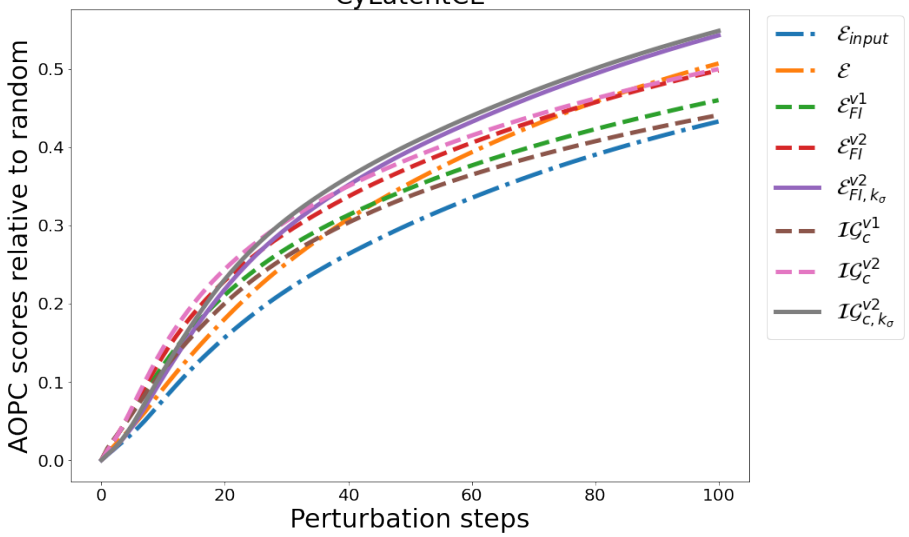
D

Appendix: Feature importance evaluations

D.1 Comparison between integration techniques

In Figures D.1a and D.1b, we display AOPC curves (for CyLatentCE) of the different variations of both the integrated counterfactual explanations and the counterfactual integrated gradient. We observe that similar results are obtained for corresponding attributions e.g. $\mathcal{E}^{v2}_{FI,k_\sigma}$ vs $\mathcal{IG}^{v2}_{c,k_\sigma}$. In addition, all integrated techniques are comparable to (or outperform) the counterfactual baseline.

(a) Pneumonia detection - X-Rays



(b) Brain tumor detection - MRI

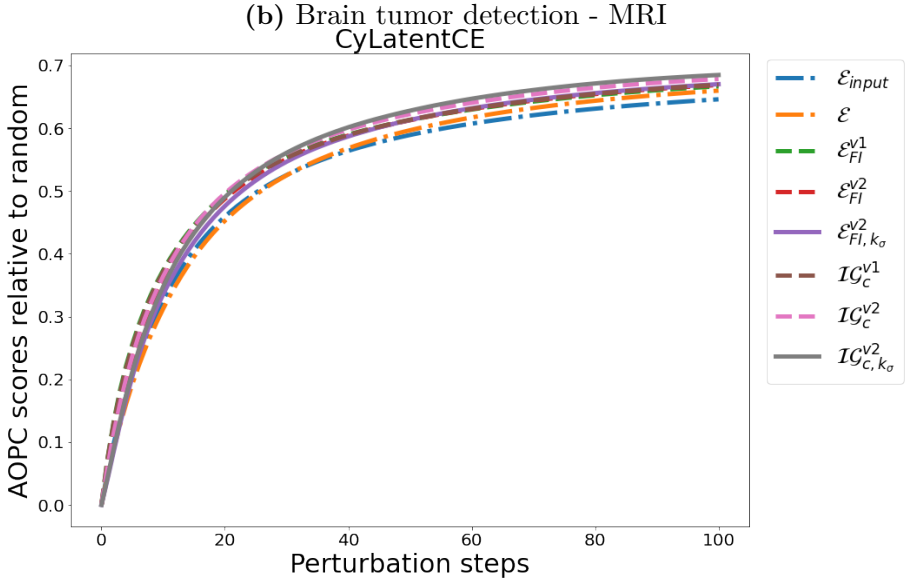


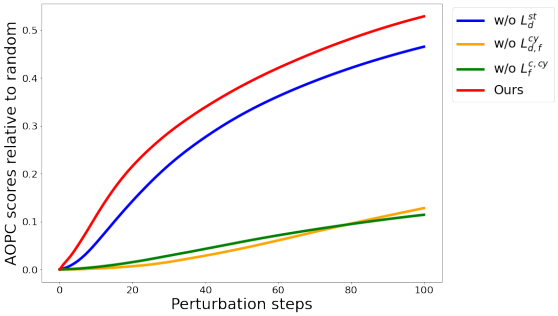
Figure D.1: AOPC scores relative to random baseline - Comparison between counterfactual integration techniques. (a) Results for the pneumonia detection problem and (b) the brain tumor detection problem. Results are given for CyLatentCE. AOPC curves (relative to random) are displayed for the baseline counterfactual explanation $\mathcal{E}$ computed against the input ($\mathcal{E}_{input}$ in blue) or the stable ($\mathcal{E}$ in orange) image, the two integrated techniques $\mathcal{E}_{FI}^{v1}$ (green) and $\mathcal{E}_{FI}^{v2}$ (red); the regularized integrated method $\mathcal{E}_{FI,k_\sigma}^{v2}$ (purple); the two counterfactual integrated versions $\mathcal{IG}_c^{v1}$ (brown) and $\mathcal{IG}_c^{v2}$ (pink); and the regularized counterfactual integrated gradient $\mathcal{IG}_{c,k_\sigma}^{v2}$ (gray).

D.2 Ablation study: CyImageCE

Here we provide additional feature importance results on the ablation study introduced in Section 8.1.2, and described in Section 8.2.1 (i.e. results on feature importance).

- Figures D.2a and D.2b display AOPC curves on both pneumonia and brain tumor detection for CyImageCE when removing the different terms of loss.
- In each case, our proposed optimization better translates the importance of input features (red lines). Compared with CyLatentCE, removing L_d^{st} retains satisfying feature importance results (so it has a smaller impact).

(a) Pneumonia detection - X-Rays



(b) Brain tumor detection - MRI

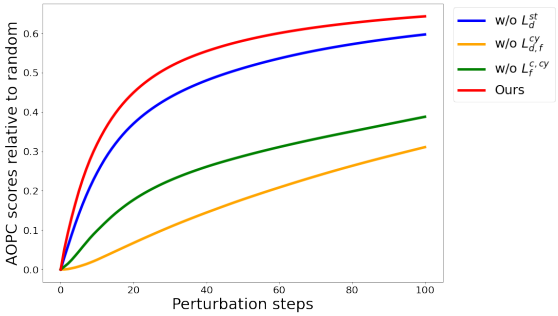


Figure D.2: AOPC scores relative to random baseline - Ablation study for CyImageCE. (a) Results for the pneumonia detection problem and (b) the brain tumor detection problem. Results for different CyImageCE optimizations are compared: optimization without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}; and our CyImageCE optimization.

D.3 Feature importance assessed against Gaussian blur perturbation

This section provides some AOPC results when using Gaussian blur to perturb the input.

- Figures D.3 and D.4 show AOPC curves for pneumonia and brain tumor detection problems respectively. For each case, we display (a) the global AOPC scores, gathering input predicted as healthy or pathological; (b) the AOPC scores for input predicted as pathological because important features are more localized and precise in this case (for our medical detection problems).
- Blur perturbation is not suited for the pneumonia problem as suggested in Figure D.3b, where the perturbated features have a very small impact on the classification prediction (see the AOPC scales) or even a poorer impact compared to a random attribution.
- In the other case, our methods are at least competitive with state-of-the-art approaches. BBMP optimized against a Gaussian blur perturbation produces the best scores.

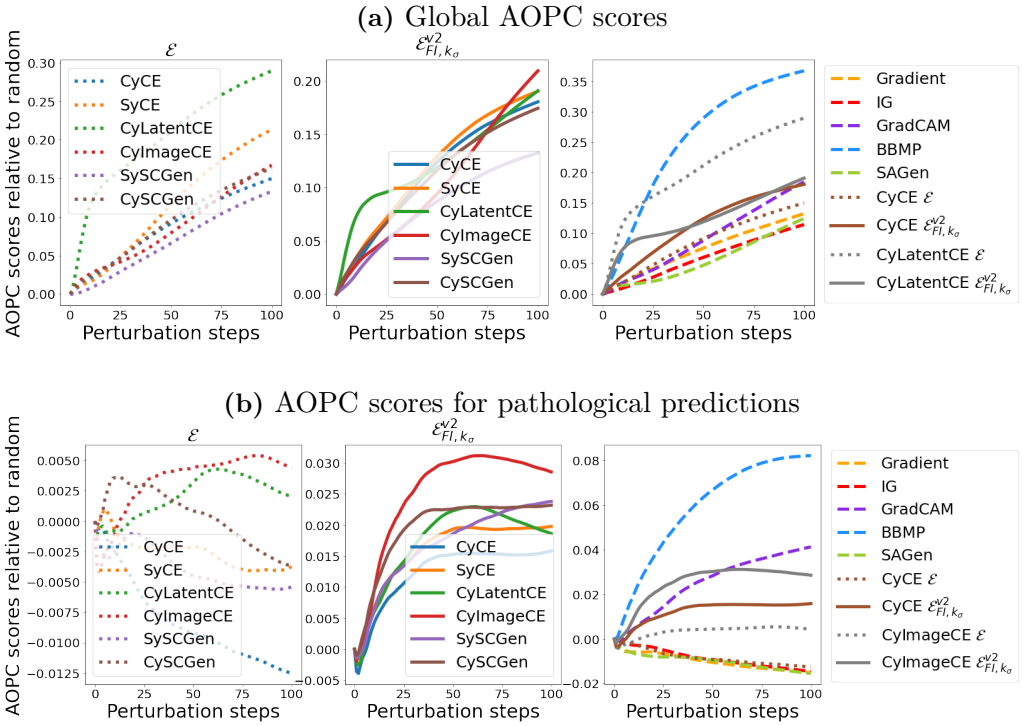


Figure D.3: Pneumonia detection - AOPC scores relative to random baseline for a Gaussian blur perturbation. (a) Results for all images, and (b) Results for images predicted as pathological. From left to right: the comparison of AOPC scores between baseline counterfactual methods $\mathcal{E}$; the comparison of AOPC scores between the regularized integrated methods $\mathcal{E}^{v2}_{FI,k_\sigma}$; and the comparison with state-of-the-art techniques as well as SAGen. In the last column, only the poorest and the best performers from the counterfactual methods are displayed. The AOPC scores are relative to random.

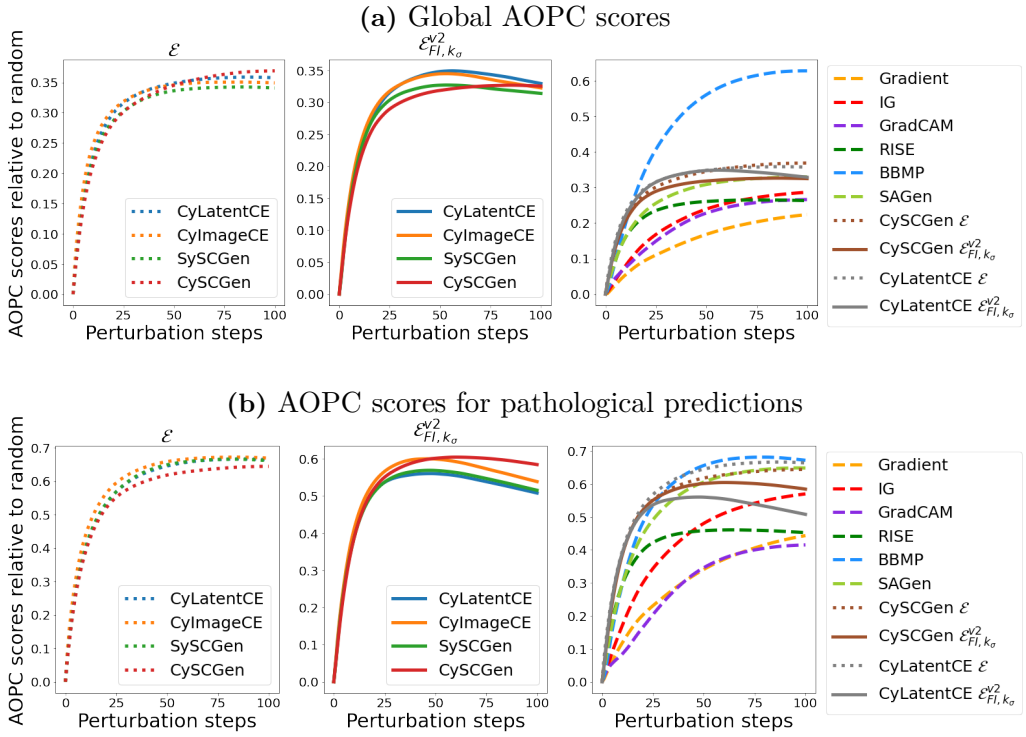


Figure D.4: Brain tumor detection - AOPC scores relative to random baseline for a Gaussian blur perturbation. (a) Results for all images, and (b) Results for images predicted as pathological. From left to right: the comparison of AOPC scores between baseline counterfactual methods $\mathcal{E}$; the comparison of AOPC scores between the regularized integrated methods $\mathcal{E}^{v2}_{FI,k_\sigma}$; and the comparison with state-of-the-art techniques as well as SAGen. In the last column, only the poorest and the best performers from the counterfactual methods are displayed. The AOPC scores are relative to random.

E

Appendix: Domain translation results

E.1 Counterfactual techniques

E.1.1 Comparison between counterfactual generations

We provide additional figures that compare counterfactual generations of the different counterfactual approaches for:

- Pneumonia detection on X-rays: in Figures E.1 and E.2 when translating inputs predicted as pathological to healthy cases; and in Figure E.6 for the opposite translation.
- Brain tumor detection on MRI slices: in Figures E.4 and E.5 for pathological to healthy translation; and in Figure E.7 for healthy to pathological translation.

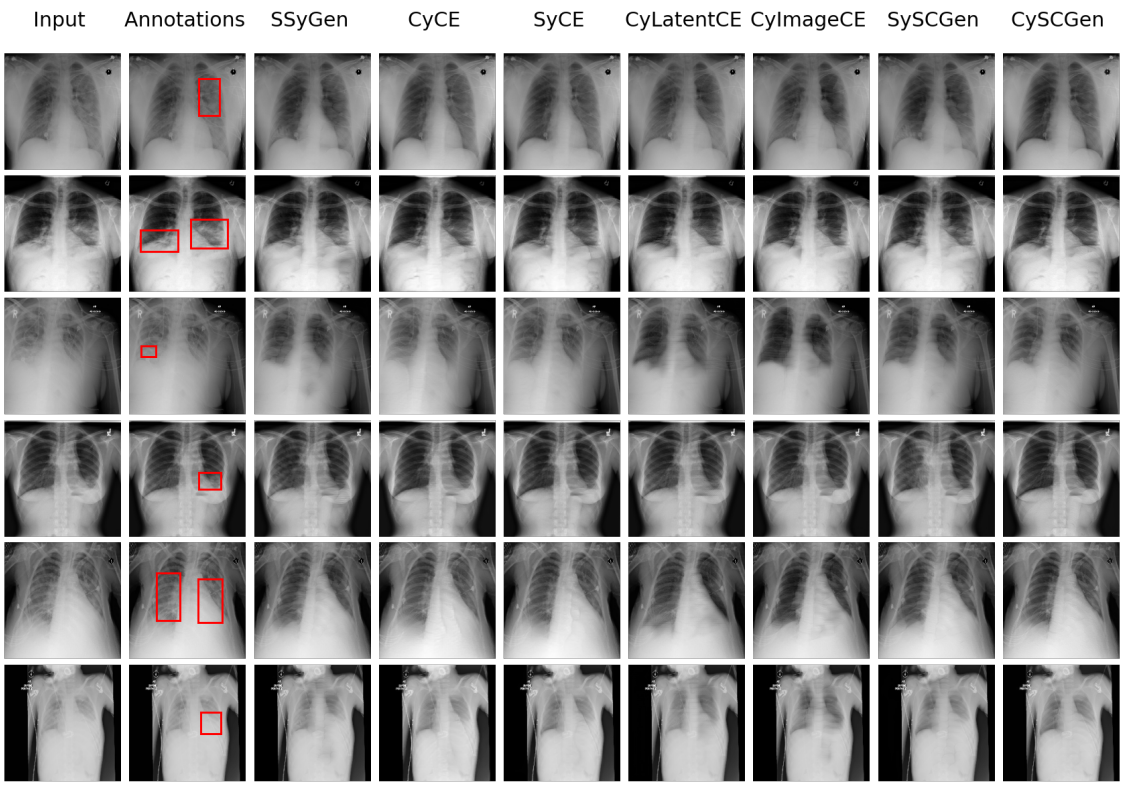


Figure E.1: Pneumonia detection (1) - Comparison between counterfactual generation techniques: From pathological to healthy image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

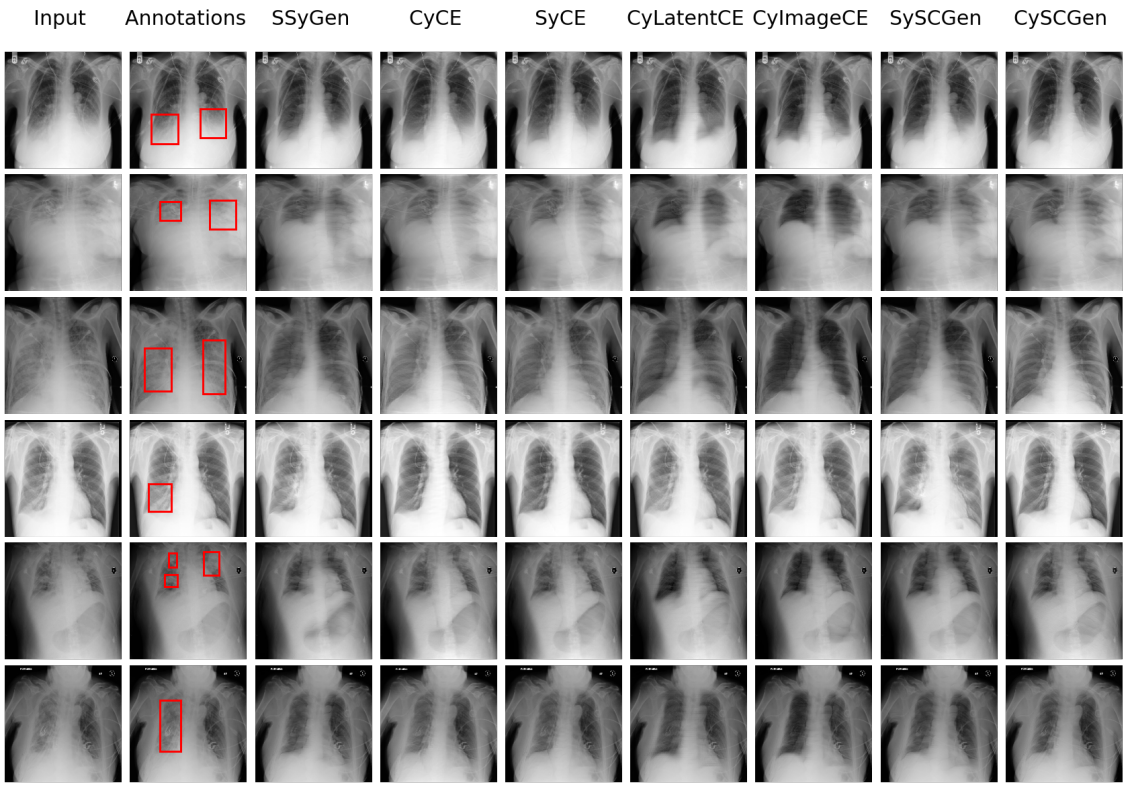


Figure E.2: Pneumonia detection (2) - Comparison between counterfactual generation techniques: From pathological to healthy image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

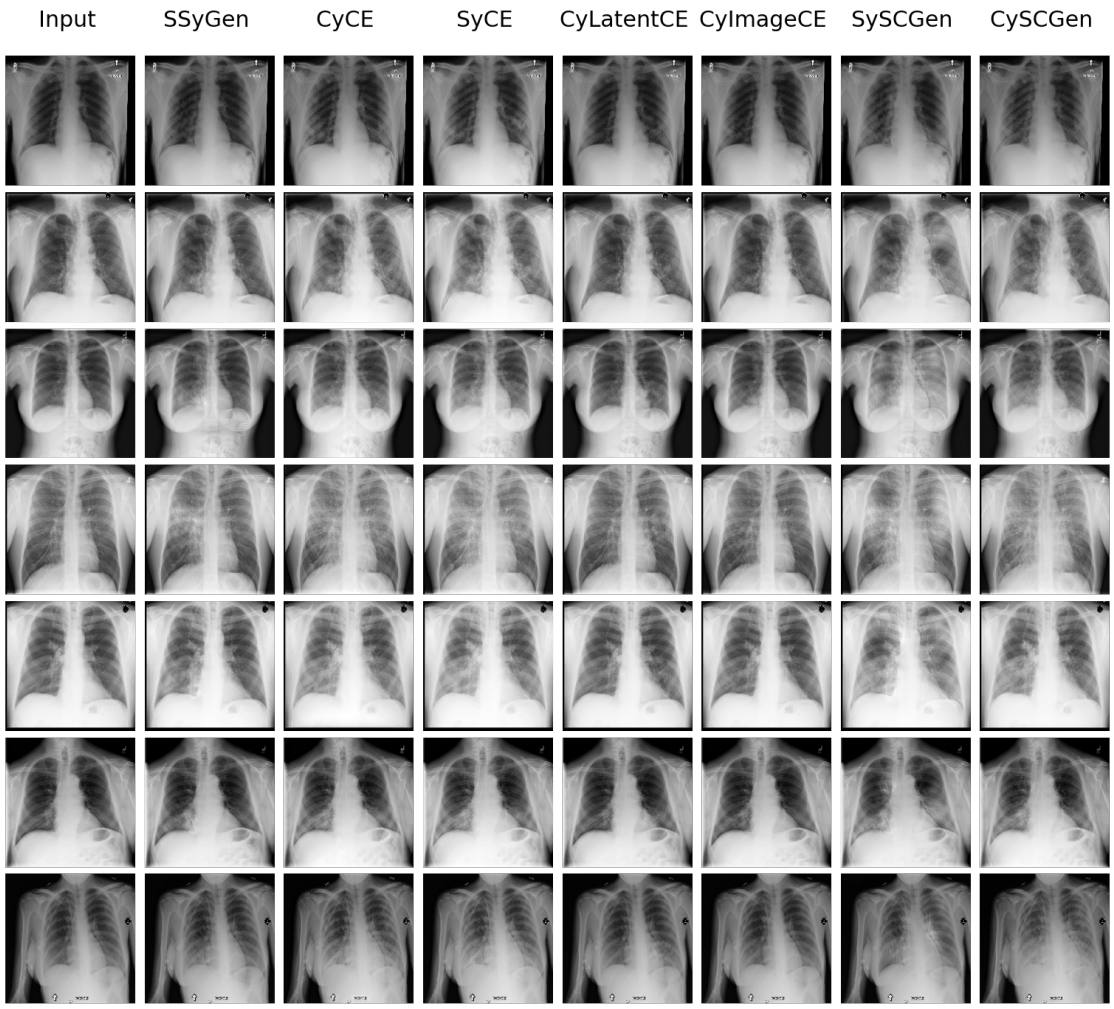


Figure E.3: Pneumonia detection - Comparison between counterfactual generation techniques: From healthy to pathological image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

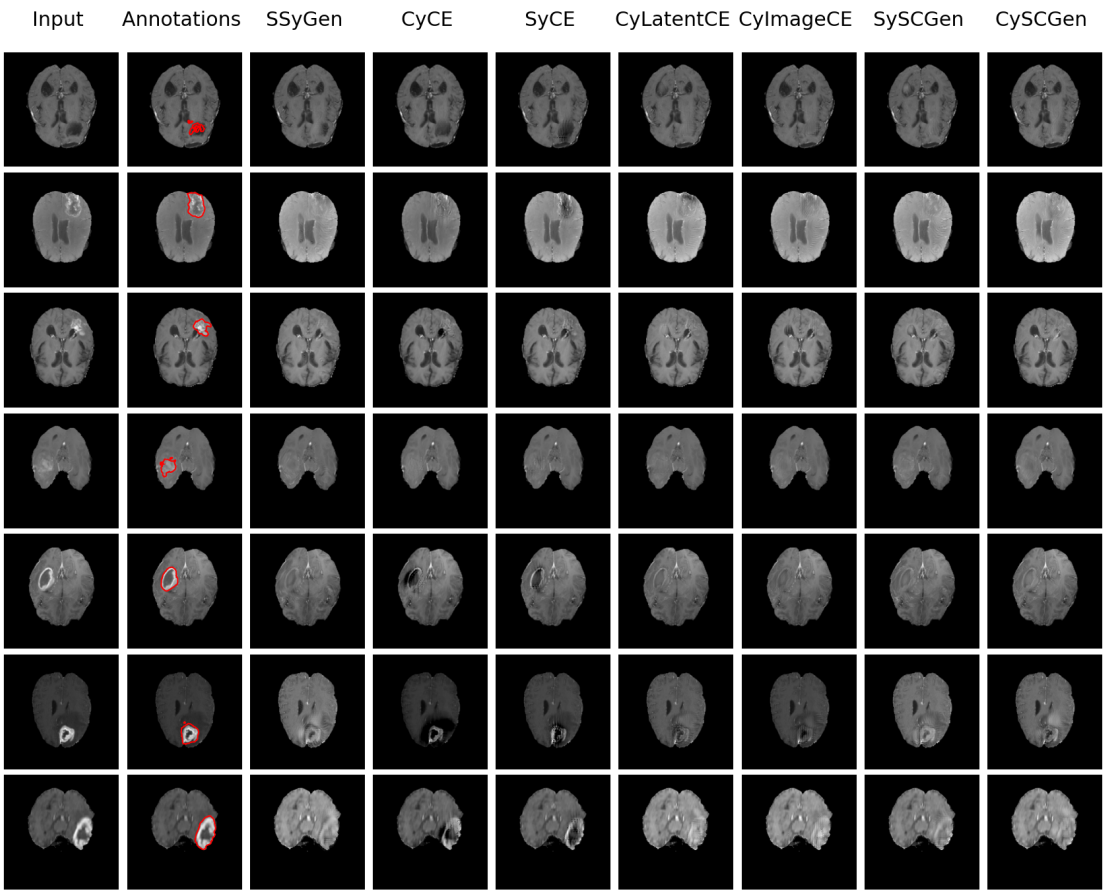


Figure E.4: Brain tumor detection (1) - Comparison between counterfactual generation techniques: From pathological to healthy image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

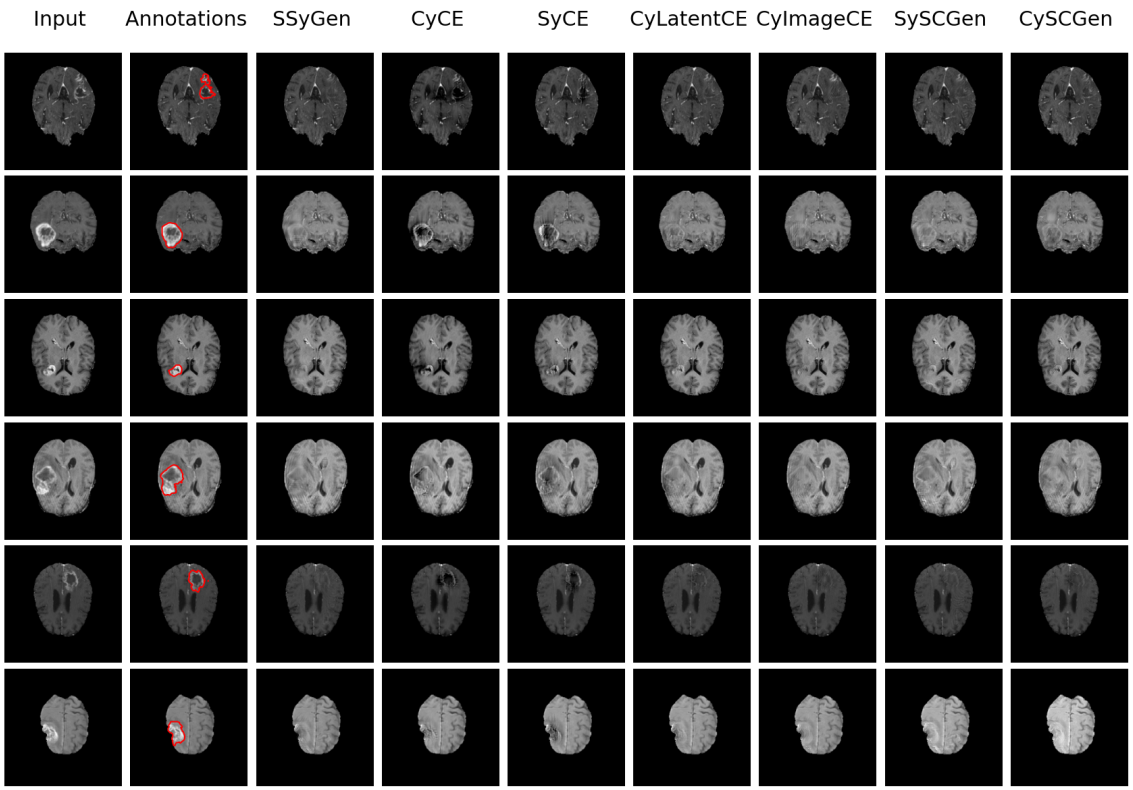


Figure E.5: Brain tumor detection (2) - Comparison between counterfactual generation techniques: From pathological to healthy image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

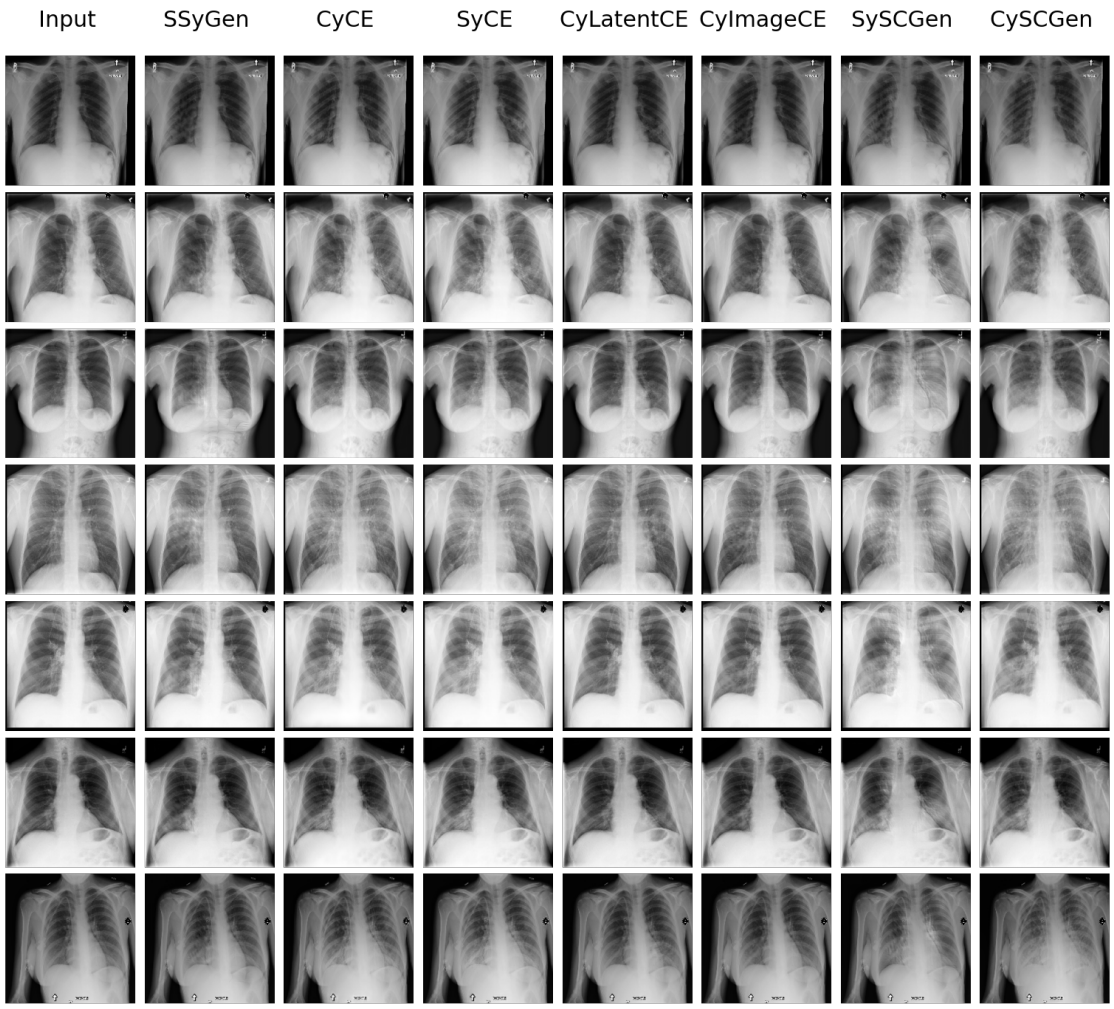


Figure E.6: Pneumonia detection - Comparison between counterfactual generation techniques: From healthy to pathological image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

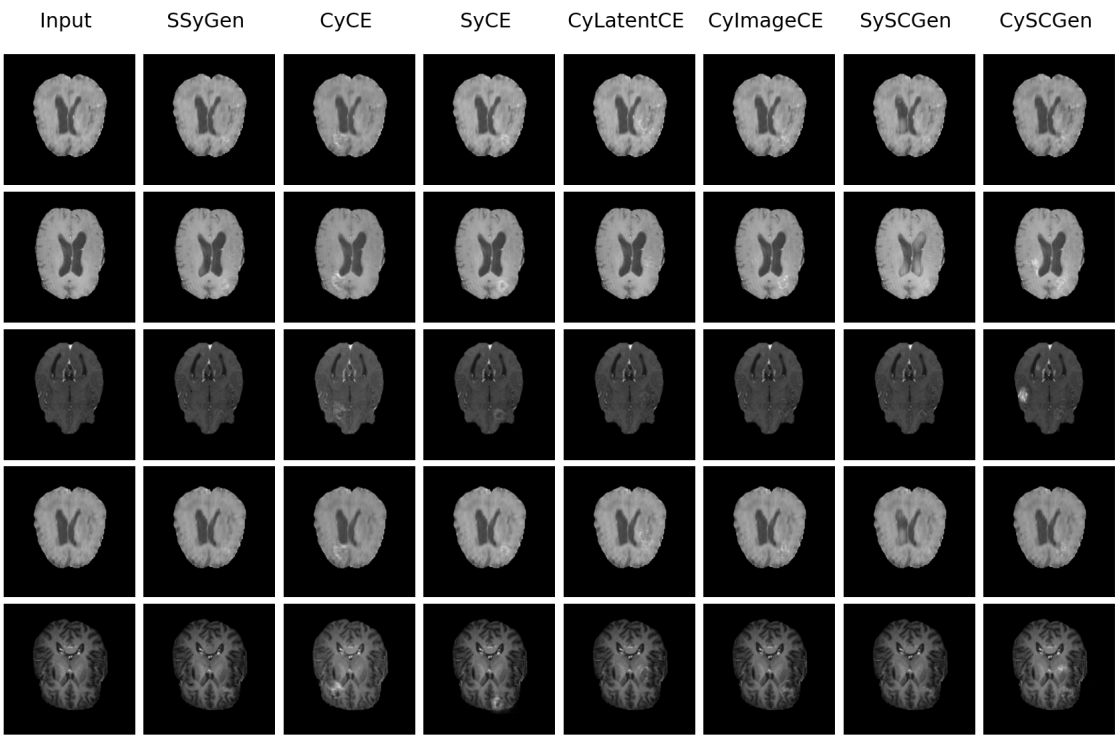


Figure E.7: Brain tumor detection - Comparison between counterfactual generation techniques: From healthy to pathological image. The first two columns: the input image and the annotated input image. From columns 3 to 9: the different counterfactual generations produced by the techniques described in chapter 5. Dual path optimization is used for all the counterfactual methods.

E.1.2 Comparison of generator architectures

Here, we provide qualitative and quantitative domain translation results for different architectures of the CyLatentCE generator.

- Figures E.8 and E.10 display counterfactual generations (for pathological cases) for pneumonia and brain tumor problems respectively. As for the localization results, the generations are comparable except for the DRIT-like architecture showing a shift in intensity for various regions of the input (not only the important ones). This supports the localization results.
- Figures E.9 and E.11 show the first two axis of the PCA projections (of the VAE latent space) for the two problems respectively and for both (a) pathological and (b) healthy counterfactual generations. In both problems, the DRIT-like architecture produces the best domain translation results (in the two directions). This architecture allows more significant input transformations, which supports findings from the main manuscript. The other architectures show mostly comparable results with differences that are not generalizable, e.g., StyleGAN2 architectures seem to produce better translation from healthy to pathological images on chest x-rays (except for DRIT-like), but not on brain MRI slices.

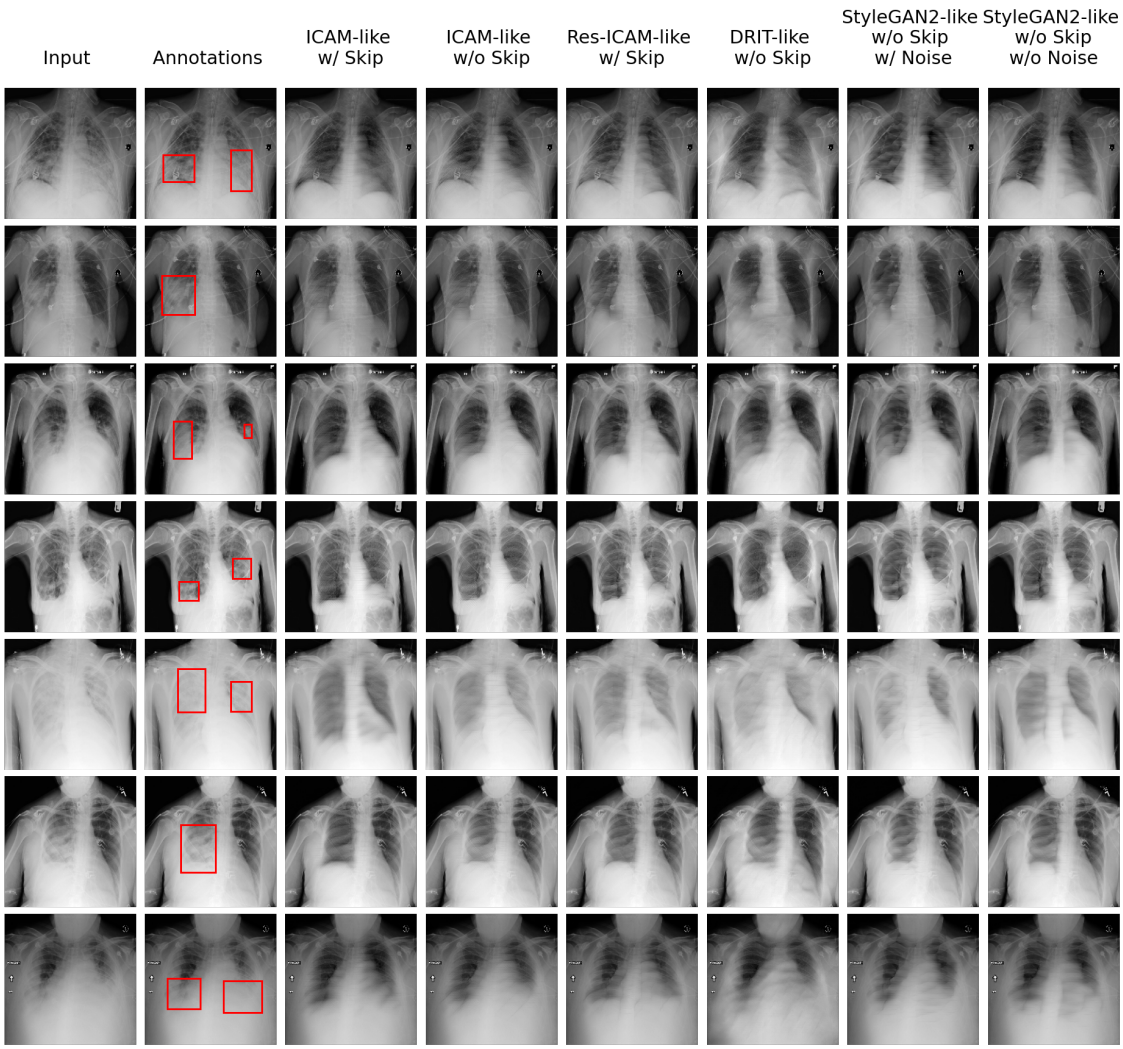
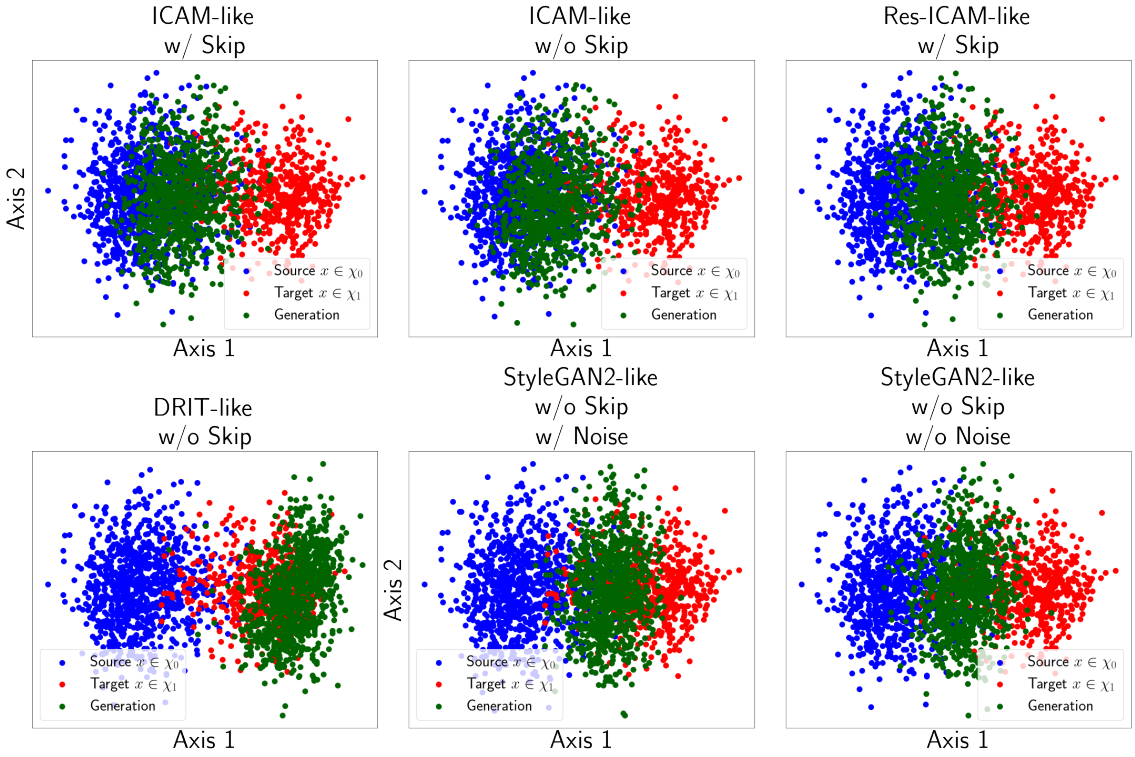
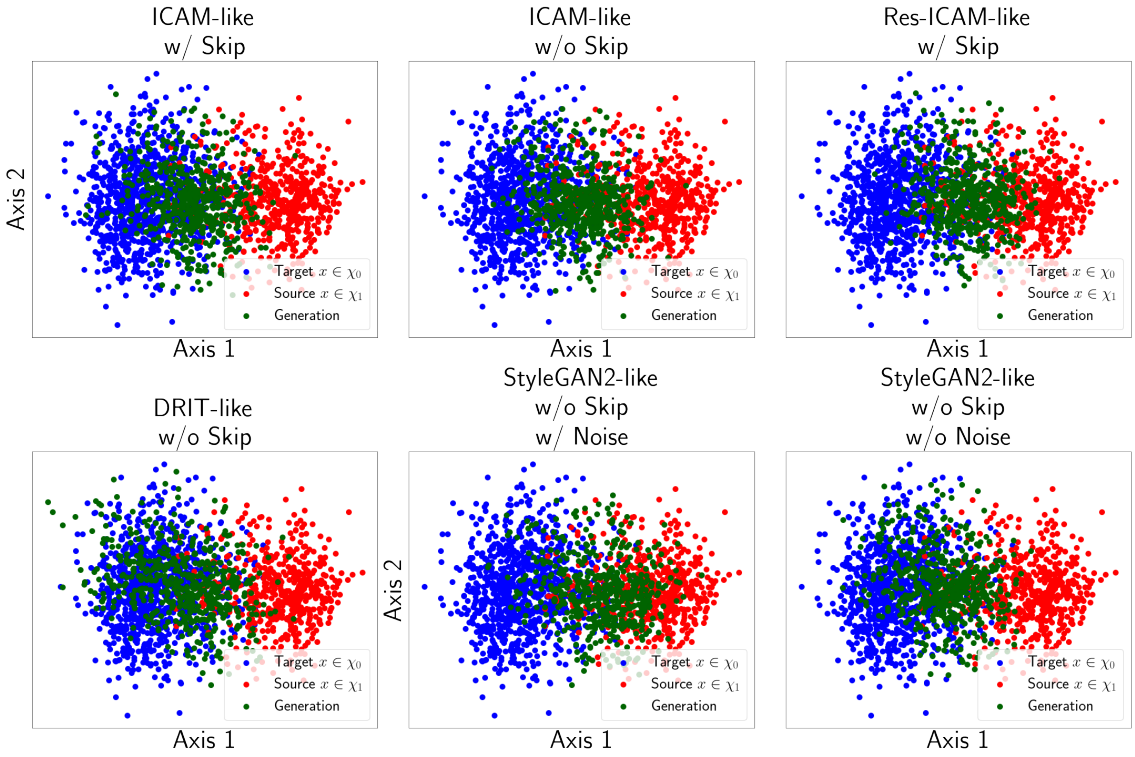


Figure E.8: Pneumonia detection - Comparison between different generator architectures for CyLatentCE. From left to right: the input image; then the counterfactual generations from CyLatentCE for different encoder-decoder architectures: Conditioning (ICAM, DRIT, or StyleGAN2-like); with or without skip connections; with residual blocks in downsampling and upsampling block (Res-) and with or without additive noise in the decoder path.



(a) $\chi_0 \longrightarrow \chi_1$



(b) $\chi_1 \longrightarrow \chi_0$

Figure E.9: Pneumonia detection - Qualitative VAE Results: Architectures comparison for CyLatentCE. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

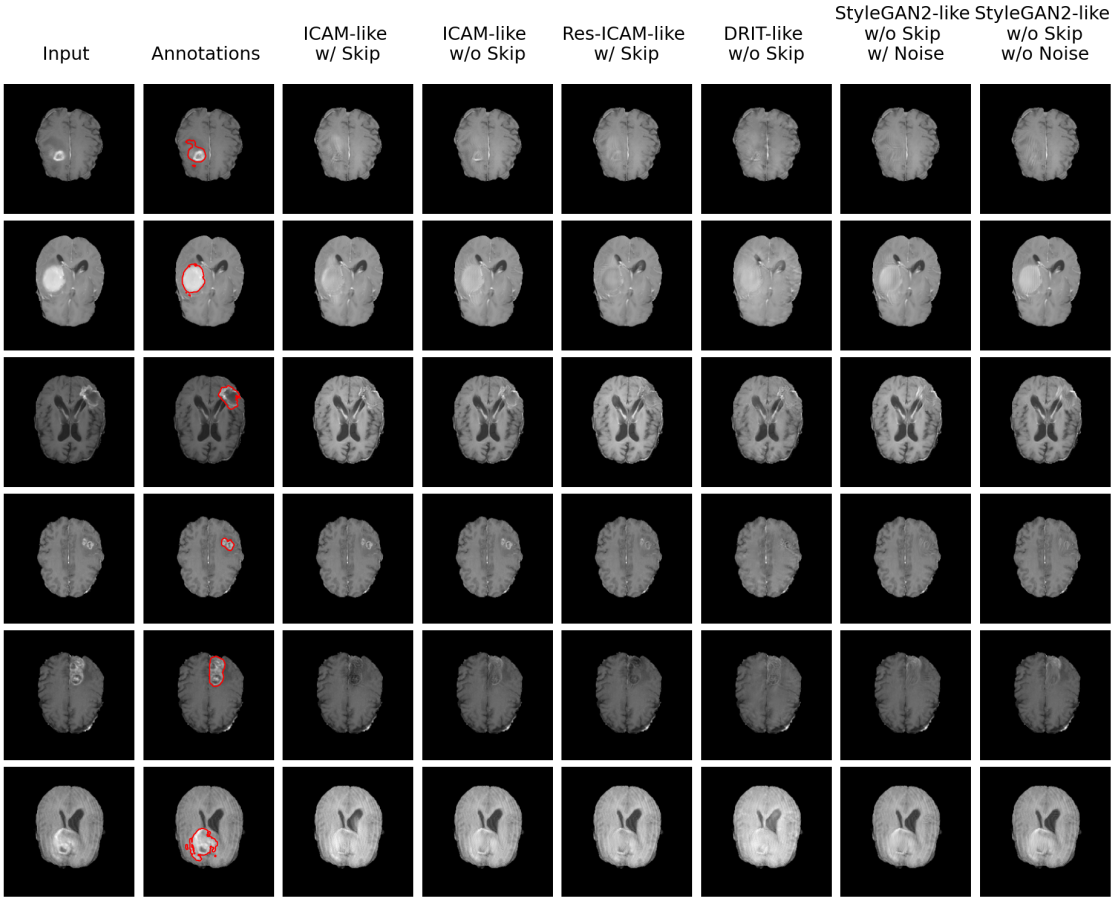
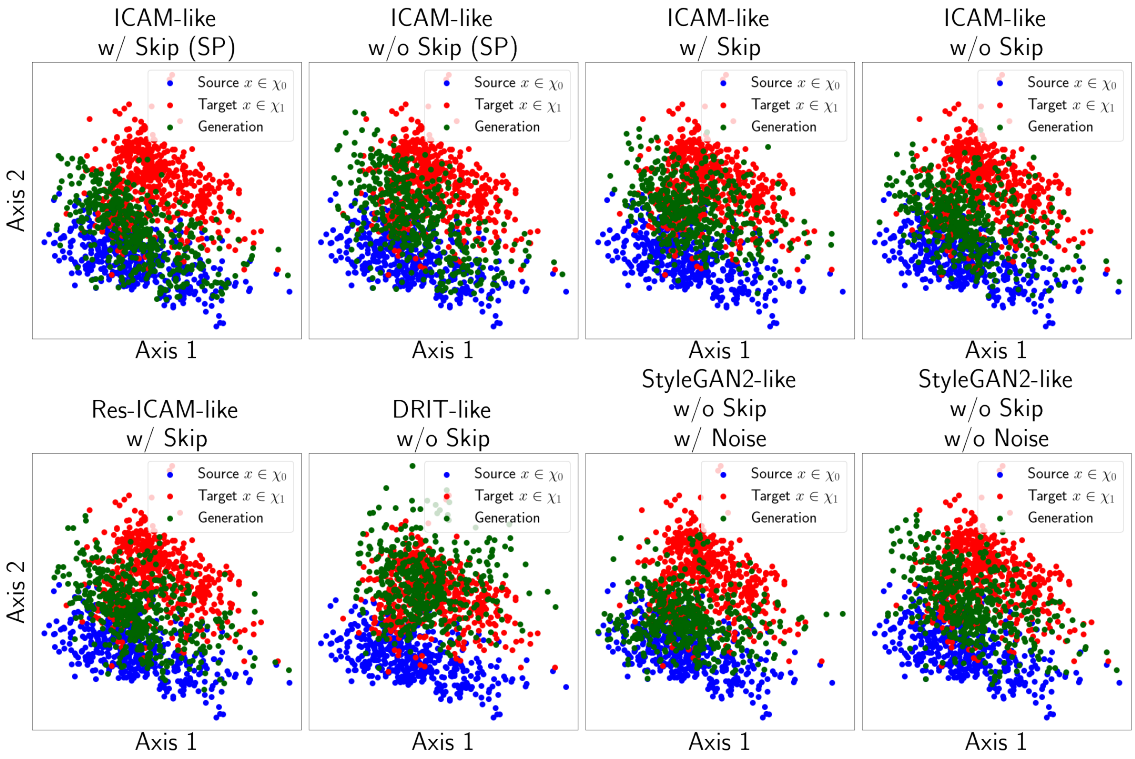
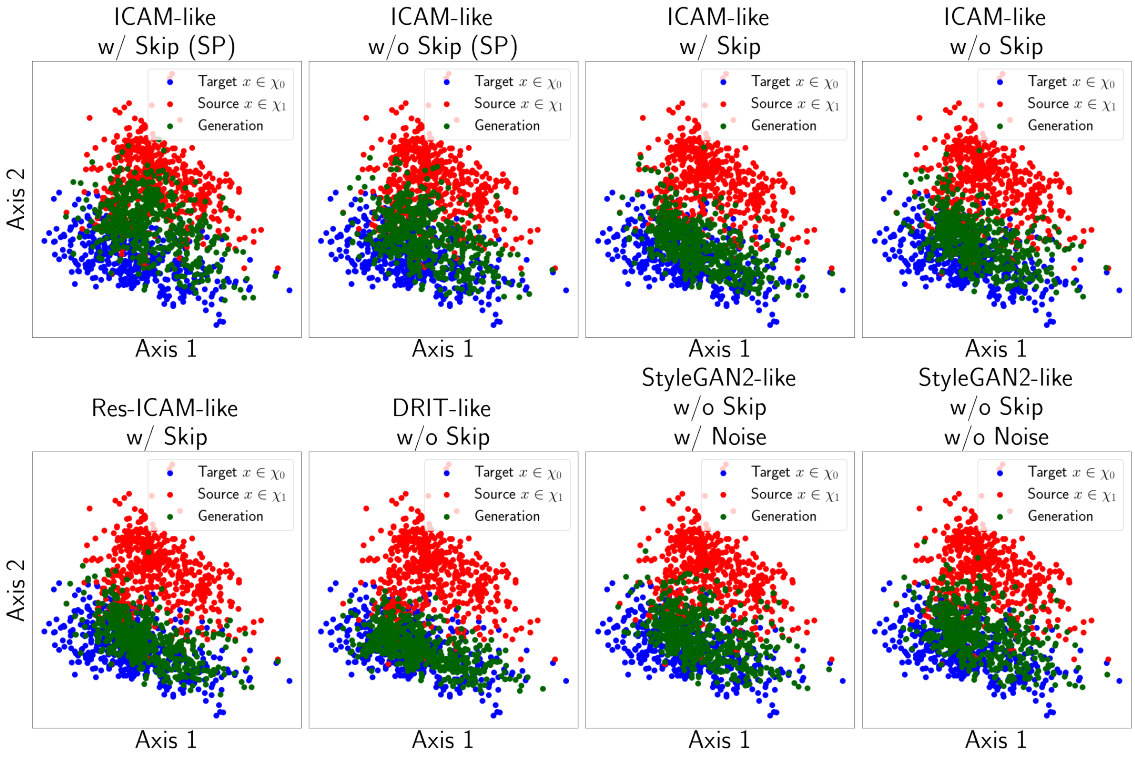


Figure E.10: Brain tumor detection - Comparison between different generator architectures for CyLatentCE. From left to right: the input image; then the counterfactual generations from CyLatentCE for different encoder-decoder architectures: Conditioning (ICAM, DRIT, or StyleGAN2-like); with or without skip connections; with residual blocks in downsampling and upsampling block (Res-) and with or without additive noise in the decoder path.



(a) $\chi_0 \longrightarrow \chi_1$



(b) $\chi_1 \longrightarrow \chi_0$

Figure E.11: Brain tumor detection - Qualitative VAE Results: Architectures comparison for CyLatentCE. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

E.1.3 Ablation study

Here we provide additional qualitative and quantitative domain translation results for the ablation study of CyLatentCE and CyImageCE.

E.1.3.1 Ablation study for CyLatentCE

- Figures E.12 and E.13 show counterfactual generations (from pathological to healthy images) for the different ablation cases and supports the findings from PCA projections (see Figures 8.26 and 8.27).
- Table E.1 displays the classification metrics on both pneumonia and brain tumor detection for CyLatentCE when removing the different terms of loss. Without the terms $L_f^{c,cy}$, the counterfactual generations are not predicted in the target class (by the classifier); the translation w.r.t. the classifier fails.

Table E.1: Classification results - Ablation study CyLatentCE. Accuracies (Acc_s, Acc_c) and "indicative" accuracies ($Acc_c^{\geq 0.2}$) computed between a target prediction and the model's prediction on the generated image.

Method	Pneumonia detection			Brain tumor detection		
	$Acc_s \uparrow$	$Acc_c \uparrow$	$Acc_c^{\geq 0.2} \uparrow$	$Acc_s \uparrow$	$Acc_c \uparrow$	$Acc_c^{\geq 0.2} \uparrow$
w/o L_d^{st}	0.995	0.965	0.986	0.998	0.971	0.980
w/o $L_{d,f}^{cy}$	0.997	0.972	0.977	0.1.0	0.982	0.988
w/o $L_f^{c,cy}$	1.0	0.002	0.005	0.998	0.150	0.193
w/o L_{GAN}	0.996	0.975	0.981	0.994	0.940	0.943
Ours	0.998	0.961	0.971	0.997	0.935	0.941

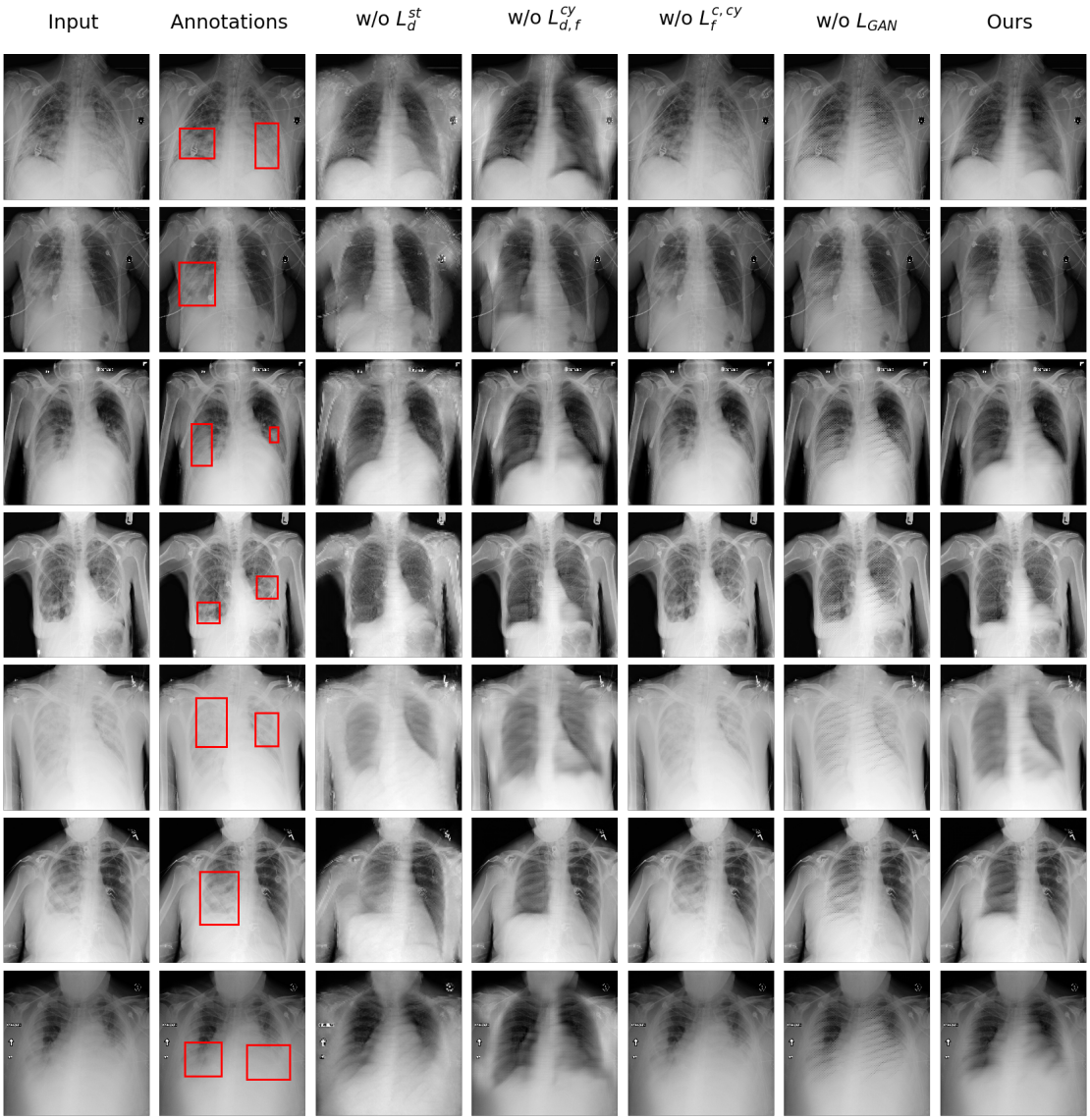


Figure E.12: Pneumonia detection - Ablation study for CyLatentCE: From pathological to healthy image. From left to right: the input image; then the counterfactual generations from CyLatentCE optimized without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}, without the realism property (the GAN term); and our CyLatentCE optimization proposed in Section 5.4.

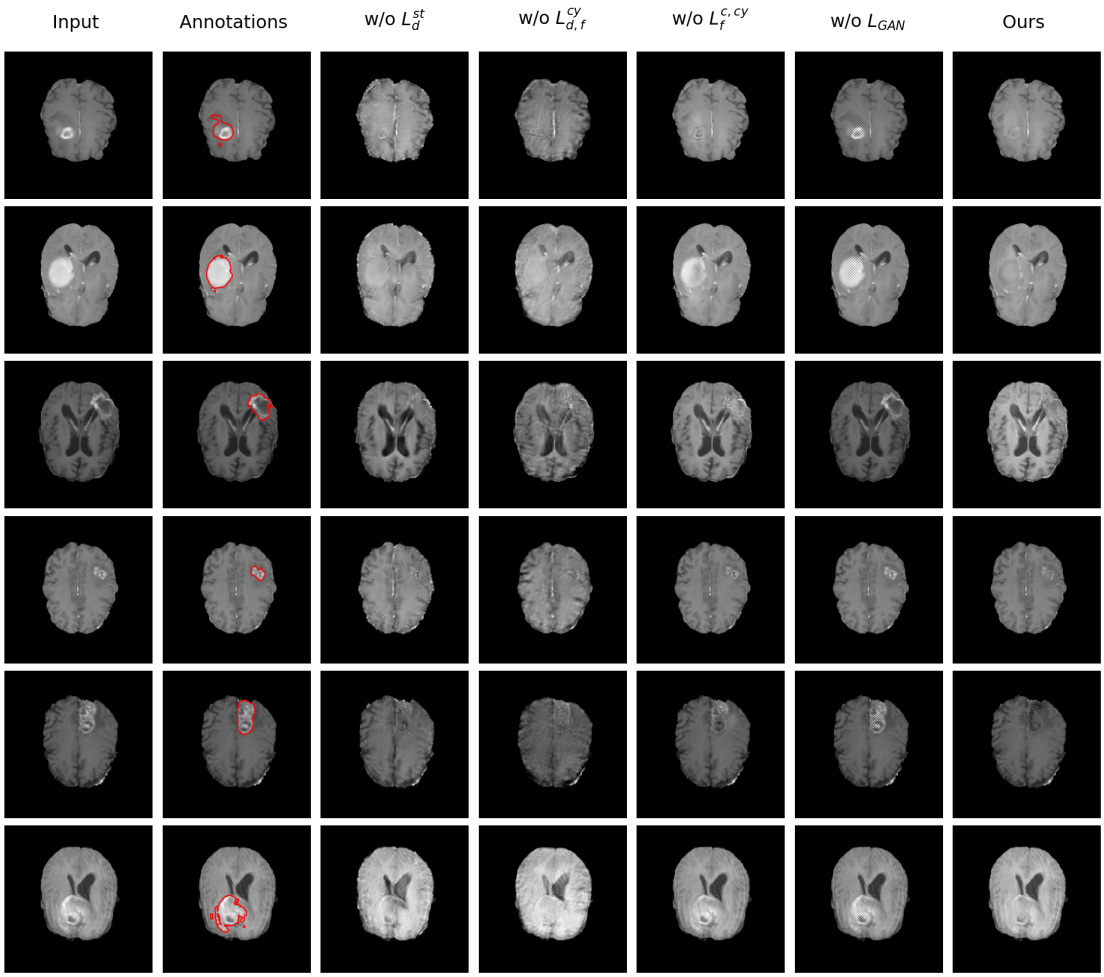


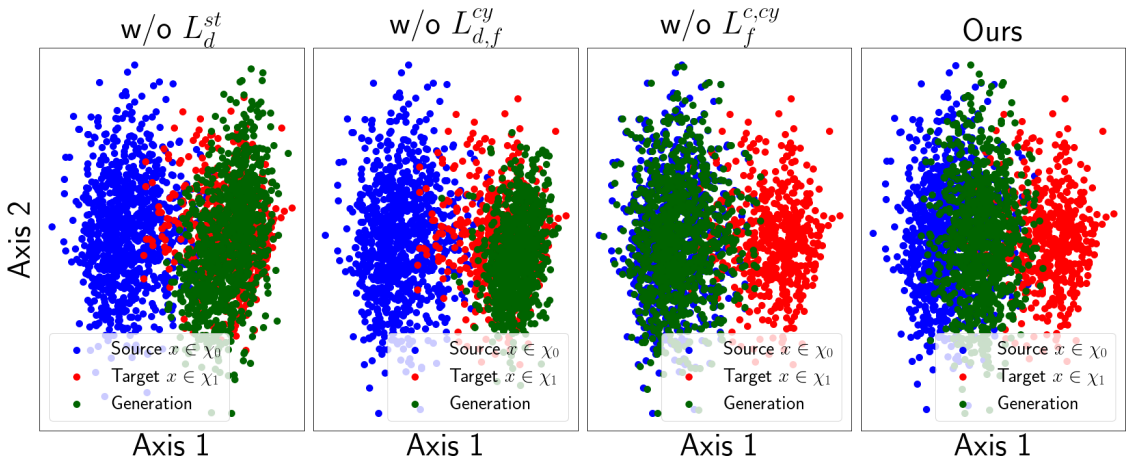
Figure E.13: Brain tumor detection - Ablation study for CyLatentCE: From pathological to healthy image. From left to right: the input image; then the counterfactual generations from CyLatentCE optimized without the stable generation (i.e. without the term L_d^{st}); without the cyclic terms (i.e. without L_d^{cy} and L_f^{cy}), without classification terms L_f^c and L_f^{cy}, without the realism property (the GAN term); and our CyLatentCE optimization proposed in Section 5.4.

E.1.3.2 Ablation study for CyImageCE

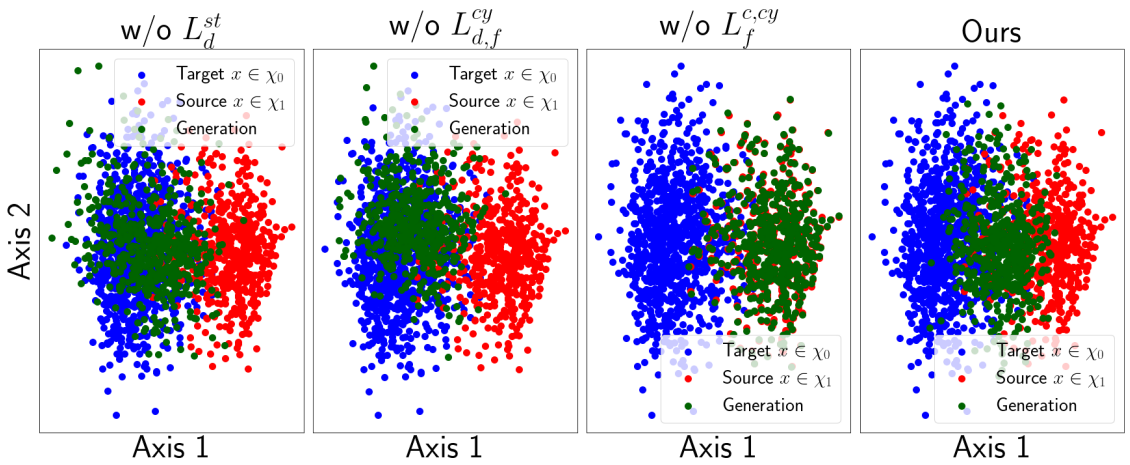
- Figures E.14 and **??** show PCA projections of CyImageCE counterfactual generations (for the two translation directions) for the different ablation cases. and supports the findings from PCA projections (see Figures 8.26 and 8.27). As for CyLatentCE, by removing terms L_d^{st} or $L_{d,f}^{cy}$, we relax the relevancy (or proximity) constraint on the counterfactual generation. The counterfactual generator can apply more transformations to the input when performing the domain translation. This results in better translation results (e.g., especially in pneumonia detection) compared to our optimization, where the translated generations remain closer to the input (i.e., satisfying the relevance property). Removing the classification terms prevents domain translation in the pneumonia problem and highly decreases the translation rate in the brain tumor problem.
- Table E.2 displays the classification metrics on both pneumonia and brain tumor detection for CyImageCE when removing the different terms of loss. Supporting previous findings, the counterfactual generations are not predicted in the target class (by the classifier) without the terms $L_f^{c,cy}$ (especially on the pneumonia problem).

Table E.2: Classification results - Ablation study CyImageCE. Accuracies (Acc_s, Acc_c) and "indicative" accuracies ($Acc_c^{\geq 0.2}$) computed between a target prediction and the model's prediction on the generated image.

Method	Pneumonia detection			Brain tumor detection		
	$Acc_s \uparrow$	$Acc_c \uparrow$	$Acc_c^{\geq 0.2} \uparrow$	$Acc_s \uparrow$	$Acc_c \uparrow$	$Acc_c^{\geq 0.2} \uparrow$
w/o L_d^{st}	0.999	0.994	0.995	0.984	0.977	0.990
w/o $L_{d,f}^{cy}$	0.999	0.998	0.998	0.999	0.993	0.994
w/o $L_f^{c,cy}$	0.994	0.006	0.008	0.989	0.342	0.400
Ours	0.999	0.968	0.984	0.999	0.950	0.955

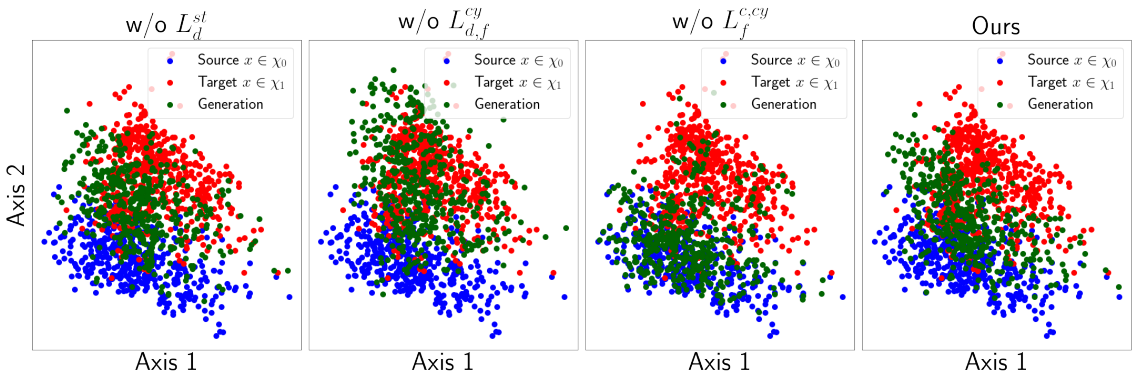


(a) $\chi_0 \longrightarrow \chi_1$

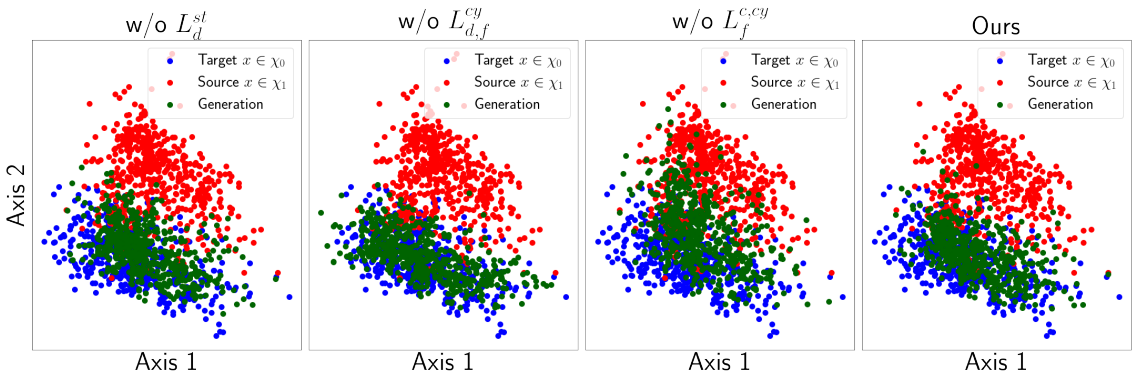


(b) $\chi_1 \longrightarrow \chi_0$

Figure E.14: Pneumonia detection - Qualitative VAE Results: Ablation study for CyImageCE. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.



(a) $\chi_0 \longrightarrow \chi_1$



(b) $\chi_1 \longrightarrow \chi_0$

Figure E.15: Brain tumor detection - Qualitative VAE Results: Ablation study for CyImageCE. The First 2 axes of the PCA applied on the embedded vector μ of the VAE for all images (real and generated) of the test set. **(a)**: Source (original) domain χ_0 and Target for counterfactual generation: χ_1. **(b)**: Source χ_1 and Target χ_0.

E.1.3.3 Stable generation impact

This section studies the stable generation and its impact on the counterfactual visual explanation.

- Tables E.3 provide the different domain proximity metrics for stable generations on (a) healthy and (b) pathological domains. For all cases, stable generations remain close to the input domain (when comparing against the values obtained for counterfactual generations in Tables 8.12). SySCGen and CySCGen produce stable images that are the closest to the inputs. We expected such findings as a specific generator branch is dedicated to the stable generation.
- Figures E.16 and E.17 show the impact of the stable generation. It mainly removes (or filters) fine-grained details from the input on the pneumonia detection problem, e.g., X-ray writings. In contrast, stable generations have different impacts on the brain tumor problem. They either differ from the input on the edges of the brain structures (SyCE and CySCGen) or may also add classification artifacts (CyLatentCE and CyImageCE). This supports findings from the localization results Section 8.1.2 and Appendix C.1.3.
- To prevent the stable generator from adding an artifact on the input image when generating a stable pathological image, we propose to pass some random noise input to the CyLatenCE generator when generating the counterfactual image and a zero input when producing the stable image. Figure E.18 illustrates this generation trick for CyLatentCE.

Table E.3: Domain Translation results for stable generations. **(a)** and **(b)**: Fréchet Distance (FD_μ) , Jenson-Shannon distances (JS) and Fréchet Inception Distance (FID_{tr}) –computed with an Inception network trained on the task– on the two medical problems. Here, we measure the distance between the stable generations and the distribution of source images.

(a) $\chi_0 \longrightarrow \chi_0$

Method	Pneumonia detection			Brain tumor detection		
	FD_μ (e-4) ↓	JS ↓	FID_{tr} ↓	FD_μ ↓	JS ↓	FID_{tr} ↓
SSyGen (SP)	0.25	0.13	0.02	8e-3	0.10	0.03
SSyGen (DP)	0.21	0.10	0.01	7e-4	0.06	0.01
SyCE	0.19	0.09	0.01	6e-4	0.05	0.01
CyLatentCE	0.86	0.13	0.02	1e-3	0.09	0.03
CyImageCE	0.14	0.07	0.01	7e-4	0.01	3e-4
SySCGen	7e-3	0.01	8e-5	6e-6	5e-3	1e-4
CySCGen	3e-4	4e-3	4e-5	7e-5	6e-3	1e-4

(b) $\chi_1 \longrightarrow \chi_1$

Method	Pneunmonia detection			Brain tumor detection		
	FD_μ ↓	JS ↓	FID_{tr} ↓	FD_μ ↓	JS ↓	FID_{tr} ↓
SSyGen (SP)	0.35	0.13	0.02	0.01	0.13	0.05
SSyGen (DP)	0.38	0.16	0.03	4e-4	0.05	0.01
SyCE	0.26	0.12	0.02	3e-4	0.05	0.01
CyLatentCE	0.63	0.12	0.03	3e-3	0.12	0.04
CyImageCE	0.33	0.09	0.02	1e-4	0.03	6e-4
SySCGen	5e-4	5e-3	9e-5	8e-6	5e-3	8e-5
CySCGen	2e-4	3e-3	6e-5	3e-6	2e-3	4e-5

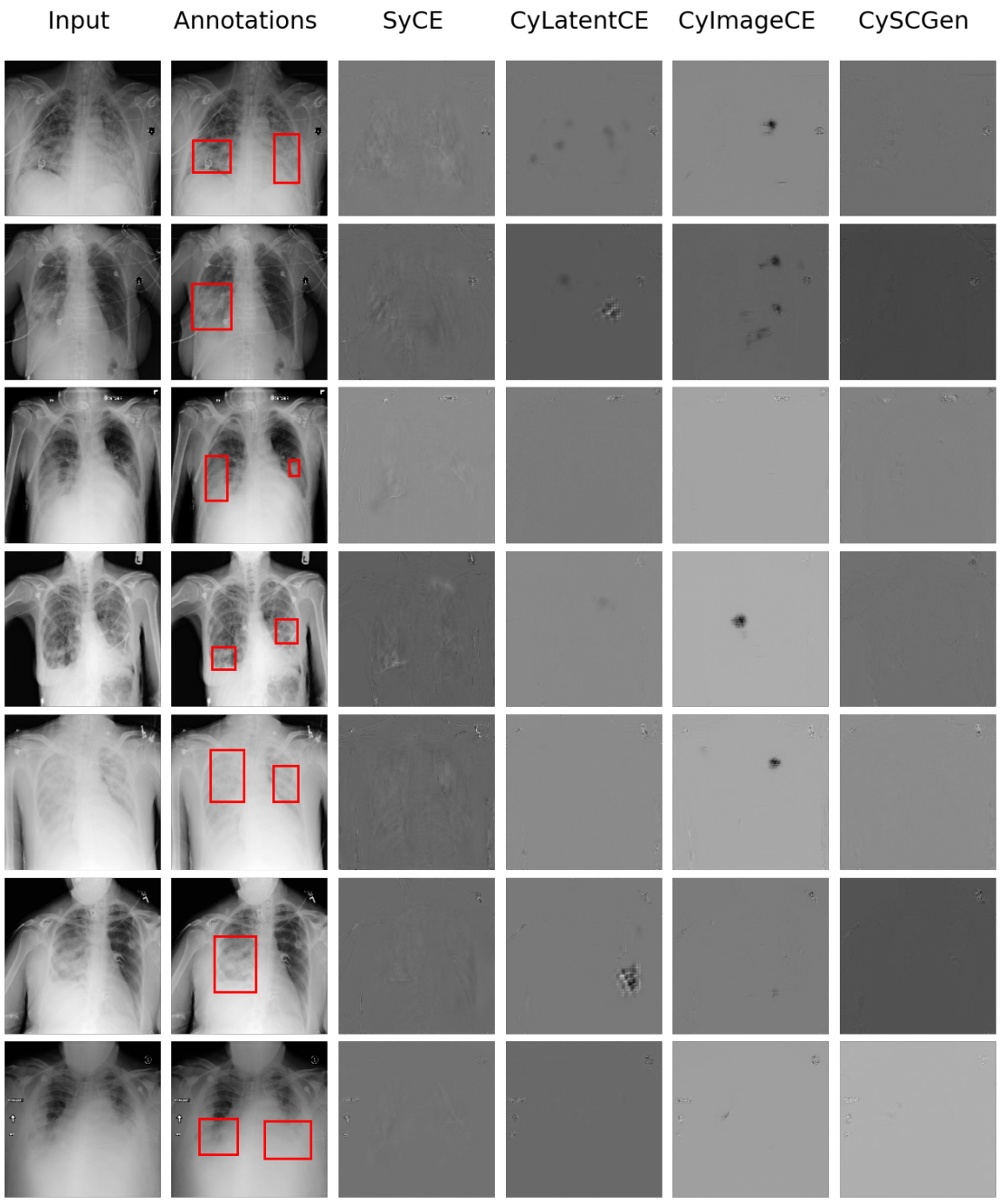


Figure E.16: Pneumonia detection - Difference between input and stable generation. From left to right: the input image, the annotated input image, then the difference between the input image and the stable generation for diverse counterfactual generation techniques.

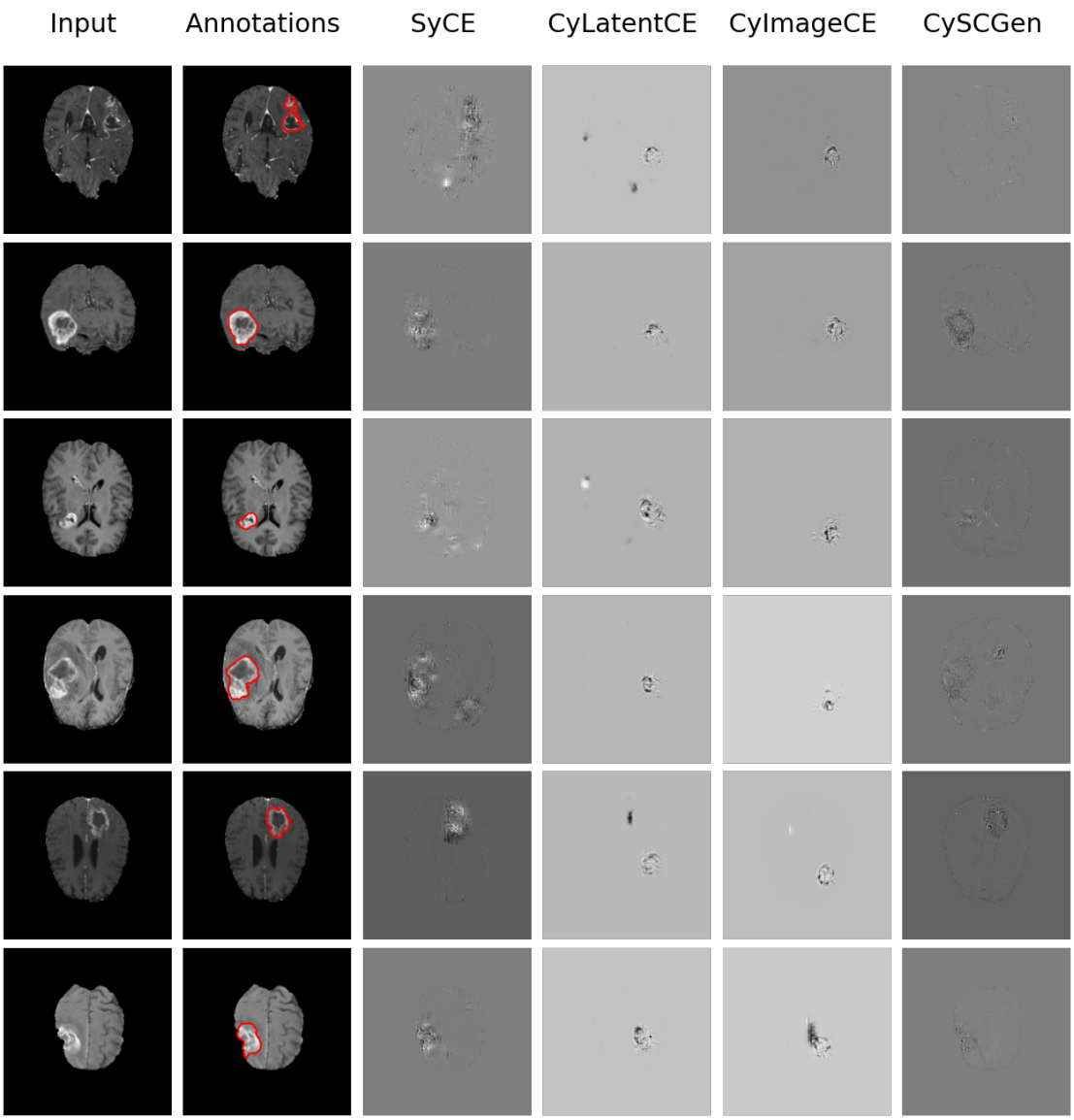


Figure E.17: Brain tumor detection - Difference between input and stable generation. From left to right: the input image, the annotated input image, then the difference between the input image and the stable generation for diverse counterfactual generation techniques.

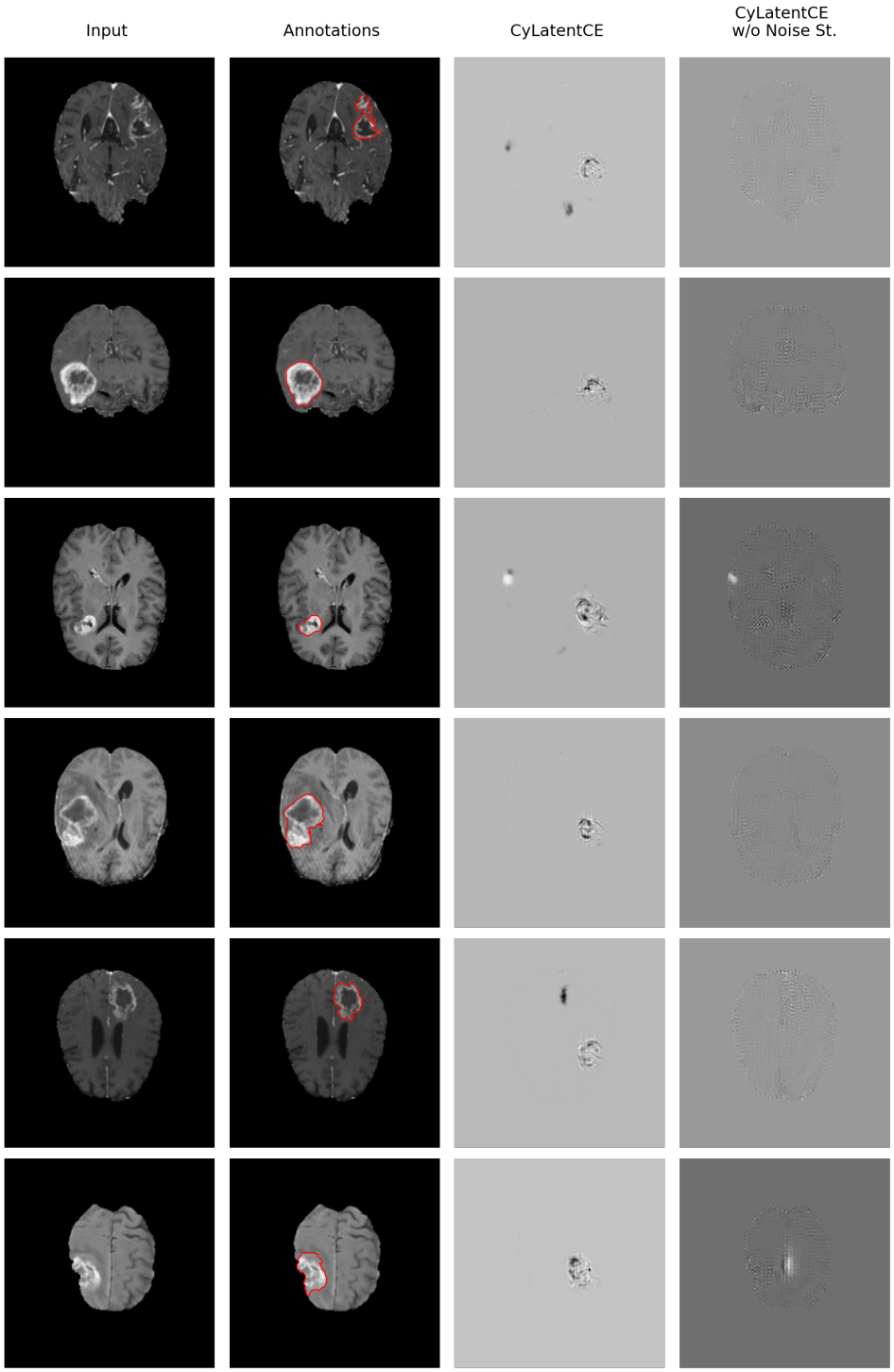


Figure E.18: Brain tumor detection - Difference between input and stable generation. From left to right: the input image, the annotated input image, then the difference between the input image and the stable generation (with zero noise trick) for diverse counterfactual generation techniques.

E.2 Comparison against state-of-the-art

Here, we provide additional figures to compare our counterfactual generations against adversarial generations (from SAGen) or perturbated images from MGen:

- Pneumonia detection: Figures E.19 and E.20 illustrate pathological to healthy translation, while Figures E.21 and E.22 shows the opposite translation.
- Brain tumor detection: idem for Figures E.23 and E.24 respectively.

E.2.1 Pneumonia detection

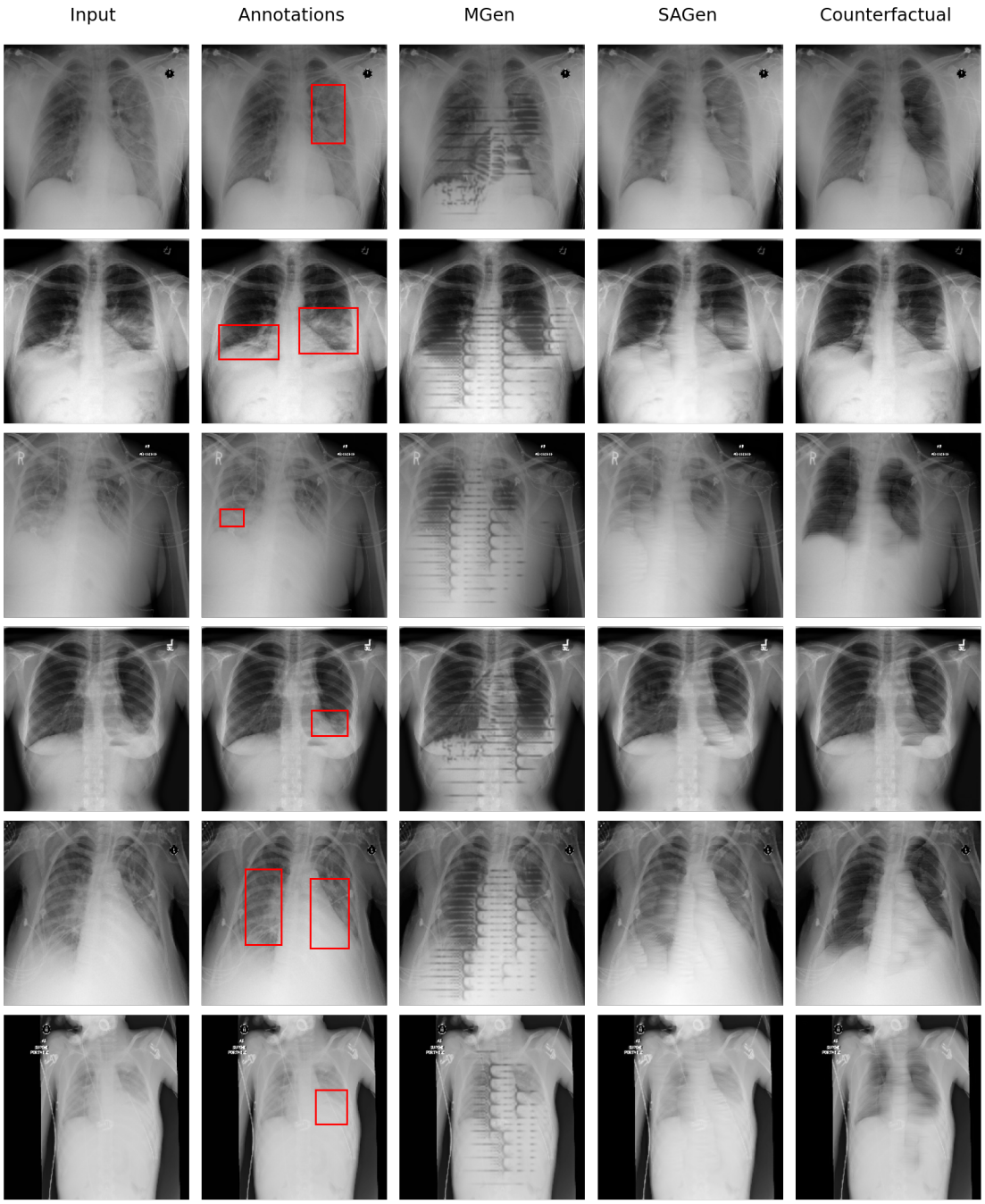


Figure E.19: Pneumonia detection (1) - Comparison with other generation / perturbation techniques: From pathological to healthy image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

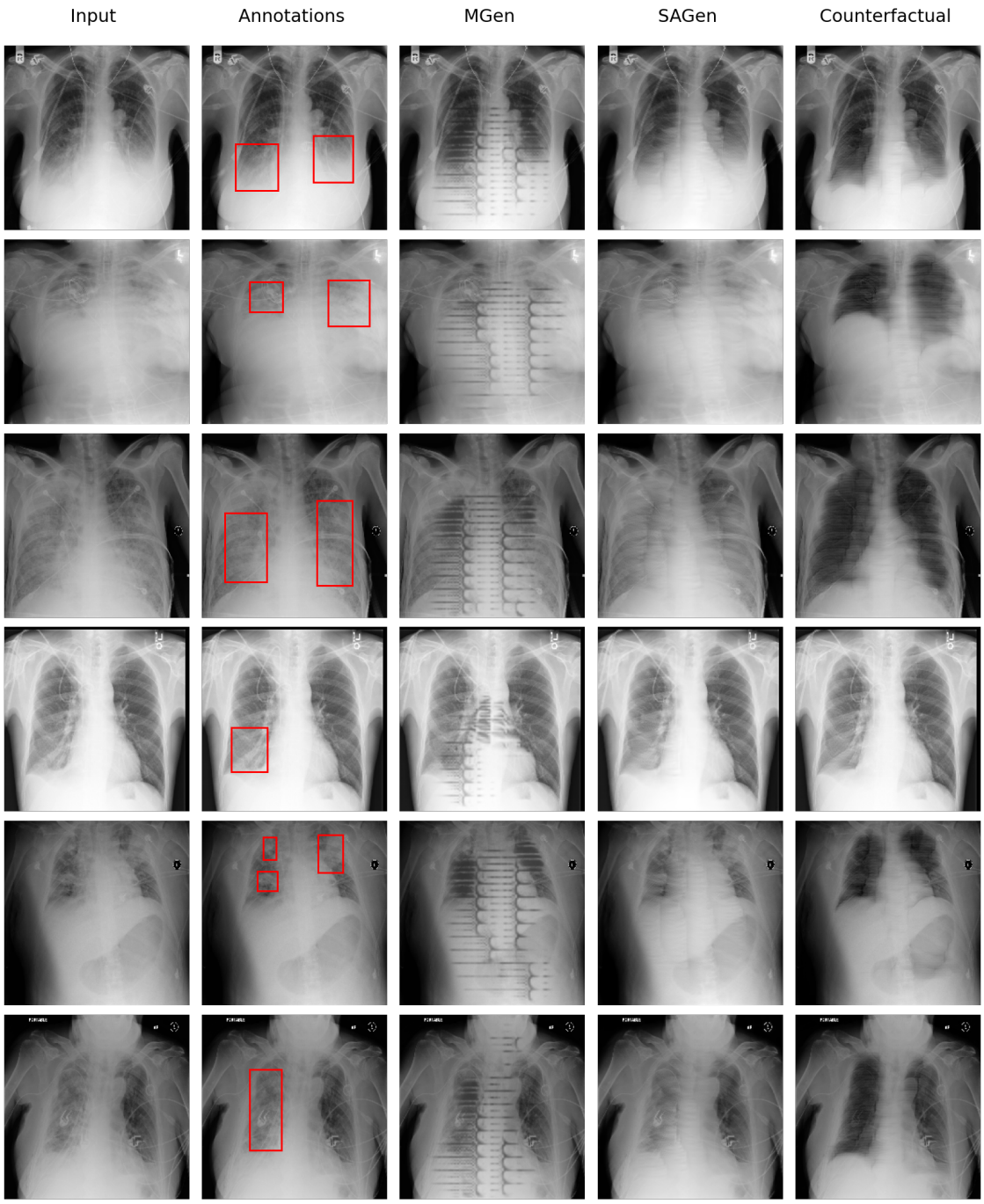


Figure E.20: Pneumonia detection (2) - Comparison with other generation / perturbation techniques: From pathological to healthy image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

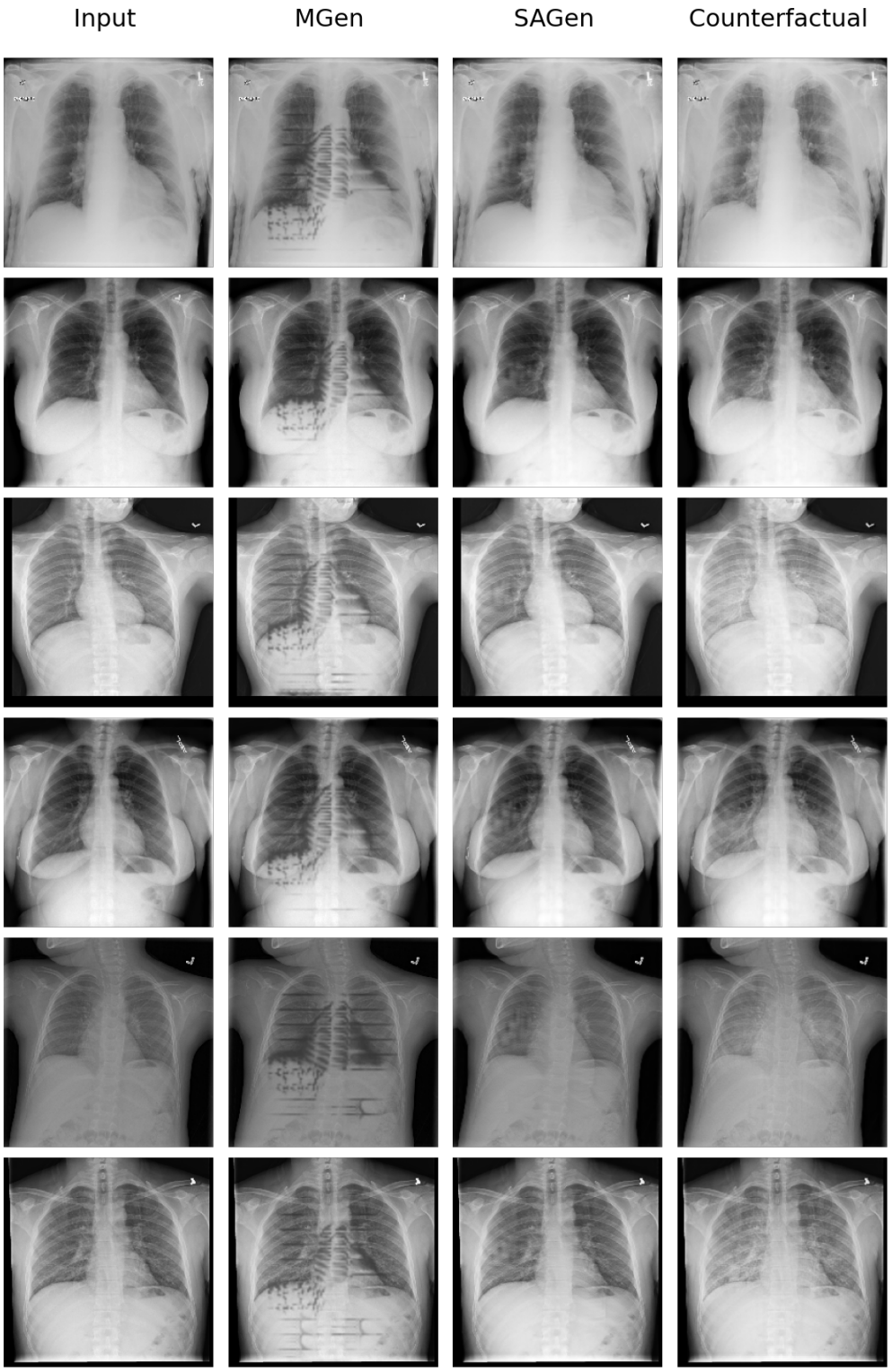


Figure E.21: Pneumonia detection (1) - Comparison with other generation / perturbation techniques: From healthy to pathological image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

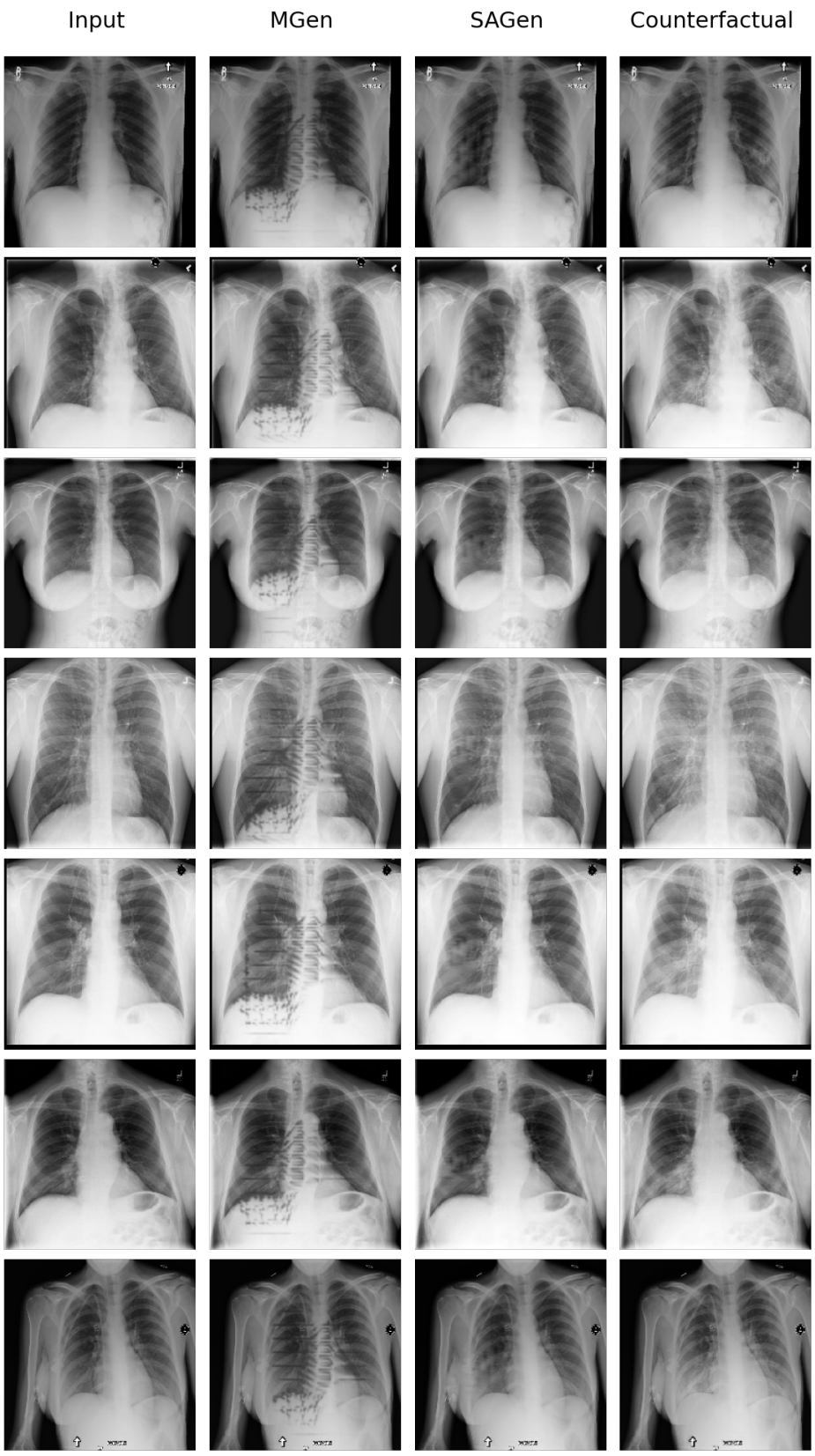


Figure E.22: Pneumonia detection (2) - Comparison with other generation / perturbation techniques: From healthy to pathological image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

E.2.2 Brain tumor detection

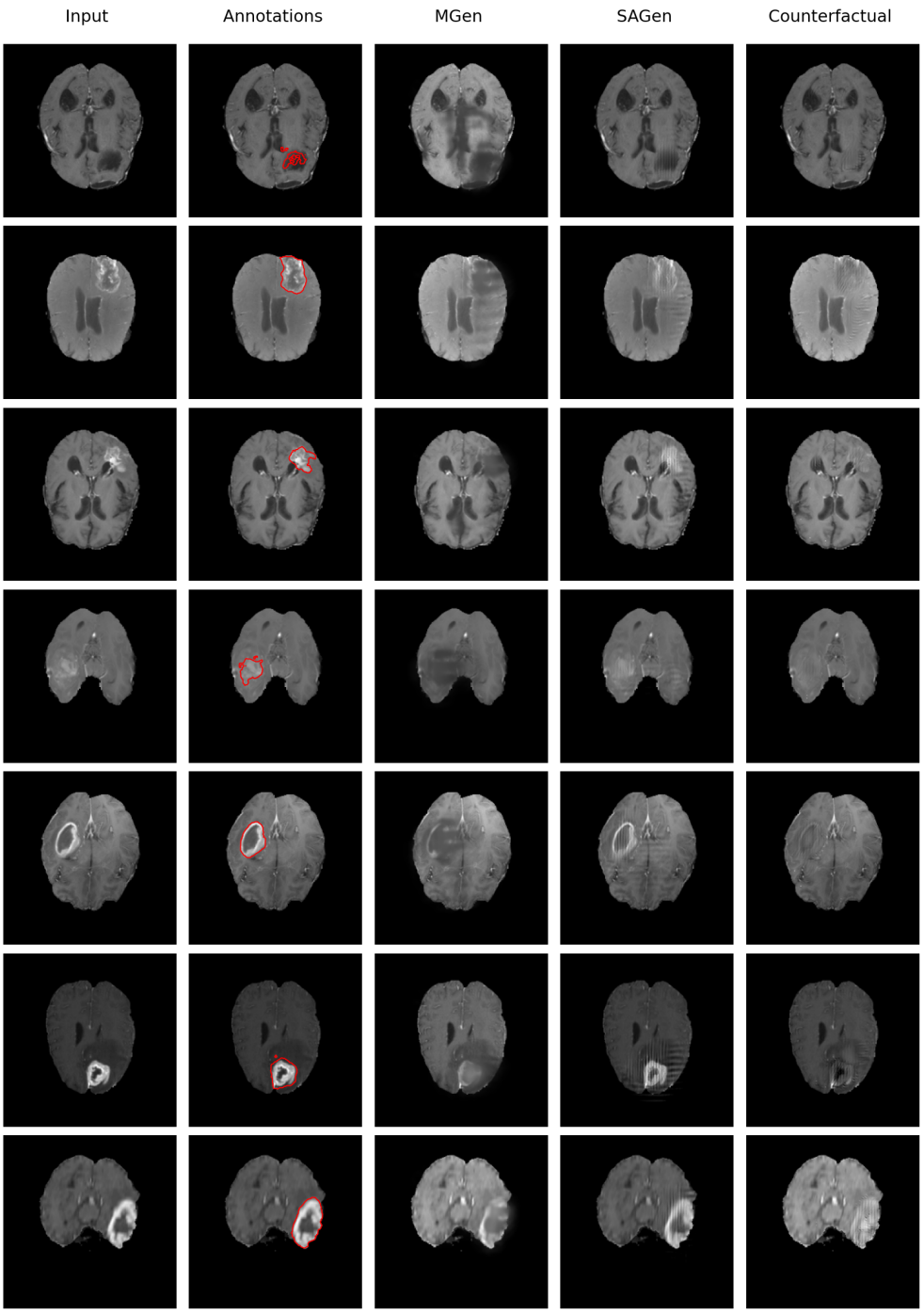


Figure E.23: Brain tumor detection (1) - Comparison with other generation / perturbation techniques: From pathological to healthy image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

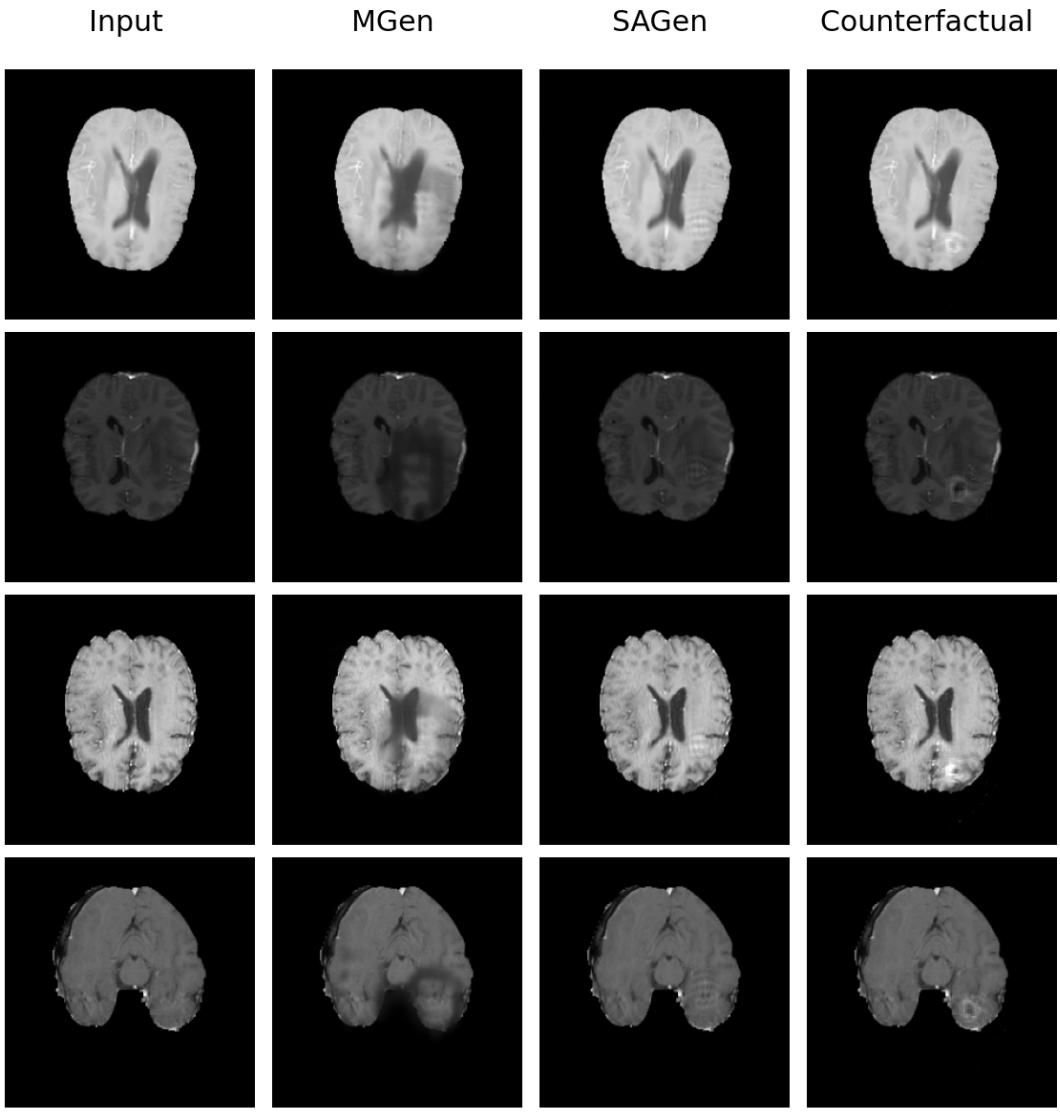


Figure E.24: Brain tumor detection (1) - Comparison with other generation / perturbation techniques: From healthy to pathological image. From left to right: the input image; the annotated input image; the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from CyImageCE.

E.2.3 MNIST 3 vs 8

Here we give additional figures of counterfactual, adversarial, and perturbated generations for the binary digit classification task (in the two directions):

- "8 $\longrightarrow$ 3" in Figure E.25.
- "3 $\longrightarrow$ 8" in Figure E.26.

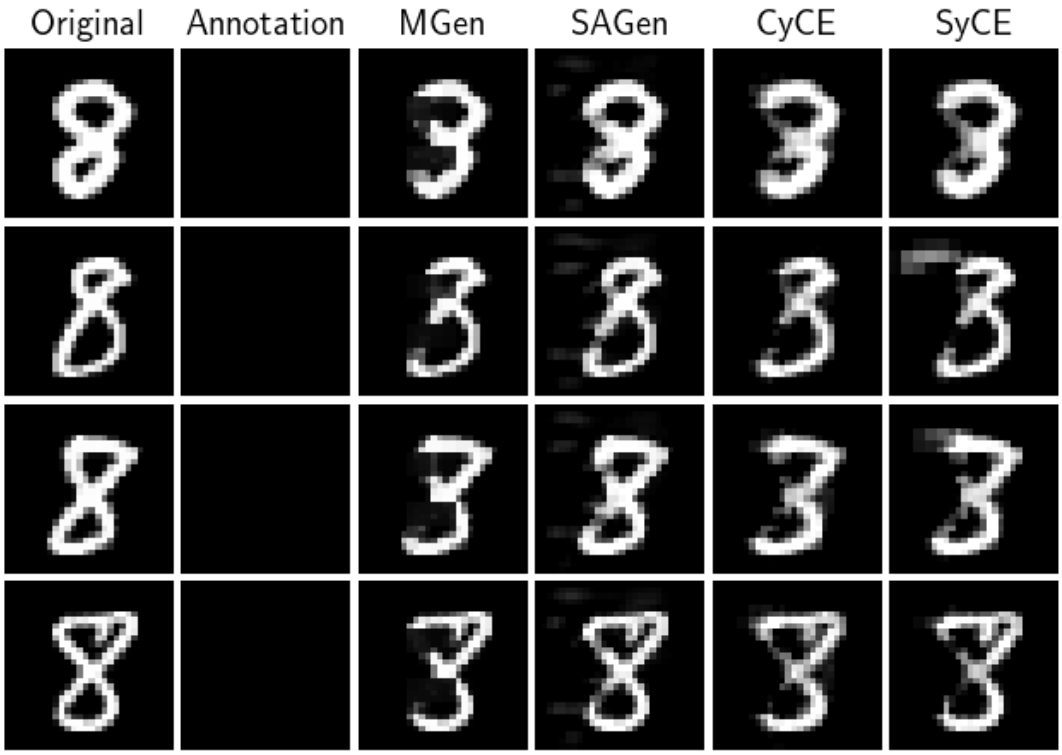


Figure E.25: MNIST 3 vs 8 - Comparison with other generation / perturbation techniques: From 8 to 3. From left to right: the input image, the annotated input image, the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from SyCE.

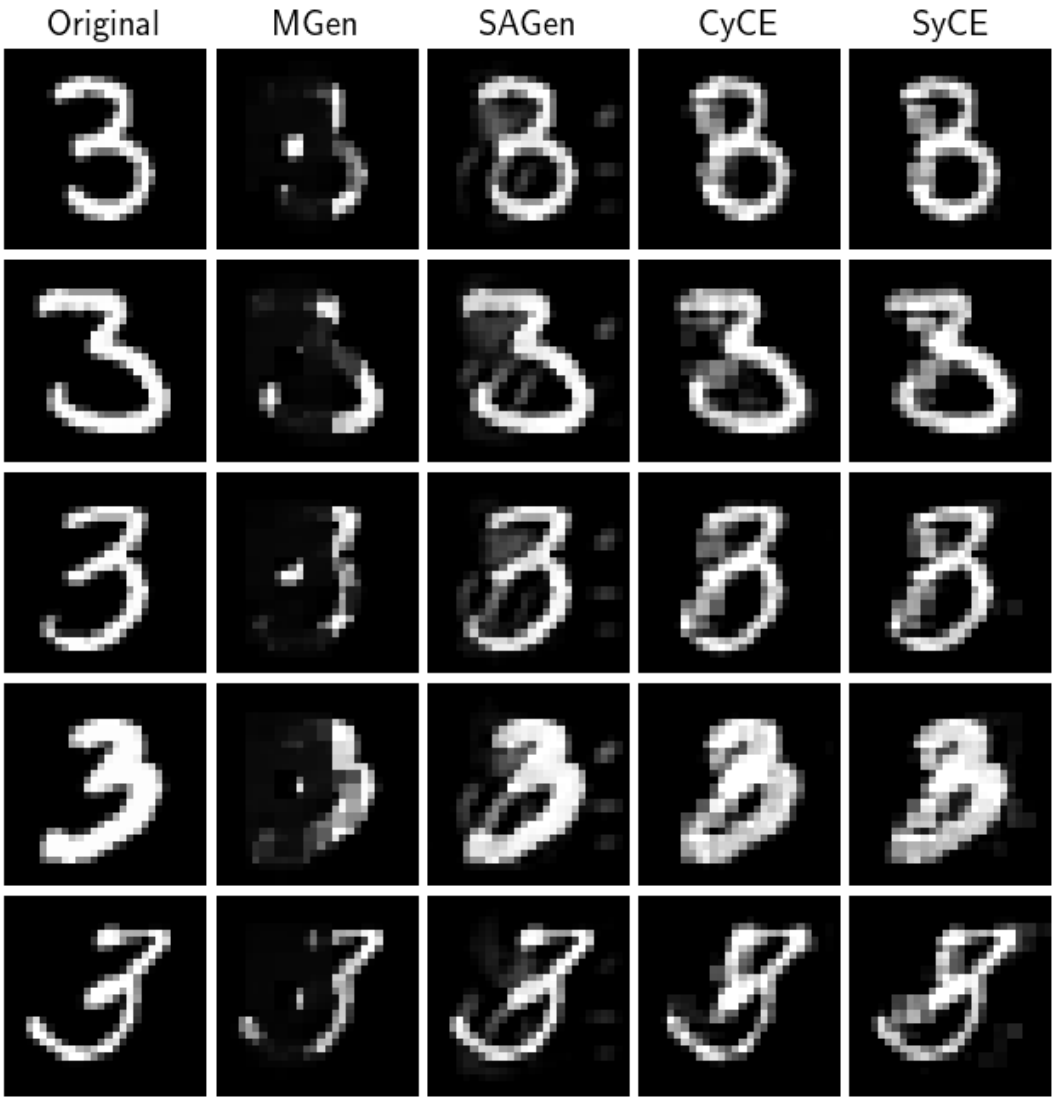


Figure E.26: MNIST 3 vs 8 - Comparison with other generation / perturbation techniques: From 3 to 8. From left to right: the input image, the annotated input image, the perturbated input through MGen, the adversarial generation from SAGen; and the counterfactual generation from SyCE.

E.2.4 Generations and Attributions for attributes classification on CelebA

We show additional figures of stable and counterfactual generations for the binary attributes classification tasks (in the two directions):

- Mustache detection in Figures E.27 and E.28. As pointed out in the manuscript (see Section 8.4), we notice a gender bias when generating a counterfactual image for a woman's face and a correlation between mustache and beard.
- Age classification in Figures E.29 and E.30.

E.2.4.1 Mustache vs. No mustache

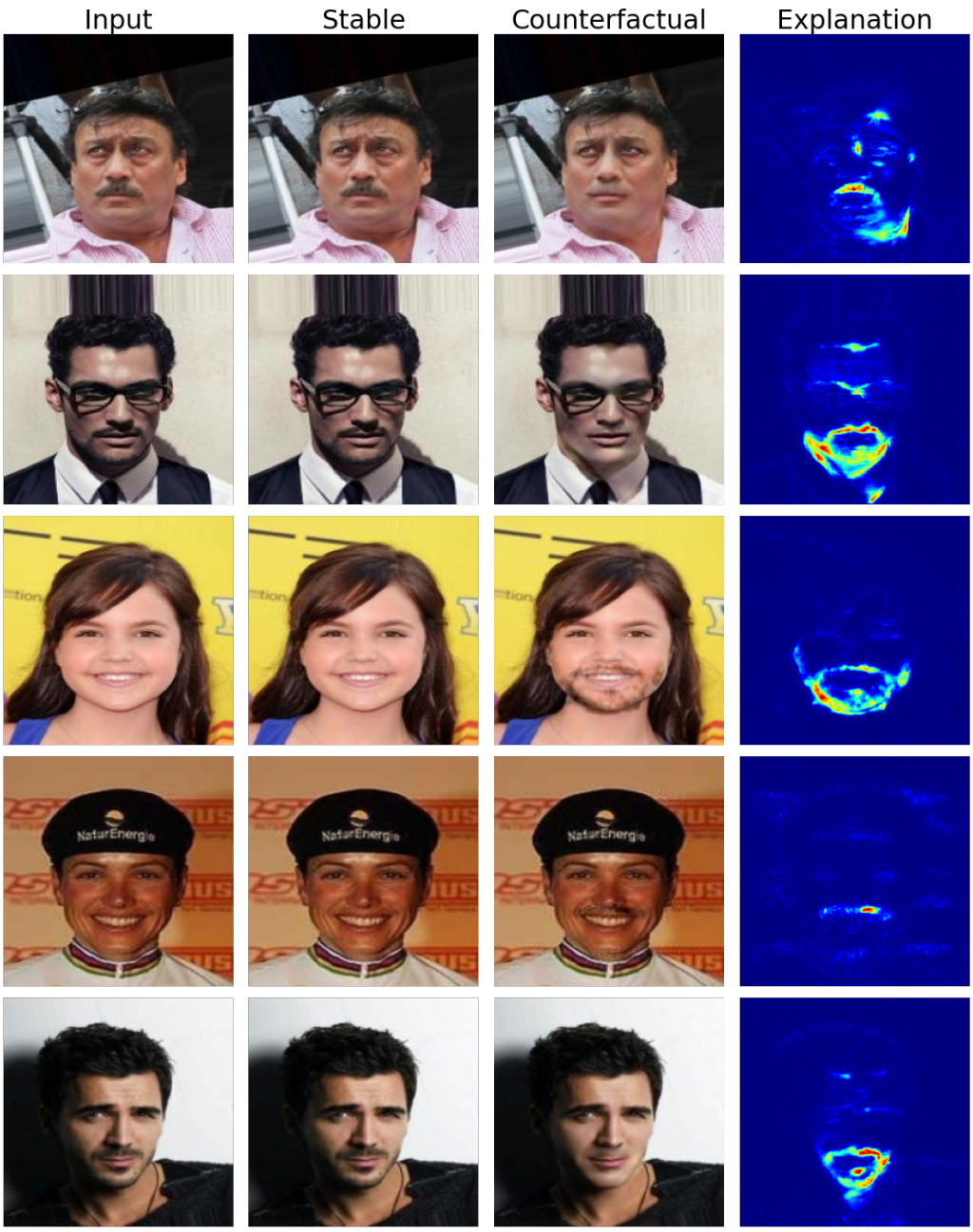


Figure E.27: Mustache vs. No mustache (1) - Generations and attributions. From left to right: the input image, the stable generation, the counterfactual generation, and the counterfactual explanation map. The results are produced with the CyLatentCE technique.

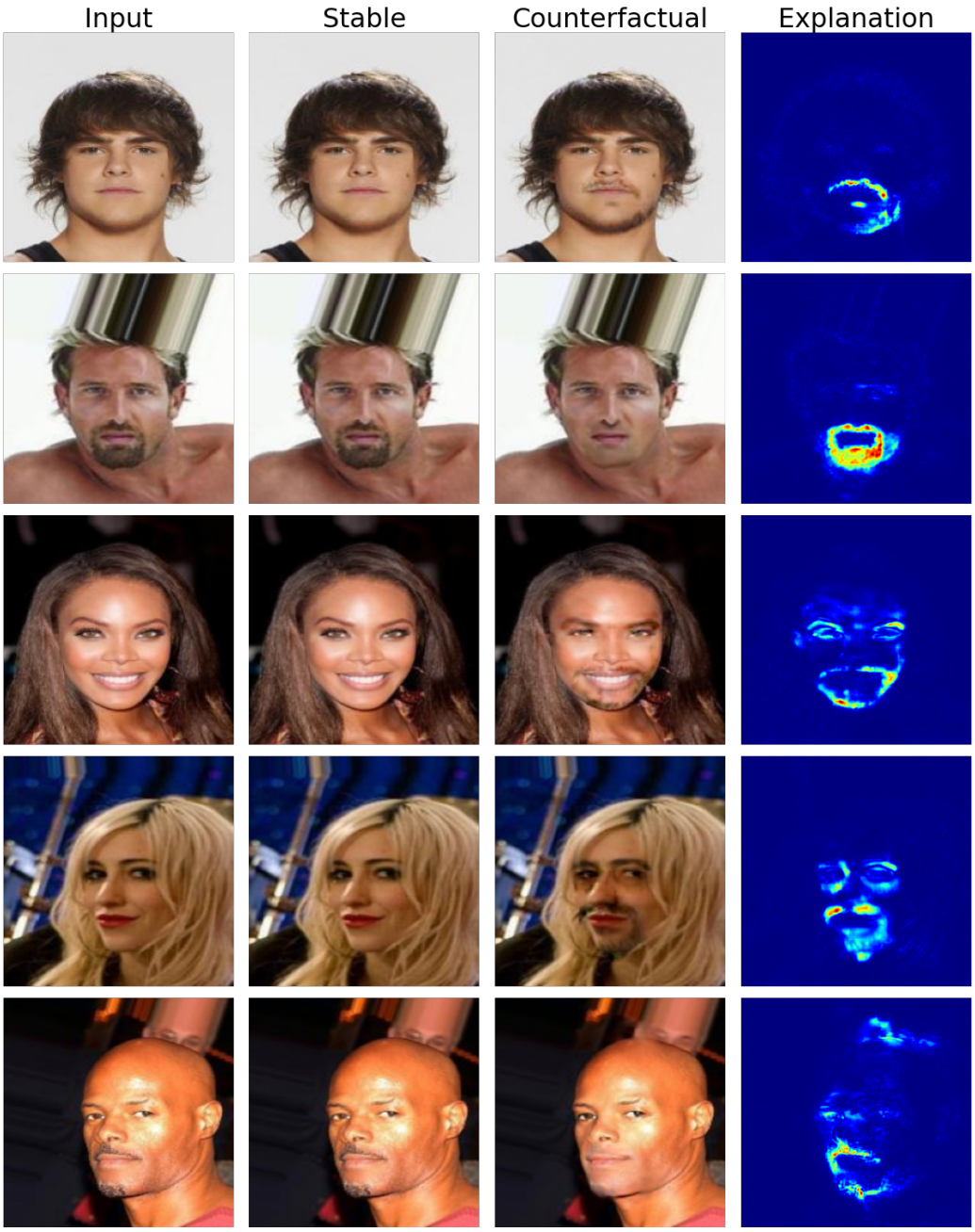


Figure E.28: Mustache vs. No mustache (2) - Generations and attributions. From left to right: the input image, the stable generation, the counterfactual generation, and the counterfactual explanation map. The results are produced with the CyLatentCE technique.

E.2.4.2 Young vs. Old

Figure E.29: Young vs. Old (1) - Generations and attributions. From left to right: the input image, the stable generation, the counterfactual generation, and the counterfactual explanation map. The results are produced with the CyLatentCE technique.

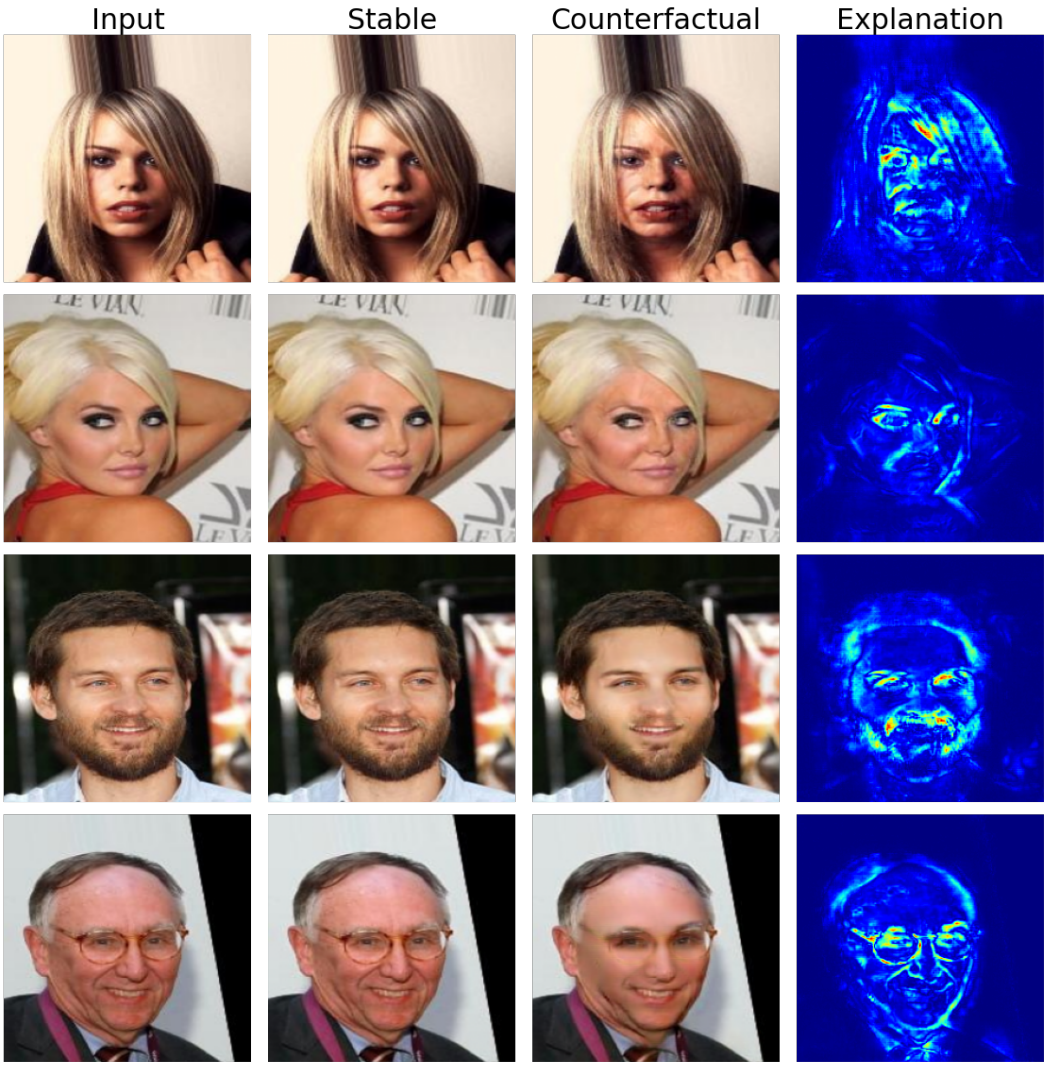


Figure E.30: Young vs. Old (2) - Generations and attributions. From left to right: the input image, the stable generation, the counterfactual generation, and the counterfactual explanation map. The results are produced with the CyLatentCE technique.

E.2.5 MNIST multi-classification

Here we give additional figures of stable and counterfactual generations for the multi-classification task on MNISt:

- Targeted counterfactual generations are shown in Figure E.31.
- Untargeted counterfactual generations are shown in Figure E.33.
- We also display targeted counterfactual generations produced when removing cyclic constraint from CyLatentCE optimization (see Figure E.32). We observe that the whole structure of the digit is changed when performing the translation, i.e., this framework does not highlight the most relevant (but minimal) regions that make the classifier decides between the input's prediction and a given target class.

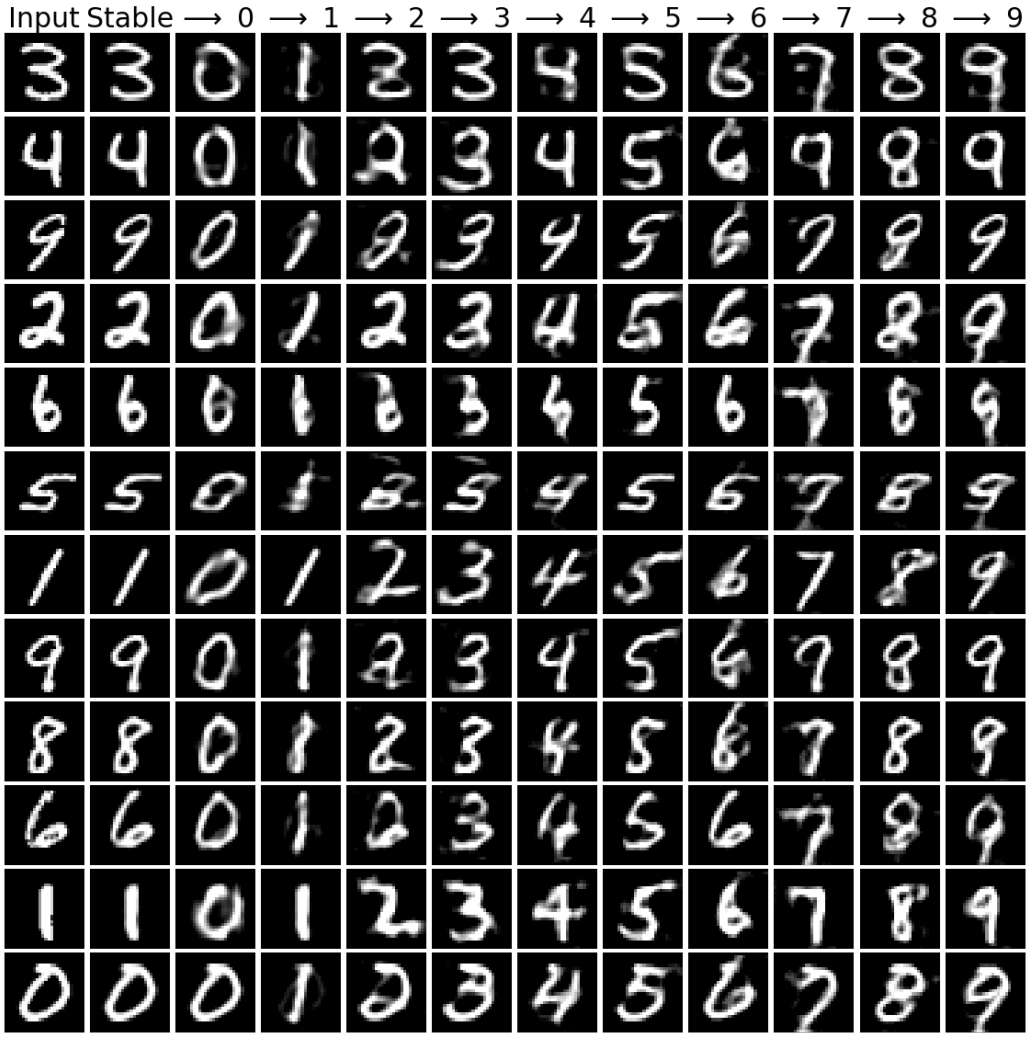


Figure E.31: Targeted counterfactual generations for multi-classification model on MNIST. From left to right: the original image; the stable image; counterfactual image with respect to each class (columns 3 to 12).

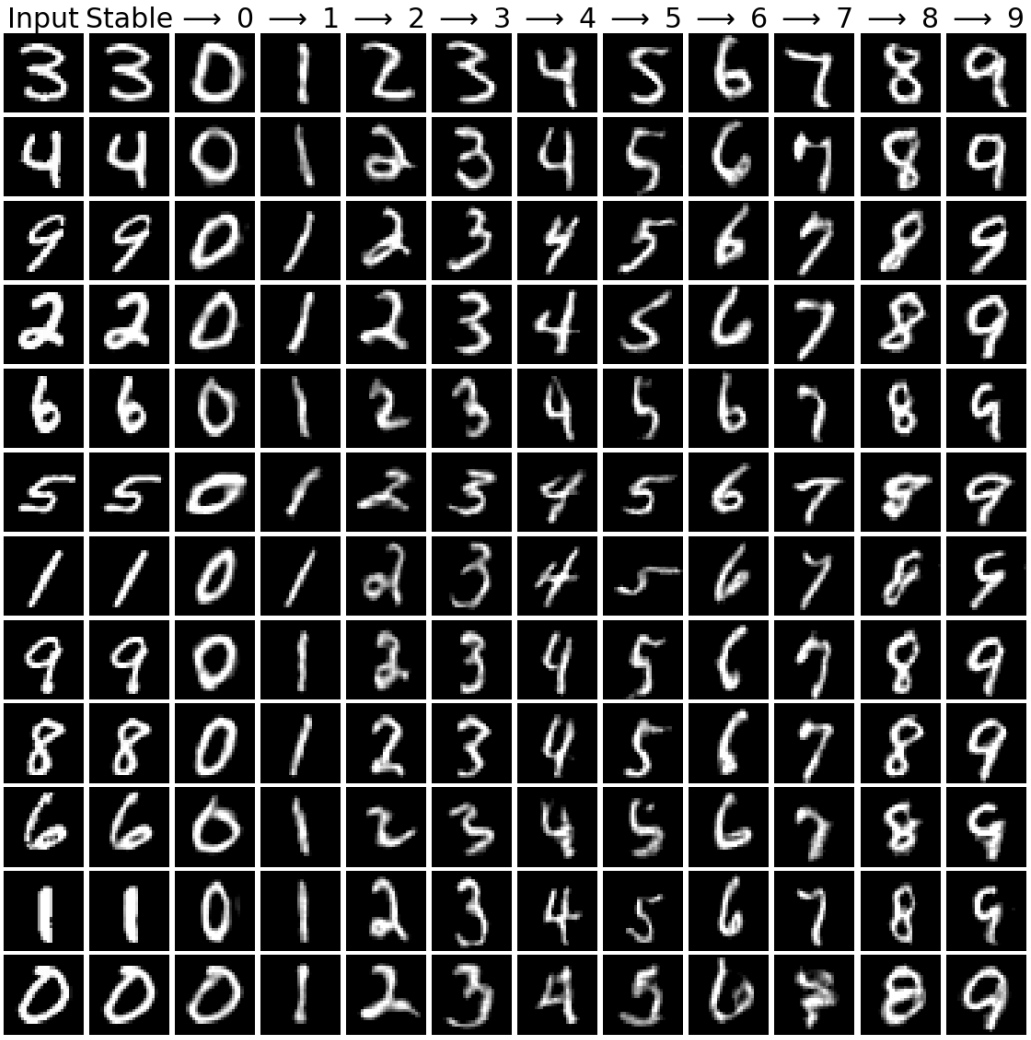


Figure E.32: Targeted counterfactual generations without cyclic constraint for multi-classification model on MNIST. From left to right: the original image; the stable image; counterfactual image with respect to each class (columns 3 to 12).

Figure E.33: Untargeted counterfactual generations for multi-classification model on MNIST. From left to right: the original image, the stable image, the untargeted counterfactual image, and the visual explanation.

F

Appendix: Errors study

Here, we compare the raw difference maps computed for different counterfactual methods for the pneumonia (F.1) and the brain tumor (F.2) detection problems. We display examples for different model's outcomes, i.e., True Positives, True Negatives, False Positives and False Negatives. We only use the "Pathological to Healthy" translation when generating counterfactuals, except for SySCGen where we cannot control the translation direction.

F.1 Pneumonia detection

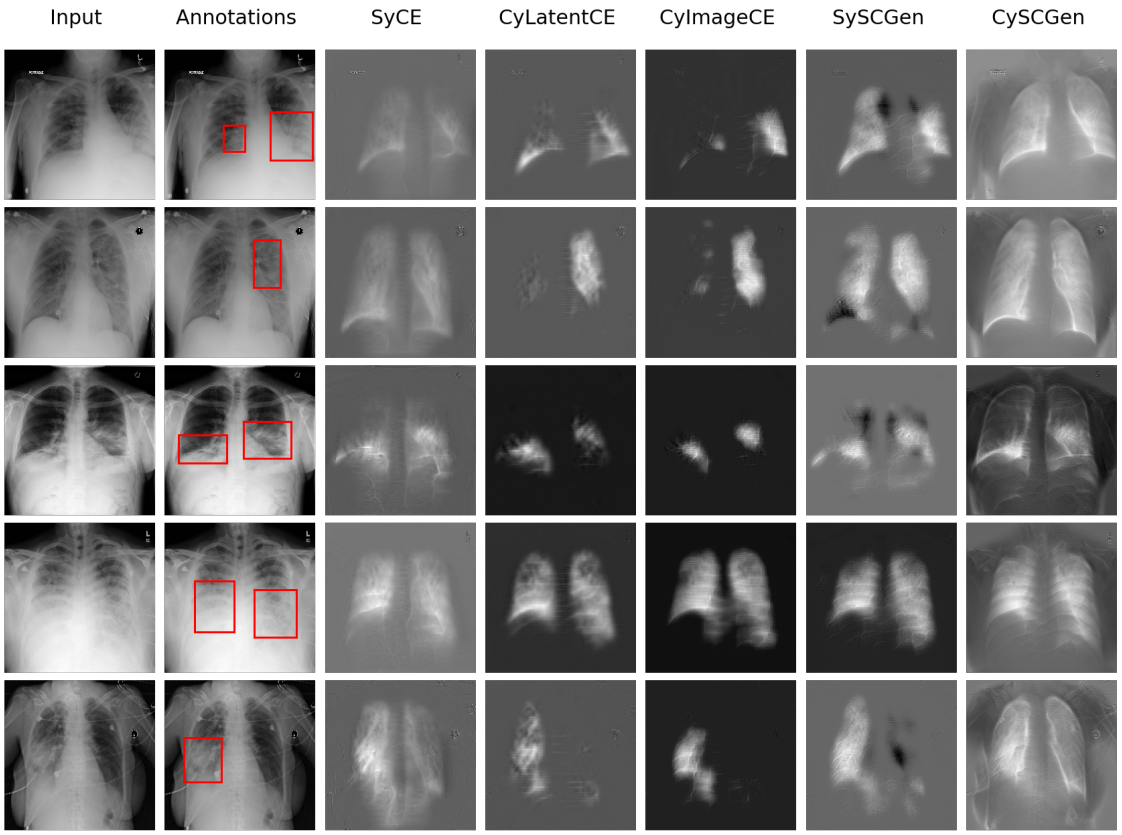


Figure F.1: Pneumonia detection - Difference maps of True Positive examples. The first two columns: the input image and the annotated input image. From columns 3 to 7: the difference maps computed for different counterfactual generation techniques. Input against SySCGen counterfactual predictions: 0.96 / 0.05 - 0.99 / 0.17 - 1.0 / 0.19 - 1.0 / 0.05 - 1.0 / 0.23. 0 being healthy and 1 pathological.

Figure F.2: Pneumonia detection - Difference map of True Negative examples. The first column: the input image. From columns 2 to 6: the difference maps computed for different counterfactual generation techniques, imposing the translation "Pathological to Healthy" (except for SySCGen). Input against SySCGen counterfactual predictions: 0.03 / 0.99 - 0.01 / 0.99 - 0.01 / 0.97 - 0.01 / 1.0 - 0.0 / 0.99 - 0.0 / 1.0. 0 being healthy and 1 pathological.

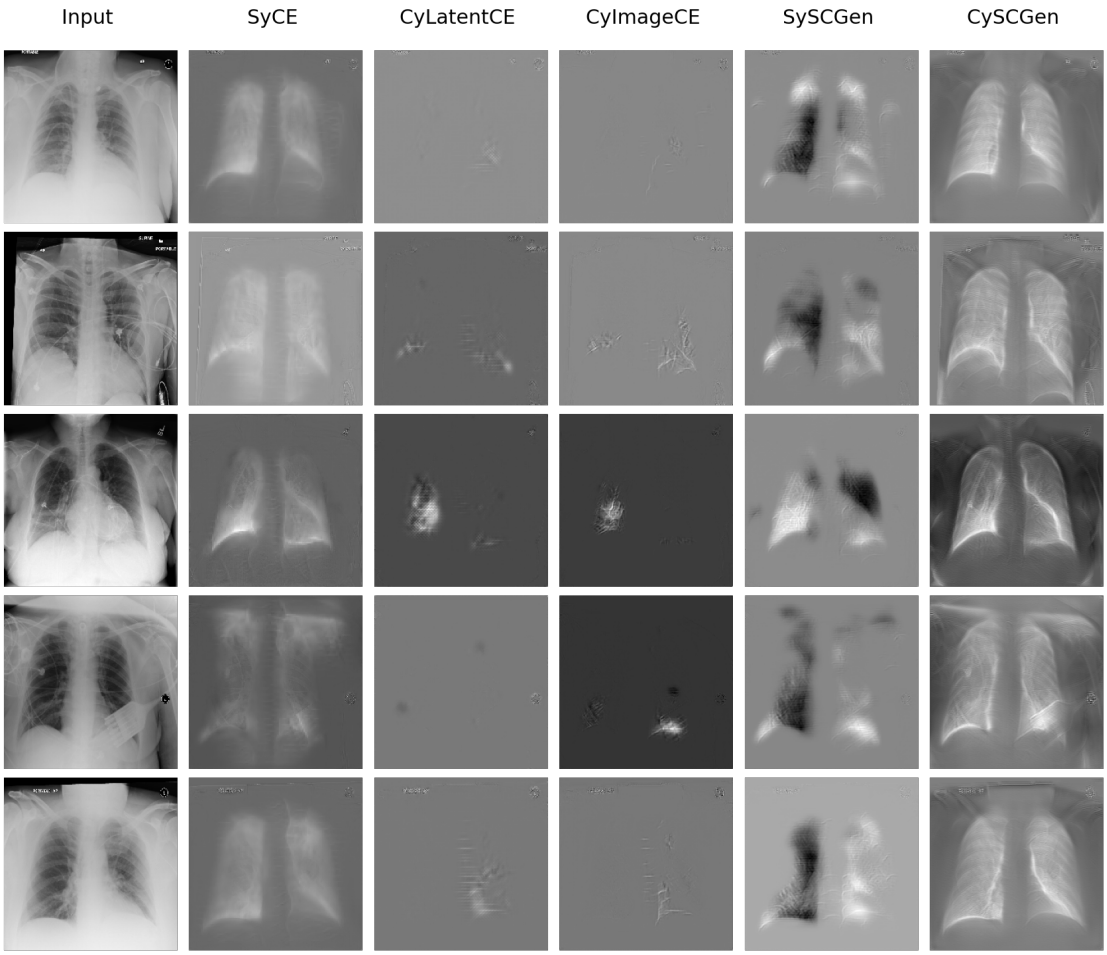


Figure F.3: Pneumonia detection - Difference map of False Positive examples. The first column: the input image. From columns 2 to 6: the difference map computed for different counterfactual generation techniques. Input against SySCGen counterfactual predictions: 0.69 / 0.65 - 0.71 / 0.11 - 0.85 / 0.04 - 0.58 / 0.99 - 0.68 / 0.52. 0 being healthy and 1 pathological.

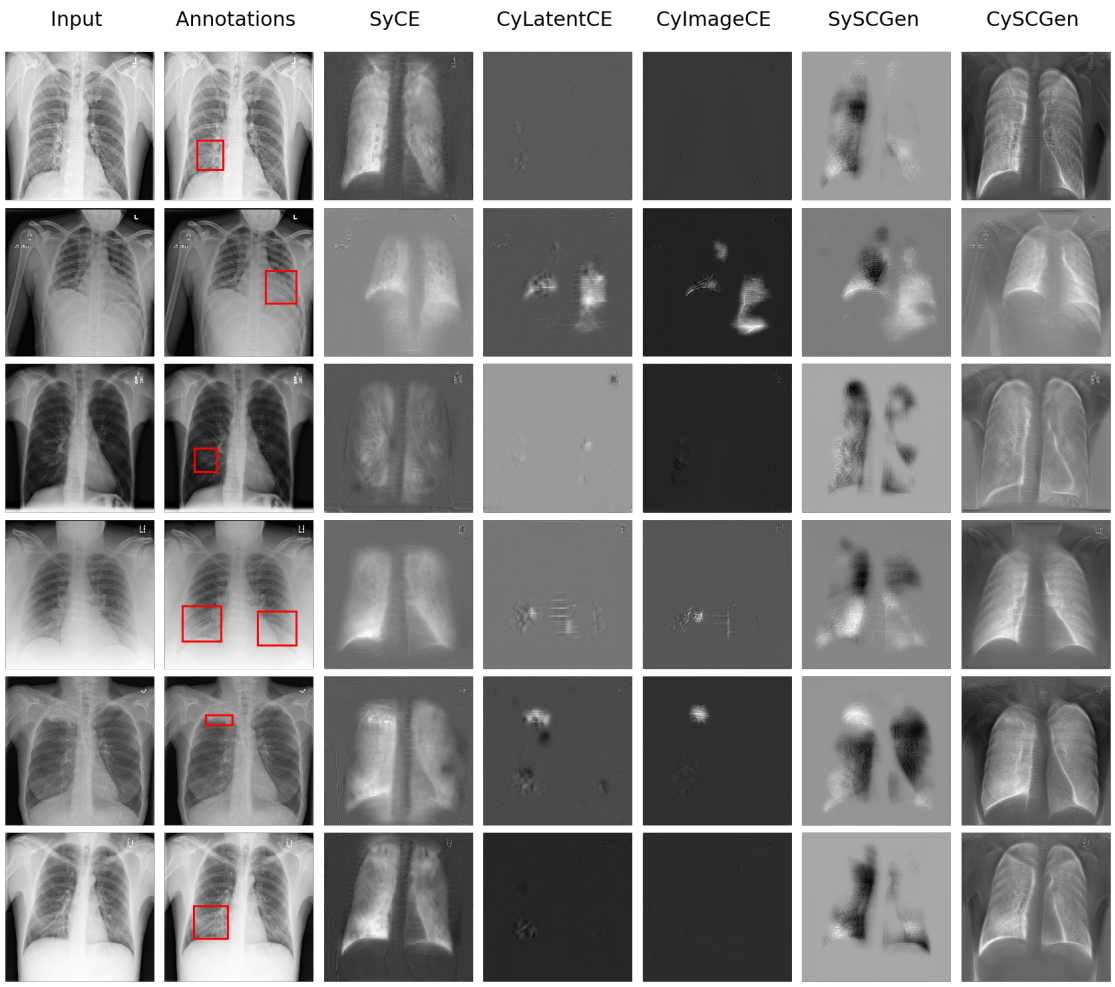


Figure F.4: Pneumonia detection - Difference map of False Negative examples. The first two columns: the input image and the annotated input image. From columns 3 to 7: the difference maps computed for different counterfactual generation techniques, imposing the translation "Pathological to Healthy" (except for SySCGen). Input against SySCGen counterfactual predictions: 0.12 / 0.91 - 0.0 / 0.95 - 0.02 / 0.83 - 0.14 / 0.02 - 0.01 / 0.90 - 0.08 / 0.99. 0 being healthy and 1 pathological.

F.2 Brain tumor detection

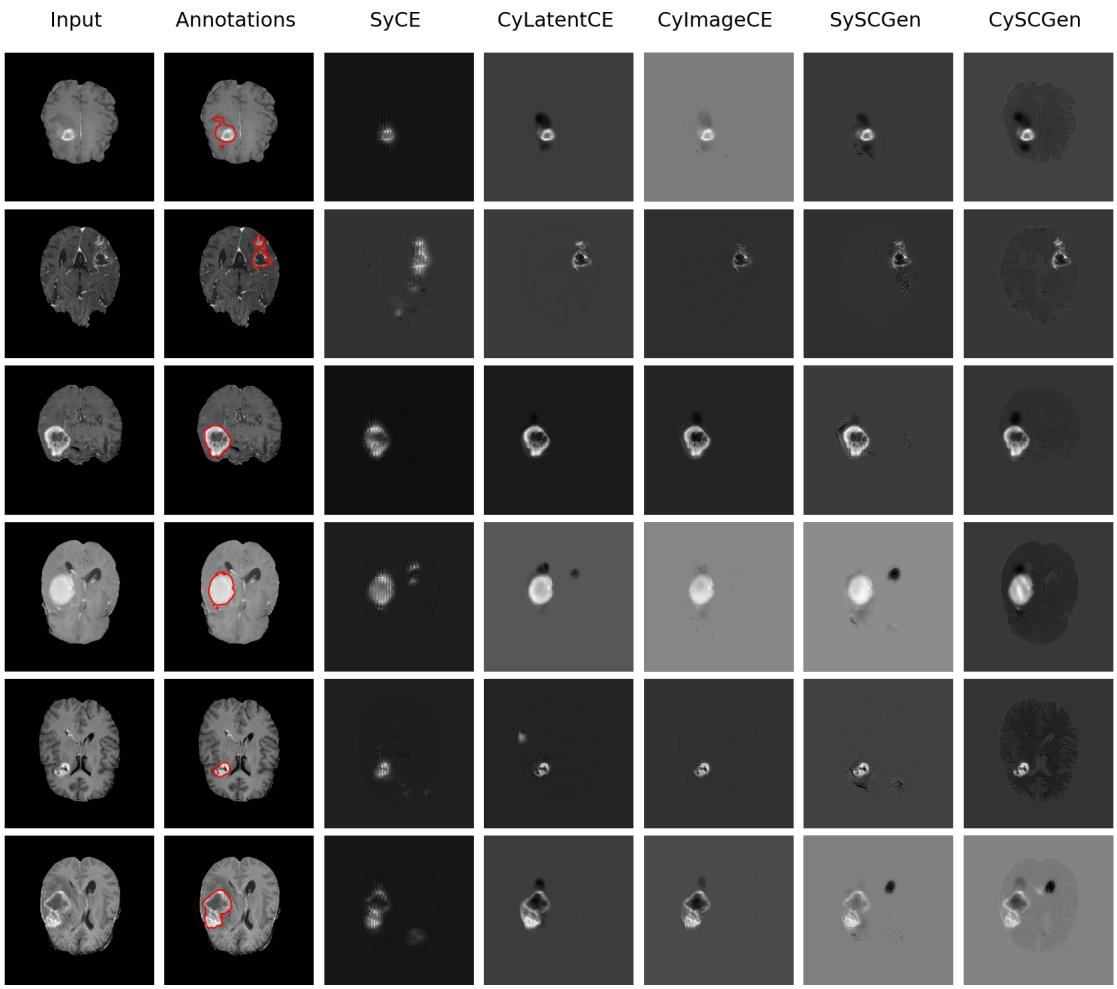


Figure F.5: Brain tumor detection - Difference maps of True Positive examples. The first two columns: the input image and the annotated input image. From columns 3 to 7: the difference maps computed for different counterfactual generation techniques. Input against SySCGen counterfactual predictions: 0.97 / 0.02 - 0.99 / 0.03 - 1.0 / 0.02 - 1.0 / 0.10 - 0.99 / 0.34 - 1.0 / 0.03. 0 being healthy and 1 pathological.

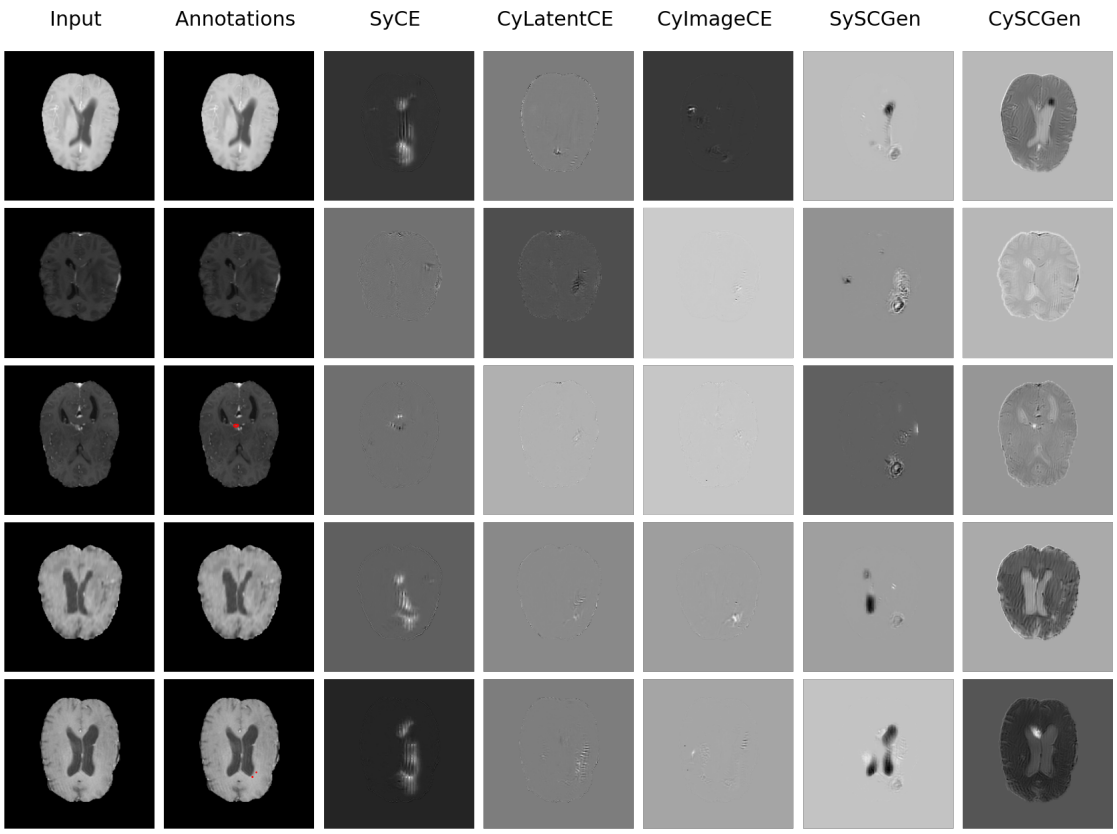


Figure F.6: Brain tumor detection - Difference map of True Negative examples. The first two columns: the input image and the annotated input image. From columns 3 to 7: the difference maps computed for different counterfactual generation techniques, imposing the translation "Pathological to Healthy" (except for SySCGen). Input against SySCGen counterfactual predictions: 0.07 / 0.92 - 0.33 / 0.79 - 0.03 / 0.71 - 0.25 / 0.96 - 0.07 / 0.64. 0 being healthy and 1 pathological.

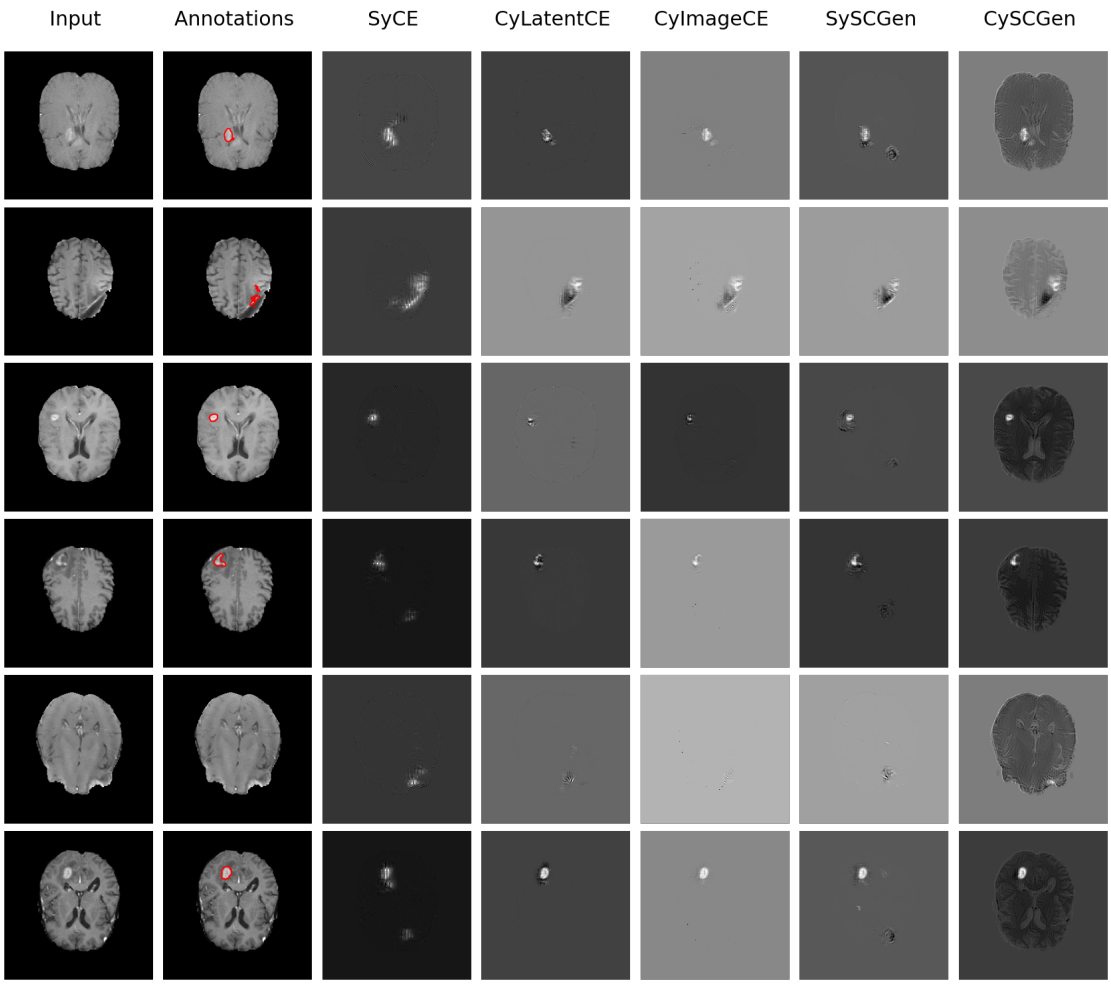


Figure F.7: Brain tumor detection - Difference map of False Positive examples. The first two columns: the input image and the annotated input image. From columns 3 to 7: the difference map computed for different counterfactual generation techniques. Input against SySCGen counterfactual predictions: 0.88 / 0.23 - 0.96 / 0.31 - 0.52 / 0.61 - 0.84 / 0.06 - 0.86 / 0.67 - 0.79 / 0.59. 0 being healthy and 1 pathological.

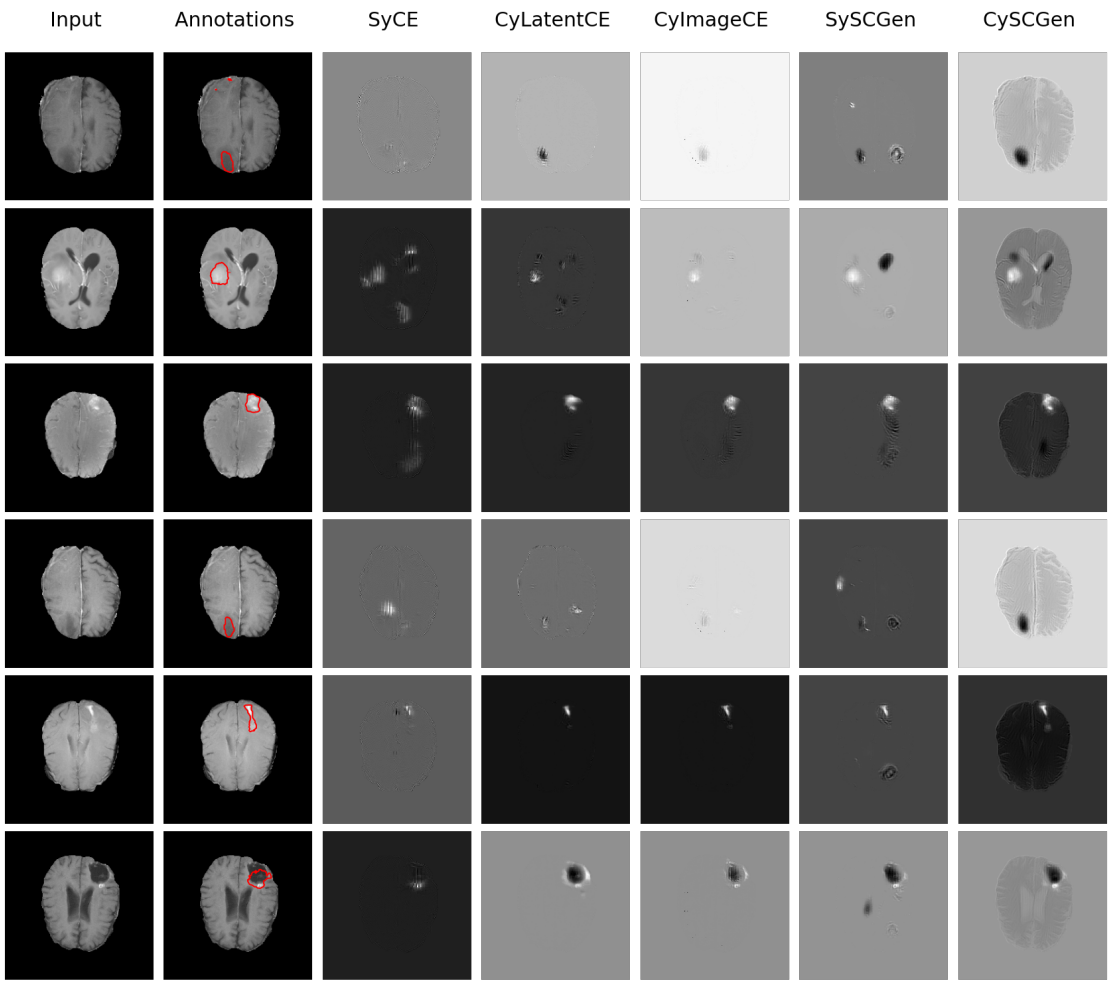


Figure F.8: Brain tumor detection - Difference map of False Negative examples. The first two columns: the input image and the annotated input image. From columns 3 to 7: the difference maps computed for different counterfactual generation techniques, imposing the translation "Pathological to Healthy" (except for SySCGen). Input against SySCGen counterfactual predictions: 0.03 / 0.54 - 0.24 / 0.54 - 0.27 / 0.05 - 0.03 / 0.79 - 0.08 / 0.92 - 0.08 / 0.16. 0 being healthy and 1 pathological.

G

Appendix: Diversity frameworks

We first provide the optimization framework for content and attributes disentanglement, in the case of random attributes sampling (see Figure G.1). Second, we summarize the findings and issues about this strategy:

- **Diversity**: generation diversity is limited when we add or remove pathology. We notice similar observations on celebrity faces tasks.
- **Normalization layers**: Most works use Instance Normalization in the content encoder to eliminate style information and no normalization in the style encoder. Style and content information are combined through conditional normalization or concatenation strategy in the decoder part. When using Instance Normalization in the content encoder, we obtain some diversity but only related to color and intensity information (extracted from a counterfactual example). Without this normalization, diversity remains poor (or absent), and the content encoding might include some class information.
- **Input proximity**: The framework has more optimization terms, and the balance is more challenging to reach. The proximity of the counterfactual to the input often decreases.
- **Attribute space**: The vector space is not clustered along with intra-class variability. As the attributes encoder extracts information from real images, we expect that it captures some diversity and specificity among the pathological domain, e.g., the different texture, intensity, size, and localization. For classifying young against old faces, we would like to see the impact of glasses, hair color, and skin texture.
- **Classification guidance**: Classification losses drive the optimization and enforce the generations to capture relevant insights for the classification model. However, it may also shrink the generation process (especially when adding a pathology) by generating an adversarial pattern or a sufficient pathology that consistently satisfies the classification objective. It prevents the generative process from producing diverse generations with very different pathological structures.

Finally, we describe the asymmetrical optimization framework of the image level disentanglement in Figure G.2.

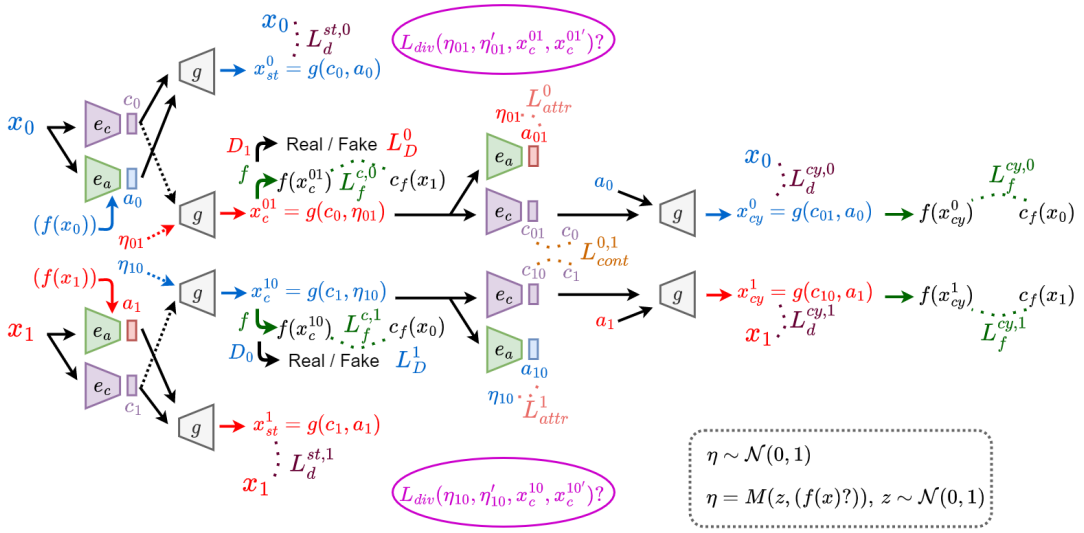


Figure G.1: Overview of Content and Attributes disentanglement framework (version 2). The terms L_i^0 (resp. L_i^1) are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). ***Stable path***: At each step, input images x_0 and x_1 are given to the content e_c and attributes e_a encoders. The attribute encoding can also be conditioned by the input's prediction $f(x_0)$. This generates content and attributes encodings c_0 and a_0. Recombining the two encodings through a generator g (with a decoder structure) produces the stable image $g(c_0, a_0)$. It is enforced to be pixel-wise close to x_0 by the term $L_d^{st,0}$. ***Counterfactual path***: Counterfactual images are produced by sampling random attributes vectors η_{01} and η_{10} (with same dimension as attributes encodings). We thus obtain $x_c^{01} = g(c_0, \eta_{01})$ (resp. $x_c^{10} = g(c_1, \eta_{10})$), the counterfactual image of x_0 (resp. x_1) in the domain χ_1 (resp. χ_0). This generated image is enforced ($L_f^{c\,0,1}$) to be classified in the opposite class $1 - c_f$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D_1 (resp. D_0) that is trained to identify real from generated images in the distribution of images predicted in class 1 (resp 0), i.e., χ_1 (resp. χ_0). Diversity is implicitly encouraged as the attributes vector differs for each sampling. An additional term L_{div} can be added to enforce generation diversity explicitly. ***Cyclic path***: x_c^{01} (resp. x_c^{10}) is also mapped back to χ_0 through cycle consistency. The counterfactual image is first encoded by e_c and e_a. A content loss enforces that the content remains the same across domains, i.e., $c_0 \approx c_{01}$ (for x_0). An attributes term enforces the attributes stability inside a domain (e.g., χ_0), i.e., $\eta_{01} \approx a_{01}$. The cycle image x_{cy}^0 (resp. x_{cy}^0) is generated (through g) by combining the counterfactual encoded content c_{01} (resp. x_{cy}^0) and the initial encoded attributes a_0. Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L_d^{cy,0}$) and ($L_f^{cy,0}$).

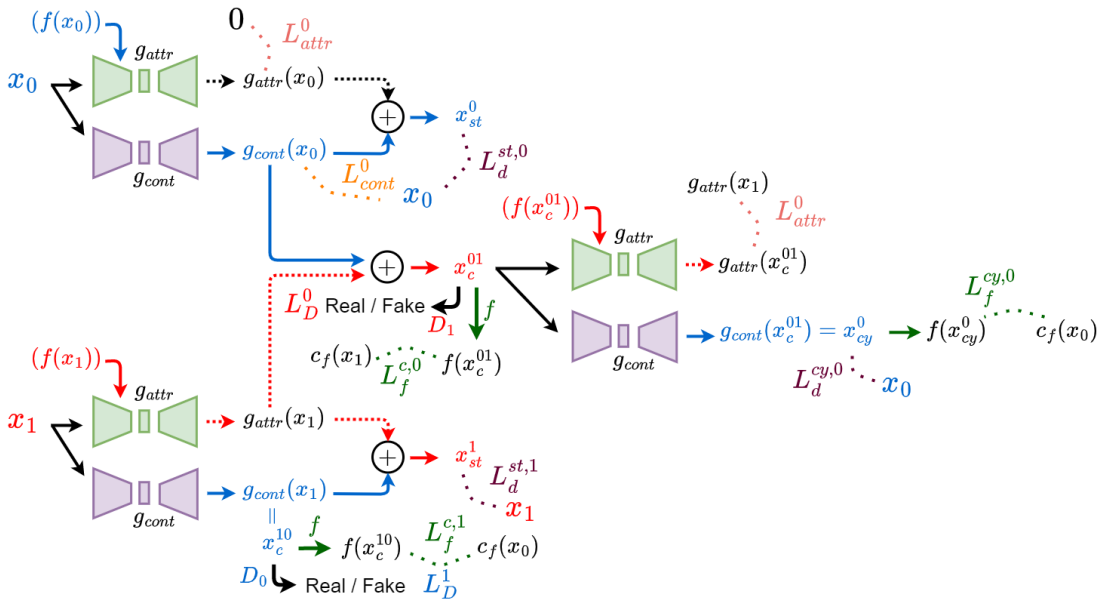


Figure G.2: Overview of the additive Content and Attributes disentanglement framework. The terms L^0_i (resp. L^1_i) are the loss parts L_i that act on $x_0 \in \chi_0$ (resp. $x_1 \in \chi_1$). The optimization framework is asymmetrical. ***Content path***: For input images x_0 and x_1, the content generator g_{cont} extracts the healthy parts of the input, i.e., it generates healthy images. Therefore, $g_{cont}(x_0)$ is enforced to be equal to x_0 (L^0_{cont}) as x_0 is predicted healthy. Instead, $g_{cont}(x_1)$ is directly the counterfactual generation x^{10}_c of x_1. It is enforced ($L^{c\,1}_f$) to be classified in the opposite class $1 - c_f(x_1)$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D_0, trained to identify real from generated images in the distribution of images predicted in class 0. ***Attribute path***: the generator g_{attr} extracts the pathological part from the input image, i.e., the resulting generation does not belong to the image distribution as it only contains pathological structures (emphasized with dashed arrows). Note that we can condition g_{attr} with the initial prediction $f(x)$. As x_0 is predicted in the "healthy" class, $g_{attr}(x_0)$ should not contain pathological elements, and tends to the null images (L^0_{attr}). The stable image $x^0_{st} = g_{cont}(x_0) + g_{attr}(x_0) \approx g_{cont}(x_0)$ is further encouraged to be close to x_0 through $L^{st,0}_d$. It also implicitly enforces the objectives on $g_{cont}(x_0)$ and $g_{attr}(x_0)$. In contrast, $g_{attr}(x_1)$ should extract the pathological regions or x_1, such that we retrieve the stable generation $x^1_{st} = g_{cont}(x_1) + g_{attr}(x_1)$ ($L^{st,1}_d$). ***Pathology addition***: to generate a pathological counterfactual x^{01}_c of x_0, we add the extracted pathology from x_1 (i.e., $g_{attr}(x_1)$) to the content of x_0 (i.e., $g_{cont}(x_0)$). x^{01}_c is enforced ($L^{c\,0}_f$) to be classified in the opposite class $1 - c_f(x_0)$ by f (green arrow). We also enforce the counterfactual image to fool the discriminator D_1 (specific to domain χ_1). ***Asymmetrical cyclic path***: Cycle consistency is only enforced for input images from χ_0. g_{attr} extract the added pathology from the counterfactual x^{01}_c, which is enforced to be close to the pathology from x_1, i.e., $g_{attr}(x_1)$ (through L^1_{attr}). g_{cont} perform the cyclic generation retrieving the healthy content from x^{01}_c, which is should be x_0. Pixel-wise proximity and classification consistency to x_0 are encouraged by the constraints ($L^{cy,0}_d$) and ($L^{cy,0}_f$).

H

Appendix: Synthèse en français

H.1 Au carrefour de la radiologie et de l'intelligence artificielle

Ces dernières années, les performances en imagerie médicale n'ont cessé d'augmenter. Alors qu'il fallait environ 30 minutes pour obtenir 40 images médicales en 1980, les machines peuvent aujourd'hui produire plus de 1000 images en 4 secondes. En parallèle, la multiplication des modalités (radiographie, scanner, échographie, IRM, TEP, ...) permet aux cliniciens de disposer de données supplémentaires et complémentaires pour étayer leur diagnostic.

La lecture d'un examen médical consiste à parcourir plusieurs volumes d'acquisition (images 2D ou 3D) potentiellement disponibles dans différentes modalités et avec différentes techniques. Pour les modalités 3D, par exemple, les cliniciens décomposent le volume en naviguant dans les différents axes d'acquisition (le plus souvent axial, coronal et sagittal). Lors de cette phase d'analyse, le radiologue rapporte tous les éléments correspondant à l'indication clinique, ainsi qu'un ensemble d'éléments appelés "incidentalomes", qui correspondent à des découvertes fortuites. Les cliniciens peuvent également étudier les rapports radiologiques antérieurs, s'ils sont disponibles et recommandés pour la prise de décision, par exemple en suivant l'évolution (dans le temps) d'une maladie, d'un traitement ou d'une opération. Ensuite, les cliniciens doivent synthétiser et reformuler tous ces éléments dans une conclusion conduisant à un diagnostic clinique. En temps réel, les cliniciens rapportent souvent ces éléments à un système d'acquisition homme-machine par l'intermédiaire d'un dictaphone. Ce processus augmente la probabilité que la nomenclature et la syntaxe dépendent fortement du radiologue. Dans certains cas, notamment en cas d'urgence, un radiologue tiers peut répéter ces étapes pour confirmer ou infirmer le diagnostic. L'analyse de toutes ces données prend du temps, surtout avec les méthodes traditionnelles. Elle peut être source d'erreurs importantes, notamment en cas de fatigue et de stress accrus, ou en fonction de la personne qui réalise l'examen, par exemple un médecin urgentiste, un radiologue expert, ou un radiologue non expert (expert de la tâche particulière en jeu).

Dans le domaine médical, et plus particulièrement en radiologie, les solutions d'intelligence artificielle (IA) peuvent avoir un impact sur l'ensemble du flux de travail, de l'acquisition de l'image (par exemple, en améliorant la qualité de l'image ou en permettant de réduire la dose d'agents de contraste) au diagnostic (par exemple, en détectant la pathologie et en améliorant les visualisations) ou même au pronostic du traitement.
La plupart de ces applications médicales sont développées à l'aide de modèles d'apprentissage profond – un sous-domaine de l'apprentissage automatique qui est lui-même un domaine de l'intelligence artificielle. Les modèles d'apprentissage profond représentent l'état de l'art pour de nombreuses tâches de vision par ordinateur (en particulier dans le domaine

des images médicales [Bien 18b, Esteva 17]). En analyse d'images médicales, ces tâches sont principalement :

- La classification soit pour détecter dans une image donnée la présence ou l'absence d'une lésion (par exemple), soit pour catégoriser entre les différentes lésions possibles.
- La segmentation pour localiser des structures spécifiques (par exemple, des organes, des pathologies) au niveau du pixel. Pour les modèles d'apprentissage automatique, elle consiste à classer chaque pixel pour déterminer s'il appartient ou non à une structure donnée. À un niveau plus faible, les modèles de localisation fournissent des boîtes comprenant la structure ciblée.
- La modélisation générative (utilisant des architectures telles que Autoencoder, Autoencoder variationnel, Generative Adversarial Networks, Normalizing Flows). Nous pouvons adopter ces techniques pour améliorer la qualité des images (par exemple, reconstruction d'images, débruitage) ; pour générer de "nouvelles" images afin d'augmenter/enrichir la base de données d'apprentissage ; pour traduire les images d'une modalité à l'autre (par exemple, CT à MRI), afin de transférer la capacité d'un modèle (par exemple, un modèle de segmentation) appris sur la première modalité dans l'autre. [Liu 21] décrit de telles applications.

Les modèles d'apprentissage profond s'adaptent bien aux problèmes d'image par rapport aux approches d'apprentissage automatique précédentes (par exemple, modèle linéaire, arbre de décision, forêt aléatoire ou modèle basé sur des règles). Les lecteurs peuvent se référer à [Goodfellow 16, Egger 22] pour une vue d'ensemble des méthodes d'apprentissage profond.

H.2 L'IA explicable : un besoin essentiel

L'entreprise Incepto construit en cocréation et distribue des outils d'intelligence artificielle pour l'analyse d'images médicales. Ces outils utilisent majoritairement des techniques par apprentissage profond qui obtiennent des résultats de plus en plus performants [Bien 18a, Esteva 17]. Cependant, ces approches utilisent des modèles avec des architectures complexes et ayant des milliers (voire des millions) de paramètres, les rendant peu explicables et/ou interprétables. Il est devenu un impératif de fournir une explication compréhensible des résultats de ces solutions d'IA ; à la fois par les radiologues, mais aussi par les concepteurs de ces solutions (en vue d'une amélioration en continue). Il est essentiel de générer la confiance autour de ces outils pour faciliter leur intégration et leur utilisation en routine clinique.

L'explication est particulièrement critique dans le cadre d'un problème de classification, où le modèle entraîné par apprentissage profond fournit une décision en prédisant un (ou plusieurs) choix, par exemple, “Présence” ou “Absence” de pathologie. Cependant, cette décision est donnée sans aucune justification ou argumentation, contrairement à la pratique des cliniciens. Les modèles de classification par apprentissage profond apparaissent comme des boîtes noires car l'utilisateur ne connaît pas le raisonnement du modèle, les régions (pertinentes ou confondantes) de l'image étudiée qui soutiennent la décision, ou le type de structures appris au cours de l'apprentissage. L'explicabilité des décisions n'a pas nécessairement les mêmes enjeux pour d'autres tâches, comme la segmentation ou la localisation, où les modèles génèrent des sorties visuelles que l'utilisateur (dans notre cas le clinicien) peut vérifier avant accepter ou rejeter les résultats produits. L'explication de

la prédiction du modèle est donc moins critique. Cependant, il est beaucoup plus simple d'obtenir des annotations de classification que des masques de segmentation (et cela prend moins de temps). Ceci explique pourquoi la classification est la tâche la plus courante. En analyse d'images médicales, les modèles de classification sont notamment appliqués pour détecter des anomalies dans les images (par exemple, une pathologie, des lésions, un artefact métallique) ou pour identifier le type de pathologie (parmi différentes classes).

Chez Incepto, KEROS appartient au premier groupe de méthodes puisqu'il vise à détecter la présence ou l'absence de lésions sur chaque structure du genou (entraînement d'un modèle de classification spécifique par structure). Pour ce type de tâche, nous aimerions que la méthode d'explication (i) mette en évidence si le modèle s'appuie sur des caractéristiques cliniques pertinentes ; et (ii) montre quels motifs/structures ont été apprises par le modèle pour différencier une classe par rapport à l'autre.

Les décisions prises sur des images médicales résultent de l'investigation visuelle de l'image. Nous nous sommes donc essentiellement concentrés sur l'élaboration d'explications visuelles des décisions d'un modèle de classification (entraîné), et non à l'explication du raisonnement interne du modèle. On parle dans ce cas d'explications "post-hoc". De nombreuses méthodes ont été proposées pour mettre en avant les régions de l'image qui supportent la décision du modèle, sous forme de cartes d'attribution (ou cartes de chaleur). Les premières approches sont basées sur des techniques de rétropropagation du gradient [Simonyan 14, Smilkov 17, Sundararajan 17], ou sur l'analyse des dernières couches du modèle de réseau de neurone [Zhou 16b, Selvaraju 17]. Cependant, ces méthodes ne sont souvent pas agnostiques aux modèles, car nous devons avoir accès structures internes du modèles (gradients et/ou couches), et produisent des visualisations souvent bruitées [Simonyan 14], ou grossières [Zhou 16b]. D'autres approches s'intéressent plutôt aux perturbations de l'image analysée [Fong 17, Dabkowski 17, Lenis 20, Fong 19]. Ces méthodes nécessitent des régularisations heuristiques pour produire des cartes de visualisation acceptables pour les humains. Elles nécessitent des adaptations manuelles importantes lors du changement de modèle de DL, ou le changement de tâche de classification. Toutes ces méthodes ont aussi été développées dans le cadre d'images naturelles et non médicale. Or, les images médicales partagent souvent un contenu similaire (par exemple le fond, les structures du corps) et diffèrent plutôt sur des motifs localisés. Ce n'est en général pas le cas pour des images naturelles (ex : chat, chien, voiture ...) avec des objets à identifier souvent très contrasté en premier plan, avec une localisation variable dans l'image, et des arrières plans très variés et avec peu de contexte. Les méthodes développées pendant cette thèse sont adaptées aux spécificités des problèmes d'imagerie médicale. Enfin, la majeures parties des méthodes de l'état de l'art produisent des visualisations pour mettre en avant les régions qui supportent la décision du modèle, mais n'étudie pas le type de structures qui ont été apprises ou qui diffèrent entre les classes.

Pour formaliser le développement de nos explications visuelles, nous rappelons tout d'abord qu'elles sont destinées aux utilisateurs finaux (les cliniciens), ainsi qu'aux développeurs des modèles afin d'identifier des faiblesses et de mettre en places des stratégies d'amélioration. La méthode d'explication devrait :

- Mettre en évidence les régions de l'image analysée pertinentes pour la décision du modèle de classification (carte d'attribution)
- Montrez comment ces régions devraient être modifiées pour produire une décision de différente tout en restant dans la distribution des données (exemple contrefactuel)
- Permettre d'identifier les signes et structures cliniques ou les biais appris par le modèle
- Rester (autant que possible) agnostique aux modèles de classification, pour être aussi

capable d'investiguer les solutions des partenaires d'Incepto.

H.3 Méthode

Dans les problèmes d'images médicales, et en particulier pour la détection de pathologies, les cliniciens recherchent des signes cliniques pour décrire les images et établir leur diagnostic (par exemple, une déchirure dans les ménisques, une opacité dans le thorax ou des tissus tumoraux dans le cerveau). La classification dépend de la présence ou de l'absence de ces motifs spécifiques. Nous supposons que la décision du modèle dépend également de la présence ou de l'absence de certaines structures (par exemple, des signes cliniques, corrélés ou confondants). Ainsi, nous proposons d'expliquer la décision du modèle en générant une image (très similaire à l'image analysée) qui change la décision du modèle. Pour respecter les objectifs fixés pour notre explication, cette image générée doit donc uniquement transformer les régions de l'image impactante dans la décision, par exemple en générant ou en supprimant une tumeur dans le cerveau. Cependant, pour générer de telles structures tout en maintenant l'image générée dans la distribution des images réelles, nous ne pouvons pas utiliser de perturbations synthétiques. Notre méthode d'explication nécessite l'accès à une base de données pour apprendre à générer ces images dites contrefactuelles. Pour produire ces explications visuelles, nous proposons une formulation générale qui s'appuie sur la génération de deux images: une stable et une contrefactuelle, appartenant à la distribution des données. Ces images étant classées similairement et différemment de l'image analysée, respectivement. Pour produire ces deux images, notre processus de génération doit respecter trois propriétés:

- **Pertinence**: L'explication visuelle doit mettre en évidence les régions de l'image analysée importantes et impactantes pour le modèle. Ainsi, les générations stable et contrefactuelle ne doivent différer que dans les régions qui sont pertinentes pour la classification.
- **Régularité**: Pour garantir la propriété de pertinence, les deux processus de génération doivent être comparables pour éviter les différences indépendantes du modèle de classification. En particulier, le bruit résiduel imputable au processus de génération doit être minimisé. D'où, l'utilisation de la génération stable qui est en fait la reconstruction de l'image analysée, mais ayant passé par le processus de génération. Si les deux processus sont comparables, le bruit résiduel de génération le sera aussi, et donc disparaîtra lorsqu'on fera la différence.
- **Réalisme**: Les générations stables et contrefactuelles doivent être réalistes dans le sens où elles doivent appartenir à la distribution des images réelles du problème étudié. Cette propriété est essentielle pour éviter (i) les artefacts générés par des exemples adversariaux [Goodfellow 15, Madry 18] ; (ii) les perturbations synthétiques qui produiraient des images complètement en dehors de la distribution des images réelles. Cette propriété révèle également les structures spécifiques (existantes dans la distribution) qui influencent le modèle. Ainsi, notre méthode exploite les techniques de transposition de domaine [Zhu 17, Choi 18] pour produire images, appartenant à la distribution des données.

Ainsi, notre explication visuelle est composée de (i) cet exemple contrefactuel, montrant les transformations réalistes qui différencient les décisions du modèle ; (ii) et une carte d'attribution basée sur la différence entre les deux images générées (stable et contrefactuelle). Cette carte d'attribution met en avant les régions de l'image les plus pertinentes pour le modèle. Les valeurs de la carte d'attribution, produite par notre explication vi-

suelle, doivent traduire l'importance pour le modèle au niveau du pixel ou à toute autre échelle. C'est-à-dire que les régions présentant les valeurs les plus élevées dans la carte d'attribution doivent être les plus pertinentes pour le modèle de classification. Pour remplir ce dernier objectif, nous encourgeons une dernière propriété s'appliquant explicitement sur la carte d'attribution: les régions mises en avant doivent être **Ordonnées par importance** pour le modèle de classification.

Nous proposons différentes implémentations de la formulation générale en ajoutant incrémentalement les propriétés. Nous validons notre méthodologie par des expériences exhaustives sur deux problèmes d'imagerie médicale. Nous démontrons que nos méthodes (i) surpassent les techniques d'attribution de l'état de l'art sur plusieurs métriques d'évaluation, (ii) peuvent identifier les biais dans l'entraînement du modèle et fournir des indications pour l'améliorer, (iii) et peuvent être étendues à d'autres problèmes de classification satisfaisant certaines contraintes.

université
PARIS-SACLAY