Towards explainable and interpretable deep neural networks

Guillaume Jeanneret Sanmiguel

Normandie Université

UNIVERSITÉ CAEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**Université de Caen Normandie**

Towards Explainable and Interpretable Deep Neural Networks

Présentée et soutenue par

GUILLAUME JEANNERET SANMIGUEL

Thèse soutenue le 25/09/2024

devant le jury composé de :

M. FREDERIC JURIE	Professeur des universités - Université de Caen Normandie (UCN)	Directeur de thèse
MME CORDELIA SCHMID	Directeur de recherche à l'INRIA - INRIA Paris	Président du jury
MME EWA KIJAK	Maître de conférences HDR - IRISA/INRIA Rennes	Membre du jury
M. LOIC SIMON	Maître de conférences - Université de Caen Normandie (UCN)	Membre du jury
M. MATTHIEU CORD	Professeur des universités - Sorbonne Université	Rapporteur du jury
MME ANTITZA DANTCHEVA	Chargé de recherche HDR - CNRS	Rapporteur du jury

Thèse dirigée par **FREDERIC JURIE** (Groupe de recherche en informatique, image et instrumentation de Caen)

MI/IΣ

GREYC
Laboratoire de recherche en sciences du numérique

Abstract

Towards Explainable and Interpretable Deep Neural Networks

by Guillaume JEANNERET SANMIGUEL

English Version: Deep neural architectures have demonstrated outstanding results in a variety of computer vision tasks. However, their extraordinary performance comes at the cost of interpretability. As a result, the field of Explanable AI has emerged to understand what these models are learning as well as to uncover their sources of error. In this thesis, we explore the world of explainable algorithms to uncover the biases and variables used by these parametric models in the context of image classification. To this end, we divide this thesis into four parts. The first three chapters proposes several methods to generate counterfactual explanations. In the first chapter, we proposed to incorporate diffusion models to generate these explanations. Next, we link the research areas of adversarial attacks and counterfactuals. The next chapter proposes a new pipeline to generate counterfactuals in a fully black-box mode, *i.e.*, using only the input and the prediction without accessing the model. The final part of this thesis is related to the creation of interpretable by-design methods. More specifically, we investigate how to extend vision transformers into interpretable architectures. Our proposed methods have shown promising results and have made a step forward in the knowledge frontier of current XAI literature.

Version en français: Les architectures neuronales profondes ont démontré des résultats remarquables dans diverses tâches de vision par ordinateur. Cependant, leur performance extraordinaire se fait au détriment de l'interprétabilité. En conséquence, le domaine de l'IA explicable a émergé pour comprendre réellement ce que ces modèles apprennent et pour découvrir leurs sources d'erreur. Cette thèse explore les algorithmes explicables afin de révéler les biais et les variables utilisés par ces modèles de boîte noire dans le contexte de la classification d'images. Par conséquent, nous divisons cette thèse en quatre parties. Dans les trois premiers chapitres, nous proposons plusieurs méthodes pour générer des explications contrefactuelles. Tout d'abord, nous incorporons des modèles de diffusion pour générer ces explications. Ensuite, nous lions les domaines de recherche des exemples adversariaux et des contrefactuels pour générer ces derniers. Le suivant chapitre propose une nouvelle méthode pour générer des contrefactuels en mode totalement boîte noire, c'est-à-dire en utilisant uniquement l'entrée et la prédiction sans accéder au modèle. La dernière partie de cette thèse concerne la création de méthodes interprétables par conception. Plus précisément, nous étudions comment étendre les transformeurs de vision en architectures interprétables. Nos méthodes proposées ont montré des résultats prometteurs et ont avancé la frontière des connaissances de la littérature actuelle sur l'IA explicable.

Acknowledgements

In these three years, I've faced many challenges, but I've also made some amazing friends and learned so much about myself and about French culture. I've had to adjust to a new culture, a new city, a foreign language, conference deadlines, the French bureaucracy, and being 7 hours away from my family and friends. It's been tough at times, but it's also been an incredible journey of growth and discovery. I'm so grateful for all the experiences I've had and all the wonderful people I've met along the way.

I'd like to start by thanking my wonderful co-bureaux and ex-neighbors Rodrigo Maulen and Jean Jacques Godeme. I've had the pleasure of working with them for two consecutive years, and I'm so grateful for their friendship and support. We had so many fun times together, even when we had to work! I'd also like to thank Ryan Webster and Benjamin Sykes for all the great chats we had. I'd also like to thank Julien Mendes-Forte, Kirill Milintsevich, and Navneet Agarwal for all the memes, coffees, beer nights, and board games we shared together. And finally, I want to thank all of you wonderful members of the GREYC laboratory - team Image. You are the ones who bring life to this place, and I am so grateful to have you all here.

I'd like to thank my supervisor, Frédéric Jurie, and Loïc Simon, who have been there for me every step of the way. I really couldn't have done it without your help! Thanks to your supervision and in-depth discussions, my research was of a really high standard. I'd also like to thank them for their warm hospitality when I first arrived in Caen. It's a city that will always have a special place in my heart.

And finally, I'd like to thank my father, Claude, my mother, Lia, my brothers, Francois and Philippe, and my family for all their love and support over the years. They were always there for me, offering words of wisdom. And finally, I'd like to thank Carolina, my amazing wife, for being my biggest supporter when I needed it most. She's the reason I'm who I am today.

Finally, I would like to thank the founding agency that made this possible. This thesis was supported by the Agence Nationale pour la Recherche (ANR) under award number ANR-19-CHIA-0017.

Dedicada a mis padres, mis hermanos y mi esposa…

Chapter 1

Introduction

1.1 Context

Image-based deep learning has seen exponential growth over the past decade. The pioneering work of Krizhevsky, Sutskever, and Hinton [97] introduced AlexNet, the first deep convolutional neural network (CNN), which significantly reduced the top 1 and top 5 classification errors on the 1000-way classification dataset ImageNet [35]. From that point on, traditional machine learning methods were almost discarded and researchers adopted neural architectures to explore solutions to all kinds of problems.

In a nutshell, neural networks are parametric functions that operate through a sequence of linear and nonlinear differentiable operations. To optimize the function's parameters, the weights are adjusted to minimize a loss function, which quantifies the disparity between the model's prediction and the target. This optimization process involves calculating the gradients of the loss function with respect to the model parameters and updating these parameters using the gradient descent algorithm. This process is commonly referred to as training or fitting a neural network.

Research has scaled up neural models to include billions of parameters to fit colossal datasets in recent years. Progress has reached the point where deep architectures are being used for virtually all computer vision tasks and beyond because of their superior performance, even surpassing humans in most tasks. However, this gain in performance comes with a drawback in terms of interpretability. The complex nature of the inner nonlinear operations prevents the decryption of the decision-making mechanisms and, hence, understanding the learned variables is nearly impossible. Ideally, we would like to know why a particular decision was made and how it can be changed to produce a different result. This scenario would shed light on the weaknesses of a target model and thus allow countermeasures to overcome any side effects. Therefore, the research field of Explainable Artificial Intelligence (XAI) emerged to *open the black box* and searches to give some insights about this problem.

XAI is driven by two concepts: explainability and interpretability. While some researchers do not distinguish between the two, explainability and interpretability

are not interchangeable. So, we settle on a common ground and define them in the context of machine learning. An *explanation* is a way to clarify how a prediction was made by a particular model, via *external methods*. As noted by Wachter, Mittelstadt, and Russell [181], an explanation has three main purposes: (i) to comprehend *why* a model predicted a particular outcome, (ii) to establish a *common ground* where multiple outcomes could be produced by the model, and (iii) to understand what could *change* in the input to produce a particular prediction. On the other hand, interpretability is not well established and can vary in the literature. In this thesis, we define this concept as the ability of the model to show, by *itself*, the decision-making process of the prediction or the learned knowledge in a transparent and human-friendly way [119].

Both interpretability and explainability approaches unravel the causal factors of a phenomenon, yet this could be achieved using multiple and diverse perspectives. So, the literature has shown different methodologies to approach the problem of interpretability and explainability in computer vision, which we will go into detail later in chapter 2. Common techniques encompass saliency maps that highlight where the model is looking, prototype-based approaches to tell how an image *looks like*, concept discovery to resume a list of notions that define a particular class, and counterfactual explanations to flip the model predictions via intuitive and understandable modifications.

1.2 Scope of the Thesis

Two main lines of research emerge along the concepts of explainability and interpretability. Related to the former, explanation generation algorithms are referred to as *Post-Hoc* explanations. These methods design a strategy to open the inner workings of the architecture under observation. In this thesis, we did not study the whole spectrum, as it is too diverse. Rather, we focused on one particular type of explanation: Counterfactual Explanations [181] (CE) for image classification in chapter 3 to chapter 5.

A CE seeks to answer the following question: *Given a model and an input, what meaningful and minimal change could be made to the input such that the model changes its original prediction?* By analyzing these changes, we can infer what variables the model is using for its prediction. While these changes could be performed in a variety of forms, counterfactuals seek three main properties in order to remain useful: realism, proximity, and diversity. However, to fulfill the aforementioned properties in the image domain, we require powerful generative approaches capable of producing realistic and diverse images. To this end, we have employed diffusion models. These generative models are trained to produce images by progressively removing noise from an instance extracted from a Gaussian distribution.

The second line of research focuses on interpretable-by-design (ID) architectures. As the name implies, these methods are closely related to our definition of interpretability. These algorithms produce predictions while exhibiting interpretable features, thus aiding the analysis of their decision-making process. We observed that most of the literature on ID methods is built for CNNs rather than vision transformers. While CNNs have been the preferred architectures in the past, next-generation neural networks are more frequently using transformer backbones. Consequently, there is a compelling need to develop transformer-based ID architectures. In chapter 6, we took a step forward in this area by developing transformer-based ID architectures for image classification.

1.3 Thesis Outline

Based on the framework outlined above, we now present the structure of this thesis.

Chapter 3 To generate CEs for image data, generative models are typically used as a regularization constraint, serving as an asset in creating the explanation. Therefore, it is crucial to have a generative approach that produces realistic changes. In 2020 and 2021, diffusion models [64] gained significant popularity, promising excellent generative capabilities. As a result, we proposed **DiME**, the first approach to generate counterfactual explanations based on diffusion models. In essence, DiME adapts the guided diffusion proposed by Dhariwal and Nichol [36] to incorporate the classifier under investigation.

Contributions:

- DiME is the first approach to generating counterfactual explanations based on diffusion models.
- We proposed a new metric to evaluate an explanator's ability to uncover spurious correlations learned by the model.
- We proposed a metric to evaluate the diversity of an explanator.

Chapter 4 From the previous work, we began to notice other uses of diffusion models. For example, DDPM can be used to remove adversarial noise [132]. In addition, the literature has shown that adversarial attacks on robust models would generate plausible features in the image. So in chapter 4, we linked the world of adversarial examples and counterfactual explanations, and proposed **ACE**. ACE operates with adversarial examples in the classical fashion to generate the CE. Rather than applying the adversarial perturbation directly to the classifier's input, our approach generates these perturbations prior to a pre-processing step via a diffusion model's

denoising process, enhancing the robustness of the target model. Consequently, the resulting output serves as a CE.

Contributions:

- We propose a novel approach to generating counterfactual explanations based on adversarial examples, effectively linking adversarial examples and counterfactuals.
- To assess our method in general data types, we extend some of the evaluation metrics to broader domains.
- We prove that ACE provides meaningful explanations by generating real-world modifications that confound the classifier's prediction.

Chapter 5 Our previous work and concurrent literature have focused on guiding counterfactual generation based on gradients through the denoising chain of diffusion models. However, these methods are computationally expensive, so there is a need to reduce this burden. To address this, in chapter 5 we proposed **TIME**. In short, we distilled the classifier's information into Stable Diffusion [151] in the form of text tokens. This approach allows us to accelerate the CE generation process by simplifying it into a straightforward denoising operation.

Contributions:

- We propose TIME, a text-to-image approach for counterfactual explanations.
- Our algorithm has the advantage of being completely black box, *i.e.* it does not rely on any feature of the model, but rather operates on the classification prediction and the input.
- TIME does not operate in the traditional setting of optimization to generate the explanation. Rather, it inputs the modification via perfect DDIM inversion [183].

Chapter 6 Finally, we shifted our focus from the counterfactual domain to the Interpretable-by-Design (ID) domain. While transformers have emerged as the architecture of choice due to their superior performance over CNN backbones, ID architectures are predominantly designed for CNNs. To address this gap, in chapter 6 we propose **HiT**, a classification model that decomposes the classification token into the contributions of each patch token, by disentangling the multi-headed attention mechanism [179], the core function in transformers.

Contributions:

- We propose an ID-transformer-like architecture that produces the importance of each token intrinsically for the final prediction.
- To do this, we disentangled the self-attention mechanism so that we could decompose the classification token as the sum of individual tokens.

1.4 Publications

According to the plan outlined above, we have managed to publish four papers; three in internationally renowned conferences and one in a journal. In addition, one article is under review. Here are our papers:

- Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Diffusion Models for Counterfactual Explanations". In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 2022.
- Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Diffusion Models for Counterfactual Explanations". In: *Computer Vision and Image Understanding*. 2024.
- Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Adversarial Counterfactual Visual Explanations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.
- Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Text-to-Image Models for Counterfactual Explanations: a Black-Box Approach". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024.
- Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Disentangling Visual Transformers: Patch-level Interpretability for Image Classification". *Under Review*. 2024.

Finally, this thesis is a based on our work published during these three years. To complement the content of each chapter, we have included a prologue and an epilogue to link all previous studies, provide some context, and state how the method relates to the current literature.

Chapter 2

Literature Review

Current XAI research can be organized into two groups: *Post-Hoc* explainability algorithms and interpretability by-design architectures. The former seeks to create algorithms to analyze a trained model, while the latter are models that create the explanation and the decision simultaneously. In this chapter, we will synthesize the current technological advances in the XAI community for both branches, along with their advantages and drawbacks. To this end, we will discuss several axes: *Post-Hoc* explanations in section 2.1, counterfactual explanations in section 2.2, and interpretable by-design (ID) architectures in section 2.3.

2.1 Post-Hoc Explanation

Formally, a *Post-Hoc* explainability method ε takes as input an architecture f, and an image x to produce an explanation e describing the outcome $f(x)$ of the model:

$$e = \varepsilon(f, x). \tag{2.1}$$

In addition, e is not restrained to a specific kind of data: it could be a textual explanation, an image, a heat map, or any combination.

The literature presents several promising explanation technologies. One primary way to categorize these methods is by distinguishing between black box, model-agnostic, and model-specific algorithms. First, black box methods allow a model to be analyzed while not having access to the weights, architectural details, or other internal features of the model. Instead, they only have access to the input and output labels. This provides a certain level of protection, such as avoiding data leakage through gradients [209]. Second, model-agnostic methods provide explanations for any architecture, but unlike black-box methods, they have access to the model's architecture, such as the probability distribution generated by the model or hidden features. Finally, model-specific algorithms are tailored to a particular model and leverage all of its features. These algorithms are more precise because they take advantage of all the specifications that make up a precise architecture. In our opinion,

if the goal is to certify the correct behavior of the model under investigation, restricting access to the model's internal workings creates an unnecessary hurdle.

Another way to distinguish *Post-Hoc* methods is between global and local explanations. Global *Post-Hoc* explanation algorithms look for a bird's eye view of the model's decision boundary and the used features. These methods are attractive because they summarize the model behavior in simple descriptions. However, today's complex models use a plethora of variables for their decision, so, extracting simple and interpretable rules can be challenging. Thus, a desired simple rule may not hold for every data sample, leading to building more complex rules. On the contrary, local explanations search to create simple descriptions of a unique instance. Clearly, local explanations are simpler than global ones as the justification accounts for the sample and not a group of instances. While this is beneficial locally, the explanation cannot be extrapolated to other instances.

Among the exhaustive list of algorithms, we highlight some of the most widely used in computer vision. To begin with, saliency maps [206] - also known as class activation maps (CAMs) - are the most popular approach to explain a vision architecture. These local explanations consist of creating a heat map to highlight the most important pixels used by the model for the final prediction. Formally, the explanation shows elevated values for the most influential pixels. In contrast, negative values indicate inhibiting features. To build these CAMs, the vast majority of methods [76, 27, 161, 101, 185, 170] use the gradients of the predicted class with respect to some features along with some post-processing steps. A small part of the literature has approached this problem by proposing different mechanisms such as Shapley values [205, 112], random [141] or super-pixel [148] perturbations, or layer-wise relevance propagation techniques [14]. In addition, there are derivatives of these maps that produce two maps between an image and a counterimage with a different prediction [177, 56]. These maps show what should change between instances so that the original image changes its prediction to the counter-image one.

CAMs provide only partial information about which features the model is using for its prediction because these explanations only highlight a region of the image and do not indicate which features were the triggering ones. As an example, we point the reader to Figure 2.1, which shows a CAM of a dog. From this CAM, the user could infer that the decision was based on the upper half of the dog's physiological features. However, this interpretation could be flawed because the user is favoring his or her beliefs over the causal factor on which the model's prediction is based, *e.g.*, a texture bias [51]. This phenomenon is called confirmation bias [2].

FIGURE 2.1: The CAM highlights the dog's top, but doesn't reveal specific features used for prediction. Example from [48].

Another branch of research in *Post-Hoc* methods is concept attribution. These algorithms can be categorized into

global or local *Post-Hoc* methods. The main goal of this field is to find the variables used by the classifier in the form of concepts [87] to give a global view of the features used by the model or to resume the decision as a list of used concepts. For example, Kim et al. [87] builds a dataset of concepts and tests whether they activate the response to a particular class, giving insight into what the model uses for a particular class. Next, Ghorbani et al. [52] and Fel et al. [47] extend this work to find the concepts automatically. Both studies work similarly: by decomposing the images into sub-regions and then finding the corresponding concepts that activate a certain class by clustering [52] or by non-negative matrix factorization [47]. Ge et al. [50] proposes to create a structural graph corresponding to the prediction of the classifier. This graph gives clues as to why the model chooses this class and not others.

Among their benefits, concept attribution methods have stronger advantages in contrast to CAMs, in our opinion. To begin with, recall that salience only provides a "where" statement. In contrast, concept attributions search to give cues into what was used for the classification. This information is more fine-grained than just showing where the model is focusing on. But, these methods search for correlations in the data and the concepts. Certifying that the found concepts are the ones used is essential. While this is informative, it is important to keep in mind that not all correlations are factors of causation.

2.2 Counterfactual Explanations

This thesis primarily explores counterfactual explanations (CE). In this chapter, we provide a comprehensive overview of recent advancements in CE methods. Counterfactuals are categorized as local *Post-Hoc* explanations. Their main goal is to modify an input instance such that the modification alters the original prediction. While this could be easily achieved using methods such as adversarial attacks [54], CE generation algorithms must satisfy three key properties to be useful.

1. **Proximity:** The counterfactuals must be close to the original input. Without this feature, we could create a naive counterfactual by replacing the original instance with one classified in a different set. In this scenario, the naive counterfactual would not provide useful information. Instead, providing proximal explanations is equivalent to showing the most important variables used by the model.

2. **Realism:** The generation pipeline should produce realistic and plausible changes to the input image. This property guarantees that the causal factors driving the decision change are plausible within real-world scenarios. From a user perspective, there is an additional property that would be ideal: the changes

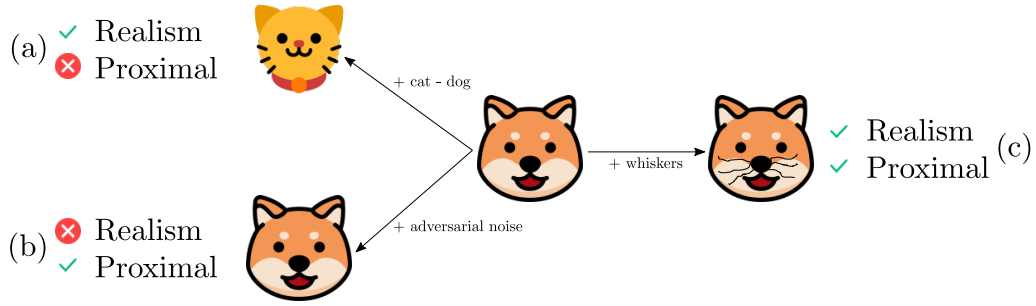


FIGURE 2.2: Examples of good counterfactual properties. (a) Shows a realistic "modification" of the input image so that the output shows a cat. These modifications are not proximal enough to provide useful insight into the model's behavior. (b) Adversarial noise in an effective approach to reversing the classifier's decision. This imperceptible noise is effectively proximal, but is not realistic and cannot be analyzed. (c) Adding whiskers to the original image causes the classifier's decision to change. This indicates that the toy model is sensitive to whiskers.

should be plausible in the sense that it should be possible to change the proposed features from the initial input state. For example, suppose there is a model that decides whether a bank can offer a loan to a user. This model takes as input the user's monthly income, gender, and age. Given the user's input information, the model decided that it could not provide a loan. As a result, the user received an explanation that if he were one year younger, he would have received the loan. This scenario illustrates the need to generate plausible changes. While the scenario may exist in the data, it's infeasible for the user to implement this solution due to the impossibility of reversing the aging process.

3. **Diversity:** CE methods should be able to generate a diverse set of explanations. This property is intended to highlight not just a single plausible scenario, but a wide range of cases. Diversity is an important aspect of CE generation because if a feature is exposed more frequently than others, we can say that it is one of the most important variables used locally by the model. Although this property is desirable, we have found that most methods in the literature neglect this property.

To exemplify the goodness of the introduced properties, in Figure 2.2, we highlight two scenarios where two counterfactual generation methods produce meaningless explanations. First, (a) shows a case where the input image has been completely replaced by another. Since the new image is in the image manifold, it is by definition realistic. But in this scenario, the analyst can ask whether the features that caused the misclassification were the color, the red nose, the collar, the shape, or some combination of these features. Since the range of possibilities is too wide, the analyst will not get any information. Second, (b) illustrates the case where the explanation is the input image with adversarial corruptions. In practice, adversarial noise seeks to be

FIGURE 2.3: Proximity ambiguity in counterfactual explanations. Both CEs are valid while the first one can be considered as not proximal to the image. The second explanation adds a barely visible pedestrian, indicated by the red arrow. Image taken from the BBD100k dataset [196] and edited by hand.

imperceptible, so it is proximal to the image both in image space and in human perception (we perceive both images equally). Again, since the image is perceptually indifferent to the original, it provides no visual cues. Finally, (c) shows an example of a useful CE. The output shows both closeness and realism since the CE method only painted the whiskers.

One of the major shortcomings of the explanatory methods introduced in section 2.1 is the lack of intervention actions, that is, the alteration of a single or a group of variables in order to observe an outcome. In contrast, counterfactuals follow this intrusive approach by attempting to alter an image to observe a change in the classifier's decision. In our view, intervention is analogous to the scientific method because it involves formulating a hypothesis (*i.e.*, does changing Z in X affect the prediction?) and empirically validating the new outcome of the model by altering the variables in question. This experiment allows certifying that Z is a variable that influences the new prediction.

From a human perspective, CEs have additional advantages over other methods. First, these explanations follow the same rationale as the human learning process and its logic for explaining phenomena; *i.e.*, in a contrastive and example-based approach [117]. Second, CEs tend to be concise because they change a few features in the input. Providing a long and complex explanation of the circumstances surrounding a particular event will overwhelm the user, potentially rendering the information as useless as providing no information about the outcome. Finally, CEs allow for a higher degree of interactivity with the user because they provide clues as to what could be changed to achieve a favorable outcome if the decision was negative.

Finally, counterfactuals are not perfect. Like all algorithms, CEs have limitations. In the first place, many of the properties are ambiguous. For example, we define property 1 as *proximity* to the input image. Intuitively, this feature makes sense, but there is no real consensus on defining what proximity is for counterfactuals. Take as an example Figure 2.3. This figure illustrates two toy counterfactuals for an autonomous driving task (stop vs. continue). One contains a red bus and the other one has a barely visible pedestrian. While several distance criteria can make a different

measurement of proximity, both cases are perceptually close to the original instance, in our opinion. Second, these CEs are only images. Even if they give information about what to change, it is partial since there may be several ways to generate the explanations. Ideally, we would like to supplement the explanation with additional descriptors, such as text, to facilitate understanding. Third, the previous point raises an important issue about explanations in general: what makes a good explanation? Miller [117] manifested that the XAI community tends to define explanations intuitively without resorting to psychology literature. This has led the community to create explanators that fit their definition for a particular work, without having a proper background. Finally, the evaluation of counterfactuals is not straightforward. It consists in generating realistic changes in the picture. Thus, the CE literature has adopted classical evaluation practices, such as the FID [63]. However, these have several weaknesses, which we will discuss in depth in subsection 2.2.2.

2.2.1 Generating Counterfactual Explanations

CEs search to minimize a distance function between the counterfactual and the original input such that their classification differs [181]. Formally, let $x \in \mathbb{R}^D$ be the D dimensional input, let $d : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^+$ be a distance function, and define the classifier as $f : \mathbb{R}^D \to \{1, 2, ..., C\}$, where C is the number of classes. Then a counterfactual $x_c \in \mathbb{R}^D$ is a solution to the minimization problem:

$$x_c = \underset{x' \in \mathbb{R}^D}{\operatorname{argmin}}\, d(x', x) \quad \text{s.t.} \quad f(x') \neq f(x). \tag{2.2}$$

The previous formulation relates to untargeted counterfactuals, where there is no specific target. In practice, counterfactuals are generated to predict an intended class $y_t \in \{1, 2, \cdots, C\}$. So, Equation 2.2 becomes

$$x_c = \underset{x' \in \mathbb{R}^D}{\operatorname{argmin}}\, d(x', x) \quad \text{s.t.} \quad f(x') = y_t. \tag{2.3}$$

Although the solution to this problem is a counterfactual, in practice this minimization is difficult to optimize. Thus, virtually all methods solve a soft version of Equation 2.3. Defining l as the loss function between the network output and the reference class, *e.g.* Cross Entropy, the minimization problem becomes

$$x_c = \underset{x' \in \mathbb{R}^D}{\operatorname{argmin}}\, l(f(x'), y_t) + \lambda d(x', x). \tag{2.4}$$

Solving this task directly would yield acceptable results in tabular data. However, in the image domain, this optimization process will produce undesired adversarial examples. To solve this issue, researchers use generative models to constrain the generation of the image manifold. Hence, when setting G as a generative model,

and z an input latent code, such that $G(z) \in \mathbb{R}^D$, Equation 2.4 becomes:

$$z_c = \underset{z'}{\operatorname{argmin}}\, l(f(G(z')), y_t) + \lambda d(G(z'), x). \tag{2.5}$$

Thus, the solution $G(z_c)$ is the CE. The previous equation is the foundation problem of counterfactual generation based on optimization for computer vision.

The literature has provided several ways to solve Equation 2.5. At the most basic level, CE methods have been equipped with different generative models. For example, Looveren and Klaise [108] used autoencoders (AE), while Rodríguez et al. [150] employed variational autoencoders [90] (VAE) to generate explanations by optimizing the latent code. As AEs and VAEs are optimized to reconstruct the input image, finding the latent code that creates the original image is trivial, achieving the proximity property easily. Since then, they have proven to be a valuable approach for low-dimensional data. However, as datasets have adapted to higher resolutions, both generative techniques have become obsolete due to their lack of generative fidelity in high-dimensional data. In our context, we need to generate high-resolution images with high fidelity (property 2), so VAEs and AEs are out of scope as a tool for CEs.

A common tool for CEs are Generative Adversarial Networks [53] (GAN). These models have been the preferred tool for visual data generation for many years. Several approaches have been proposed for CE generation. Singla et al. [167] used conditional GANs to modify the image in a multi-step fashion. Nemirovsky et al. [130] proposed using residual GANs to generate a delta such that the explanation is the original image plus the delta. Similarly, Jacob et al. [74] proposed a conditional GAN that uses segmentation maps as additional input features in the generator. OCTET [200] uses BlobGAN [41] to specify the location and type of objects. Luo et al. [113] and Khorram and Fuxin [85] used generated images from GANs to optimize them directly in the latent space, without resorting to original images.

GANs have several advantages when building CEs. For example, these architectures have been extensively studied and have achieved impressive quality, optimal for the realism property. Additionally, GANs generate an image in a single pass, ensuring fast generation. Nevertheless, these models present some weaknesses. First, recovering a faithful estimate of the input is not trivial and requires an optimization process. Without this, property 1 (proximity) may be difficult to achieve. Second, GANs are not as diverse as diffusion models [131] and, as explained previously, we are looking for methods that satisfy property 3 (diversity).

We believe that diffusion models are an ideal tool for CE, given the advantages and drawbacks of the previous generative models. These architectures are known to generate faithful instances of the training data (property 2), and diffusion models produce diverse images compared to previous approaches (property 3). Although, as with GANs, recovering the input image is a complex task. Nevertheless, diffusion

models benefit from certain features, such as the generation of images from partially noisy instances created from clean ones. This is advantageous because we can use the diffusion model to recover a clean image from this noisy state while preserving coarse details.

In the literature on diffusion models for counterfactual explanations, explainers have adopted image-based and latent-based diffusion approaches. On the one hand, image-based diffusion approaches in pixel space are computationally intensive. For example, DiME (chapter 3) uses the noise chain to propagate the gradient to the noisy instance. DVCE [10] used the blended diffusion strategy [12] and an adversarially robust model to regularize the gradients of the target model. ACE (chapter 4) drastically reduces the sampling steps to compute the gradients through all steps. These three methods have in common that they circumvent the gradient through the complete denoising chain by approximating the gradients in different manners. On the other hand, Stable Diffusion [151] has opened the door to incorporating text-driven counterfactuals. LDCE [46] and CoLa-DCE [123] took a similar approach to pixel-wise algorithms by using guided diffusion to generate the explanations. LANCE [146] used image captioning methods [102] and LLMs [22] to find concepts that change the classifier's decision in a brute force manner. To add these concepts, the approach used standard inpainting techniques [62, 118]. Similarly, DiG-IN [11] used null-inversion [118] and optimized the entire noise chain to create the explanation. Finally, Dataset Interface [180] and TIME (chapter 5) and uses textual-inversion [49] along with perfect inversion [183], making it simpler than previous approaches.

2.2.2 Evaluation Metrics

Evaluating counterfactual explanations has proven to be a difficult task. As discussed in the previous section, counterfactual explanations aim to change the classifier's decision by generating instances that satisfy three properties: realism, proximity, and diversity. In addition, counterfactuals can be used as tools for uncovering spurious correlations learned by a model, so, the literature also attempts to evaluate this aspect. In table Table 2.1 we summarize the metrics used to assess counterfactuals.

Now, let us define some notation that will be used in this section. Let $x \in \mathbb{R}^D$ be a D-dimensional instance in the image manifold and $y_t \in \{1, 2, ..., C\}$ be its target class in the label space, where C is the number of labels. Further, let $f : \mathbb{R}^D \to \{1, 2, ..., C\}$ be the model to explain, x_c the CE produced by a generation method ε. Finally, let y be the predicted class by the model of an image x, *i.e.* $f(x) = y$. As a side note, y_t can be different for each image x, but, for simplicity, we will use y_t as a general notation for the target.

Realism	Proximity	Diversity	Spurious Correlations
· FID [63]	· ℓ_p	· σ_L [79]	· Attribute Mixing [167]
· sFID [78]	· FVA [167]		· CD [79]
· IM1 [108]	· FS [78]		
· IM2 [108]	· S^3 [78]		

TABLE 2.1: Counterfactual Explanations evaluation metrics.

Flip ratio. First, we quantify the main goal of CEs: to change the classifier's decision. To quantify the accuracy of the CE algorithm, we measure the number of images that correctly changed the classifier's original prediction to the target label. Naturally, this metric is defined as

$$FR = \frac{1}{N} \sum_{x \in X} \mathbb{1}(f(x_c) = y_t). \tag{2.6}$$

Realism. A common approach to evaluate this property is to use standard metrics from the generative literature. To this end, research in CEs for visual data has adopted the Fréchet Inception Distance (FID) [63]. This metric computes a similarity score between the real and fake distributions in the feature space of an inception network. Note that when we refer to the real distribution, we mean the original dataset, and the fake distribution is the generated instances, or in our case, the counterfactuals. Although the FID is a common choice when evaluating generative methods, it has some weaknesses that need to be addressed. First, the FID is biased towards certain classes and will respond more strongly to particular textures or objects [82]. Second, the FID assumes that both distributions are Gaussian, which is not the case. Third, it fails to show some desired characteristics such as precision and recall. Fourth, the FID cannot measure the realness of the inserted feature at a local level but looks at all images together. Finally, the FID metric will favor small changes (area-wise), since smaller changes mean that the image is closer to the original.

Although we did not address the first two weaknesses of the FID, in chapter 4 we modified -at least partially- the last two by proposing the sFID. This metric divides the data set into two equal parts and computes the FID between the counterfactuals of one group and the original instances of the second set, thus removing the similarities between the two.

In addition, there are some metrics that the community no longer uses for realism: IM1 and IM2 [108]. The first metric, IM1, quantifies whether the image is in the distribution of the target class compared to the source class. To do this, IM1 computes the ratio between the reconstruction errors of two AEs, one trained on the target class y_t and the other trained on the source class y. If the target AE can recover the counterfactual, it means that the CE is in the target distribution, while the second AE should have trouble recovering the explanation. So, setting both AEs as

AE_{y_t} and AE_y, the IM1 is:

$$IM1(x_c, AE_{y_t}, AE_y) = \frac{\|x_c - AE_{y_t}(x_c)\|_2^2}{\|x_c - AE_y(x_c)\|_2^2 + \epsilon}. \tag{2.7}$$

The second metric, IM2, compares the reconstruction of the target autoencoder and a general one called AE_{global}. Looveren and Klaise [108]'s rationale is that if the counterfactual lies on the target distribution, it should also exist on the general distribution. Thus, their proposed metric is

$$IM2(x_c, AE_{y_t}, AE_{global}) = \frac{\|AE_{y_t}(x_c) - AE_{global}(x_c)\|_2^2}{\|x_c\| + \epsilon}. \tag{2.8}$$

These metrics have many weaknesses. First, it requires the cumbersome task of training $C + 1$ autoencoders. In addition, it requires checking that all AEs are working correctly. Second, the measurement is in pixel space, which is a poor statistic for distance measurement. Finally, the metric assumes that all images are correctly classified. Therefore, the source AE would not be able to correctly reconstruct the CE if the original instance was already on a different distribution.

Proximity. The second property to evaluate is a distance metric between the input instance and the counterfactual. Naively, we can compute a simple ℓ_p norm between x_c and x. However, similar to the IM1 and IM2 metrics, measuring distances in pixel space is not an ideal way to measure proximity between two images. Recall Figure 2.3, where we show two plausible counterfactuals. In this scenario, the explanation 1 (containing a bus) is further away than the other explanation. In our opinion, both valid cases are equidistant from the original input.

As a result, the literature has resorted to using specialized metrics to assess proximity for specific types of visual data. A common dataset for evaluating CEs is attribute recognition for face images (*e.g.* CelebA [107] and CelebAHQ [99]). Singla et al. [167] proposed two complementary ways to assess proximity. First, they checked whether the face changed its identity via the Face Verification Accuracy (FVA) [24]. Second, they measured the attributes that changed on the counterfactual. This metric is called the Mean Number of Attributes Changed (MNAC). Both metrics complement each other, since the former checks if the identity remains equal, while the latter checks if the attributes remain unchanged. In chapter 4, we noticed that the FVA metric was saturated, reaching almost 100% accuracy. So we skipped the tresholding in the FVA network and reported the cosine similarity in the embedding space. This metric is dubbed Face Similarity (FS). Please refer to chapter 4 for more details.

For general-purpose datasets (*e.g.* BDD100k [197] or ImageNet [35]), there was only a single approach to evaluate proximity. Khorram and Fuxin [85] suggested extending the insertion/deletion metric [141] for counterfactuals. In simple words, this metric, called COUT, gradually adds the changes to the original images and

checks how the probability of the target class changes. Additionally, in chapter 4 we complement the metrics of general case scenarios by adapting the metrics of face data. Specifically, we proposed the SimSiamSimilarity (S^3), which computes the perceptual distance between two images, as in FS. In this case, we used the SimSiam network [31] as it was trained to minimize the distance between two different views of an instance. All of these metrics share similar issues with the realism metrics, namely, they depend on the performance of specialized models.

Diversity. At the beginning of this thesis, there was no evaluation of diversity in CE for images in the literature. For tabular data, Mothilal, Sharma, and Tan [122] used the mean distance between explanations of the same instance to compute diversity. In chapter 3, we follow a similar approach to Mothilal, Sharma, and Tan [122] and compute the diversity in the same way but using the LPIPS [203] distance. We refer to this metric as σ_L.

Spurious Correlations. As explained above, CEs are tools that should be able to detect spurious correlations learned by the model. Singla et al. [167] proposed to evaluate this criterion by artificially partially mixing face attributes of two classes and detecting whether the model can uncover both attributes. However, mixing two or more attributes is an extreme view of finding spurious correlations. Therefore, in chapter 3, we highlight this problem and link this criticism to the MNAC metric. Namely, the MNAC contradicts the goal of detecting spurious correlations, since targeting one attribute may change others if they are correlated. Therefore, we propose the Correlation Difference metric to evaluate spurious correlations while maintaining the good properties of MNAC.

2.3 Interpretable by Desing architecture

Unlike *Post-Hoc* explanations, an interpretable by-design (ID) architecture f uses the explanation e to compute the prediction y:

$$e, y = f(x). \tag{2.9}$$

In this thesis, we categorize interpretable architectures into two distinct groups. The first group encompasses architectures that are inherently interpretable, *i.e.* their internal mechanisms and decision-making processes are intrinsically interpretable by design. For instance, a simple linear model is interpretable because we can directly examine the weights of the model to understand which input features contribute more significantly towards a particular class prediction. Similarly, decision trees, especially shallow ones, are interpretable as we can trace the path from input features to the final decision, following the learned decision rules. Finally, support vector

machines with simple kernel functions exhibit interpretability similar to linear models, as we can analyze the feature weights to comprehend which features push the prediction toward the positive or negative class regions.

The second group consists of ID architectures. Although these models use the same underlying mathematical operations as traditional neural networks, *i.e.* linear and nonlinear transformations, their decision-making process is developed to provide a more intuitive and interpretable rationale. For example, prototype-based approaches calculate a prediction based on the reasoning "this looks like this" [30]. In a nutshell, the decision process compares the spatial features with prototype features extracted from training images. Thus, the final decision is derived from the similarity between known features of training images and the tested instance. Later works [184, 128, 127, 26, 19, 175, 153, 186, 129, 39, 34, 59, 193] build on this logic and propose extensions to increase both performance and interpretability. In a similar vein, some methods have similar tendencies and compute the final score based on parts of the object in the image [92] or following a bag-of-words-like strategy [21]. Other approaches in the literature step away from this paradigm and propose novel algorithms for interpretability. Finally, some works highlight what a filter is responding to [103, 104, 163, 16, 23] or produce the final decision based on concepts [93, 133]. Finally, other literature incorporates self-attention layers before the decision layer to show the similarities between features and concepts [66, 88, 136, 149].

Most of the previous methods are built for CNN backbones. We believe that this common approach is attractive because the inductive bias of convolutions allows localized features to be built without any bells and whistles. However, only a small fraction of the literature ventures into building ID transformer-based architectures. For example, Xue et al. [193] proposed incorporating prototype layers [30] into transformers, and Böhle et al. [23] extended their previous work [16] to include self-attention layers in their algorithm. This opens up the possibility of exploring how to incorporate arbitrary interpretable features into transformers. To this end, in chapter 6 we propose an architecture based on transformers for classification.

Evaluating the performance of ID architectures typically involves a quantitative assessment of accuracy on the task at hand, in addition to visual inspection. This raises the question of how to properly evaluate these models, as merely looking at accuracy and validating the interpretable properties of the model visually is insufficient. However, the diverse nature of these methods and their different interpretability properties challenge a unified evaluation framework. This phenomenon has resulted in each method proposing a unique way to evaluate the method.

Finally, when considering the accuracy in each task, *e.g.* classification, many of these interpretable models offer increased transparency, but, they often exhibit a trade-off in terms of performance compared to their black-box counterparts. We speculate that this gap may arise from the simplification of certain processes in order to improve interpretability. Nevertheless, recent approaches have shown promising

results, achieving performance levels comparable to their black-box counterparts.

Chapter 3

Diffusion Models for Counterfactual Explanations

Prologue

Generating counterfactual explanations in the image domain typically leverages generative models. To effectively utilize these architectures, the generative models must be capable of producing diverse and realistic images. In this work, we addressed the challenge of integrating diffusion models [64] for generating CEs. These generative models emerged as an alternative tool for image generation during my first academic year, yielding comparable or superior results to generative adversarial networks (GANs). Beyond their improved qualitative outcomes, diffusion models also exhibited signs of more diversified generated instances.

In the context of counterfactual explanation generation, diffusion models can generate an image from partially noisy image states, with the generated instance retaining similar low-frequency data as the original. Thus, in our specific use case, we can partially corrupt the input image and then denoise it, guiding [36] it towards the counterclass while constraining it to remain close to the input. However, a naive adaptation of Dhariwal and Nichol [36]'s method is insufficient for generating effective CEs. Indeed, the main challenge in this work was to adapt the guided diffusion pipeline [36] to accept classifiers trained without noise.

Finally, this chapter extends beyond the mere generation of counterfactuals, addressing the challenge of their evaluation - an area that has seen limited advancement. In this work, we additionally critically examined current evaluation protocols and modified some of their weaknesses to correctly evaluate one of the key goals. From this work, we published a conference paper (ACCV 2022) and a journal article (CVIU 2024) in which we included extensive ablation studies on the behavior of the model. Here we include the journal article and omit the conference paper since the former contains all information for the latter.

3.1 Introduction

Convolutional neural networks (CNNs) have achieved remarkable performance levels by leveraging large, deep architectures comprising hundreds of layers and billions of trainable parameters. However, elucidating their decisions poses a challenge due to their pronounced non-linearity and excessive parameterization. Moreover, in real-world scenarios, if a model capitalizes on spurious correlations within data to make predictions, it can significantly erode end-user confidence in its decisions. This concern is particularly acute in high-stakes domains such as medicine or critical systems. Hence, there is a pressing need for Machine Learning (ML) models to ensure the utilization of accurate features while avoiding spurious associations in prediction. Consequently, the field of Explainable Artificial Intelligence (XAI) has witnessed substantial growth in recent years, aiming to unravel the decision-making mechanisms inherent in deep learning models.

This chapter focuses on *post-hoc* explanation methods, with particular emphasis on the burgeoning field of Counterfactual Explanations (CE) [181]. The primary objective of CEs is to introduce minimal yet meaningful perturbations to an input sample, thereby altering the original decision made by a black-box model. In essence, CE methods aim to achieve three key properties: *(i)* generating proximal images with sparse modifications, *i.e.* instances with the smallest perturbation; *(ii)* ensuring explanations are both realistic and comprehensible to humans; and *(iii)* producing diverse instances. Generally, counterfactual explanations strive to unveil the learned correlations underpinning the model's decisions.

Numerous studies on CEs leverage generative models to effect tangible alterations in images [158, 150, 80]. Additionally, these architectures discern the factors responsible for generating images close to the image manifold [9].

Given the recent advances within the image synthesis community, we propose DiME: Diffusion Models for counterfactual Explanations. DiME harnesses the denoising diffusion probabilistic models [64] to produce CEs. For simplicity, we will refer to these models as diffusion models or DDPMs. To the best of our knowledge, we are the first to exploit these new synthesis methods in the context of CEs.

Diffusion models offer several advantages over alternative generative models like GANs. Firstly, DDPMs incorporate multiple latent spaces, each governing coarse and fine-grained details. We exploit low-level noise latent spaces to effect semantically meaningful changes in the input image, a feature only recently explored by Meng et al. [115] for inpainting. Secondly, owing to their probabilistic nature, they yield a diverse array of images. This stochasticity proves advantageous for CEs, as multiple explanations may elucidate a classifier's error modes. Thirdly, as suggested by Nichol and Dhariwal [131], DDPMs cover a broader spectrum of the target image distribution. They observed that, for similar Frechet Inception Distance (FID)[63],

the recall is significantly higher on the improved precision-recall metrics [98]. Finally, the training of DDPMs is more stable compared to state-of-the-art synthesis models, notably GANs. However, due to their relatively recent development, DDPMs remain underexplored, with several aspects yet to be fully understood.

This work contributes to the XAI community by delving into the low-level noise latent spaces of DDPMs within the context of counterfactual explanations. Our contributions span across three main axes:

- **Methodology:** *(i)* DiME uses the recent diffusion models to generate counterfactual examples. Our algorithm relies on a single unconditional DDPM to achieve instance counterfactual generation. To accomplish this, *(ii)* we derive a new way to leverage an existing (target) classifier to guide the generation process instead of using one trained on noisy instances, as proposed by Dhariwal and Nichol [36]. Additionally, *(iii)* to reduce the computational burden, we take advantage of the forward and backward diffusion chains to transfer the gradients of the classifier under observation.

- **Evaluation**: We show that the standard MNAC metric is misleading because it does not account for possible spurious correlations. Consequently, we introduced a new metric, dubbed Correlation Difference, to evaluate subtle spurious correlations in a CEs setting.

- **Performance**: We set a new state-of-the-art result on several datasets, outperforming previous work on several standard metrics.

To further boost research on counterfactual explanations, our code and models are publicly accessible on https://github.com/guillaumejs2403/DiME.

3.2 Related Work

This work contributes to the field of XAI, within which two families can be distinguished: interpretable-by-design and *post-hoc* approaches. The former includes, at the design stage, human interpretable mechanisms [7, 202, 30, 126, 6, 70, 18]. The latter aims at understanding the behavior of existing ML models without modifying their internal structure. Our method belongs to this second family. The two have different objectives and advantages; one benefit of *post-hoc* methods is that they rely on existing models that are known to have good performance, whereas XAI by design often leads to a performance trade-off.

Post-hoc methods: In the field of *post-hoc* methods, there are several explored directions. Model Distillation strategies [171, 50] approach explainability through fitting an interpretable model on the black-box models' predictions. In a different vein, some methods generate explanations in textual form [61, 134, 191]. When it comes to

explaining visual information, feature importance is arguably the most common approach, often implemented in the form of saliency maps computed either using the gradients within the network [161, 76, 185, 101, 27, 206] or using the perturbations on the image [141, 142, 178, 199]. Concept attribution methods seek the most recurrent traits that describe a particular class or instance. Intuitively, concept attribution algorithms use [87] or search [52, 194, 50, 207] for human-interpretable notions such as textures or shapes.

Counterfactual Explanations (CEs): CEs is a branch of *post-hoc* explanations. They are relevant to legally justifying decisions made automatically by algorithms [181]. Some recent methods [187, 56] exploit the query image's regions and a different classified picture to interchange semantic appearances, creating counterfactual examples. Despite using the same terminology, this line of work [56, 188, 177] is diverging towards a task where it merely highlights regions that explain the discrepancy of the decision between the two real images, significantly differing from our evaluation protocol setup. Other works [181, 160] leverage the input image's gradients with respect to the target label to create meaningful perturbations. Conversely, Akula, Wang, and Zhu [4] find patterns via prototypes that the image must contain to alter its prediction. Similarly, Poyiadzi et al. [144] and Looveren and Klaise [108] follow a prototype-based algorithm to generate the explanations. Even Deep Image Priors [172] and Invertible CNNs [71] have shown the capacity to produce counterfactual examples. Furthermore, theoretical analyses [72] found similarities between counterfactual explanations and adversarial attacks.

Due to the nature of the problem, the generation technique used is the key element to produce data near the image manifold. For instance, Dhurandhar et al. [37] optimizes the residual of the image directly using an autoencoder as a regularizer. Other works propose to use generative networks to create the CEs, either unconditional [130, 150, 164, 204, 80] or conditional [176, 167, 106, 74]. In this chapter, we adopt more recent generation approaches, namely *diffusion models*; an attempt never considered in the past for counterfactual generation.

Diffusion Models: Diffusion models have recently gained popularity in the image generation research field [64, 168]. For instance, DDPMs approached inpainting [154], conditional and unconditional image synthesis [131, 64, 32], super-resolution [155], even fundamental tasks such as segmentation [15], providing performance similar or even better than State-of-the-Art generative models. Further, Song et al. [169] and Huang, Lim, and Courville [67] show score-based approaches and diffusion are alternative formulations to denoise the reverse sampling for data generation. Due to the recursive generation process, DDPMs sampling is expensive. Many works have studied alternative approaches to accelerate the generation process [95, 190].

The recent method of Dhariwal and Nichol [36] targets conditional image generation with diffusion models, which they do by training a specific classifier on noisy

instances to bias the generation process. Our work bears some similarities to this method, but, in our case, explaining an existing classifier trained uniquely in clean instances poses additional challenges. In addition, unlike past diffusion methods, we perform the image editing process from an intermediate step rather than the final one. To the best of our knowledge, no former study has considered diffusion models to explain a neural network counterfactually.

3.3 Methodology

3.3.1 Diffusion Model Preliminaries

Let us first introduce the generation process of diffusion models. DDPMs are based on two Markov chain sampling schemes that are inverse to each other. In the forward direction, the sampling starts from a natural image x and iteratively samples $z_1, \cdots, z_T$ by replacing part of the signal with white Gaussian noise. More precisely, if β_t is a given variance, the forward process follows the recursive expression:

$$z_t \sim \mathcal{N}(\sqrt{1-\beta_t}\, z_{t-1}, \beta_t\, I), \tag{3.1}$$

where $\mathcal{N}$ is the normal distribution, I the identity matrix, and $z_0 = x$. This process can be simulated directly from the original sample with

$$z_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x, (1-\bar{\alpha}_t)I), \tag{3.2}$$

where $\bar{\alpha}_t := \prod_{k=1}^{t}(1-\beta_k)$. For clarification, through the rest of the chapter, we will refer to clean images with an x, while noisy ones with a z.

In the reverse process, a neural network recurrently denoises z_T to recover the previous samples $z_{T-1}, \cdots, z_0$. This network takes the current time step t and a noisy sample z_t as inputs, and produces an average sample $\mu(t, z_t)$ and a covariance matrix $\Sigma(t, z_t)$, shorthanded as $\mu(z_t)$ and $\Sigma(z_t)$, respectively. Then z_{t-1} is sampled with

$$z_{t-1} \sim \mathcal{N}(\mu(z_t), \Sigma(z_t)). \tag{3.3}$$

So, the DDPM algorithm iteratively employs Equation 3.3 to generate an image z_0 with zero variance, *i.e.* a clean image. Some diffusion models use external information, such as labels, to condition the denoising process. However, in this chapter, an unconditional DDPM is employed.

In practice, the variances β_t in Equation 3.1 are chosen such that $z_T \sim \mathcal{N}(0, I)$. Further, the DDPM's trainable parameters are fitted so that the reverse and forward processes share the same distribution. Readers wishing to know further details on DDPM can refer to Ho, Jain, and Abbeel [64] and Nichol and Dhariwal [131]. Once the network is trained, one can rely on the reverse Markov chain process to generate

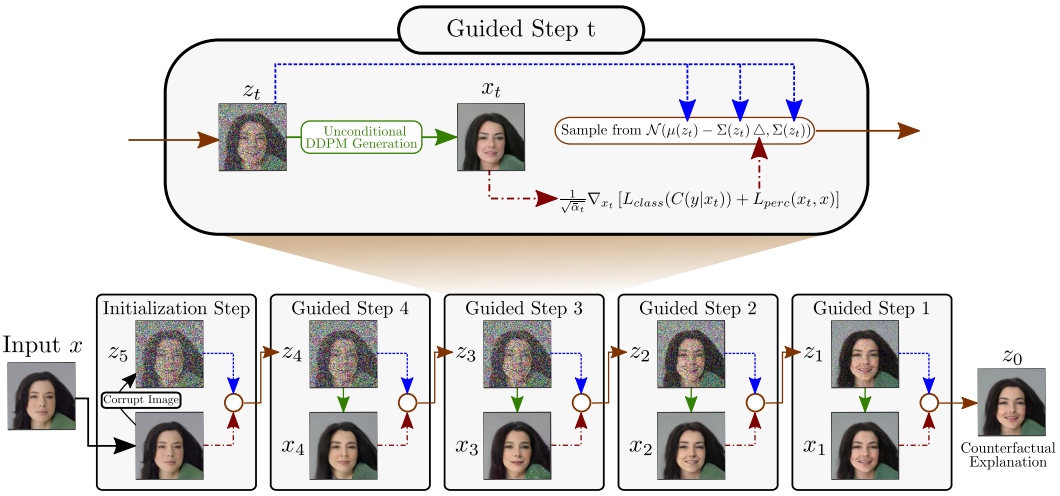


FIGURE 3.1: **DiME: Diffusion Models for Counterfactual Explanations.** Given an input instance x, we perturb it following Equation 3.2 to get z_τ (here $\tau = 5$). At time step t, the DDPM model is used to generate a clean image x_t, allowing to obtain the clean gradient L_{class} and L_{perc}. Finally, we sample z_{t-1} using the guiding optimization process on Equation 3.4, using the previously extracted clean gradients.

a clean image from a random noise image z_T. Besides, the sampling procedure can be adapted to optimize some properties following the so-called *guided diffusion* scheme proposed by Dhariwal and Nichol [36]:

$$z_{t-1} \sim \mathcal{N}(\mu(z_t) - \Sigma(z_t)\,\nabla_{z_t} L(z_t; y), \Sigma(z_t)), \tag{3.4}$$

where L is a loss function using z_t to specify the wanted property of the generated image, for example, to condition the generation on a prescribed label y. Note that in the work of Dhariwal and Nichol [36], guided diffusion is restricted to a specific classification loss. However, for the sake of generality and conciseness, this work shows how it can be extended to arbitrary losses.

3.3.2 DiME: Diffusion Models for Counterfactual Explanations

DiME takes an image editing perspective on CE generation. In order to generate a CE for a query image, DiME uses guided diffusion to direct the generation towards the target class. To remove the dependency of the guided diffusion on a particular classifier and to include the classifier under scrutiny, this study proposes a new way to compute the steering gradients $\nabla_{z_t} L$ (Equation 3.4).

Consider $t \in \{1, 2, ..., T\}$ as an intermediate time step in the noise chain and z_t as an instance in the corresponding noise state. To compute valid gradients, DiME iteratively denoises z_t to produce a clean image x_t using Equation 3.3. Thus, x_t can be processed by the classifier as it is no longer noisy. As a result, it is now possible to compute a loss function L and back-propagate the gradients through the denoising

iterative phase. Hence, $\nabla_{z_t} L(z_t)$ is computed as:

$$\nabla_{z_t} L(z_t) = \left(\frac{Dx_t}{Dz_t}\right)^T \cdot \nabla_{x_t} L(x_t). \tag{3.5}$$

To reduce the computational burden of the Jacobian computation $^{Dx_t}/_{Dz_t}$, we rely on the single-step forward process in Equation 3.2. Using the reparametrization trick [90], one obtains

$$z_t = \sqrt{\bar{\alpha}_t} x_t + \sqrt{1 - \bar{\alpha}_t}\epsilon, \ \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{3.6}$$

So, by solving x_t from z_t, we can leverage the gradients of the loss function with respect to the noisy input, a consequence of the chain rule. Henceforth, the gradients of L with respect to z_t become

$$\nabla_{z_t} L(z_t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \nabla_{x_t} L(x_t). \tag{3.7}$$

Now, we need to choose the loss function to optimize. Building upon previous approaches for CEs based on other generative models [167, 181, 74], we rely on a loss function composed of two components to steer the diffusion process: a classification loss L_{class}, and a perceptual loss L_{perc}. The former guides the image edition into imposing the target label, and the latter drives the optimization in terms of proximity. Accordingly, the loss function is

$$L(x_t) = \lambda_c L_{class}(C(y|x_t)) + \lambda_p L_{perc}(x_t, x), \tag{3.8}$$

where L_{class} is a classification loss used to guide the image edition toward the target label, a perceptual loss L_{perc} to optimize the proximity, λ_c and λ_p are regularization constants, and $C(y|x_t)$ is the posterior probability of the category y given x_t computed with the classifier C. Additionally, we incorporate an ℓ_1 loss, $\lambda_1 \|z_t - x\|_1$, between the noisy image z_t and the input x to slightly boost the similarity between the counterfactual and the query image in the pixel space.

After explaining how DiME uses the target classifier, Figure 3.1 shows how DiME makes counterfactual explanations. The algorithm starts by corrupting the input instance $x = x_\tau$ according to Equation 3.2 up to the noise level $t = \tau$, with $1 \leq \tau \leq T$, creating z_t. Then the following two steps are iterated from $t = \tau$ to $t = 1$: *(i)* First, we guide the diffusion process to obtain z_{t-1} using Equation 3.4 with the gradients computed in Equation 3.7 using the previous clean instance x_t. *(ii)* Next, we estimate the clean image x_{t-1} for the current time step z_{t-1} using the unconditional generation pipeline of DDPMs. The final image is the counterfactual.

3.3.3 Discussion

This section discusses the methodological contributions of DiME. First, it explains how we adjusted guided diffusion to include the classifier being studied without having to modify it. Then it assesses the proposed technique for speeding up gradient computation and finally, it provides feedback on the loss function.

Adapting the Guided Diffusion The classifiers to be analysed by XAI methods are trained on images that do not contain any degree of noise, and should not be modified in any way. So, it is expected that they would not produce robust predictions for noisy samples such as those used by guided diffusion [36]. This phenomenon leads to uninformative gradients $\nabla_{z_t} L_{class}(C(y|z_t))$ when the classifier C is used directly with z_t, which misguides the generation of the counterfactual. For this purpose, generating a clean instance x_t from z_t and computing the gradients with x_t yields an approximate but useful steering gradient. This experiment is fact-checked quantitatively in subsection 3.5.1.

Efficient Gradient Computation The dependence of the loss on x_t, rather than directly from z_t, renders the gradient computation more expensive (Equation 3.5). Indeed, this procedure requires retaining Jacobian information throughout the entire computation graph, which is very deep when t is close to τ. As a result, backpropagation is too memory-intensive to be considered an option. So, the proposed approximation greatly relaxes this computational constraint, enabling the generation of counterfactual explanations in regular-user GPU setups.

A note on the Loss function Diffusion models are inherently stochastic, *i.e.* each denoising step operates on a noise sampled from the normal distribution. Thus, x_t is also stochastic, and, the loss function in Equation 3.8 should contain an expectation, as in

$$\tilde{L}(z_t; y, x) = \mathbb{E}\left[L(x_t; y, x)\right]. \tag{3.9}$$

In practice, computing the loss gradient would require sampling several realizations of x_t and taking an empirical average. We restrict ourselves to a single realization per step t for computational reasons and argue that this is not an issue. Indeed, we can partly count on an averaging effect along the time steps to cope with the lack of individual empirical averaging. Besides, the stochastic nature of our implementation is an advantage since it introduces more diversity in the produced CEs, a desirable feature as advocated by Rodríguez et al. [150].

3.4 Experiments

3.4.1 Datasets

This work assesses the DiME's effectiveness using established methodologies across three datasets: CelebA, CelebA HQ, and BDD100k. Next, we will describe each dataset and their standardized protocols, including the classification task, model architecture, and image resolution.

CelebA [107] is a dataset for attribute classification. The dataset contains approximately 200.000 images, each containing a label for the 40 attributes. For the CE generation, the protocols are similar to the one employed by [150, 167, 74]. The architecture of choice is a DenseNet121 [68] employed to extract the attributes of images with a resolution of 128×128. Finally, along the 40 attributes, the CE will be generated for the *smile* and *young* attributes.

Jacob et al. [74] proposed extending the evaluation protocols to CelebA HQ [100], increasing to a higher degree of difficulty. This dataset contains 30.000 high-definition face images, each labelled with 40 attributes, similar to CelebA. For the task, Jacob et al. [74] followed a setup comparable to the one utilized for CelebA, covering equivalent target classes and classifier but with 256×256 resolution instances.

Finally, the study also includes experiments conducted on BDD100k [196] following the conventions proposed by Jacob et al. [74]. The BDD100k dataset comprises 10 tasks for autonomous driving, such as object detection, lane marking, and instance semantic segmentation. It contains 100.000 driving scene videos at a 720p resolution. Due to the lack of a classification task, Jacob et al. [74] proposed using a DenseNet121 [68], as well, in the BDD-OIA [192] extension set for the *moving forward / slowing down* task. The images are resized at a resolution of 512×256 pixels.

3.4.2 Implementation Details.

As seen in subsection 3.3.2, DiME's CE generation involves several key hyperparameters. Firstly, there are the hyperparameters for the classification, perceptual, and ℓ_1 losses: $\lambda_c, \lambda_p, \lambda_1$, respectively. Regarding the classification mixing factor, if the counterfactual generation does not find a valid explanation, DiME re-runs its algorithm but with an increased value for λ_c. Secondly, τ controls the amount of noise added to the initial instance for the denoising process.

For all datasets, DiME uses $\lambda_p = 30$, and $\lambda_1 = 0.05$. Further, DiME's diffusion model was trained to generate images using 500 diffusion steps from the normal distribution. Yet, when generating the explanations, DiME re-spaced the sampling process to 200 time-steps, reducing the inference time. For CelebA and CelebA HQ, DiME's diffusion model uses a $\tau = 60$. Hyperparameter-wise, for the former dataset, DiME sets $\lambda_c \in \{8, 10, 15\}$ to iteratively find the counterfactuals, while for the latter it used a $\lambda_c \in \{25, 30, 40\}$. Regarding BDD100k, DiME sets $\tau = 45$ and

$\lambda_c \in \{50, 55\}$. Lastly, all unconditional DDPM models were trained with the publicly available code of Dhariwal and Nichol [36].

3.4.3 Evaluation Metrics

Building on previous studies [150, 167], the subsequent paragraphs summarize the principles of current evaluation metrics.

As explained in section 3.1, the goal of CEs is to create explanations that *mislead* the classifier under observation. We assess this goal by computing the flip ratio (FR). This measure is computed as the proportion of instances in the dataset that flip the classifier's decision. On another note, the CEs must have several properties. Following the image synthesis research literature, previous methods adapted the FID [63] as a measure of the *realism* of the image distribution. Furthermore, the second property of CEs is to create *proximal and sparse* images. Among other tools, the XAI community has adopted the Face Verification Accuracy [24] (FVA) and the Mean Number of Attributes Changed (MNAC) [150] for face images. On the one hand, the MNAC metric looks at the face attributes that changed between the input image and its counterfactual explanation, regardless of whether the individual's identity changed. On the other hand, the FVA looks at the identity of the individual without considering the difference in attributes.

To compute these measures, the FVA uses the cosine similarity between the input image and its produced counterfactual on the feature space of a ResNet50 [60] pretrained model on VGGFace2 [24]. This metric considers that two instances share identity if the similarity is higher than 0.5. So, the FVA is the mean number of faces sharing the same identity with their corresponding CE. Computing the MNAC requires fine-tuning the VGGFace2 model on the CelebA dataset or CelebA HQ. This fine-tuned model is dubbed *oracle*. Thus, the MNAC is the mean number of attributes for which the oracle switch decision under the action of the CE. For a fair comparison with the state-of-the-art, all classifiers were trained using the DiVE's [150] available code for the CelebA dataset, including the fine-tuned ResNet50 for the MNAC assessment. For the CelebA HQ and BDD100k sets, STEEX's[74] publicly available code provided their models. Finally, previous studies [150, 167] compute the FID, the FVA, and the MNAC metrics considering only those successful counterfactual examples.

3.4.4 Quantitative Evaluation

This section quantitatively evaluates the performance of DiME against previous methods. Foremost, we assess the FP of the STEEX [74] and DiVE [150] algorithms. While STEEX reports a 99.5% FR, DiVE does not report theirs. This raises concerns about the fairness of comparing the quantitative measures between the proposed method, STEEX, and DiVE. Since some metrics depend highly on the number of

samples, especially FID, it was necessary to recompute their CEs. Surprisingly, their FR was relatively low: 44.6% for the smile category. In contrast, DiME achieved a success rate of 97.6 and 98.9 for the smile and young attributes, respectively. Hence, by changing their optimization steps to 100 - now referred to as DiVE^{100} -, the new FR performance rise to 92.0% for the smile attribute and to 93.4% for the young one.

Table 3.1 shows DiME's metrics in for CelebA. The proposed method beats the previous literature in five out of six metrics. For instance, there is an approximately 3-fold improvement on the FID metric for both smile and young attributes. DiME's generation process does not require entirely corrupting the input instances; hence, the coarse details of the image remain. The other methods rely on latent space-based architectures. Thus, they require compacting essential information and removing outlier data. Consequently, the generated CEs cannot reconstruct the missing information, losing significant visual components of the image statistics.

	Smile			Age		
Method	**FID** (↓)	**FVA** (↑)	**MNAC** (↓)	**FID** (↓)	**FVA** (↑)	**MNAC** (↓)
xGEM+ [80]	66.9	91.2	-	59.5	97.5	6.70
PE [167]	35.8	85.3	-	53.4	72.2	3.74
DiVE [150]	29.4	97.3	-	33.8	**98.2**	4.58
DiVE^{100}	36.8	73.4	4.63	39.9	52.2	4.27
STEEX [74]	10.2	96.9	4.11	11.8	97.5	3.44
DiME (Ours)	**3.17**	**98.3**	**3.72**	**4.15**	95.3	**3.13**

TABLE 3.1: **CelebA results.** DiME's performance against the state-of-the-art on the FID, FVA, and MNAC metrics. The values in **bold** are the best results. The proposed method has a 3-fold improvement on the FID metric. All results were extarcted from Jacob et al. [74]'s manuscript.

All results from CelebA HQ dataset are in Table 3.2. For both attributes, DiME outperforms STEEX in the FID and MNAC metrics, while achieving competitive performance on the FVA metric. Yet, the former method has an advantage with respect to the latter. STEEX uses additional segmentation maps to generate the counterfactual, while DiME merely uses the target classifier. This is advantageous as the proposed method can be applied out of the box without requiring a segmentation model.

The evaluation of the BDD100k dataset does not enjoy the benefits of the FVA and MNAC metrics since these are for face data. Thus, the FID is the only metric available to evaluate the BDD100k dataset. Jacob et al. [74] noted that Rodríguez et al. [150]'s work does not converge on the BDD dataset. Hence, we compare ourselves only against STEEX [74]. Performance-wise, DiME achieves a 7.35 FID compared to 58.8 for STEEX. Nevertheless, DiME achieves a 95.6% FR, while STEEX claims a 99.5%.

	Age			Smile		
Method	**FID** (↓)	**FVA** (↑)	**MNAC** (↓)	**FID** (↓)	**FVA** (↑)	**MNAC** (↓)
DiVE [150]	107.0	35.7	7.41	107.5	32.3	6.76
STEEX [74]	21.9	**97.6**	5.27	26.8	**96.0**	5.63
DiME (Ours)	**18.1**	96.7	**2.63**	**18.7**	95.0	**2.10**

TABLE 3.2: **CelebA HQ results.** DiME's performance compared to the state-of-the-art on the FID, FVA, and MNAC metrics. The values in **bold** are the best results. The proposed CE algorithm has small FID and MNAC performance improvement while being competitive in the FVA metric. We extracted all results from the work of Jacob et al. [74].

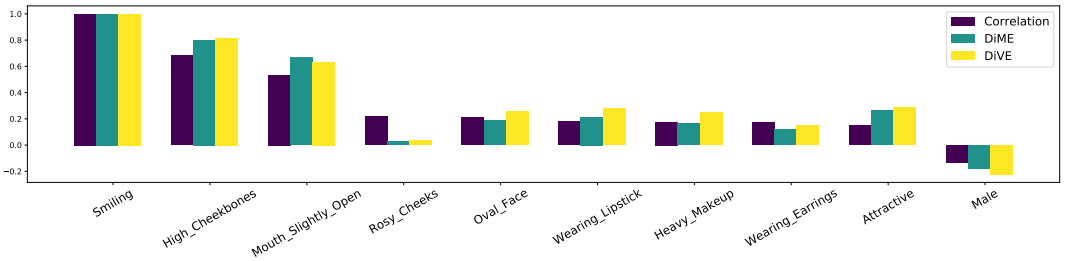


FIGURE 3.2: **Spurious Correlation Detection.** We show the top 9 most correlated attributes in the label space with "smile". We obtained the Pearson Correlation Coefficient from the ground truth on the training set. Albeit the difference in the MNAC performance, DiME and DiVE achieve to detect the spurious correlation similarly.

3.4.5 Discovering Spurious Correlations

One end goal of CE is to uncover the modes of error of a target model, in particular its reliance on spurious correlations. Previous evaluation protocols [167] search to assess the counterfeit dependencies by inducing artificial entanglements between two supposedly uncorrelated traits, such as the smile and gender attributes. Such an extreme experiment does not shed light on the ability to reveal spurious correlations for two reasons. First, the introduced entanglement is complete, in the sense that in this experiment the two considered attributes are fully correlated. Second, the entanglement is restricted to two attributes. In fact, as depicted in Figure 3.2, in real datasets such as CelebA, many labels are correlated at multiple levels. As a result, this phenomenon calls the previously proposed correlation experiment into question.

At the same time, the interpretation of some standard metrics can be challenged when spurious correlations are present. This is the case for MNAC. Arguably, the classical interpretation is that, between two CE methods, the one displaying the smaller MNAC is reckoned as the better one. This interpretation is at odd with the fact that the alternative method may display a higher MNAC because it actually reveals existing spurious correlations.

Consequently, this work designed a new metric called *Correlation Difference* (CD),

verifying the following principles: **(i)** it quantifies how well a counterfactual routine captures spurious correlations. **(ii)** It should apply an oracle to predict the (unknown) attributes of counterfactual examples. **(iii)** To mitigate potential errors of the oracle, the metric should preferably rely on attribute prediction changes between the original example and its explanation, rather than solely on the prediction made on the counterfactual. In other words, principle (i) amends the failure of MNAC by estimating the correlation between two attributes after applying the counterfactual algorithm, while (ii) and (iii) maintain its desirable features.

To do so, let $c_{q,a}$ be the Pearson correlation coefficient between the target attribute q and any other attribute a. Denoting X a random image sample along with its two associated binary attribute labels Y_q and Y_a, $c_{q,a}$ is defined as

$$c_{q,a} = \text{PCC}(Y_q, Y_a), \tag{3.10}$$

where PCC is the Pearson correlation coefficient operator. To cope with principle (i) we would like to estimate correlations between attributes q and a as well as we would like our estimation to rely on the CE method M targeting the attribute q. The main issue is that we do not know the actual attributes for the CE, $M(X, q)$, obtained from an image X. Yet, following principle (ii), we may rely on an oracle to predict these attributes. More precisely, letting $O_a(X)$ be the oracle prediction for a given image X and for the label a, we could simply compute the correlation coefficient between $O_q(M(X, q))$ and $O_a(M(X, q))$. Such an estimate would be prone to potential errors of the oracle, and following principle (iii) we would prefer to rely on attribute changes $\delta^M_{q,a}(X) = O_a(M(X, q)) - O_a(X)$.

Interestingly, one can show that $c_{q,a}$ can be reformulated as follows:

$$c_{q,a} = PCC(\delta_q, \delta_a), \tag{3.11}$$

where $\delta_a = Y_a - Y'_a$ and $\delta_q = Y_q - Y'_q$, with (X, Y_q, Y_a) and (X', Y'_q, Y'_a) two independent samples. In other words, $c_{q,a}$ can be interpreted as the correlation between changes in attributes q and a among random pairs of samples.

Theorem 1. *Let Y_q and Y_a be the distributions of the query and target attributes* **q** *and* **a**, *respectively, over the random variable X. Similarly, let Y'_q and Y'_a be the distributions of each attribute over the X' i.i.d. than X. Thus, if*

$$c_{q,a} := \frac{\text{Cov}(Y_q, Y_a)}{\sqrt{\text{Cov}(Y_q, Y_q)\text{Cov}(Y_a, Y_a)}}, \tag{3.12}$$

then,

$$c_{q,a} = \frac{\text{Cov}(\delta_q, \delta_a)}{\sqrt{\text{Cov}(\delta_q, \delta_q)\text{Cov}(\delta_a, \delta_a)}}, \tag{3.13}$$

where $\delta_a := Y_a - Y'_a$*, with* X' *is i.i.d. than* X*, and* Cov *is the covariance operator.*

Proof. By definition, the covariance between Y_q and Y_a is:

$$\text{Cov}(Y_q, Y_a) = \mathbb{E}[(Y_q - \mathbb{E}[Y_q])(Y_a - \mathbb{E}[Y_a])]. \tag{3.14}$$

Thus,

$$\begin{aligned}
\text{Cov}(Y_q, Y_a) &= \mathbb{E}[(Y_q - \mathbb{E}[Y_q])(Y_a - \mathbb{E}[Y_a])] \\
&= \mathbb{E}[Y_q(Y_a - \mathbb{E}[Y_a])] + \underbrace{\mathbb{E}[\mathbb{E}[Y_q](Y_a - \mathbb{E}[Y_a])]}_{=0 \text{ by symmetry}} \\
&= \mathbb{E}[Y_q(Y_a - \mathbb{E}[Y_a])] \\
&= \mathbb{E}[Y_q(Y_a - Y'_a)] \\
&= \mathbb{E}[Y_q \delta_a] \\
&= \mathbb{E}\left[\frac{1}{2}(2\,Y_q - Y'_q + Y'_q)\delta_a\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(\delta_q + Y_q + Y'_q\right)\delta_a\right] \\
&= \frac{1}{2}\mathbb{E}[\delta_q \delta_a] + \frac{1}{2}\underbrace{\mathbb{E}[(Y_q + Y'_q)\delta_a]}_{=0 \text{ by symmetry}} \\
&= \frac{1}{2}\mathbb{E}[\delta_q \delta_a].
\end{aligned} \tag{3.15}$$

Given that $\mathbb{E}[\delta_i] = 0$ for $i \in \{a, q\}$, then

$$\mathbb{E}[\delta_q \delta_a] = \text{Cov}(\text{ffi}_\text{q}, \text{ffi}_\text{a}). \tag{3.16}$$

Therefore,

$$\text{Cov}(Y_q, Y_a) = \frac{1}{2}\text{Cov}(\text{ffi}_\text{q}, \text{ffi}_\text{a}). \tag{3.17}$$

Note that the same derivation holds when $a = q$. Consequently,

$$c_{q,a} = \frac{\text{Cov}(\delta_q, \delta_a)}{\sqrt{\text{Cov}(\delta_q, \delta_q)\text{Cov}(\delta_a, \delta_a)}} \tag{3.18}$$

□

Following the result of Theorem Theorem 1, $\delta^M_{q,q}$ and $\delta^M_{q,a}$ are used as replacements for δ_q and δ_a in Equation 3.11 to obtain the estimate $c^M_{q,a}$ of $c_{q,a}$. Finally, CD for label q is merely:

$$CD_q = \sum_a |c_{q,a} - c^M_{q,a}|. \tag{3.19}$$

So, this sections assesses DiME and DiVE^{100}'s explanations on the CelebA dataset. The proposed method got a CD of 2.30 while DiVE^{100} 2.33 on CelebA's validation set,

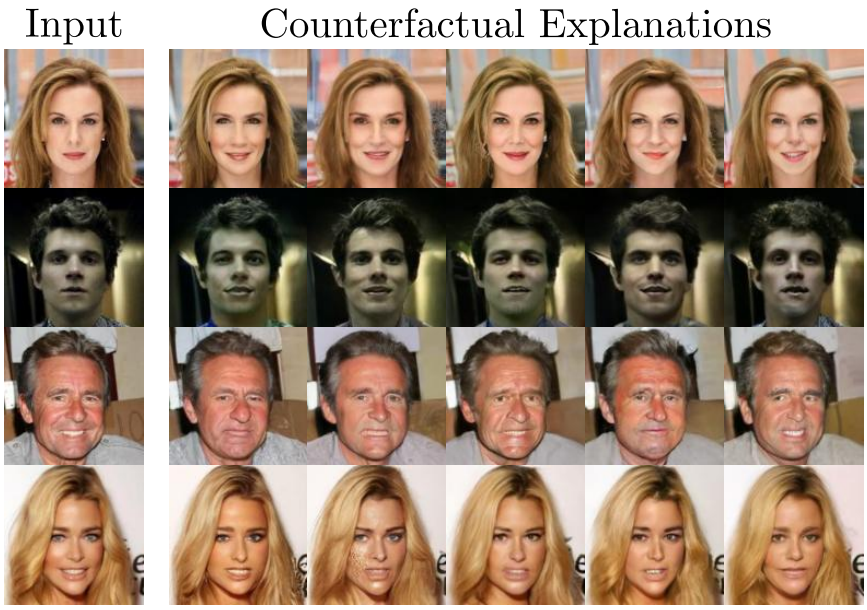


FIGURE 3.3: **Diversity Counterfactual examples.** The classifier predicts the first two input images as non-smiley and the last two as smiley. In this example, all explanations fool the classifier. DiME is capable of synthesizing diverse counterfactuals without any additional mechanism.

meaning that DiVE^{100} lags behind DiME. However, the margin between the two approaches is only slender. This reveals our suspicions: the MNAC results presented in Table 3.1 give a misleading impression of a robust superiority of DiME over DiVE^{100}.

3.4.6 Diversity Assessment

One of the most crucial traits of counterfactual explanations is to create multiple and diverse examples [150, 122]. So, this chapter proposes the σ_L metric to assess trait by computing the mean pair-wise LPIPS [203] metric between five independent runs. Formally, setting N as the length of the dataset and $n = 5$ as the number of different CE runs, the diversity metric is:

$$\sigma_L = \frac{1}{N}\sum_{i=1}^{N}\frac{2}{n(n-1)}\sum_{j=1}^{n}\sum_{k=j+1}^{n} LPIPS(x_j^i, x_k^i), \tag{3.20}$$

where x_j^i is the j counterfactual of the instance i. Therefore, a higher σ_L means increased perceptual dissimilarities between the explanations, hence, more diversity. In practice, σ_L uses all counterfactual examples, even the unsuccessful instances, to compute the evaluation metric because we search for the capacity of exploring different traits. Additionally, the original instance is excluded since we search for the dissimilarities between the counterfactuals.

Accordingly, we assess DiME's performance with DiVE^{100} and its Fisher Spectral variant on a small partition of the validation subset. As well, Figure 3.3 visualizes

some examples. On the one hand, DiME achieved a diversity of 0.213. On the other hand, DiVE [150] reached 0.044 and its Spectral Fisher variant got 0.086. In addition, Table 3.3 shows five different runs using the FID, FR, and ℓ_1 metrics. Even when we set different initial conditions for each iteration, DiME is robust to many instantiations.

TABLE 3.3: **Diversity experiments.** The proposed method was tested five times, varying the initial seed. The results show that DiME is robust to the different initial conditions, although the visual elements vary significantly.

Seed	**FID**(↓)	**FR**(↑)	ℓ_1(↓)
1	20.51	97.9	0.0430
2	20.60	97.6	0.0430
3	20.72	97.9	0.0431
4	20.67	97.7	0.0431
5	20.46	98.2	0.0430

3.4.7 Qualitative Results

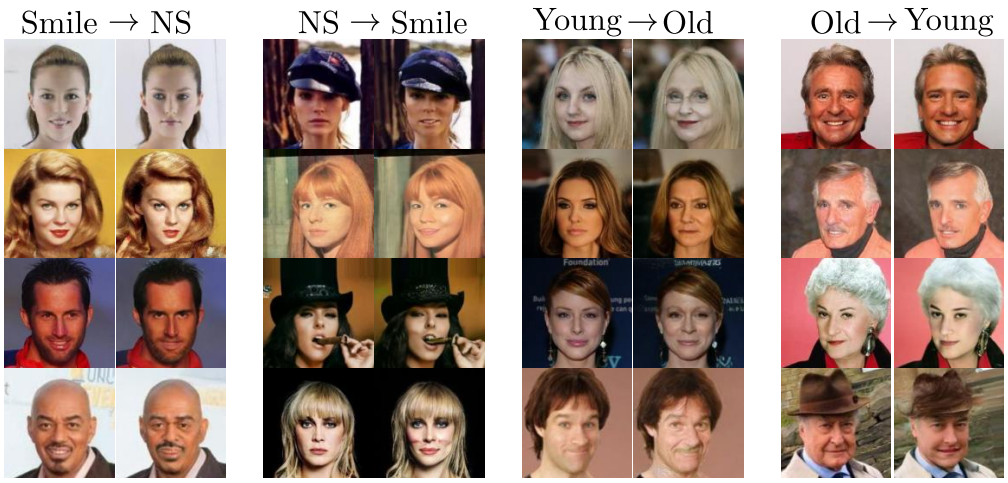


FIGURE 3.4: **Qualitative Results.** This figure visualizes some images and their corresponding counterfactual explanation produced by our proposed approach. The proposed methodology achieves to incorporate small but perceptually tangible changes in the image. NS stands for Non-Smiley.

Figure 3.4 shows some inputs (left) and the counterfactual examples (right) produced by DiME for all datasets. At first glance, the results reveal that the model performs semantic editings into the input image. In addition, uncorrelated features and coarse structure remain almost unaltered. Further, some out-of-distribution objects (e.g. items, pendants, or hands) present slight variations. Yet, DiME fails to reconstruct the exact shape of these objects, but the essential aspect remains the same.

3.5 Ablation studies

This section studies all DiME components to analyze each individual contribution. To validate all distinct variations, all evaluations were performed on a small and randomly selected mini-val. Several metrics were considered to compute all scores: the FR, the FID, and the ℓ_1 metric of successful CEs. However, in order to make the FID values more comparable among all variants, we condition its computation only on the successful CEs and keep the same number of samples for all methods to mitigate the bias in FID related to the number of samples, denoted as FID$^+$.

3.5.1 Impact of the noise-free input of the classifier

As a major contribution, this chapter proposed an adjustment to the guided diffusion process. It consists in applying the classifier on noise-free images x_t rather than on the current noisy version z_t to obtain a robust gradient direction. To assess the role of this part, consider several alternatives to DiME's approach. The first alternative, dubbed *Direct*, uses the gradient of the classifier applied directly to the noisy instance z_t. The second version, called *Naive*, uses the gradient of the original input image at each time step to guide the optimization process. The last variation is a near duplicate of DiME except for the fact that it ends the guided diffusion process as soon as x_t fools the classifier, dubbed *Early Stopping*. In addition, the results contain the evaluation of the DDPM generation without any guiding beginning from the corrupted image at time-step τ - named *Unconditional*. This assessment marks a reference of the performance of the DDPM model.

Method	**FR** ($\uparrow$)	**FID**$^+$($\downarrow$)	ℓ_1($\downarrow$)
Direct	19.7	**50.51**	0.0454
Naive	70.0	98.93 $\pm$ 2.36	0.0624
Early Stopping	97.3	51.97 $\pm$ 0.77	0.0467
Unconditional*	8.6	53.22 $\pm$ 0.98	0.0492
DiME	**97.9**	**50.20** $\pm$ 1.00	**0.0430**

TABLE 3.4: **DiME variations.** This table shows the advantages of the proposed adjustment to incorporate the classifier under observation. Including the clean gradients benefits DiME on all metrics, especially the FR. *: FID$^+$ and ℓ_1 are computed with the same number of samples as the rest, but without filtering out unsuccessful CEs.

Table 3.4 shows the results of the different versions. The most striking point is that when compared to the Naive and Direct approaches, the unimpaired version of DiME is the most effective in terms of FR by a large margin. This observation validates the need for our adjustment of the guided diffusion process. Further, DiME is also superior to all other variations in terms of the other metrics. At first glance, it was expected that the unconditional generation to have better FID than DiME

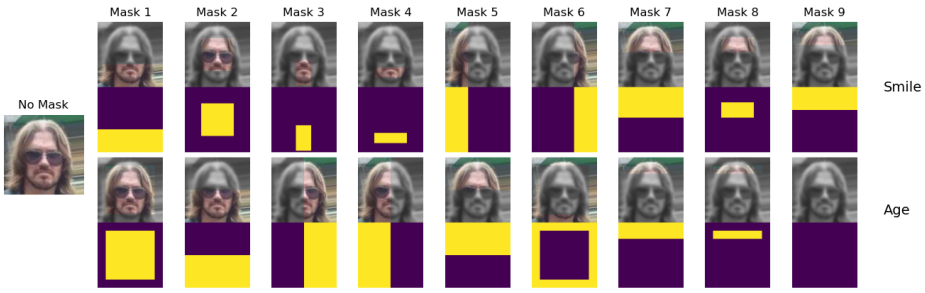


FIGURE 3.5: **Regional Masks to detect adversarial noise.** Several different masks were created to localize our counterfactual explanations and the adversarial attack. In the top and third rows, we visualize an image example with smooth/gray-out and color regions. The former corresponds to the untargeted regions, while the latter is the localization of interest. The second and final rows correspond to the corresponding masks. The masks are ordered from left to right by their impact level on the *Smile* and *Age* attributes.

and the ablated methods. However, we believe that the perceptual component of our loss is beneficial in terms of FID. Therefore, the unconditional FID is higher. Based on the same rationale, one can explain the slightly higher FID displayed by the early stopping variant. Moreover, most instances merely shifted the decision boundary, reporting low confidence of the posterior probability. These instances are semi-factual [84] and contain features from both attributes, making them hard to analyze in the context of explainability, in our opinion.

3.5.2 Is DiME relying on adversarial noise to flip the label?

Conceptually, adversarial attacks (AA) [114] and counterfactual images share a common principle [138]: to find a minimal modification of the input image that changes the decision of the model. However, the objective is very different: adversarial attacks produce images whose modification is barely visible, while the modification of counterfactual images must be valid and easily understood by a human. As previously stated, subsection 3.3.2 hypothesized that the use of a diffusion model in DiME would allow it to stay in the distribution of natural images and thus produce only images that are possible in the real world, which makes the difference with adversarial images. This section presents experimental results that confirm this.

To assess whether DiME relies on adversarial noise to flip the prediction, this section compares the FR performance of regionally constrained counterfactuals to localized AA. If DiME does not rely on adversarial noise, there are two possible outcomes. On the one hand, if the spatial constraint targets a region containing features that are supposedly uncorrelated with the target label, one would expect to see a significant performance gap in FR between the regionally constrained CE and the AA. On the other hand, if DiME uses semantic changes, the FR will be lower than the AA, but the gap will be narrow.

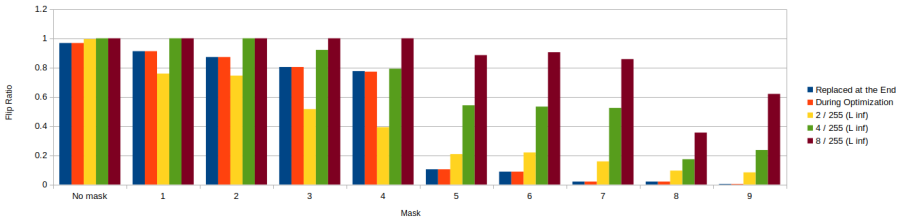


FIGURE 3.6: **Searching for Adversarial Noise (*Smile* Attribute).** We visualize the results of localized counterfactual explanations and PGD noise. We observe that for regions containing pixels that are not correlated with the *Smile* class (e.g., mask 9), DiME fails to produce a counterfactual image, which is the desired behavior, while there were possible adversarial attacks.

The described experiment focuses on the *Smile* and *Age* labels of the CelebA dataset. Figure 3.5 defines a set of 9 localization masks for the smile attribute and 8 for age. Thereafter, the modifications that will be made to the input images will be made only on the pixels corresponding to the bright part of the masks, those of the dim region remaining unchanged.

To enable localized editing, Equation 3.7 is slightly modified to constrain the generation in the form of masks. Recall that each update step computes the gradients of the loss function with respect to the noisy image z_t following Equation 3.7. Instead, L is modified to include the mask M as follows:

$$\begin{aligned} \nabla_{z_t} L(z_t; y, x, M) &= \frac{1}{\sqrt{\bar{\alpha}_t}} \nabla_{x_t} \lambda_c L_{class}(C(y|x_t^M)) + \lambda_p L_{perc}(x_t, x) \\ x_t^M &= x_t \odot M + x \odot (1 - M) \end{aligned} \tag{3.21}$$

where $\odot$ is the element-wise product operation, y the target label, and x and x_t are the query and current clean image at timestep t, respectively. This method is denoted as *During Optimization*. Alternatively, we generate the sample by replacing the target region of the image with the counterfactual generated using Equation 3.7 - called *Replaced at the End*. This setting is included to avoid generating out-of-distribution noise in the untouched regions for reference.

Regarding the adversarial algorithm, the attack of choice is the traditional PGD [114]. Including the mask in the PGD attack merely requires changing the iterative process to:

$$x'_{i+1} = \prod_{B(x,\epsilon,\infty)} x'_i - M \odot \gamma \; sgn(\nabla_{x'_i} L_{class}(C(y|x'_i))), \tag{3.22}$$

where $\prod_{B(x,\epsilon,\infty)}$ is the projection function on the ϵ-ball over x under the ℓ_∞ norm, x'_i is the i'th iteration with $x'_0 = x$, sgn is the sign operation, and γ is the step size. We implemented PGD with 50 iterations with $\epsilon \in \{2/255, 4/255, 8/255\}$, which is sufficient to fool an undefended model.

Figure 3.6 shows the FR of the different methods presented above for the smile attribute. The most important observation of these experiments can be seen for mask 9

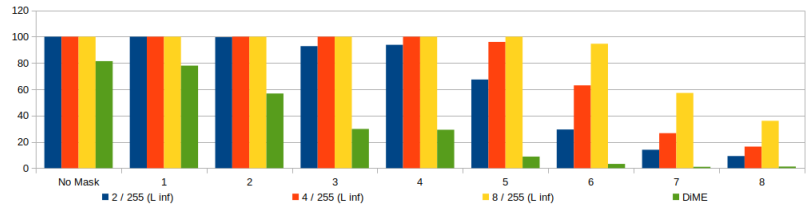


FIGURE 3.7: **Searching for adversarial noise (*Age* Attribute).** We visualize the results of the localized counterfactual explanations and PGD noise. We observe that for areas that contain pixels that are not correlated with the 'Age' class (e.g. mask 8), DiME fails to produce a counterfactual image, which is the desired behavior, while there were possible adversarial attacks.

which corresponds to the forehead region. It is possible to produce adversarial examples by modifying the forehead region, while DiME fails to produce them. Indeed, this is a desirable behavior, since it should not be possible to make the person smile by modifying her/his forehead. The second observation is that both methods for creating spatially constrained counterfactuals have similar performance. In fact, all values have the same value except for mask 4, where the *replacing at the end* method has a marginally better performance by 0.004 points. From these experiments, we conclude that the DDPM is a great regularizer that performs semantic editing while avoiding the addition of adversarial noise.

Figure 3.7 shows the same experiment but for the age attribute. In a general fashion, the experiment follows the same trend as the smile ablation. Therefore, the same conclusion can be drawn from the previous comments.

3.5.3 Trade-off between Quality and Inference Speed

The computational complexity of DiME is a limitation for on-the-fly use. This reduces its applicability in real-world scenarios where speed is paramount. Fortunately, DDPMs have a mechanism to reduce the number of sampling steps. However, this strategy comes at the expense of image quality. Therefore, this section explores the trade-off between sampling strategy and image quality.

T (S)	**FR**(↑)	**FID**(↓)	ℓ_1(↓)	**Speed**(↓)
200 (60)	97.5	20.45	0.043	82.92s
100 (30)	97.6	20.70	0.044	21.27s
50 (15)	96.1	22.67	0.046	5.76s
20 (6)	93.9	27.49	0.059	1.10s
DiVE^{100}	92.3	50.71	0.108	5.14s

TABLE 3.5: **Inference Speed.** Diffusion models are capable of sampling using different timesteps without any fine-tuning. Here, several sampling strategies were tested without changing the initial step to total number of timestep ratio. T and S stand for timesteps and steps, respectively. DiME's original sampling step is 200 (60).

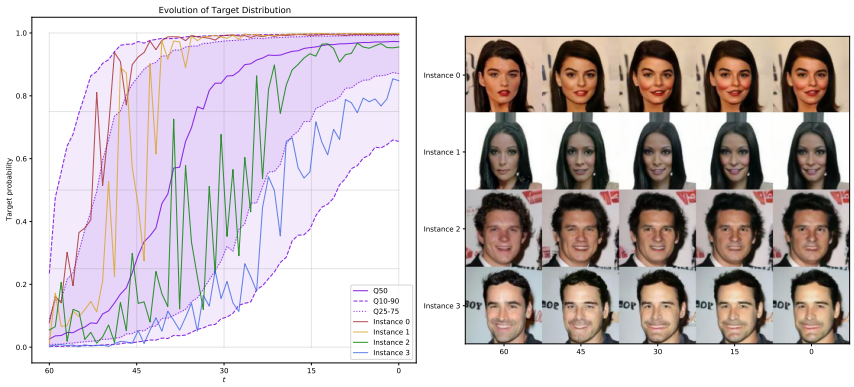


FIGURE 3.8: **Target distribution at each time-step.** This figure visualizes the evolution of the target labels' probability. Each purple line represents a quantile of the probabilities. The colored curves are cases shown on the right. In expectancy, the clean image probability increases at each time step. Nevertheless, the curves are typically sporadic but, with an increasing tendency.

Table 3.5 shows the results of the ablation. There are three observations: **(i)** As expected, reducing the number of iterations worsens each metric while decreasing the inference time. **(ii)** Also, halving the number of steps reduces the inference speed by a factor of four. This result is due to the square growth associated with the number of steps. **(iii)** Finally, compared to DiVE^{100}, using a schedule of 50 steps provides a similar inference time. However, even in this case, DiME enjoys better quantitative metrics. That being said, it is recommended to use more steps for better qualitative results for the end user.

3.5.4 Distribution Overtime

The proposed pipeline uses the unconditional DDPM to enable the use of the classifier under observation. At each step, the classifier uses the generated image to compute the gradient with respect to the target label. Therefore, this image reveals information about the optimization process at each time step.

Figure 3.8 visualizes the evolution of the probability of the target labels over time, along with some examples. There is a clear average increase in probability over time. However, the examples show sporadic and non-steady trends. Yet, there is an increasing behavior. The most unstable behavior is near the first steps. Nevertheless, the optimization starts to settle when it reaches a time step close to 0 (about $t = 20$). This behavior results from an averaging effect over time; when the image generation reaches the final steps, the variance nearly disappears. This observation is related to the comments of Equation 3.8, where we argue for using a single realization of x_t at each time step. As mentioned there, the lack of averaging at each step is partially mitigated in terms of the optimization objective by an averaging effect over time. But thanks to the randomness inherited from the early steps ($t \approx \tau$), the overall CE generation process still shows some diversity.

3.5.5 Hyperparameter Ablations

Continuing with DiME ablation study, this subsection analyzes the effect of the set of DiME hyperparameters introduced in subsection 3.3.2.

Initial Step Ablation

The first variable of interest is the initialization step τ. This hyperparameter is responsible for the initial noise level. Accordingly, Table 3.6 reports this hyperparameter exploration. The results show that on the one hand, a higher τ opens more opportunities to modify the image. The image generation has more optimization steps when the initial noise level increases. Thus, it easily reaches a counterfactual that fools the classifier at the cost of decreasing CE sparsity, an unwanted effect in the CE community. On the other hand, this increased generation power can be detrimental to the resulting image quality. Thus, a low τ increases sparsity, but the CEs are not as successful. Choosing $\tau = 60$ provides a good trade-off.

Steps	**FID$^+$**(↓)	**FR**(↑)	ℓ_1(↓)
50	**20.19**	92.4	**0.0406**
60	20.94	97.9	0.0430
70	23.21	**99.7**	0.0479

TABLE 3.6: **Initialization Step.** This table shows the result of different τ choices. The ℓ_1 and FID metrics are computed solely from the successful counterfactual explanations. Using $\tau = 60$ provides the best trade-off between image quality, Flip Ratio, and similarity. We computed the FID$^+$ taking the same number of samples for the experiment with fewer instances ($\tau = 50$).

Gradients' Scale Ablation

The scaling parameter λ_c guides the generation process to go beyond the decision frontier of the classifier. Accordingly, this section investigates the influence of this variable in the guidance process. To this end, this experiment runs DiME with three different scales, namely $\lambda_c \in \{8, 10, 15\}$, to study its effect on CE generation.

Table 3.7 reports the outcomes of this experiment. In this particular scenario, the results show a trade-off between success rate and image quality. This is because setting a low λ_c produces fewer modifications with better qualitative properties. Accordingly, since most of the explanations were created with the lowest scale, DiME excelled in these metrics assessed with FID and ℓ_1. Now, by increasing to higher scales, DiME boosts the FR at the cost of lowering the image quality. From a qualitative point of view, adding too much gradient introduces out-of-distribution noise. We believe that the DDPM cannot detect this distortion, so it produces artifacts on the image. However, these artifacts may coincide with patterns that affect the classifier's response.

λ_c	**FID**$^+(\downarrow)$	**FR**$(\uparrow)$	$\ell_1(\downarrow)$
8	22.93	80.1	**0.0427**
10	23.32	88.0	0.0432
15	25.87	97.7	0.0446
DiME	**22.48**	**97.9**	0.0430

TABLE 3.7: **Gradient Scales.** This table shows the impact of different scale choices. Increasing the gradient scale λ_c decreases the FID and ℓ_1. From the Flip Ratio results, we see that most explanations are produced with a low scale value, hence producing similar results in the pixel space with high fidelity. Harder instances require the use of an increased scale to successfully produce the counterfactual example. The FID$^+$ is computed by taking the same number of samples for the experiment $\lambda_c = 8$.

Distance Losses Ablations

ℓ_1	L_{perc}	**FID**$(\downarrow)$	**FR**$(\uparrow)$	$\ell_1(\downarrow)$
χ	χ	24.05	**98.8**	0.051
✓	χ	23.48	98.0	0.048
χ	✓	22.32	98.0	0.046
✓	✓	**20.51**	97.9	**0.043**

TABLE 3.8: **Distance Loss ablations.** Effect of ℓ_1 and L_{perc} losses. Each loss contributes to enhancing the FID individually, while jointly providing a great qualitative and quantitative boost.

The CE method seeks to produce the smallest plausible changes in the counterfactual. To this end, recall that DiME computes the perceptual loss L_{perc} between the input instance x and the current instance x_t at timestep t to increase their similarity. Additionally, it computes an ℓ_1 loss between the input sample x and the noisy instance z_t. Thereby, this section ablates the effect of adding the distance losses ℓ_1 and L_{perc} to DiME.

Table 3.8 shows the results of this experiment. At first glance, we notice that the FR is greater than adding any distance loss. This is an expected result, since removing any distance loss relaxes the optimization problem. Thus, DiME can modify as many aspects because the proximity of the input instance is irrelevant. In contrast, adding every single loss reduces marginally the success of our model. Despite this, the FID and ℓ_1 metrics improve favorably. Note that the perceptual loss improves both metrics more than the ℓ_1 loss. Finally, adding both losses at the same time significantly decreases both metrics.

3.6 Limitations

This chapter shows the benefits of using DiME, but many aspects are far from being achieved for the XAI community. For example, DiME has two limitations. On the one hand, the proposed approach adopts the most problematic aspect of DDPMs: the inference time. Namely, DiME needs to use the DDPM model about 1800 times to generate a single explanation. This aspect is undesirable whenever the user needs an explanation on the fly. On the other hand, we require access to the training data; a limitation common to many previous studies. However, this aspect is crucial in domains with sensitive data. Although access to the training data is allowed in many cases, we restrict ourselves to using the data without any labels.

3.7 Conclusion

This work explores the novel diffusion models in the context of counterfactual explanations. By harnessing the conditional generation of guided diffusion, we achieve successful counterfactual explanations through DiME. These explanations follow the requirements given by the XAI community: a small but tangible change in the image while remaining realistic. The performance of DiME is confirmed using a battery of standard metrics. Furthermore, the current approach to validate the sparsity of CE has significant conflicts with the assessment of spurious correlation detection. The proposed metric, Correlation Difference, correctly measures the impact of measuring the subtle correlation between labels. Moreover, DiME also exhibits strong diversity in the explanation produced. This is partly inherited from the intrinsic properties of diffusion models, but also results from a careful design of our approach. Finally, we hope that our work opens up new ways to compute and evaluate counterfactual explanations.

Epilogue

At the time of the conference publication, no other works were focusing on counterfactual explanations using diffusion models. Concurrent [10, 156] and subsequent [46, 123] works have demonstrated that guided diffusion is indeed a valuable tool for constructing counterfactuals. The main differences between these works and DiME are how they handle the gradient backpropagation from the clean instance to the noisy one. While DiME opted to approach this in a brute-force manner, the previously cited papers use the same technique as Avrahami, Lischinski, and Fried [12] to create the gradients.

Several non-diffusion approaches, such as ZOOM [113] and C3LT [85], generate counterfactual explanations from strictly generated images. While these methods

resemble DiME in their use of direct optimization processes to create counterfactuals, other non-diffusion approaches incorporate additional elements. For instance, some utilize segmentation maps [74] or employ generation via scene decomposition [200] to partition the scene into multiple components. We posit that DiME and our subsequently proposed approaches could benefit from adopting similar strategies because the inclusion of object-aware generators facilitates the creation of more complex explanations, such as those required for datasets like BDD100k [197].

A key discussion is how DiME compares to contemporary approaches for CE generation. In our opinion, DiME continues to produce acceptable counterfactuals, even by current standards. Moreover, it generates perceptually diverse explanations when executed with different initializations, a characteristic that only the work of Rodríguez et al. [150] approximated for vision CEs. However, DiME relies on a computationally intensive approach: generating a clean image to produce guiding gradients at every step in the diffusion process, a trait that scales poorly as image dimensions increase. Consequently, adopting alternative strategies like latent diffusion [151] to accelerate the diffusion process could be beneficial, especially given the prevalent trend towards higher image resolutions. Finally, DiME is based on one of the earliest instantiations of diffusion models. Leveraging modern approaches, such as incorporating transformer-based backbones [139], could potentially enhance its performance and efficiency.

Chapter 4

Adversarial Counterfactual Visual Explanations

Prologue

Our first approach presented in the previous chapter, DiME, was a good proof of concept for future approaches to generating counterfactual explanations using diffusion models. Now, with the experience gained from the previous work, we thought about different ways to generate these explanations. At the time, some literature suggested potential links between counterfactual explanations and adversarial attacks [44]. Of course, both approaches share the same goal: to create distortions in the image so that the classifier's prediction changes. Conversely, CE must input plausible modifications, while adversarial examples do not have this hard constraint. In this chapter, our goal is to exploit these attacks to generate CE.

Previous literature on adversarial robustness has shown that adversarial attacks on robust models produce changes understandable by humans [157, 210, 20]. To illustrate this phenomenon, Figure 4.1 visualizes some examples where the adversarial attack produced semantic changes in the image. Thus, robustizing a brittle model should produce human-understandable changes when being attacked. This is advantageous because if we succeed in robustizing the model under consideration, we can use adversarial attacks to generate CEs. The challenge, however, is to

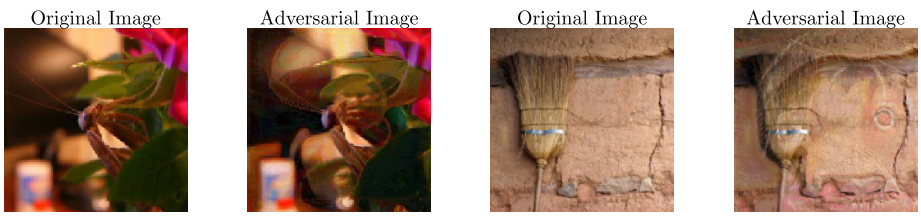


FIGURE 4.1: Adversarial attacks on robust models show human-like features. The first image was originally classified as *mantis*, while the adversarial attack is classified as *sombrero*. Similarly, in the second set of images, the original image was classified as *broom*, while the attack is classified as *Chesapeake Bay Retriever*. In both cases, the attack shows a few characteristics of the selected class. All images were extracted from Pérez et al. [140] study.

achieve this robust state while keeping the target model intact.

To leverage adversarial attacks for CEs, we must first understand how these attacks are constructed. The generation of adversarial attacks follows an iterative approach, with each iteration adding small amounts of adversarial corruption to the input. To remove this corruption from the image, we can use diffusion models. This process involves adding sufficient Gaussian noise to the adversarially corrupted image and to then denoising it using a diffusion model, resulting in an approximate instance closer to the original clean input. Following this logic, we can make the model more robust without altering its weights. So, by incorporating distance regularizations and using the diffusion model to steer the noisy image back to the image manifold, we expect the adversarial attack to create realistic image edits. This reasoning led to the development of the current chapter of this thesis, which was published in CVPR 2023.

One of the main results of this paper is Figure 4.4. Unlike previous and recent studies, and to the best of our knowledge, this work is the only one that shows results of actionability in real-life scenarios. This result is perhaps the most important in this work, as it shows that ACE explanations are realistic and that we were able to find a weakness in a model and test it through an intervention, even if it was in a simple scenario.

4.1 Introduction

The research branch of explainable artificial intelligence has yielded remarkable results, gradually opening the machine learning black boxes. The production of counterfactual explanations (CE) has become one of the promising pipelines for explainability, especially in computer vision [150, 79, 74, 167]. As a matter of fact, CE are an intuitive way to expose how an input instance can be minimally modified to steer the desired change in the model's output. More precisely, CE answers the following: *what does X have to change to alter the prediction from Y to Y'?* From a user perspective, these explanations are easy to understand since they are concise and illustrated by examples. Henceforth, companies have adopted CE as an interpretation methodology to legally justify the decision-making of machine learning models [181]. To better appreciate the potential of CE, one may consider the following scenario: a client goes to a photo booth to take some ID photos, and the system claims the photos are invalid for such usage. Instead of performing random attempts to abide by the administration criteria, an approach based on CE could provide visual indications of what the client should fix.

The main objective of CE is to add minimalistic semantic changes in the image to flip the original model's prediction. Yet, these generated explanations must accomplish several objectives [181, 79, 150]. A CE must be *valid*, meaning that the CE has to change the prediction of the model. Secondly, the modifications have to be *sparse*

and proximal to the input data, targeting to provide simple and concise explanations. In addition, the CE method should be able to generate *diverse* explanations. If a trait is the most important for a certain class among other features, diverse explanations should change this attribute most frequently. Finally, the semantic changes must be *realistic*. When the CE method inserts out-of-distribution artifacts in the input image, it is difficult to interpret whether the flipping decision was because of the inserted object or because of the shifting of the distribution, making the explanation unclear.

Adversarial attacks share a common goal with CE: flipping the classifier's prediction. For traditional and non-robust visual classifiers, generating these attacks on input instances creates imperceptible noise. Even though it has been shown that it contains meaningful changes [73] and that adversarial noise and counterfactual perturbations are related [44, 72], adversarial attacks have lesser value. Indeed, the modifications present in the adversaries are unnoticeable by the user and leave him with no real feedback.

Contrary to the previous observations, many papers (*e.g.*, Pérez et al. [140]) evidenced that adversarial attacks toward *robust* classifiers generate semantic changes in the input images. This has led works [157, 210] to explore robust models to produce data using adversarial attacks. In the context of counterfactual explanations, this is advantageous [20, 160] because the optimization will produce semantic changes to induce the flipping of the label.

Then two challenges arise when employing adversarial attacks for counterfactual explanations. On the one hand, when studying a classifier, we must be able to explain its behavior regardless of its characteristics. So, a naive application of adversarial attacks is impractical for non-robust models. On the other hand, according to Tsipras et al. [174], robustifying the classifier yields an implicit trade-off by lowering the *clean accuracy*, as referred by the adversarial robustness community [33], a particularly crucial trait for high-stakes areas such as the medical field [116].

The previous remarks motivate our endeavor to mix the best of both worlds. Hence, in this chapter, we propose robustifying brittle classifiers *without* modifying their weights to generate CE. This robustification, obtained through a filtering preprocessing leveraging diffusion models [64], allows us to keep the performance of the classifier untouched and unlocks the production of CE through adversarial attacks.

We summarize the novelty of this work as follows: (i) We propose Adversarial Counterfactual Explanations, ACE in short, a novel methodology based on adversarial attacks to generate semantically coherent counterfactual explanations. (ii) ACE performs competitively with respect to the other methods, beating previous state-of-the-art methods in multiple measurements along multiple datasets. (iii) Finally, we point out some defects of current evaluation metrics and propose ways to remedy their shortcomings. (iv) To show a use case of ACE, we study ACE's meaningful

and plausible explanations to comprehend the mechanisms of classifiers. We experiment with ACE findings producing actionable modifications in real-world scenarios to flip the classifier decision.

Our code and models are available on GitHub.

4.2 Related Work

Explainable AI. The main dividing line between the different branches of explainable artificial intelligence stands between *Ad-Hoc* and *Post-Hoc* methods. The former promotes architectures that are interpretable by design [153, 17, 18, 70] while the latter considers analyzing existing models as they are. Since our setup lies among the Post-Hoc explainability methods, we spotlight that this branch splits into global and local explanations. The former explains the general behavior of the classifier, as opposed to a single instance for the latter. This work belongs to the latter. There are multiple local explanations methods, from which we highlight saliency maps [76, 185, 101, 27, 89, 205], concept attribution [87, 52, 94] and model distillation [171, 50]. Concisely, these explanations try to shed light on *how* a model took a specific decision. In contrast, we focus on the on-growing branch of counterfactual explanations, which tackles the question: *what* does the model uses for a forecast? We point out that some novel methods [177, 56, 188, 187] call themselves counterfactual approaches. Yet, these systems highlight regions between a pair of images without producing any modification.

Counterfactual Explanations. CE have taken momentum in recent years to explain model decisions. Some methods rely on prototypes [108] or deep inversion [172], while other works explore the benefits of other classification models for CE, such as Invertible CNNs [71] and Robust Networks [20, 160]. A common practice is using generative tools as they give multiple benefits when producing CE. In fact, using generation techniques is helpful to generate data in the image manifold. There are two modalities to produce CE using generative approaches. Many methods use conditional generation techniques [176, 167, 108] to fit what a classification model learns or how to control the perturbations. Conversely, unconditional approaches [150, 130, 79, 164, 204, 85] optimize the latent space vectors.

We'd like to draw attention to Jeanneret *et al.* [79]'s counterfactual approach, which uses a modified version of the guided diffusion algorithm to steer image generation towards a desired label. This modification affects the DDPM generation algorithm itself. In contrast, while we also use DDPM, we use it primarily as a regularizer before the classifier. Instead of controlling the generation process, we generate semantic changes using adversarial attacks directly on the image space, and then post-process the image using a standard diffusion model. Furthermore, we use a refinement stage to perform targeted edits only in regions of interest.

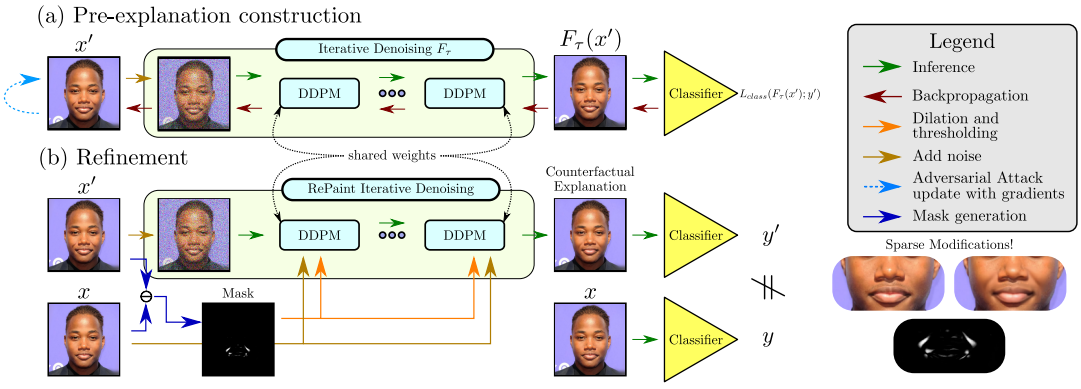


FIGURE 4.2: **Pre-explanation Construction and Refinement** ACE generates the counterfactual explanation in a two-step sequence. Initially, (a) To generate semantic updates in the input image, the DDPM processes the instance before computing the loss function $L_{class}(F_\tau(x'); y')$, where y' is the target label. To simplify the process, we omit the distance loss between the perturbed image x' and the input image x. Then, we compute the gradients with respect to x' and update it using the adversarial attack. Finally, (b) we generate a binary mask using the magnitude's difference between the explanation and input image to refine the pre-explanation using RePaint's inpainting method.

Adversarial Attacks and their relationship with CE. Adversarial attacks share the same main objective as counterfactual explanations: flipping the forecast of a target architecture. On the one hand, *white-box* attacks [55, 114, 25, 121, 33, 77] leverage the gradients of the input image with respect to a loss function to construct the adversary. In addition, universal noises [120] are adversarial perturbations created for fooling many different instances. On the other hand, *black-box* attacks [208, 143, 8] restrain their attack by checking merely the output of the model. Finally, [132] study DDPMs from a robustness perspective, disregarding the benefits of counterfactual explanations.

In the context of CE for visual models, the produced noises are indistinguishable for humans when the network does not have any defense mechanism, making them useless. This lead works [72, 3, 137] to approach the relationship between these two research fields. Compared to previous approaches, we manage to leverage adversarial attacks to create semantic changes in undefended models to explore their semantic weaknesses perceptually in the images; a difficult task due to the nature of the data.

4.3 Adversarial Counterfactual Explanations

The key contribution of this chapter is our novel Adversarial Counterfactual Explanations (ACE) method. ACE produces counterfactual images in two steps, as seen in Figure 4.2. We briefly introduce these two steps here and detail them in the following sections.

Step 1. Producing pre-explanation images (subsection 4.3.1). Let $L_{class}(x; y)$ be a function measuring the agreement between the sample x and class y. This function is typically the cross-entropy loss of the classifier we are studying with respect to y. With ACE, generating the pre-explanation image of (x, y) for the target class $y' \neq y$ consists in finding x' minimizing $L_{class}(F(x'); y')$ using the adversarial attack as the optimizer. Here, $F(x')$ is a filtering function that constrains the attack to stay in the manifold of the training images. In a nutshell, the filtering process F robustifies the fragile classifier under examination to generate semantic changes *without* modifying its weights.

Step 2. Bringing the pre-explications closer to the input images (subsection 4.3.2). The pre-explanation generation restricts only those pixels in the image that are useful in switching the output label from y to y'. The rest of the pixels are only implicitly constrained by the design of F. Accordingly, the purpose of this second step is to keep these non-explicitly constrained pixels identical to those of the input image.

4.3.1 Pre-explanation generation with DDPMs

To avoid generating adversarial noise and producing useful semantics, the previously introduced function F should have two key properties. (i) Removing high-frequency information that traditional adversarial attacks generate. Indeed, these perturbations could change the classifier's decision without being actionable or understandable by a human. (ii) Producing in-distribution images without distorting the input image. This property seeks to maintain the image structures not involved in the decision-making process as similar as possible while avoiding giving misleading information to the user.

Denoising Diffusion Probabilistic Models [64], commonly referred to as DDPM or diffusion models, achieve these properties if used properly. On the one hand, each inference through the DDPM is a denoising process; in particular, it removes high-frequency signals. On the other hand, DDPMs generate in-distribution images.

As a reminder, DDPMs rely on two Markov chains, one inverse to the other. The forward chain *adds* noise from a state t into $t+1$ while the reverse chain *removes* it from $t+1$ to t. Noting x_t the instance at time step t, the forward chain is directly simulated from a clean instance x_0 through

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon,\ \epsilon \sim \mathcal{N}(0, I), \tag{4.1}$$

where $\bar{\alpha}_t$ is a time-dependent constant. At inference, the DDPM produces a mean $\mu_t(x_t)$ and a deviation matrix $\Sigma_t(x_t)$. Using these variables, the next less noisy image is sampled from

$$x_{t-1} = \mu_t(x_t) + \Sigma_t(x_t)\, \epsilon,\ \epsilon \sim \mathcal{N}(0, I). \tag{4.2}$$

Thus, the DDPM denoising algorithm iterates the previous step until $t = 0$ arriving at an image without noise. Please refer to previous works [36, 64] for a thorough understanding of diffusion models.

ACE pre-explanation generation. Starting from a query image x, we can obtain a filtered version by applying the forward DDPM process up to level τ (Equation 4.1) and then denoise it recursively thanks to the iterative DDPM denoising steps (Equation 4.2) starting from level $t = \tau$. In this case, to highlight the use of this intermediate step τ, we denote the diffusion filtering process as $F = F_\tau$ (Figure 4.2a). Thus, we optimize the image through the DDPM filtering process, F_τ, before computing the classification loss. Henceforth, we obtain the pre-explanations by optimizing

$$\underset{x'}{\operatorname{argmin}}\ L_{class}(F_\tau(x'); y') + \lambda_d\, d(x', x) \tag{4.3}$$

using the adversarial attack of choice. Here, λ_d is a regularization constant and d a distance function.

4.3.2 Bringing the pre-explanations closer to the input images

By limiting the value of τ, the DDPM will not go far enough to generate a normal distribution, and the reconstruction will somehow preserve the overall structure of the image. However, we noted that a post-processing phase could help keep irrelevant parts of the image untouched. For example, in the case of faces, the denoising process may change the hairstyle while targeting the smile attribute. Since hairstyle is presumably uncorrelated with the smile feature, the post-process should neutralize those unnecessary alterations.

To this end, we first compute a binary mask m delineating regions that qualify for modifications. To do so, we consider the magnitude difference between the pre-explanation and the original mask, we dilate this gray-scale image and threshold it, yielding the desired mask. This matter being settled, we need to fuse the CE inside the mask along with the input outside the mask to accomplish our objective.

With that aim, a natural strategy is using inpainting methods. So, we leverage RePaint's recent technique [111], originally designed for image completion, and adapt it to our *picture-in-picture* problem (Figure 4.2b). This adaptation straightforward and integrates very well with the rest of our framework. It starts from the noisy pre-explanations x_τ and iterate the following altered denoising steps:

$$x_{t-1} = \mu_t(x'_t) + \Sigma_t(x'_t)\,\epsilon,\ \epsilon \sim \mathcal{N}(0, I), \tag{4.4}$$

where $x'_t = x_t \cdot m + x^i_t \cdot (1 - m)$ is the raw collage of the current noisy reconstruction x_t and the noisy version of the initial instance x^i_t at the same noise level t, obtained with Equation 4.1. The final image, x_0, will be our counterfactual explanation – identical to the input sample outside the mask, and very similar to the

Smile							
Method	FID	sFID	FVA	FVA_d	MNAC	CD	COUT
DiVE	29.4	-	97.3	-	-	-	-
$DiVE^{100}$	36.8	-	73.4	-	4.63	2.34	-
STEEX*	10.2	-	96.9	-	-	-	-
DiME	3.17	4.89	98.3	0.729	3.72	2.30	0.5259
ACE ℓ_1	**1.27**	**3.97**	**99.9**	**0.874**	*2.94*	*1.73*	**0.7828**
ACE ℓ_2	*1.90*	*4.56*	**99.9**	*0.867*	**2.77**	**1.56**	*0.6235*
Age							
Method	FID	sFID	FVA	FVA_d	MNAC	CD	COUT
DiVE	33.8	-	*98.2*	-	4.58	-	-
$DiVE^{100}$	39.9	-	52.2	-	4.27	-	-
STEEX*	11.8	-	97.5	-	3.44	-	-
DiME	4.15	5.89	95.3	0.6714	*3.13*	3.27	0.4442
ACE ℓ_1	**1.45**	**4.12**	**99.6**	*0.7817*	3.20	*2.94*	**0.7176**
ACE ℓ_2	*2.08*	*4.62*	**99.6**	**0.7971**	**2.94**	**2.82**	*0.5641*

TABLE 4.1: **CelebA Assessement.** Main results for CelebA dataset. *STEEX uses use additional data for their counterfactual explanations. Hence we are not comparable with them directly. We extracted the results from DiME and STEEX papers. In **bold** and *italic* we show the best and second best performances.

pre-explanation within the mask. In the supplementary material (Appendix A), we added an overview of ACE.

4.4 Experimentation

4.4.1 Evaluation Protocols and Datasets

Datasets. In line with the recent literature on counterfactual images [150, 167, 79, 80], first, we evaluate ACE on CelebA [107], with images of size of 128×128 and a DenseNet121 classifier [69], for the 'smile' and 'age' attributes. Following Jacob *et al.* [74], we experimented on CelebA HQ [99] and BDD100k [197]. CelebA HQ has a higher image resolution of 256×256. BDD100k contains complex traffic scenes as 512×256 images; the targeted attribute is 'forward' vs 'slow down'. The decision model is also a DenseNet121, trained on the BDD-IOA [192] extension dataset. Regarding the classifiers for which we want to generate counterfactuals, we took the pre-trained weights from DiME [79] source for CelebA and from STEEX [74] for CelebA HQ and BDD100k, for fair comparisons.

Evaluation criteria for quantitative evaluation.

Validity of the explanations is commonly measured with the Flip Rate (FR), *i.e.* how often the CE is classified as the targeted label.

Diversity is measured by extending the diversity assessment from Mothilal *et al.* [122]. As suggested by Jeanneret *et al.* [79], the diversity is measured as the average LPIPS [203] distance between pairs of counterfactuals (σ_L).

Sparsity or proximity has been previously evaluated with several different metrics [150, 167], in the case of face images and face attributes. On the one hand, the mean number of attributes changed (MNAC) measures the smallest amount of traits changed between the input-explanation pair. Similarly, this metric leverages an oracle network pretrained on VGGFace2 [24] and then fine-tuned on the dataset. Further, Jeanneret *et al.* [79] showed the limitations of the MNAC evaluation and proposed the CD metric to account for the MNAC's limitations. On the other hand, to measure whether an explanation changed the identity of the input, the assessment protocol uses face verification accuracy [24] (FVA). To this end, the evaluation uses a face verification network. However, FVA has 2 main limitations: i) it can be applied to face related problems only, ii) it works at the level of classifier decisions which turns out to be too rough when comparing an image to its CE, as it involves only a minimal perturbation. For face problems, we suggest skipping the thresholding and consider the mean cosine distance between the encoding of image-counterfactual pairs, what we refer to as Face Similarity (FS). To tackle non-face images, we propose to extend FS by relying on self-supervised learning to encode image pairs. To this end, we adopted SimSiam [31] as an encoding network to measure the cosine similarity. We refer to this extension as SimSiam Similarity (S^3). Finally, also for classifiers that are not related to faces, Khorram *et al.* [85] proposed COUT to measure the transition probabilities between the input and the counterfactual.

Realism of counterfactual images [167] is usually evaluated by the research community with the FID [63] between the original set and the valid associated counterfactuals. We believe there is a strong bias as most of the pixels of counterfactuals are untouched and will dominate the measurement, as observed in our ablation studies (subsection 4.4.6). To remove this bias, we split the dataset into two sets, generating the CE for one set and measuring the FID between the generated explanations and the other set, iterating this process ten times and taking the mean. We call this metric sFID.

Implementation details. One of the main obstacles of diffusion models is transferring the gradients through all the iterations of the iterative denoising process. Fortunately, diffusion models enjoy a time-step re-spacing mechanism, allowing us to reduce the number of steps at the cost of a quality reduction. So, we drastically decreased the number of sampling steps to construct the pre-explanation. For CelebA [107], we instantiate the DDPM [36] model using DiME's [79] weights. In practice, we set $\tau = 5$ out of 50 steps. For CelebA HQ [99], we fixed the same τ, but we used the re-spaced time steps to 25 steps. For BDD100k [197], we follow the same settings as STEEX [74]: we trained our diffusion model on the 10.000 image subset of BDD100k. To generate the explanations, we used 5 steps out of 100. Additionally, all our methods achieve a success ratio of 95% at minimum. We will detail in the Appendix A all instructions for each model on every dataset. We adopted an ℓ_1 or ℓ_2 distance for the distance function. Finally, for the attack optimization, we chose

Smile							
Method	FID	sFID	FVA	FS	MNAC	CD	COUT
DiVE	107.0	-	35.7	-	7.41	-	-
STEEX	21.9	-	*97.6*	-	5.27	-	-
DiME	18.1	27.7	96.7	0.6729	2.63	**1.82**	**0.6495**
ACE ℓ_1	**3.21**	**20.2**	**100.0**	**0.8941**	**1.56**	2.61	0.5496
ACE ℓ_2	*6.93*	*22.0*	**100.0**	*0.8440*	*1.87*	*2.21*	*0.5946*
Age							
Method	FID	sFID	FVA	FS	MNAC	CD	COUT
DiVE	107.5	-	32.3	-	6.76	-	-
STEEX	26.8	-	96.0	-	5.63	-	-
DiME	18.7	*27.8*	*95.0*	0.6597	2.10	*4.29*	**0.5615**
ACE ℓ_1	**5.31**	**21.7**	**99.6**	**0.8085**	**1.53**	5.4	0.3984
ACE ℓ_2	*16.4*	28.2	**99.6**	*0.7743*	*1.92*	**4.21**	*0.5303*

TABLE 4.2: **CelebAHQ Assessment.** Main results for CelebA HQ dataset. We extracted the results from STEEX's paper. In **bold** and *italic* we show the best and second-best performances, respectively. ACE outperforms most methods in many assessment protocols.

Method	FID	sFID	S^3	COUT	FR
BDD-OIA					
DiME	13.70	26.06	0.9340	0.3188	91.68
ACE ℓ_1	**2.09**	**22.13**	**0.9980**	*0.7404*	99.91
ACE ℓ_2	*3.3*	*22.75*	*0.9949*	**0.7840**	100.0
BDD100k					
STEEX	58.8	-	-	-	99.5
DiME	7.94	11.40	0.9463	0.2435	90.5
ACE ℓ_1	**1.02**	**6.25**	**0.9970**	*0.7451*	99.9
ACE ℓ_2	*1.56*	*6.53*	*0.9946*	**0.7875**	99.9

TABLE 4.3: **BDD Assessement.** Main results for BDDOIA and BDD100k datasets. We extracted STEEX's results from their paper. In **bold** and *italic* we show the best and second-best performances, respectively.

the PGD [114] without any bound and with 50 optimization steps.

4.4.2 Comparison Against the State-of-the-Art

In this section, we quantitatively compare ACE against previous State-of-the-Art methods. To this end, we show the results for CelebA [107] and CelebA HQ [99] datasets in Table 4.1 and Table 4.2, respectively. Additionally, we experimented on the BDD100k [197] dataset (Table 4.3). To extend the study of BDD, we further evaluated our proposed approach on the BDD-IOA [192] validation set, also presented in Table 4.3. Since DiME [79] showed superior performance over the literature [150, 167, 80], we compare only to DiME.

DiME experimented originally on CelebA only. Hence, they did not tune their parameters for CelebA HQ and BBD100k. By running their default parameters,

DiME achieves a flip rate of 41% in CelebA HQ. We fix this by augmenting the scale hyperparameter for their loss function. DiME's new success rate is 97% for CelebA HQ. For BDD100k, our results showed that using fewer steps improves the quality. Hence, we used 45 steps out of their re-spaced 200 steps. Unfortunately, we only managed to increase their success ratio to 90.5%.

These experiments show that the proposed methodology beats the previous literature on most metrics for all datasets. For instance, ACE, whatever the chosen distance, outmatches DiME on all metrics in CelebA. For the CelebA HQ, we noticed that DiME outperforms ACE only for the COUT and CD metrics. Yet, our proposed method remains comparable to theirs. For BDD100k, we remark that our method consistently outperforms DiME and STEEX.

Two additional phenomena stand out within these results. On the one hand, we observed that the benefit of favoring ℓ_1 over ℓ_2 depends on the characteristics of the target attribute. We noticed that the former generates sparser modifications, while the latter tends to generate broader editing. This makes us emphasize that different attributes require distinct modifications. On the other hand, these results validate the extensions for the FVA and FID metrics. Indeed, the difference between the FVA values on CelebA are small (from 98.3 to 99.9). Yet, the FS shows a major increase. Further, for the Age attribute on CelebA HQ, ACE ℓ_2 shows a better performance than DiME for the FID metric. The situation is reversed with sFID as DiME is slightly superior.

To complement our extensive experimentation, we tested ACE on a small subset of classes on ImageNet [35] with a ResNet50. We selected three pairs of categories for the assessment, and the task is to generate the CE targeting the contrary class. For the FID computation, we used only the instances from both categories but not external data since we are evaluating the in-class distribution.

We show the results in Table 4.4. Unlike the previous benchmarks, ImageNet is extremely complex and the classifier needs multiple factors for the decision-making process. Our results reflect this aspect. We believe that current advancements in CE still need an appropriate testbed to validate the methods in complex datasets such as ImageNet. For instance, the model uses the image's context for forecasting. So, choosing the target class without any previous information is unsound.

4.4.3 Diversity Assessment

In this section, we explore ACE's ability to generate diverse explanations. Diffusion models are, by design, capable of generating distributions of images. Like [79], we take advantage of the stochastic mechanism to generate perceptually different explanations by merely changing the noise for each CE version. Additionally, for a fair comparison, we do not use the RePaint's strategy here because DiME does not

Method	FID	sFID	S^3	COUT	FR
Zebra – Sorrel					
ACE ℓ_1	84.5	122.7	0.9151	-0.4462	47.0
ACE ℓ_2	67.7	98.4	0.9037	-0.2525	81.0
Cheetah – Cougar					
ACE ℓ_1	70.2	100.5	0.9085	0.0173	77.0
ACE ℓ_2	74.1	102.5	0.8785	0.1203	95.0
Egyptian Cat – Persian Cat					
ACE ℓ_1	93.6	156.7	0.8467	0.2491	85.0
ACE ℓ_2	107.3	160.4	0.7810	0.3430	97.0

TABLE 4.4: **ImageNet Assessement.** We test our model in ImageNet. We generated the explanations for three sets of classes. Producing CE for these classes remains a challenge.

have any local constraints and can, as well, change useless structures, like the background. To validate our approach, we follow [79] assessment protocol. Numerically, we obtain a diversity score of $\sigma_L = 0.110$ while DiME reports 0.213. Since DiME corrupts the image much more than ACE, the diffusion model has more opportunities to generate distinct instances. In contrast, we do not go deep into the forward noising chain to avoid changing the original class when performing the filtering.

To circumvent the relative lack of diversity, we vary the re-spacing at the refinement stage and the sampled noise. Note that later in the text, we show that using all steps without any re-spacing harms the success ratio. So, we set the new re-spacing such that it respects the accuracy of counterfactuals and fixed the variable number of noise to maintain the ratio between τ and the re-spaced number of sampling steps ($5/50$ in this case). Our diversity score is then of 0.1436. Nevertheless, DiME is better than ACE in terms of diversity, but this is at the expense of the other criteria, because its diversity comes, in part, from regions of the images that should not be modified (for example, the background).

4.4.4 Qualitative Results

We show some qualitative results in Figure 4.3 for all datasets, included some ImageNet examples. From an attribute perspective, some have sparser or coarser characteristics. For instance, age characteristics cover a wider section of the face, while the smile attribute is mostly located in small regions of the image. Our qualitative results expose that different distance losses impose different types of explanations. For this case, ℓ_1 loss exposes the most local and concrete explanations. On the other hand, the ℓ_2 loss generates coarser editing. This feature is desired for certain classes, but it is user-defined. Additionally, we note that the generated mask is useful to spot out the location of the changes. This is advantageous as it exemplifies which changes were needed and where they were added. Most methods do not indicate the

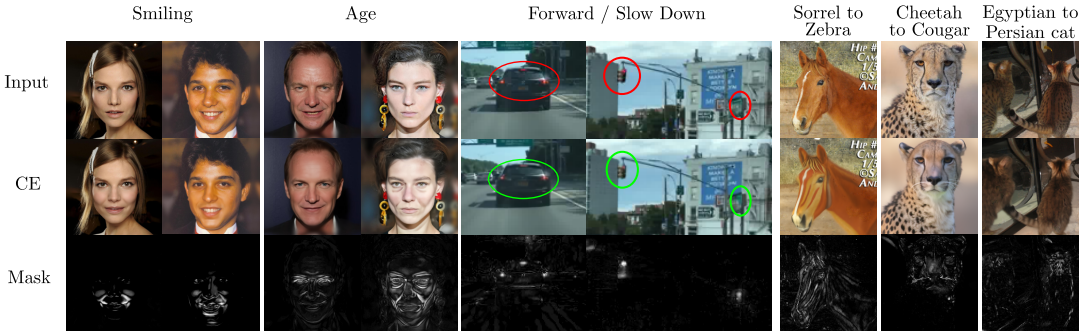


FIGURE 4.3: **Qualitative Results.** ACE create sparse but realistic changes in the input image. Further, ACE enjoys from the generate mask, which helps in understanding which and where semantic editing were added. The first row displays the input images, the second one the counterfactual explanations and the third the corresponding mask.

localization of the changes, making them hard to understand. In the supplementary material (Appendix A), we included more qualitative results.

4.4.5 Actionability

Counterfactual explanations are expected to teach the user plausible modifications to change the classifier's prediction. In this section, we study a batch of counterfactual-input tuples generated with our method. If ACE is capable of creating useful counterfactual explanations, we should be qualified to understand some weaknesses or some behaviors of our classifier. Additionally, we should be able to fool the classifier by creating the necessary changes in real life. To this end, we studied the CelebA HQ classifier for the age and smile attributes.

After surveying some images and their explanations, we identified two interesting results (Figure 4.4). Many of the counterfactual explanations changing from 'young' to 'old' evidence that frowning could change the prediction of the classifier. So, we tested this hypothesis in the real life. We took a photo one individual before and after the frown, avoiding changing the scenery. We were successful and managed to change the prediction of the classifier. For smile, we identified a spurious correlation. Our counterfactuals show that the classifier uses the morphological trait of high cheekbones to classify someone as smiling as well as having red cheeks. So, we tested whether the classification model wrongly predicts as smiling someone with high cheekbones even when this person is not smiling. We also tested whether we can enhance it with some red make up in the cheeks. Effectively, our results show that having high cheekbones is a realistic adversarial feature toward the smiling attribute for the classifier. Also, the classifier confidence (probability) can be strengthened by adding some red make up in the cheeks. These examples demonstrate the applicability of ACE in real scenarios.

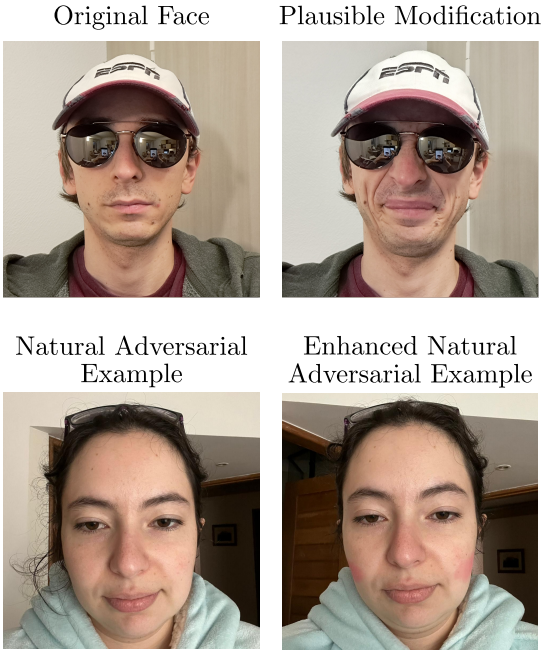


FIGURE 4.4: **Actionability.** From browsing our counterfactuals, we found two weaknesses of the scrutinized classifier. Row 1: We tested if a frown could change the classification from young to old. Row 2: we checked if having high cheekbones flipped is enough to classify someone as smiling. Both experiments were successful.

4.4.6 Ablation Studies

In this section, we scrutinize the differences between the pre-explanation and the refined explanations. Then we explore the effects of using other types of adversarial attacks. Finally, we show that the S^3 metric gives similar results as the FVA, as a sanity check.

Pre-Explanation *vs* Counterfactual Explanations. We explore here, quantitatively and qualitatively, the effects of the pre-explanations (Pre-CE). Also, we apply the diffusion model for the explanations with and without the re-spacing method, using no inpainting strategy, referred as FR-CE and F-CE, respectively. Finally, we compare them against the complete model (ACE). To quantitatively compare all versions, we conducted this ablation study on the CelebA dataset for both 'smile' and 'age' attributes. We assessed the components using the FID, sFID, MNAC, CD, and FR metrics. We did not include the FVA or FS metrics, as these values did not vary much and do not provide insightful information; the FVA is $\sim$99.9 and FS $\sim$0.87 for all versions.

We show the results in Table 4.5. We observe that pre-explanations have a low FID. Nonetheless, their sFID is worse than the F-CE version. As said before, we noticed that including both input and counterfactual in the FID assessment introduces a bias in the final measurement, and this experiment confirms this phenomenon. Additionally, one can check that the MNAC metric between the pre-explanation and

Smile					
Method	FID	sFID	MNAC	CD	FR
Pre-CE	1.87	4.63	3.48	3.05	99.82
FR-CE	8.31	10.30	3.43	1.68	99.97
F-CE	2.64	4.61	3.16	1.56	93.37
ACE	1.27	3.97	2.94	1.73	99.86
Age					
Pre-CE	3.93	6.71	3.76	3.17	99.55
FR-CE	7.10	9.09	3.13	2.66	99.77
F-CE	4.23	6.20	3.53	3.04	93.50
ACE	2.08	4.62	2.94	2.82	99.35

TABLE 4.5: **Refinement Ablation.** We show the importance of each component from ACE. FR stands for flip rate.

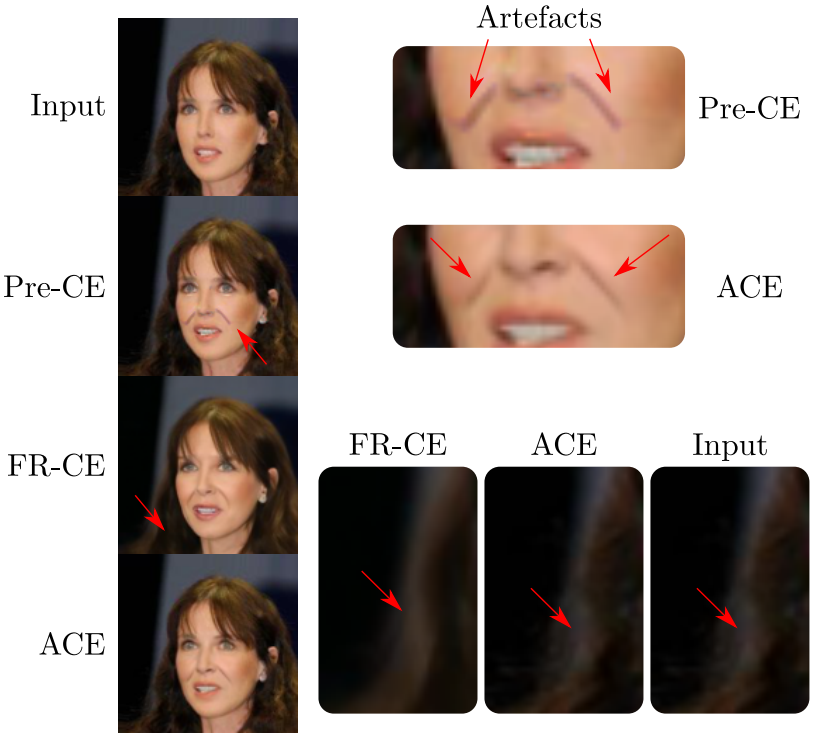


FIGURE 4.5: **Refinement Ablation.** We observe that pre-explanations can have out-of-distribution artifacts. After filtering them, the diffusion process creates in-distribution data, but there are unnecessary changes such as the background. ACE is capable of changing the key features while avoiding modifying unwanted structures.

the FR-CE version does not vary much, yet, the CD metric for the FR-CE is much better. This evidences that the generative model can capture the dependencies between the attributes. Also, we notice that the flip rate (FR) is much lower when using all diffusion steps instead of the re-spaced alternative. We expected this behavior, since we create the pre-explanation to change the classifier's prediction with re-spaced time steps within the DDPM.

Qualitatively, we point out to Figure 4.5, where we exemplify the various stages of ACE. For instance, we see that the pre-explanation contains out of distribution

Metric	Random	ACE	Pre-CE	DiME
Smile				
FS	0.2649 (4)	0.8941 (2)	0.9200 (1)	0.6729 (3)
S^3	0.4337 (4)	0.9876 (2)	0.9927 (1)	0.9396 (3)
Age				
FS	0.2649 (4)	0.7743 (2)	0.8300 (1)	0.6597 (3)
S^3	0.4337 (4)	0.9417 (2)	0.9870 (1)	0.9379 (3)

TABLE 4.6: **S^3 equivalence to FS.** The S^3 metric and the FS are equivalent in a similar context. We show the metric and the order (in parentheses) and observe that both orderings are equal.

artifacts and how the refinement sends it back to the image distribution. Also, we highlight that the filtering modifies the hair, which is not an important trait for the classifier. The refinement is key to avoid editing these regions.

Effect of Different Adversarial Attacks. At the core of our optimization, we have the PGD attack. PGD is one of the most common attacks due to its strength. In this section, we explore the effect of incorporating other attacks. Thus, we tested C&W [25] and the standard gradient descent (GD). Note that the difference between PGD and GD is that GD does not apply the *sign* operation.

Our results show that these attacks are capable of generating semantic changes in the image. Although these are as successful as the PGD attack, we require optimizing the pre-explanation for twice as many iterations. Even when our model is faster than [79] –3.6 times faster–, we require about 500 DDPM iterations to generate an explanation.

Validity of the S^3 Metric. In this experiment, we show that the S^3 and the FS metrics are equivalent when used in the same test bed, *i.e.*, CelebA HQ. To this end, we assess whether the ordering between ACE, pre-explanation, and DiME are equal. To have a reference value, we evaluate the measurements when using a pair of random images. So, we show the values (ordering) for both metrics in Table 4.6 for the Age and Smiling attribute. As we expect, the ordering is similar between both metrics. Nevertheless, we stress that FS is adequate for faces since the network was trained for this task.

4.5 Conclusion

In this chapter, we proposed ACE, an approach to generate counterfactual explanations using adversarial attacks. ACE relies on DDPMs to enhance the robustness of the target classifier, allowing it to create semantic changes through adversarial attacks, even when the classifier itself is not robust. ACE has multiple advantages regarding previous literature, notably seen in the counterfactual metrics. Moreover, we highlight that our explanations are capable of showing natural feature to find sparse and actionable modifications in real life, a characteristic not presented before.

For instance, we were able to fool the classifier with real world changes, as well as finding natural adversarial examples.

4.6 Epilogue

Concurrent work has demonstrated that incorporating adversarial-based approaches yields promising results. Similar to our reasoning, *i.e.*, including adversarial approaches for CE explanations, Boreiko et al. [20] proposed using robust models to generate CEs. However, their method is restricted to robust models only, as they solely employ traditional attacks and a distance loss to generate counterfactuals for a robust classification model.

Similarly, Augustin et al. [10] proposed an approach akin to DiME's (chapter 3), but generates the clean sample to guide the generation process in a single step using the diffusion model. In their work, they utilize a robust model to adjust the gradients of the target (fragile) model, counteracting the poor quality of the single-step generation. Nevertheless, the approach by Augustin et al. [10] necessitates training a robust model on the same dataset to exploit the robust gradients, a process that proves cumbersome even in small datasets.

After the ACE's publication, we tried linking DiME's gradients shortcut to ACE to remove the demand of the gradient computation through the denoising chain of the diffusion process. However, we noticed that the resulting output was not able to change the prediction of the classifier as often. Likewise, the explanation contained bizarre patterns, showing it an unfeasible line of work. In addition, we tried mixing Stable Diffusion [151] and ACE together, to leverage its generative power. Even though we got some interesting results, passing the gradients through a few steps of this colossal generative model was impossible without resulting in memory-reducing techniques, such as the checkpoint [57] method, leading to the next chapter.

Chapter 5

Text-to-Image Models for Counterfactual Explanations: a Black-Box Approach

Prologue

Text-to-image generative models and the classifier learn different concepts that could be summarized equally well in text form. Thus, creating a counterfactual image with a textual prompt containing the feature to be changed may not reveal what the model is using. Consider the case scenario used in the previous chapters: a classifier that detects whether a face is smiling or not. This classifier is trained on celebrity faces, while Stable Diffusion [151] is trained on general faces. There are significant differences between the features presented in the images generated by the generative approach and the images in the dataset, as seen in Figure 5.1. Thus, inpainting the Stable Diffusion's learned concept of "smile" into an image to flip the classifier's prediction will not match the model's learned concept of "smile". This phenomenon calls for fine-tuning approaches in the text space of the generative model.

In this chapter, we propose Text-to-Image Models for Counterfactual Explanations (TIME), a two-step solution to the presented conceptual misalignment problem for generating CEs. First, we use a common approach to distill the information

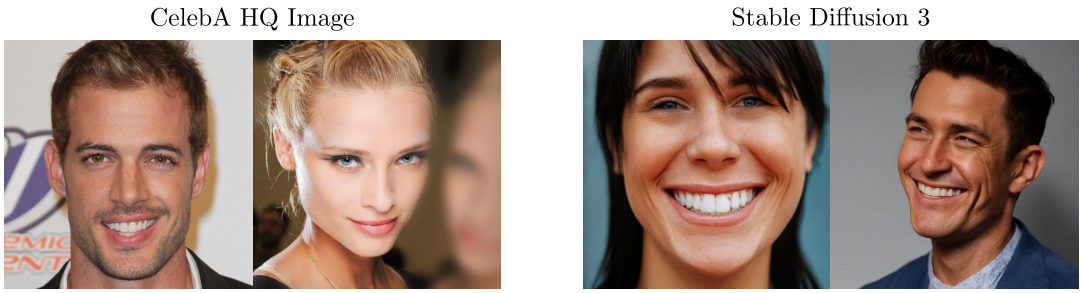


FIGURE 5.1: Stable Diffusion 3 [42] and CelebA HQ [99] smiling images. The textual concept of a smile is the same, but the features of each case are different. Images generated using the prompt *A close-up picture of a face smiling*.

from the classifier to the generative model in the form of textual tokens, *i.e.* textual inversion [49]. To do this, we compute the prediction-image tuple for the training set of the model and train new tokens in the embedding layer of the text interface of Stable Diffusion [151], aligning the learning of the model and the generative model. Second, we leverage perfect inversion techniques [183] to inpaint the corresponding changes in the image. We do this to keep the modified output image as close as possible to the original input.

Our approach has several advantages over previous methods in the literature. First, it is completely black-box. In this chapter, we branch away from the standard definition of black-box models. Traditionally, a black-box model refers to an architecture whose inner workings cannot be understood. Here, we define a black-box model as an architecture where we do not have access to its internal components (weights, gradients, architecture, etc.), but only to the input image and the resulting prediction class. Second, our method eliminates the need to train a diffusion model from scratch. Instead, we simply fine-tune a new text token in the text interface of the generative approach, significantly reducing the training time. Finally, our approach does not rely on optimization during the generation process. Instead, we use DDIM noise inversion techniques that speed up the CE production phase.

5.1 Introduction

Recently, deep neural networks (DNN) have seen increased attention for their impressive forecasting abilities. The use of deep learning in critical applications, such as driving automation, made the scientific community increasingly involved in what a model is learning and how it makes its predictions. These concerns shed light on the field of Explainable Artificial Intelligence (XAI) in an attempt to "open the black-box" and decipher its induced biases.

Counterfactual explanations (CEs) are an attempt to find an answer to this previous problem. They try answering the following question: *What do we need to change in X to change the prediction from Y to Z?* Because CEs give intuitive feedback about what to change to get the desired result, two applications use these explanations: feedback recommendation systems and debugging tools. Take an automated loan approval system as an example. From a user's point of view, if it gets a negative prediction, the user would be more interested in knowing what plausible changes can be made to get a positive result, rather than having an exhaustive list of explanations for why the result is unfavorable. From the debugger's point of view, it can look for biases that were considered in the decision when they should not have been, thus revealing the classifier's weaknesses.

While there are multiple ways to address this question for visual systems, *e.g.* by adding adversarial noise [54], the modifications must be sparse and comprehensive

Method	Model	Training	Specificity	Optim.
DiVE [150]	VAE	Days	Only DNN	Yes
STEEX [74]	GAN	Days	Only DNN	Yes
DiME [79]	DDPM	Days	Only DNN	Yes
ACE [78]	DDPM	Days	Only DNN	Yes
TIME (Ours)	T2I	Hours	Black-Box	No

TABLE 5.1: **Advantages of the proposed methodology.** TIME uses a pre-trained T2I model and trains only a few textual embeddings, requiring hours of training instead of days. It does not require access to the target model (completely black-box) and does not involve any optimization during counterfactual generation.

to provide insight into which variables the model is using. To this end, most studies for CEs use generative models, such as GANs [53], Denoising Diffusion Probabilistic Models (DDPMs) [64], or VAEs [90], as they provide an intuitive interface to approximate the image manifold and constrain the generation in an appropriate space. Although they have several advantages, training these generative models is cumbersome and may not yield adequate results, especially when the data is limited [83]. To this end, we expect that the use of large generative models trained on colossal datasets, such as LAION-5B [159], can provide a sufficient tool to generate CEs. On the one hand, these generative models have shown remarkable qualitative performance, an attractive feature to exploit. Second, since the generative model is already optimized, it can be used to capture data set specific concepts - *e.g.* textual inversion [49] captures the main aspects of a target object when subject to only three to five images.

In this chapter, we explore how to take advantage of Text-to-Image (T2I) generative models for CEs - specifically, using Stable Diffusion [151]. To do so, we take a distillation approach to transfer the learned information from the model into new text embeddings to align the concept class in text space. Second, we use inversion techniques [183] to find the optimal noise to recover the original instance. Finally, with our distilled knowledge, we denoise this optimal point to recover the final instance using the target label, thus generating the CE. This is advantageous because we can tackle the challenging scenario of explaining a black-box model, *i.e.* having access only to its predictions.

Our proposed approach has three main advantages over previous literature, as shown in Table 5.1. First, we only train some textual embeddings, making the training efficient, while previous methods require training a generative model from scratch. Second, we do not require an optimization loop when generating the final counterfactual, which reduces the generation time. Finally, our explainability tool works in a completely black-box environment. While most modern approaches [150, 74, 79, 200, 78] are DNN-specific, because they rely on gradients, our approach, which uses only the output and input as cues, can be used to diagnose any model regardless of its internal functioning. This setting is crucial for privacy-preserving

applications, such as medical data analysis, since eliminating access to the gradients could prevent data leakage [209], as it helps protect personal or confidential information.

We summarize our contributions as follows[1]:

- We propose TIME: Text-to-Image Models for Counterfactual Explanations, using Stable Diffusion [151] T2I generative model to generate CEs.
- Our proposed approach is completely black-box.
- Our counterfactual explanation method based on a distillation approach does not require any optimization during inference, unlike most methods.
- From a quantitative perspective, we achieve similar performance to the previous state-of-the-art, while having access only to the input and the prediction of the target classifier.

5.2 Related Work

5.2.1 Explainable Artificial Intelligence

The research branch of XAI broads multiple ways to provide insights into what a model is learning. As a bird's view analysis, there are two main distinctions between methods: *Interpretable by-design* architectures, and *Post-Hoc* explainability methods. The former searches to create algorithms that directly expose why a decision was made [128, 39, 30, 17, 18, 70, 87]. Our research study is based on the latter. *Post-hoc* explainability methods study pretrained models and try to decipher the variables used for forecasting. Along these lines, there are saliency maps [81, 141, 27, 161], concept attribution [87, 47, 52], or distillation approaches into interpretable by-design models [50]. In this chapter, we study the on-growing branch of CEs [181]. In contrast to previous methods, these explanations are simpler and more aligned with human understanding, making them appealing to comprehend machine learning models.

5.2.2 Counterfactual Explanations

The seminal work Wachter, Mittelstadt, and Russell [181] defined what a counterfactual explanation is and proposed to find them as a minimization problem between a classification loss and a distance loss. In the image domain, optimizing the image's raw pixels produces adversarial noises [54]. So, many studies based their work on Wachter, Mittelstadt, and Russell [181]'s optimization procedure with a generative model to regularize the CE production, such as variational autoencoders [150], generative adversarial networks [74, 200, 85, 113, 167], and diffusion models [79, 156,

[1]Code is available at https://github.com/guillaumejs2403/TIME

78, 10]. In contrast to these works, our proposed approach, TIME, is a distillation approach for counterfactuals. Our method does not require any optimization loop when building the explanation, since we transfer the learning into the T2I model. Furthermore, we do not require access to the gradients of the target model but only the input and output, making it black-box, unlike previous methods.

Co-occurent works analyze dataset biases using T2I models to create distributional shifts in data [180, 145]. Although a valid approach to debug datasets, we argue that these approaches do not search what a model learned but instead a general strategy for the biases in datasets under distributional shifts (*e.g.* it is normal to misclassify a dog with glasses since the model was not trained to classify dog with glasses). Further, their proposed approaches are computationally heavy, since they require fine-tuning Large Language Models or optimizing each inversion step on top of Stable Diffusion. Instead, ours requires training a word embedding, and the inference merely requires Stable Diffusion without computing any gradients, which fits into a single small GPU.

5.2.3 Customization with Text-to-Image Models

Due to the interest in creating unimaginable scenarios with personalized objects, customizing T2I diffusion models has gained attention in recent literature. Textual Inversion [49] and following works [152, 38, 58, 201, 124] are popular approaches to learn to generate specific objects or styles by fine-tuning all or some part of the T2I model. Thus, the new concept can be used in a phrase such that the T2I model will synthesize it.

One of the most difficult problems is editing real-world images with T2I models. The pioneer work of Song *et al.* [168] proposed a non-stochastic variant of DDPMs, called Denoising Diffusion Implicit Models (DDIM). Hence, a single noise seed yields the same image. So, to find an approximate noise, DDIM Inversion noises the image using the diffusion model. Yet, some problems arise with this approximation. So, novel works [118, 135] modify the inversion process by including an inner gradient-based optimization at each noising step, making it unfeasible when analyzing a bundle of images. Finally, Wallace *et al.* [183] proposed to modify the DDIM algorithm into a two-stream diffusion process, reaching a "perfect" inversion. We take advantage of these works and distill the learned information from a classifier to generate counterfactual explanations of real images, a step to interpret the target classifier.

5.3 Methodology

This section explains the proposed methodology for generating counterfactuals using T2I generative models. In subsection 5.3.1, we briefly introduce some useful

preliminary concepts of diffusion models. Then we describe our proposed method in a three-step procedure. First, we explain how to transfer what the classifier has learned into the generative model as a set of new text tokens (subsection 5.3.2). Second, using recent advances in DDIM Inversion, we revert the image to its noise representation using the original prediction of the classifier. Finally, we denoise the noisy latent instance using the target label (subsection 5.3.3).

5.3.1 DDPM Preliminaries

Diffusion models [64] are generative architectures that create images by iteratively *removing* noise. DDPMs are based on two inverse Markov chains. The forward chain *adds* noise, while the reverse chain *removes* it. Thus, the generation process is reverse denoising, starting from a random Gaussian variable and removing small amounts of noise until a plausible image is returned.

Formally, given a diffusion model ϵ_θ and a fixed set of steps T, ϵ_θ takes as input a noisy image x_t, the current step t to compute a residual shift, and a textual conditioning C, in our case. For the generation, ϵ_θ updates x_t following:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t, C)\right) + \sigma_t \epsilon, \tag{5.1}$$

where σ_t, α_t and $\bar{\alpha}_t$ are some predefined constants, and ϵ and x_T are extracted from a Gaussian distribution. This process is repeated until $t = 0$. To train a DDPM, for a given an image-text pair (x, C), each optimization step minimizes the loss:

$$L(x, \epsilon, t, C) = \|\epsilon - \epsilon_\theta(x_t(x, t, \epsilon),\ t, C)\|^2, \tag{5.2}$$

with

$$x_t(x, t, \epsilon) = \sqrt{\bar{\alpha}_t}\, x + \sqrt{1-\bar{\alpha}_t}\, \epsilon. \tag{5.3}$$

The pioneering work of Ho, Jain, and Abbeel [64] focused on training and evaluating these models in the pixel space, making them computationally heavy. Latent Diffusion Models [151] proposed to reduce this burden by performing the diffusion process in the latent space of a Quantized Autoencoder [43]. Further, they augment the generation by using textual conditioning C at its core to steer the diffusion process, as well as increasing the quality of the generation using Classifier-Free Guidance [65] (CFG).

The CFG [65]'s core modifies the sampling strategy in Equation 5.1 by replacing ϵ_θ with ϵ^f_θ, a shifted version defined as follows:

$$\epsilon^f_\theta(x_t, t, C) := (1 + w)\, \epsilon_\theta(x_t, t, C) - w\, \epsilon_\theta(x_t, t, \varnothing), \tag{5.4}$$

where $\varnothing$ is the empty conditioning and w is a weighting constant, resulting in a qualitative improvement.

5.3.2 Distilling Knowledge into Stable Diffusion

To use large generative models, and in particular Stable Diffusion [151], we chose to distill the learned biases of the target classifier into the generative model to avoid any gradient-based optimization during the CE formation.

A model is subject to several biases as it learns, of which we distinguish two. The first is a *context bias*. This bias refers to the way images are formed. For example, ImageNet images [35] tend to have the object (*e.g.*, animals, cars, bridges) in the center, while CelebA HQ images [99] are human faces. The second bias is class-specific, and it relates to the semantic cues extracted by the classifier to make its decision, *e.g.* white and black stripes for a zebra.

So, we take a textual inversion approach to distill the context bias and the knowledge of the target classifier into the textual embedding space of Stable Diffusion. In a nutshell, textual inversion [49] links a new text-code c^* and an object (or style) such that when this new code is used, the generative model will generate this new concept. To achieve this, Gal *et al.* [49] proposed to instantiate a new text embedding e^*, associate it to the new text-code c^*, and then train e^* by minimizing the loss

$$\mathbb{E}_{(x,C)\sim D, t\sim U[1,T], \epsilon\sim\mathcal{N}(0,I)} \left[L(x, t, \epsilon, C) \right]. \tag{5.5}$$

Here, D is the set of images containing the concept to be learned, U is the uniform distribution of natural numbers between 1 and T, and C is a text prompt containing the new text code c^*.

Accordingly, to distill the context bias into Stable Diffusion, we follow [49] practices and learn a new textual embedding $e^*_{context}$ minimizing Equation 5.5 using as the conditioning the phrase `A` $c^*_{context}$ `picture`. Here, $c^*_{context}$ is the textual code related to textual embedding $e^*_{context}$. In our setup, we used the complete training set of images with no labels where the model was trained.

So far, we have not been required to use the classifier. To transfer the knowledge learned by the classifier to the T2I generation pipeline, we follow a similar approach. In this case, we train a new textual embedding e^*_i for each class i and represent its text token with c^*_i. However, instead of using the full training dataset D, we used only those images that the classifier predicted to be the source class i. As for the conditioning sentence, we take the previously learned context token and add the new class token to the sentence. Thus, we optimize Equation 5.5 with the new phrase `A` $c^*_{context}$ `image with a` c^*_i and the filtered dataset. For the rest of the text, we will refer to this prompt as C_i.

5.3.3 Counterfactual Explanations Generation

Now we want to use the learned embeddings to generate explanations. Current research on diffusion models has attempted to recover input images by retrieving the best noise, such that when the DDIM sampling strategy is used, it generates the initial instance. This is advantageous for our goal, since we can use current technological advances to generate this optimal latent noise and then inpaint the changes necessary to flip the classifier.

Since we need to perform perfect recovery to avoid most changes in the input image, we use EDICT [183]'s perfect inversion technique. In fact, they showed that inverting an image with a caption (Equation 5.8) and then denoising it (Equation 5.7) with a modified version of the original caption will produce semantic changes in the image. In short, EDICT modifies the DDIM [168] sampling strategy for diffusion models into a two-flow invertible sequence. By introducing a new hyperparameter $0 < p < 1$, setting x_0 and y_0 as the target image, and new variables:

$$\begin{aligned} a_t &= \sqrt{\bar{\alpha}_{t-1}/\bar{\alpha}_t} \\ b_t &= \sqrt{1-\bar{\alpha}_{t-1}} - \sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)/\bar{\alpha}_t}, \end{aligned} \tag{5.6}$$

the denoising phase becomes:

$$\begin{aligned} x_t^{inter} &= a_t\, x_t + b_t\, \epsilon_\theta^f(y_t, t, C) \\ y_t^{inter} &= a_t\, y_t + b_t\, \epsilon_\theta^f(x_t^{inter}, t, C) \\ x_{t-1} &= p\, x_t^{inter} + (1-p)\, y_t^{inter} \\ y_{t-1} &= p\, y_t^{inter} + (1-p)\, x_{t-1}. \end{aligned} \tag{5.7}$$

In a similar vein, the inversion phase is the inverse of Equation 5.7:

$$\begin{aligned} y_{t+1}^{inter} &= (y_t - (1-p)\, x_t)\ /\ p \\ x_{t+1}^{inter} &= (x_t - (1-p)\, y_{t+1}^{inter})\ /\ p \\ y_{t+1} &= \frac{1}{a_{t+1}}(y_{t+1}^{inter} - b_{t+1}\, \epsilon_\theta^f(x_{t+1}^{inter}, t+1, C)) \\ x_{t+1} &= \frac{1}{a_{t+1}}(x_{t+1}^{inter} - b_{t+1}\, \epsilon_\theta^f(y_{t+1}^{inter}, t+1, C)). \end{aligned} \tag{5.8}$$

We can see a clear connection between Wallace *et al.* [183]'s work and our main objective. If we invert an image using the caption with our context and source class tokens and then denoise it by changing the prompt to include the target token (learned in subsection 5.3.2), we can hope to generate the necessary changes to flip the classifier's decision.

However, while adapting the EDICT method, we noticed a major problem with this approach. Although the chosen algorithm recovers the input instance, many

images were difficult to modify. To circumvent this issue, we had to adjust the scores of the CFG in Equation 5.4. As diffusion models are seen as score-matching models, the term

$$w(\epsilon_\theta(x_t, t, C_i) - \epsilon_\theta(x_t, t, \varnothing)) \tag{5.9}$$

in Equation 5.4 are gradients pointing to the target distribution conditioned on C_i. We call this the positive drift. Thus, by including a negative drift term,

$$-w(\epsilon_\theta(x_t, t, C_j) - \epsilon_\theta(x_t, t, \varnothing)), \tag{5.10}$$

we can lead the generation process *away* from the source distribution conditioned in C_j. Therefore, we reformulate the CFG scores ϵ_θ^f, and rename it to ϵ_θ^c, as follows:

$$\begin{aligned}\epsilon_\theta^c(x_t, t, C_i, C_j) = (1 + w)\, \epsilon_\theta(x_t, t, C_i) \\ - w\, \epsilon_\theta(x_t, t, C_j).\end{aligned} \tag{5.11}$$

As a result, and given the previously introduced notions, we propose **T**ext-to-**I**mage **M**odels for counterfactual **E**xplanations (TIME), illustrated in Figure 5.2. To leverage these big generative models, we first distill the context bias into the pipeline's text embedding space by training a text embedding with the complete dataset. Then, we transfer the knowledge of the classifier by training a new embedding but using solely the instances with the same predictions. Finally, given an input image classified as i and the target j, we invert the image (Equation 5.8) using ϵ_θ^c as the score network (Equation 5.11) using as the positive and negative drift C_i and C_j, respectively. Then, we denoise the noisy state using Equation 5.7 but switching textual conditionings.

Practical considerations. To avoid large changes in the image, the inversion stops at an intermediate step τ instead of T. In addition, we have found that using more than a single embedding for the context and class biases yield further expressiveness. Also, if we fail to find a valid counterfactual, we choose a new τ and w to rerun the algorithm. We will give the implementation details later in section 5.4.

5.4 Experimental Validation

Datasets and Models. We evaluate our counterfactual method in the popular dataset CelebAHQ [99]. The task at hand is classifying smile and age attributes from face instances, computed with a DenseNet121 [69] with an image resolution of 256×256 as in [74, 78]. The evaluation is performed on the test set. To make the assessment

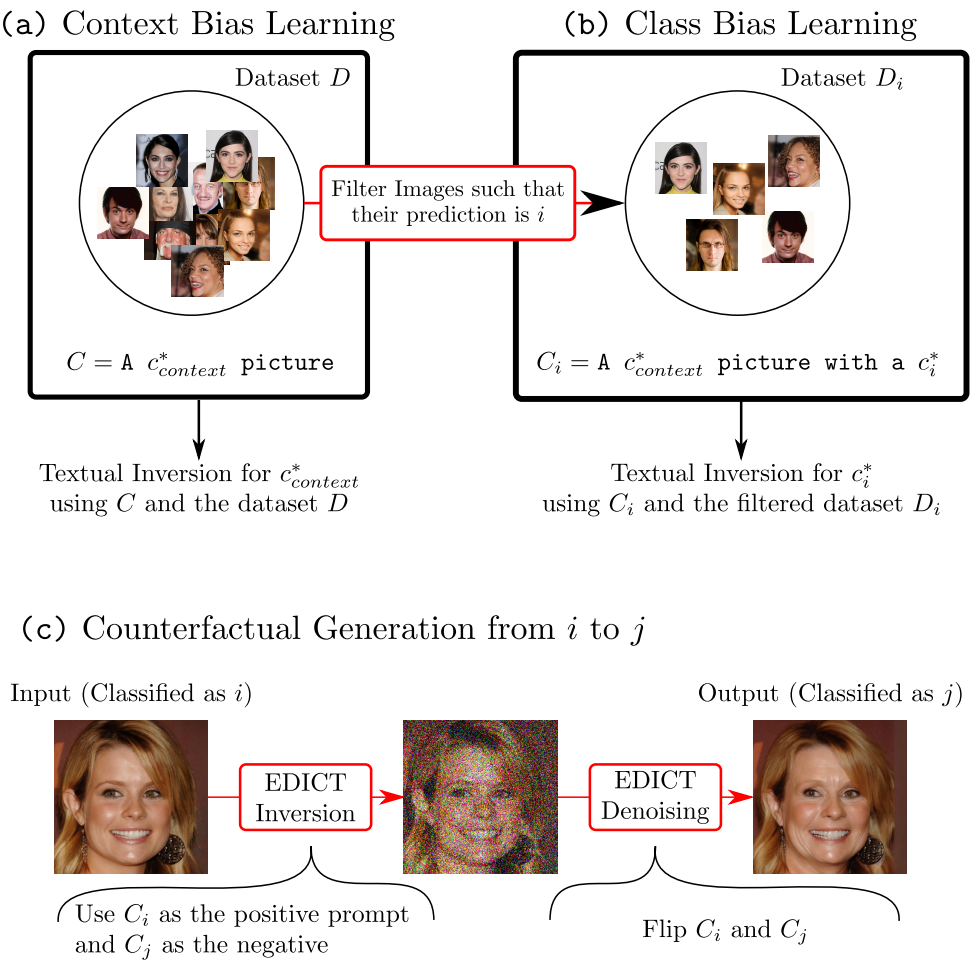


FIGURE 5.2: **TIME Overview.** Our proposed method consists of three steps: (a) We learn a context token for the whole dataset using textual inversion. (b) We filter out the images that the classifier predicts as source class i and learn a new embedding. (c) Finally, to generate the counterfactual explanation, we invert the input image using a prompt containing the source embedding and then denoise it using the target embedding.

fair with previous methods, we used the publicly available classifiers for CelebA HQ dataset from previous studies [74].

Implementation Details. We based our approach on Stable Diffusion V1.4 [43]. For all dataset, we trained three textual embeddings for the context and class biases for 800 iterations with a learning rate of 0.01, a weight decay of 1e-4, and a batch size of 64. For the inference, we used the default EDICT's hyperparameter $p = 0.93$ and a total of 50 steps. For the smiling attribute, we begin the CE generation with $(\tau, w) = (25, 3)$. In case of failure, we increased the tuple to $(30, 4)$, $(35, 4)$ or $(35, 6)$. For the age attribute, we used $(\tau, w) \in \{(30, 4), (30, 6), (35, 4), (35, 6)\}$. We performed all training and inference in a Nvidia GTX 1080.

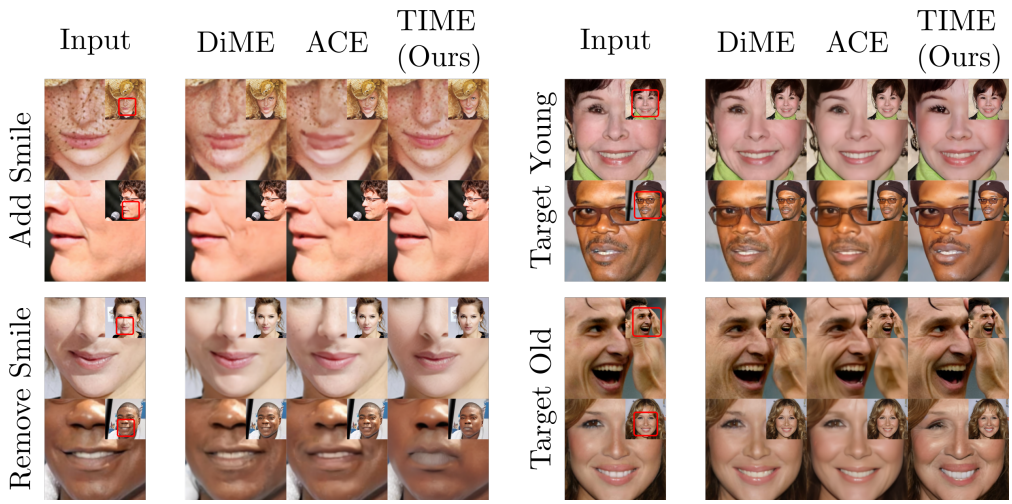


FIGURE 5.3: **Qualitative Results.** We present qualitative examples and compare them to the previous state of the art. DiME generates some out-of-distribution noise, while ACE creates blurry image sections. In contrast, TIME produces more realistic changes by harnessing the generative power of the T2I model.

5.4.1 Quantitative Assessment

Assessing counterfactuals presents inherent challenges. Despite this, several metrics approximate the core objectives of counterfactual analysis. We will now provide a concise overview of each objective and its frequent evaluation protocol, reserving an in-depth exploration of these metrics for Appendix B.

Validity. First, we need to quantify the ability of the counterfactual explanation method to flip the classifier. This is measured by the Success Ratio (SR aka Flip Rate).

Sparsity and Proximity. A counterfactual must have sparse and proximal editions. Several metrics have been proposed to evaluate this aspect, depending on the data type. For face images[150, 78, 167, 79], there are the face verification accuracy (FVA), face similarity (FS), mean number of attributes changed (MNAC), and Correlation Difference (CD). For general-purpose images, like BDD100k [197], the quantitative assessment is done via the SimSiam Similarity (S^3) [78] and the COUT metric [85].

Realism. The CE research adapts its evaluation metrics from the generation field. Hence, the realism of CEs is commonly measured with the FID [63] and sFID [78] metrics but only in the correctly classified images.

Efficiency. An efficiency analysis is often omitted by many methods. A crucial criterion for counterfactual generation techniques is to minimize computation time for generating explanations in "real time". We evaluate this by contrasting efficiency

Smile								
Method	FID (↓)	sFID (↓)	FVA (↑)	FS (↑)	MNAC (↓)	CD (↓)	COUT (↑)	SR (↑)
DiVE [150]	107.0	-	35.7	-	7.41	-	-	-
STEEX [74]	21.9	-	97.6	-	5.27	-	-	-
DiME [79]	18.1	27.7	96.7	0.6729	2.63	1.82	0.6495	97.0
ACE* ℓ_1 [78]	26.1	36.8	99.9	0.8020	2.33	2.49	0.4716	95.7
ACE ℓ_1 [78]	3.21	20.2	100.0	0.8941	1.56	2.61	0.5496	95.0
ACE* ℓ_2 [78]	26.0	35.2	99.9	0.8010	2.39	2.40	0.5048	97.9
ACE ℓ_2 [78]	6.93	22.0	100.0	0.8440	1.87	2.21	0.5946	95.0
TIME (Ours)	10.98	23.8	96.6	0.7896	2.97	2.32	0.6303	97.1
Age								
Method	FID (↓)	sFID (↓)	FVA (↑)	FS (↑)	MNAC (↓)	CD (↓)	COUT (↑)	SR (↑)
DiVE [150]	107.5	-	32.3	-	6.76	-	-	-
STEEX [74]	26.8	-	96.0	-	5.63	-	-	-
DiME [79]	18.7	27.8	95.0	0.6597	2.10	4.29	0.5615	97.0
ACE* ℓ_1 [78]	24.6	38.0	99.6	0.7680	1.95	4.61	0.4550	98.7
ACE ℓ_1 [78]	5.31	21.7	99.6	0.8085	1.53	5.4	0.3984	95.0
ACE* ℓ_2 [78]	24.2	34.9	99.4	0.7690	2.02	4.29	0.5332	99.7
ACE ℓ_2 [78]	16.4	28.2	99.6	0.7743	1.92	4.21	0.5303	95.0
TIME (Ours)	20.9	32.9	79.3	0.6282	4.19	4.29	0.3124	89.9

TABLE 5.2: **CelebAHQ Evaluation.** While TIME does not outperform the state-of-the-art metrics, our proposed method provides competitive performance while being completely black-box, *i.e.* having access only to the input and output of the model. ACE* is [78]'s method without their post-processing method.

using floating point operations (FLOPs) per explanation - lower values signify faster inference - and by measuring the average time taken to generate an explanation, specifically within our cluster environment.

Main Results.

Table 5.2 shows the results of TIME and compares them to the previous literature. Although we do not outperform the state-of-the-art in any metric, we found that our results are similar even when our proposed method is restricted to be black-box. Further, it does not require training of a completely new generative model and does not rely on any optimization for CE generation. For the realism metric, we expected to get a low FID [63] and sFID [78] due to the use of Stable Diffusion and beat ACE [78]. However, ACE uses an inpainting strategy to post-process their counterfactuals. This reduces this metric because they keep most of the original pixels in their output. If we remove the post-processing, the FID increases dramatically. With these results, we confirm that T2I generative models are a good tool to explain classifiers counterfactually in a black-box environment.

Qualitative Results

We show some qualitative results in Figure 5.3 and added more instances in Appendix B. First, we see that DiME [79], ACE [78], and TIME generate very realistic

Steps	GS	SR (↑)	FID (↓)	FS (↑)	CD (↓)
25	3	30.1	35.26	0.8957	2.82
	4	41.0	30.23	0.8570	2.61
	5	50.1	27.39	0.8231	2.33
30	3	62.1	23.15	0.8147	2.34
	4	74.0	22.51	0.7710	2.66
	5	80.8	23.51	0.7300	2.85
35	3	87.1	21.69	0.7227	2.63
	4	92.9	24.37	0.6731	3.03
	5	95.0	27.53	0.6306	3.54

TABLE 5.3: **Steps-Scale trade-off.** We analyze the trade-off between our hyperparameters τ and w. Our results show that increasing τ gives a strong boost in SR while impacting the other metrics and increasing the generation time. In contrast, w has a similar effect but is less potent without any effect on the generation time.

Context	SR (↑)	FID (↓)	FS (↑)	CD (↓)
Without	73.9	23.47	0.7480	2.41
With	92.9	24.37	0.6731	3.03

TABLE 5.4: **Context token ablation**. Here, we check the effect of including the context embeddings into our pipeline. The main advantage is increasing the success ratio. This result suggests that we can reduce τ to reach similar results while being more efficient - less number of EDICT iterations.

counterfactuals, and the differences are mostly in the details. However, the most notable changes are between ACE and our method. When we check the regions where ACE made the changes, they are blurred. This is due to their over-respacing to create the counterfactual. For DiME, we checked and found that some of their modifications seem out-of-distribution, for many cases. However, TIME produces realistic changes most of the time. Finally, in our opinion, TIME alterations can be spotted with more ease.

Efficiency Analysis

We continue our analysis and study the efficiency of TIME when creating the CE with respect to previous state-of-the-art methods, DiME [79] and ACE [78]. We estimated that TIME uses 98 TFLOPs and 45 seconds to create a single counterfactual, using $\tau = 35$ as the worst case scenario. In contrast, ACE took 279 TFLOPs and 62 second per CE while DiME took 1004 TFLOPs and 163 seconds.

5.4.2 Ablations

To show the effectiveness of each component, we realized thorough ablation experiments. To this end, we first show the hyperparameter exploration between the depth

of the chain of noise τ and the guidance scale w. Additionally, we will show the effect of including multiple textual tokens, the context tokens, and, finally, the effect of adding our negative drift – please refer to the practical consideration in subsection 5.3.3 for the variable τ. Unless explicitly told, we set $\tau = 35$ and $w = 4$ for all the ablations. For the dataset, we did the ablation using 1000 instances of the CelebA HQ validation dataset for the smiling attribute. As the quantitative metrics, we used the SR, the FID, the FS, and the CD.

Regarding the FID metric, please note that this metric is very sensible to the number of images. When using fewer images, the FID becomes less reliable to compare two methods, and hardly becomes intelligible if the two approaches are evaluated on different number of images. Since we use the FID to compare counterfactual on only those instances that flipped the classifier, comparing FIDs where the SR varies significantly does not give any cues.

Steps and scale trade-off. To begin with, we investigate the effect of the number inversion steps and the scale of the guidance. We jointly explore both variables to check the best trade-off, as shown in Table 5.3. At first glance, we notice that adding a higher guidance scale or more noise inversion steps produces more successful counterfactuals, assessed with the SR. Yet, it comes with a trade-off in other compartments: namely, the quality of the CE, and the amount of editions into the image. Generally, increasing τ or w reflects a decrease in the quality of the image and the increasing numbers of editions.

Learning the Context Token. Continuing with our study, we analyze the inclusion of our novel context token into our counterfactual generation pipeline. To ablate this component, we test whether using our learned context tokens has any advantage in contrast to giving a generic description. The results are in Table 5.4. As we can see, including our tokens provides the best performance gains in terms of SR. Qualitatively, the images are similar, yet, the images without context present some artifacts in some cases. Furthermore, we see that removing the context provides a boost in the CD and FS metrics. Although it seems counterintuitive to include this component, we can easily reach these values by decreasing τ or w (*e.g.* setting $\tau = 30$ and $w = 4$, check Table 5.3), and reducing the inference time.

Effect of the guidance. We further explore the inclusion of the negative drift term in Equation 5.10 and show the results in Table 5.5. From the quantitative assessment, we initially observed that using the classifier-free guidance (CFG in the Table) decreases the SR. When denoising the current stage x_t at time t, the CFG in Equation 5.4 estimates gradients of the log-likelihood conditioned on C_j, $-\nabla_{x_t} \log(p(x_t|C_j))$, [65] thus, pushing the generation *toward* the distribution of C_j. In contrast, incorporating the negative guidance (NG) helps steer the generation *away* from the distribution

Guidance	SR (↑)	FID (↓)	FS (↑)	CD (↓)
CFG	75.9	21.58	0.7749	2.34
NG	92.9	24.37	0.6731	3.03

TABLE 5.5: **Negative Guidance**. Here, we check the effect of performing the negative guidance (NG) instead of the classifier-free guidance (CFG). The main advantage is increasing the success ratio. This result suggests that we can reduce τ to reach similar results while being more efficient.

Tokens	SR (↑)	FID (↓)	FS (↑)	CD (↓)
Single	88.1	22.02	0.7177	3.02
Multiple	92.9	24.37	0.6731	3.03

TABLE 5.6: **Multiple-tokens Ablation.** We test if using multiple tokens in our pipeline provides any advantage. The results show an increase in SR.

conditioned on C_i. Therefore, the combined effect results in moving the instance from the boundary decision. From a qualitative perspective, we did not see major differences.Nonetheless, as noted in the context of ablation, this can be easily mitigated by reducing w and τ.

Multi-token Inclusion. Finally, we explore using multiple tokens instead of a single one for both the context and class embeddings, shown in Table 5.6. Without any surprise, we noticed that using a single token reduces the SR by a small factor. This aligns with the observations given by [49], a token catches enough information of an object or style - or in this case, inductive biases. Like in previous analyses, including multiple tokens will increase the efficiency of the model, since we can reach similar performances by tuning τ or w. Qualitatively, the most notable change between the images is sharpness.

Recommendations. Given the previous results, we propose several recommendations for the user and the model debugger, as explained in the introduction. Recall that the counterfactual explanations are used as well to recommend changes to the user to get a positive outcome. So, for the user, we recommend using the lower amount of iterations τ and guidance scale w. This results in a similarity increase and fewer edited characteristics (as evidenced by the CD and FS metrics). If the algorithm fails, it is preferable to adjust the guidance scale rather than the number of steps. For the debugger, always use the context, the negative guidance, and multiple tokens. When building the counterfactuals, follow the same recommendations for the user.

FIGURE 5.4: **BDD100k, a limit for TIME.** TIME changes the entire scene when generating the counterfactuals. Nevertheless, it still gives some insight into what the models have learned, as illustrated by the features inside the red boxes.

5.4.3 Limitations.

To test TIME in more complex scenarios, we generate CEs in the BDD100k [197] dataset using a DenseNet121 [69] trained in a *move-forward/stop* binary classification, as in [74]. We show the quantitative evaluation in Table 5.7. When generating the explanations, we noticed that TIME modifies most parts of the image, unfortunately, as shown by the S^3 metric. This is expected, as this task is challenging since it requires multiple factors to decide if to stop or to move forward. Nevertheless, we believe that these explanations still give some useful insights as a debugging tool. For example, Figure 5.4 shows that removing the red lights and adding motion blur will change the classification from *stop* to *move*, as evidenced in [78], or adding objects in front will flip the prediction to *stop*.

We believe that counterfactual methods for tasks dependent on complex scenes, where the decision is impacted by large objects or co-occurrences of several stimuli,

Method	FID (↓)	sFID (↓)	S^3 (↑)	COUT (↑)	SR (↑)
STEEX	58.8	-	-	-	99.5
DiME	7.94	11.40	0.9463	0.2435	90.5
ACE ℓ_1	1.02	6.25	0.9970	0.7451	99.9
ACE ℓ_2	1.56	6.53	0.9946	0.7875	99.9
TIME (Ours)	51.5	76.18	0.7651	0.1490	81.8

TABLE 5.7: **BDD Assessement.** We evaluate the performance of TIME on the complex BDD100k benchmark. On this dataset, there is still room for improvement for black-box counterfactual methods.

require specific architectures. In fact, we noticed that ACE [78] mainly adds some small modifications (*e.g.* changing the red lights), which is not inaccurate but is too constrained and cannot explore more insights about the learned features. Indeed, the work of Zemni *et al.* [200] focuses only on the object aspect of counterfactuals, in this case using an object-centric generator, BlobGAN [41]. This suggests that general-purpose counterfactual methods are not adapted for these tasks.

5.5 Conclusion

In this work, we present TIME, a counterfactual generation method to analyze classifiers disregarding their architecture and weights, only by looking at their inputs and outputs. By leveraging T2I generative models and a distillation approach, our method is capable of producing CEs for black-box models, a complex scenario not tackled before. Further, we show the advantages and limitations of TIME and shed light on possible future works. We believe that our approach opens the door to research focus on counterfactual methods in the challenging scenario of the black-box models.

Epilogue

As in the previous chapter, we will discuss the relationship with current methods for Counterfactual Explanations (CEs). Modern approaches for CEs are largely based on Text-to-Image (T2I) models, following two distinct lines of work: noise optimization approaches and text manipulation. Noise optimization-based approaches [46, 123] are similar to DiME in that they control the iterative generation process using guided diffusion [36]. The text manipulation line, however, shares more features with the approach proposed in this chapter. On one hand, Prabhu et al. [146] and Vendrow et al. [180] proposed creating counterfactual *examples* (rather than explanations) by using T2I to create shifts in generated [180] or real images [146], *e.g.*, adding sunglasses to a dog. Both methods rely on modifying text descriptions [146] by adding

different adjectives or learning a text embedding per class [180]. Closer to our work is DiG-IN [11]. This method optimizes different conditioning embeddings for each time step in the diffusion process, requiring gradient transfer through all diffusion steps.

Regarding the results, TIME demonstrates a clear advantage in certain aspects compared to DiME (chapter 3) and ACE (chapter 4). Although the results for our previous methods are similar in face images, we highlight the distinct performance in the BDD100k dataset. DiME and ACE both produced subtle modifications, such as adding a red light or turning off brake lights on a car. While valid, these approaches were unable to insert new objects into the scene. In contrast, TIME is capable of achieving this (please refer to Figure 5.4 or Appendix B). This is a crucial aspect as disregarding these cases would yield only partial information about what the model learned. However, TIME's flip rate is lower compared to our previous methods. Additionally, we observe limited diversity, unlike DiME. Finally, the quality of the generated instances could be enhanced, for example, by employing modern versions of stable diffusion [42].

Chapter 6

Disentangling Visual Transformers: Patch-level Interpretability for Image Classification

Prologue

In this chapter, we branch out from the current trajectory of our thesis to explore the domain of interpretable by-design architectures (ID). We have noticed that most methods for interpretable architectures focused their efforts on CNN-based models. While these backbones were once dominant, transformer architectures have since emerged as the new standard due to their superior performance. Consequently, the research community has widely adopted transformer backbones for various visual tasks. However, integrating interpretable layers into these transformer architectures remains unexplored.

So, we aim to integrate ID models with transformer backbones. Yet, the multi-headed attention mechanism (MHA), the core function of transformers, challenges this goal. In a few words, this function mixes the information within image patches, resulting in token-to-token comparisons. Consequently, when multiple MHA layers are applied, the final output wraps n-way relationships between data points, effectively capturing long-range dependencies. Accordingly, extracting visual representations from the entangled tokens challenges this process as one token represents a complex interaction in the tokenized image.

Our first track to tackle the task of ID transformers was to incorporate prototypical layers. We had two possible directions to accomplish this. First, we wanted to create a prototype bank to replace the keys and values in the original ViTs to tell what a token looks like. However, this would only work for the first MHA operation, not the later ones. This is due to the n-way data aggregation of tokens, as discussed earlier. In fact, some approaches in the literature [66, 88, 136] have created

similar mechanisms to this idea, implementing a single self-attention operation on top of convolutional features. Second, we could incorporate prototype layers, as in the work of Chen et al. [30]. This second approach has been addressed by Xue et al. [193], who proposed ProtoPFormer. ProtoPFormer bases its decision on the image tokens from the last layer, and as explained by Raghu et al. [147], when a transformer performs the classification on the image tokens, they tend to contain global information. So they have no guarantees that the information contained in a token is mostly local.

From the previous discussion, we have extracted a common weakness of both prototypical approaches: the need to recover local information. In this chapter, we attempt to resolve the entanglement of tokens in the output of the MHA operation to find the contribution of individual tokens. To this end, we proposed Hindered Transformers (HiT), a transformer-based architecture capable of disentangling the final classification token into individual image patches. As a result, the final classification can be seen as the sum of the individual contributions per patch. So, by rearranging the individual contributions of each token in its corresponding spatial location, we can create a saliency map of the decision.

6.1 Introduction

Deep learning architectures have made breakthroughs in speech, sound, and vision domains. These achievements have sparked interest in using these models in real-world applications. Therefore, understanding the decision-making process of neural networks is essential, particularly in high-stakes scenarios, to ensure that decisions are based on the correct variables and not on spurious correlations. This has led researchers to search for trustworthy architectures.

Interpretable-by-Design (ID) architectures seek to make decision-making interpretable without the need for external methods. These architectures have (ideally) similar performance to traditional opaque methods and are a good choice when it is necessary to understand their internal workings or the decisions they produce. Nonetheless, so far, these architectures use convolutional neural networks as feature extractors, which have shown weaker performance than transformer frameworks. Therefore, a natural transition is to create an interpretable architecture based on transformers.

However, the literature lacks studies on the explainability and interpretability of Vision Transformers (ViT) [40]. Even though, attention maps are considered interpretable by some authors, many works [162, 75] agree that they give little to no clues to explain the decision of the network. We agree that transformers' attention maps give partial cues about the decision-making process of ViTs, but their incomplete nature is insufficient when interpretability is paramount. We don't go further in this debate, but instead propose a new interpretable transformer-like architecture.

In this chapter, we push the knowledge frontier of ViTs by studying the flow of individual image patches in ViTs to unravel the classification token `CLS` into each individual token contribution. Consequently, our proposed architecture, dubbed Hindered Transformed (HiT), creates its prediction as the sum of the individual contributions of each token, without the need of external methods [5] or gradients [165], categorizing it as an ID method.

We summarize our contributions as follows:

- We propose the Hindered Transformer (HiT) backbone, a variant of vision transformers that is inherently interpretable.
- Empirically, we validate our architecture quantitatively and qualitatively on 4 different datasets: ImageNet [35], CUB 2011 [182], Stanford Dogs [86], and Stanford Cars [96], validating its interpretability capabilities over recent ID transformer based models.

To contribute to future research on this topic, we will release our code and weights upon publication.

6.2 Related Work

The performance of neural networks on computer vision tasks is well-established, but the complexity of these models can make them difficult to understand. This lack of transparency is problematic and has motivated the scientific community to develop methods for making neural networks more interpretable. There are two main approaches to this problem: post-hoc methods, which seek to analyze an already-trained model, and interpretable by design architectures, which aim to create models whose decision-making processes are inherently transparent.

The literature has proposed many post-hoc methods, including counterfactual explanations [10, 78, 200], saliency maps [189, 76, 161, 141], and model distillation [50, 171]. Closer to our work, a few efforts have been made to explain a ViT architecture via post-hoc algorithms. For instance, Abnar and Zuidema [1] proposed to compute a score per token by recursively propagating the attention maps in a top-bottom approach. In addition, some methods extended the LRP [14] paradigm to include the attention heads [28, 29]. The previous methods leverage the mechanisms of transformers to estimate the individual contribution of each token. We just mention a few of them, without further comment, as our aim here is to develop inherently explainable methods.

The field of interpretable by design architectures is diverse, as there is no single approach to explaining the complex behaviors neural networks. While many methods have been proposed, there has been a recent focus on prototypical part networks, such as ProtoPNets introduced by Chen et al. [30]. ProtoPNets computes

class predictions based on the distances between patches of the final feature map and some prototypes, which can be visualized. Additionally, there have been many variations of ProtoPNets proposed in the literature [184, 128, 127, 26, 19, 175, 153, 186, 129, 39, 34, 59], but, these provide just a few refinements to the original network to increase its performance and interpretability capabilities. Nonetheless, all of these works share the common characteristic: operating exclusively on convolutional neural networks (CNNs) [60, 166].

In addition to ProtoPNets, there are other methods that use alternative forms of prototypes. For example, PDiscoNet [92] automatically detects parts of objects and uses them for the final classification. Similarly, BagNet [21] mimics the Bag-of-Features approach to understand the decision-making process. Concept Bottleneck Models [93, 133] use concepts to explain their decisions. Finally, a family of networks propose interpretable layers, such as B-cos networks [16], that learns an easily interpretable input-dependent linear transformation. The work of Zhang, Wu, and Zhu [202] proposed convolutional interpretable layers. Finally, some works use some variation of decoupled networks [104, 103, 163] to highlight what a filter is looking at.

Concerning visual transformers, several recent works have attempted to make transformers more interpretable by modifying their architecture. Some papers try to push the boundaries to include transformer-based heads [66, 136, 149, 88] for prototype interpretability. However, they only include a single self-attention mechanism on top of a CNN backbone. ProtoPFormer [193] suggests including a ViT backbone in the prototype setup. To do this, ProtoPFormer includes a prototype layer on top of both the classification and image tokens. However, this architecture does not guarantee that the tokens contain purely local information, especially since computing a classification on top of image tokens increases the receptive fields of the tokens [147]. B-cos networks V2 [23] use the same rational as the original B-cos [16] approach, but they extend it to transformers. In a few words, B-cos networks summarize their inner workings as a single linear function, creating the attribution map by simply multiplying the input and the summarized network. However, their proposed extension to transformers requires convolutional layers as first layers to produce sparse explanations.

In contrast with this literature, we propose an architecture interpretable by design, by adapting the main building blocks of vision transformers to disentangle the contributions of each image patch. This enables us to compute the salient regions of the image without the need for any non-traditional training or invasive methods. Unlike other approaches, we do not rely on using the attention mechanism of the multi-headed attention (MHA) block to produce our maps. Instead, our network inherently generates them as part of its decision-making process.

6.3 Methodology

In this section, we present our proposed approach and the rationale behind it. First, we will introduce in §6.3.1 the preliminaries for the multi-headed attention mechanism and ViTs. Next, in §6.3.2, we will show that attention and multi-headed attention output can be decomposed into individual contributions of the inputs. Finally, in §6.3.3 we present our novel architecture, the Hindered Transformer (HiT). The core of our method is to minimize the mixing of patch-level information during the computation of predictions from a vision transformer. This allows us to simplify the classification token (`CLS`) in ViTs to the sum of individual tokens, a direct outcome of §6.3.2. In other words, this simplification allows us to check the contribution of each token individually.

6.3.1 Preliminaries: Transformers, ViTs and Notations

The transformer architecture is built upon the Scaled Dot-Product Attention operation [179], commonly referred to as the *attention*. Given a query token sequence $x^q \in \mathbb{R}^{L_q \times d_{model}}$ and a target sequence (or key-value sequence) $x^t \in \mathbb{R}^{L_t \times d_{model}}$, where L_q and L_t are their respective sequence lengths and d_{model} is the token dimension, the attention mechanism is computed as follows:

$$Attn(x^q, x^t) = softmax\left(\frac{(x^q W_Q + b_Q)(x^t W_K + b_k)^T}{\sqrt{d_k}}\right)(x^t W_V + b_v), \tag{6.1}$$

where the output is a sequence of the same length as x^q, d_k is the dimension of the linear transformations, and $W_i \in \mathbb{R}^{d_{model} \times d_k}$ and $b_i \in \mathbb{R}^{d_k}$ are the weights of the linear projection $i \in \{Q, K, V\}$. In addition, Vaswani et al. [179] proposed to compute the attention mechanism h times in parallel, setting $d_k = d_{model}/h$ for each individual attention operation. The resulting vectors of each individual attention, formally called heads, are concatenated and linearly post-processed to obtain the final result. This operation is called multi-headed attention, and it is described as follows:

$$MHA(x^q, x^t) = \underbrace{[Attn^1(x^q, x^t); ...; Attn^h(x^q, x^t)]}_{\texttt{Concatenate } h \texttt{ times - channel dim}} W_o + b_o, \tag{6.2}$$

with $Attn^i$ being the i^{th} attention mechanism in the MHA, and $W_o \in \mathbb{R}^{d_{model} \times d_{model}}$ and $b_o \in \mathbb{R}^{d_{model}}$ the linear transformation parameters.

In computer vision, to incorporate image data into this sequence-based formulation, the ViT first partitions the input image into N^2 equal-sized patches and linearly projects them to create the patch token sequence[1]. Additionally, following standard practice, a learnable classification token `CLS` is prepended to the patch sequence.

[1] For the rest of the chapter, we will use the terms token and patch interchangeably, referring to the image patch tokens.

Furthermore, each patch token is summed with a positional embedding to encode its spatial location within the image. For the remainder of the work, the sequence $x \in \mathbb{R}^{(N^2+1)\times d_{model}}$ denotes the concatenation of the patch tokens and the `CLS` token, where $x[0]$ corresponds to the `CLS` token.

The main ViT block builds on the MHA operation, followed by a token-wise MLP block, as in text-based transformers. Formally, given a set of patches x_l at layer l, the ViT block first computes a globalized set of tokens using the MHA block. The resulting output is summed with a skip connection. Then, the output is fed into a token-wise MLP to post-process each token, followed, again, by a skip connection. This block is summarized as follows

$$\begin{aligned} x'_l &= x_l + MHA(x_l, x_l) \\ x_{l+1} &= x'_l + MLP(x'_l) \end{aligned} \tag{6.3}$$

Note that before the MHA and MLP blocks, a LayerNorm [13] operation is applied to the data sequence, but for simplicity, we omit this operation. Finally, the `CLS` token is fed into a LayerNorm followed by a linear classifier to produce the logits of the classification task.

6.3.2 Multi-Headed Attention and Patch Mixing in Transformers

In this section, we aim to decompose the MHA operation to demonstrate that it is possible to retrieve the individual contributions of each token. In this way, we aim to lay the foundation for our architecture, which is described in the next section.

Let's start by focusing on the attention operation (Equation 6.1). Since we will focus on the `CLS` token later, and to simplify the analysis, let's assume that the query sequence has length $L_q = 1$. Consequently, the attention mechanism can be rewritten as

$$Attn(x^q, x^t) = \sum_{v \in x^t} a(v, x^q, x^t)(v\, W_V + b_v), \tag{6.4}$$

where $a(v, x^q, x^t)$ is the attention of a single token $v \in x_t$. Here, Equation 6.4 shows that we can decompose the attention mechanism into separately processed patches - each patch v in x^t adds $a(v, x^q, x^t)(v\, W_V + b_v)$. Accordingly, if x^t contains purely local information, the output of the attention is *a sum of local data*.

To continue, we incorporate the previous observation into multi-headed attention and verify that we can still unroll this operation into a *sum of separate vectors*. One might be concerned that the concatenation-linear operation will mix each token. However, we argue that the result is still valid, since concatenating and linearly transforming the resulting vector is equivalent to linearly transforming each head and adding them together. Formally, by denoting W_v^i and b_v^i as the weights of the linear transformation generating the value sequence of i^{th} head, and breaking apart W_o into h separate matrices, $W_o = [W_o^1; W_o^2; ...; W_o^h]$, with $W_o^i \in \mathbb{R}^{d_k \times d_{model}}$, then, the

MHA becomes

$$MHA(x^q, x^t) = b_o + \sum_{v \in x^t} v'(v) \text{ where } v'(v) = \sum_{i=1}^{h} a(v, x^q, x^t)(v\, W_v^i + b_v^i) W_o^i. \quad (6.5)$$

The previous result implies that we can still decompose the MHA result as the sum of vector patches, regardless of the number of heads in the MHA. So the same conclusion holds as in Eq. 6.4: if the content in x^t is local, then we can unravel the MHA mechanisms into *local contributions*.

6.3.3 Untangling Visual Transformers

Unlike single MHA layers, ViTs operate on global features. To integrate local information, these architectures use two mechanisms: the MHA layers, which spread the information within tokens; and the nonlinear MLPs, which introduce complex correlations even when applied to a linear combination of local contributions. For better explainability, it would be ideal if the decisions if the visual classifier could be expressed as a combination of information from individual patches, allowing a more interpretable understanding of how local information contributes to global predictions.

In this section, we describe our proposed architecture: Hindered Transformer (HiT). By constraining the image tokens to contain only local information along all inference blocks, and by avoiding mixing the `CLS` token, our novel method is able to partition the `CLS` token into each individual patch, a direct outcome of the previous section. Figure 6.1 shows the difference between a ViT block, and our novel block.

The first challenge is then constraining the data flow between patches. To do so, we create an intermediate architecture that uses `CLS` token $x_l[0]$ as the query in the MHA operation, and the rest of the sequence x_l as the key-value input. So, the output from the MHA is a single token that is summed to $x_l[0]$. Then, as in ViTs, we will post-process each token in the sequence with the MLP. Thus, the ViT update function in

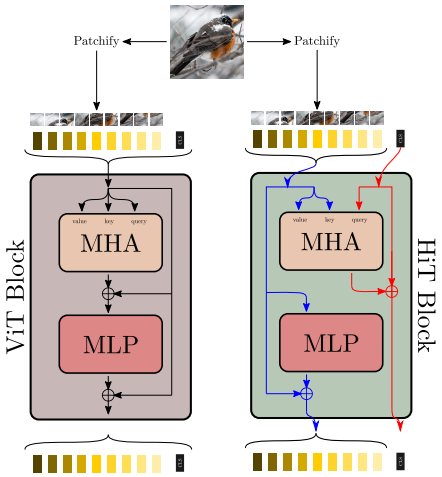


FIGURE 6.1: **ViT and HiT blocks.** While the ViT block mixes the patch data, HiT uniquely updates the `CLS` via the MHA, but avoids post-processing the classification token in the MLP, allowing the `CLS` to be unrolled at the last layer as individual patch contributions.

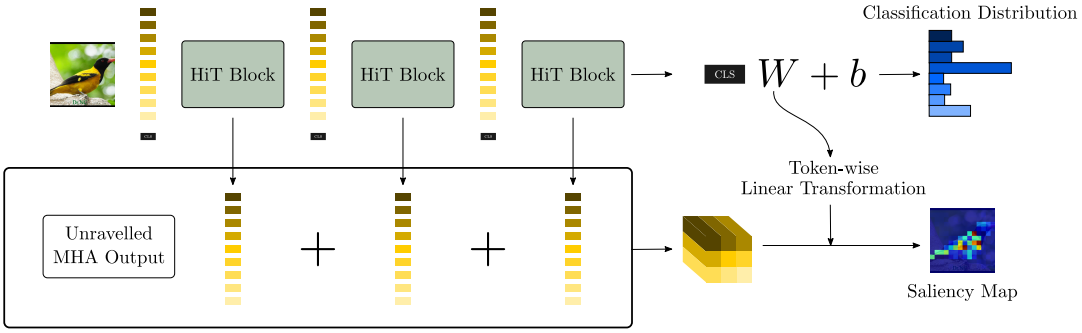


FIGURE 6.2: **Saliency Maps computation using HiT.** From the results from §6.3.2 and the definition of our architecture, HiT enables to extract the individual contribution per token and per layer. By adding together all tokens per layer, we can rearrange the tokens in a spatial layout and use the linear layer *à la* CAM [206] to extract the contribution of each token.

Equation 6.3 is transformed to

$$\begin{aligned} x'_l[0] &= x_l[0] + MHA(x_l[0], x_l) \\ x'_l[1:] &= x_l[1:] \\ x_{l+1} &= x'_l + MLP(x'_l). \end{aligned} \tag{6.6}$$

Please note that by simplifying this function, we reduce the number of operations from $\mathcal{O}(|x|^2)$ to $\mathcal{O}(|x|)$.

The previous model solves one problem by limiting the merging of data in local patches. However, processing the `CLS` token through the MLP mixes the local information provided by the MHA block, as well as the value and output operations. Since our goal is to disentangle the data flow into individual contributions, we need to further constrain this processing. To do this, we simply avoid updating the `CLS` token through the MLP and passing it to the target sequence. So, our block inference is

$$\begin{aligned} x_{l+1}[0] &= x_l[0] + MHA(x_l[0], x_l[1:]) \\ x_{l+1}[1:] &= x_l[1:] + MLP(x_l[1:]) \end{aligned} \tag{6.7}$$

We call the final architecture the Hindered Transformer (HiT). In a nutshell, HiT only updates the `CLS` token via the MHA, while the MLP blocks update the image patches. These restrictions help to preserve purely local information in each token, while allowing the `CLS` token to be unrolled.

Since the classification token is not post-processed with MLP or MHA, the final image classification is the sum of the individual tokens in all layers, as shown in

§6.3.2. Therefore, the CLS in the last layer is

$$\begin{aligned} x_L[0] &= x_0[0] + \sum_{l=0}^{L-1} MHA(x_l[0], x_l[1:]) \\ &= x_0[0] + \sum_{l=0}^{L-1} \left[b_o^l + \sum_{v \in x_l[1:]} v_l'(v) \right] \\ &= \sum_{l=0}^{L-1} \sum_{v \in x_l[1:]} \left[v_l'(v) + \frac{b_o^l}{N^2} + \frac{x_0[0]}{LN^2} \right]. \end{aligned} \tag{6.8}$$

Please note that we distribute the biases b_o^l of the projection operation in the MHA head evenly to each patch $v_l'(v)$. In a similar fashion, we spread $x_0[0]$ into all tokens for all layers.

One advantage of this architecture is that we can easily compute saliency maps, as shown in Figure 6.2. The double sum in Equation 6.8 can be viewed as a tensor $\mathrm{R}^{L \times N^2 \times d_{model}}$, where the final result is the sum over the first and second dimensions, *i.e.* the block and token dimension, respectively. So we can add all vectors in the first dimension of this tensor to get the series of local tokens. Thus and similarly to CAM [206], to compute the regions of interest used by the model for an input image, we simply run the linear classifier on each patch to get the map. This rationale is similar to the LRP [14] method in the sense that the sum of value in the saliency is equal to the output logit for that specific class.

6.4 Experiments

6.4.1 Datasets and Evaluation Metrics

We evaluated our novel architecture on four diverse image classification datasets: i) ImageNet [35]: A large-scale dataset with 1.2 million images and 1,000 classes, often used as a benchmark. ii) CUB-2011 [182]: A challenging dataset containing 200 bird classes with only 30 training samples per class on average. iii) Stanford Dogs [86]: A dataset with 120 dog classes and 10,000 training and test images. iv) Stanford Cars [96]: A dataset featuring 196 car classes with 8,100 training and validation examples.

To assess the performance of our proposed architecture, we computed the top-1 accuracy in their respective validation/test sets. To evaluate HiT's interpretability capacity, we used the insertion and deletion metrics proposed by Petsiuk, Das, and Saenko [141]. The insertion metric adds in succession the regions of the image into a perturbed version[2] from most to least influential patches. In parallel, it stores the class probability of the originally predicted label at each step, resulting in a curve per image, where the x-axis is the percentage of inserted tokens, and the y-axis the

[2]This perturbed image is a zero-filled or blurred image.

probability. Finally, it computes the area under the curve (AUC) of the mean curve for all images in the validation set. The deletion metric complements the insertion one by examining the probability decrease by adding the perturbed patches to the original image.

6.4.2 Implementation Details

To train HiT on ImageNet [35], we used the official DeiT3 codebase [173] and followed a similar setup to their method. For HiT-B and HiT-S[3], we trained our models for 600 epochs using the AdamW optimizer [109] with a learning rate of 8×10^{-4}, a weight decay of 0.05, a batch size of 4096, 20 warm-up epochs, a cosine annealing scheduler [110], and ThreeAugment [173] data augmentation. Unlike the DeiT3 training regime, we did not use the binary cross entropy loss or any LayerDrop [45] regularization, but the traditional cross entropy with a smoothing of 0.1 and an attention dropout of 0.2. To fine-tune HiT in CUB, Stanford Dogs, and Stanford Cars, we trained our models similarly to the ImageNet's configuration, but instead we set the batch size to 512, the number of epochs to 300 and the learning rate to 5×10^{-5}, 1×10^{-4}, and 4×10^{-4} for Stanford Dogs, CUB 2011, and Stanford Cars, respectively.

6.4.3 Evaluating HiT

In Table 6.1 we show the accuracy performance of our new architectures and their efficiency, measured as FLOps. The results show the expected behavior: all ViT architectures outperform our proposed model. However, this performance gap comes with a trade-off in efficiency and interpretability. On the one hand, HiT reduced the computational cost by 25% compared to ViTs, even surpassing specialized architectures for efficient inference (A-ViT). On the other hand, HiT provides an intuitive way to compute its interpretability contribution. In addition, and in our opinion, its performance is still reasonable, even in complex scenarios like ImageNet.

Next, we analyze the insertion-deletion metrics [141] of the saliency maps extracted from HiT in Table 6.2 and compare them to GradCAM [161] and an adapted rollout matrix [1], both computed on HiT. Please refer to the Appendix C to visualize the curves. These metrics verify that the salient regions are the ones used by the model to compute the predicted class. This is evidenced by the steep increase/decrease in both metrics, as only a few of the most salient tokens are needed to compute most of the predicted class probability. In contrast to state-of-the-art post-hoc methods, we observe a clear performance gap, indicating that these algorithms do not always reliably indicate the regions used for classification. With respect to the probability curves, we observed that they are concave. We conjecture that including more context in the image introduces confounding factors from other

[3]Model-wise, HiT uses the same configuration as the ViT with 16×16 patches. We just remove the last MLP block as it is not used during inference.

Model	ImageNet	CUB	Dogs	Cars	GFLOps	Params (M)
DeiT3-B	83.6	84.9	94.0	92.8	17.7	86.4
DeiT-B	81.1	84.9	93.4	93.0	17.6	86.4
B-cos (ViT-B)	74.4	-	-	-	17.6	86.9
HiT-B	71.5	76.3	80.2	84.7	13.1	81.8
DeiT3-S	81.4	83.1	90.6	93.0	4.6	22.1
DeiT-S	79.8	83.0	89.6	92.4	4.6	22.1
A-ViT-S	78.6	-	-	-	3.6	22.1
B-cos (ViT-S)	69.2	-	-	-	4.6	22.0
HiT-S	67.3	76.1	77.1	83.9	3.3	20.8

TABLE 6.1: **HiT Performance.** Our proposed models have a clear performance drop in contrast to other ViTs. Yet, the ViT gains come at the cost of efficiency and interpretability. Please note that a ViT-B and a HiT-B share the same MLP submodules. A single MLP is the most resource-heavy part of both ViTs and HiT models using 0.93 GFLOps to compute its outputs. The efficiency gains come from the MHA submodule, where standard ViTs use 0.465, ours uses 0.234 GFLOps.

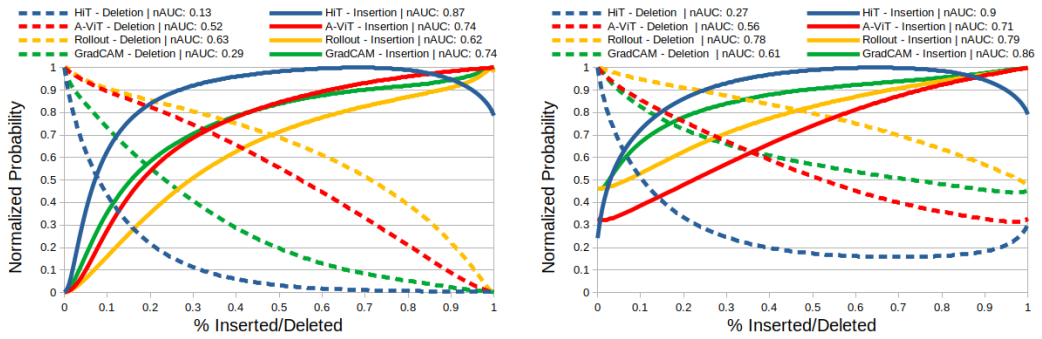
Method	ImageNet		CUB 2011		Stanford Cars		Stanford Dogs	
	Ins-Z	Del-Z	Ins-Z	Del-Z	Ins-Z	Del-Z	Ins-Z	Del-Z
HiT	**0.65**	**0.08**	**0.56**	**0.04**	**0.72**	**0.05**	**0.64**	**0.07**
HiT + Rollout	0.39	0.21	0.43	0.09	0.49	0.12	0.47	0.19
HiT + GradCAM	0.36	0.15	0.40	0.09	0.34	0.11	0.40	0.15
	Ins-B	Del-B	Ins-B	Del-B	Ins-B	Del-B	Ins-B	Del-B
HiT	**0.67**	**0.16**	**0.59**	**0.11**	**0.65**	**0.15**	**0.62**	**0.18**
HiT + Rollout	0.47	0.31	0.50	0.22	0.50	0.29	0.50	0.32
HiT + GradCAM	0.48	0.29	0.52	0.21	0.51	0.29	0.52	0.31

TABLE 6.2: **HiT and Explainability methods:** We quantitatively compare HiT maps and those created by GradCAM and the modified rollout matrix (mean attention). The assessment shows that HiT maps are in fact more faithful to those generated by GradCAM and the rollout matrix. Higher insertion is better, while lower deletion is better. Ins and Del refers to the Insertion and Deletion metrics, respectively. Z is the zero-corrupted image, while B is the blurred corruption strategy.

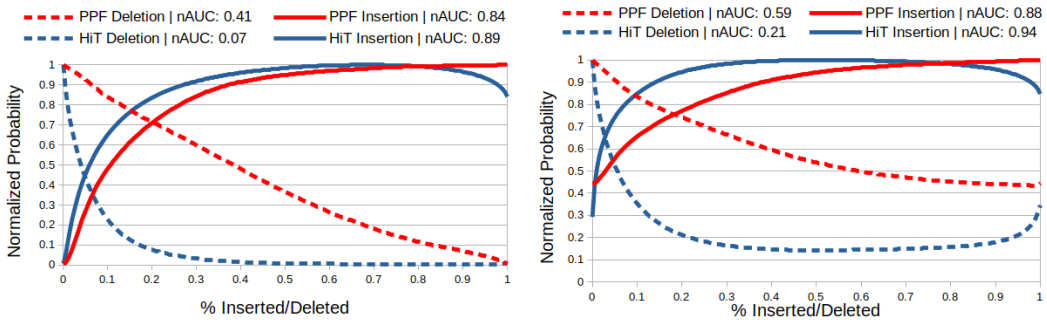
classes, thus marginally altering the class probability when inserting/deleting the entire image.

To complement our analysis and compare with other state-of-the-art ID transformer-based methods, we assessed A-ViT [195], the rollout matrix [1], and GradCAM [161] post-hoc explanation on DeiT-B on ImageNet. As for A-ViT, this method was not designed for interpretability, but it shows interesting interpretability properties. A-ViT discards some tokens at certain layers. So, to create its saliency map, we took the layer where the token was discarded, as in their paper. In addition, we evaluate ProtoPFormer [193] on CUB-2011. ProtoPFormer relies on a rollout matrix to filter out the useless tokens for its final computation. So we took their rollout matrix as their salient region for the insertion-deletion computation.

In view to compare saliency maps computed on alternate networks, which shows



(A) ImageNet



(B) CUB-2011

FIGURE 6.3: **ProtoPFormer vs A-ViT vs Rollout vs GradCAM vs HiT.** We tested whether HiT's saliency maps provide better information than ProtoPFormer's, A-ViT's maps, the rollout attention, and GradCAM. The results indicate that our methods are indeed more interpretable. Every subfigure contains two different scenarios. The left one represents the substitution with zeros case, while the right subfigure represents the blur replacement.

different probabilities values, we amend both metrics by computing the normalized AUC (nAUC), *i.e.* the AUC divided by the maximum value of the curve. Non-normalized metrics are difficult to interpret and may lead to erroneous conclusions. Still, please take into account that comparing different architectures may not reflect better interpretability capabilities since different architectures have different properties, *e.g.*, ViTs are known to be robust to occlusions [125]. We show the results of this experiment in Figure 6.3a for ImageNet and Figure 6.3b for CUB-2011. The adjusted assessment score suggests that HiT had the upper hand in terms of interpretability over other ID methods, even when their detection is significantly higher than ours.

6.4.4 Qualitative Results

Next in our study, we show some results of the saliency maps generated by HiT in Figure 6.4. We visualize both correctly and incorrectly classified images. In general, we found that our method focuses on the object, regardless of whether the prediction is accurate or not. For the Dogs dataset, we found that misclassification is generally due to similar features in the image class. For example, the first misclassified image (first column, second row) is labeled as *Pembroke*, while the model predicted its label

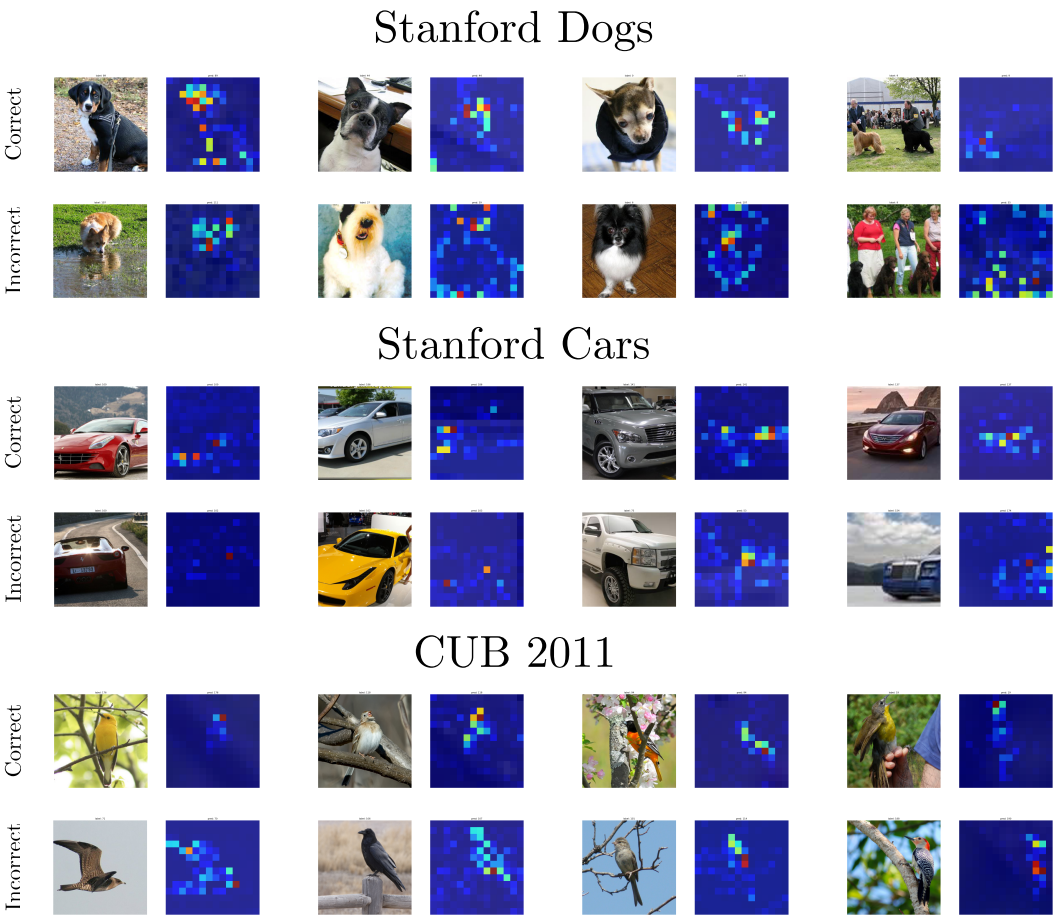
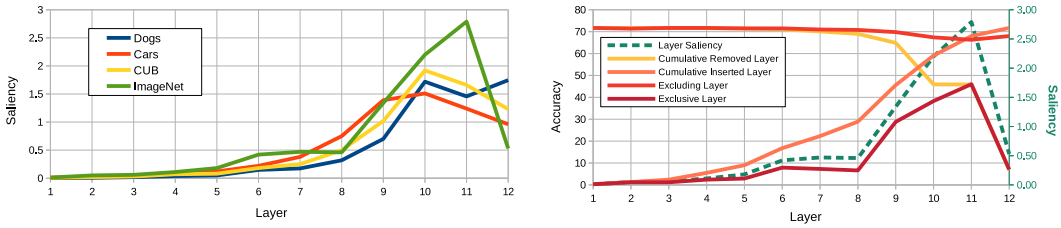


FIGURE 6.4: **Qualitative Examples.** We show the image and its saliency map produced by HiT. We noticed that HiT tends to use the object's features in the image for its prediction, independently if its prediction is erroneous or not.

as *Chow*. To our untrained eyes, the Chow has long hair, like the dog in the image. Similarly, the last misclassified image for the CUB dataset shows an image where the correct label and the incorrect prediction are both *woodpecker*. Physiologically, both birds have black wings with white spots. According to the explanations, these factors may be the largest contributors to the incorrect prediction, but further analysis is necessary.

Another advantage of HiT is that we can compute the contribution of each layer. Similar to computing the saliency maps spatially, we can create the layer-wise output tokens and look for their individual contributions. To this end, we show the results in Figure 6.5a for all tested dataset. Without any surprise, we can see that most of the discriminative features are in the final layers.

To ensure that our results are valid, we tested several ablations of our trained model on the ImageNet dataset, shown in Figure 6.5b. For instance, we tested the accuracy drop by removing or adding a layer of choice (*Excluding/Exclusive Layer* in the figure). Similarly, we check the performance loss by removing/adding layers in a cascaded manner, dubbed *cumulative removed/inserted layer*. The results corroborate



(A) Layer saliencies per dataset.

(B) Empirical validation of layer-wise saliency.

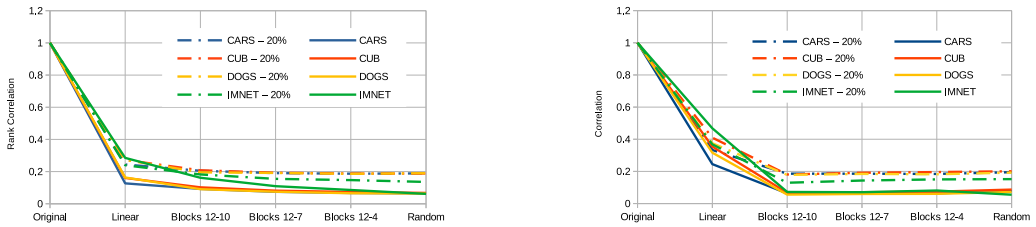
FIGURE 6.5: **Layer Saliency.** HiT has more advantages than just image saliency. (a) The first experiment shows that we can compute the contribution per layer. Without any surprise, the final layers have a greater contribution. (b) We empirically validate our findings in ImageNet with a variety of experiments. Indeed, the results show that by removing certain layers, we obtain larger expected results congruent with the layer saliency.

our previous conclusions: our architecture is capable of showing the contribution of each individual layer without relying on external methods, such as Linear Probing [5].

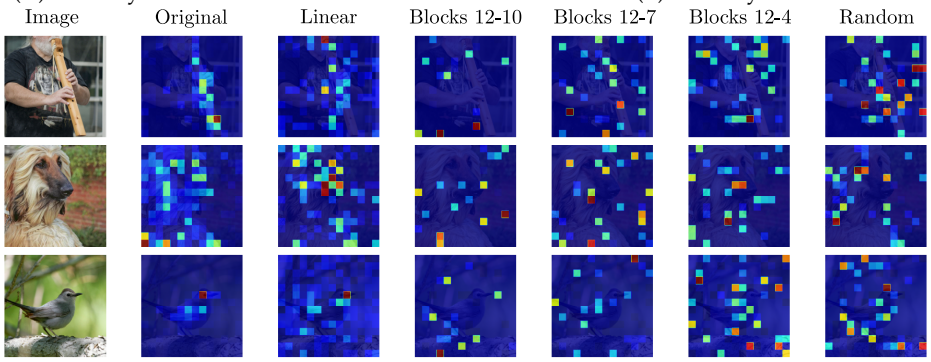
6.4.5 Sanity Check

Adebayo *et al.* [2]'s work points out that some saliency mapping methods do not show what the model is looking at. Therefore, their work proposes a sanity check to verify that a saliency method does indeed exhibit the underlying process of the network and not some alignments of the framed object. Their proposed approach consists of randomizing the layers in the network in a cascade fashion (deep to shallow layers) and examining the changes in the output saliency maps. We follow the same setup as Adebayo et al. [2], and compute the absolute rank correlation between the original and the cascaded-randomized saliency map for all four datasets in Fig. 6.6a. To complement the evaluation, we also added the absolute Pearson correlation coefficient (Figure 6.6b). Additionally, to avoid any bias due to the high amount of low salient regions, we computed the same metrics over the top 20% tokens - labeled as `DSET - 20%` in both figures.

Foremost, we noticed that the rank correlation has a steeper slope than the Pearson correlation for the linear classification layer. This shows that there is a greater similarity with the Pearson correlation. However, the values are relatively low (less than 0.5), indicating large variations. Secondly, both metrics reach a plateau for the subsequent randomized models. This low similarity suggests that the salient regions highlighted by our model are indeed what the model sees. Finally, Fig. 6.6c shows some qualitative examples produced by the randomization of all blocks, showing that, effectively, the weights' randomization reflect a large variation in the produced saliency.



(A) Saliency Rank Correlation (B) Saliency Correlation



(C) Example Images

FIGURE 6.6: **Sanity Checks on HiT.** We measure the (a) rank Correlation and (b) the Pearson correlation of randomized HiT saliency maps against the unmodified model. The results show that our saliency maps reflect the areas that the model uses for its final classification. (c) Some visual examples of the randomized saliency models.

6.4.6 Performance Loss

A key limitation of our architecture is its reduced performance on common classification benchmarks. We investigate the cause of this accuracy drop, hypothesizing that the lack of inter-token connections is the primary factor. To test this, we empirically ablated our architecture by adding pooling operations. First, we introduced 2×2 average pooling layers. This involved arranging the tokens in their spatial layout and applying the pooling operation as if they were convolutional features. Additionally, we implemented the pooling strategy used by IdentityFormer [198]: a 3×3 convolution with a stride of 2. We focus on this architecture because it shares similar characteristics with HiT, where each token contains its own information. Additionally, we theorize that optimizing HiT architectures is challenging. To address this, we adopt an approach similar to DeiT3 [173], training our model for 300 and 600 epochs. Finally, we observed that using binary cross-entropy adversely affects the model's performance, contrary to its effect on DeiT3.

We present the results in Table 6.3. The findings align with our suspicions: the lack of transferred information between patches significantly reduces the model's accuracy. For instance, by merely adding the convolutional pooling layer of IdentityFormer, we increase the accuracy of a HiT-s18[4] from 65% to 75%. However, these

[4]Please refer to [198]'s work for details about the architecture.

Architecture	Pooling	Loss	Epochs	Val ImageNet
HiT-S	None	XE	300	65.6
HiT-S	None	XE	600	67.8
HiT-S	2x2 AvgPool	XE	300	69.3
HiT-S	2x2 AvgPool	XE	600	71.4
HiT-S	None	BCE	400	59.9
HiT-S	None	BCE	800	62.6
HiT-B	None	XE	600	71.5
HiT-B	2x2 AvgPool	XE	600	75.0
HiT-s18	2x2 AvgPool	XE	300	65.6
HiT-s18	2x2 AvgPool	XE	600	69.3
HiT-s18	3x3 Conv	XE	300	75.9

TABLE 6.3: **Performance loss ablation.** HiT's performance loss stems from the limited information shared between tokens. Concurrently, the results suggest that HiT's optimization problem is more challenging, as extended training periods lead to more significant performance improvements.

convolutional layers compromise our model's interpretability by entangling information between tokens. While this argument could be applied to the average pooling operation, we contend that it does not pose a similar problem. Tokens in higher layers do not share image regions in the receptive field, whereas deeper tokens in IdentityFormer do share information. The only drawback of the average pool strategy is that the resulting final map would be coarse.

Unlike DeiT3 training schemes, our network benefits significantly from increasing the number of epochs. We believe that this result indicates HiT has not yet converged. In addition, the binary cross entropy reduces its performance. These results suggest that HiT presents novel research opportunities to investigate optimal training strategies for such models and let it for future work.

6.5 Conclusions

This chapter proposed a novel Vision Transformer (ViT) architecture: Hindered Transformer (HiT). On the one hand, this architecture enhances interpretability by decoupling contributions from individual image patches, enabling the extraction of natural saliency maps without external tools, making it an inherently interpretable-by-design architecture. On the other hand, it offers improved efficiency by reducing the computational complexity of the multi-headed attention mechanism, maintaining good performance compared to traditional ViTs. Extensive experiments across multiple datasets demonstrated the enhanced efficiency and improved interpretability benefits from HiT with a reasonable drop in overall performance. This architecture presents a promising approach to address critical challenges in ViTs, offering favorable trade-offs for applications where efficiency, interpretability, or both are essential

requirements.

Epilogue

In this chapter, we aimed to incorporate transformers into Interpretable-by-Design (ID) architectures. While HiT's decision-making process directly utilizes saliency, its interpretability features are predominantly limited to Class Activation Maps (CAM). As discussed in chapter 2, saliency maps have several limitations. The most significant drawback is that these maps indicate where the model focuses without providing more detailed insights. Ideally, we would like to enhance the model's interpretability by incorporating comprehensive visual cues. For example, we could disentangle HiT's classification token in its spatial layout and use this arrangement to include a prototype layer. However, we left this extension for future work.

HiT produces its saliency maps equivalently to the original CAMs, proposed by Zhou et al. [206]. Notwithstanding, CAMs were initially designed for architectures employing average pooling at the final layer. Following our rationale, traditional CNNs could be considered ID architectures as they generate saliency during their decision-making process. However, these architectures blend information via their 3×3 convolutions, gradually diffusing the information. Consequently, the resulting map cannot be entirely attributed to specific regions, as these architectures are incapable of disentangling the contributions of individual patches up to the shallow layers, unlike HiT.

In a similar vein, IdentityFormer [198] is a ViT-like architecture [40] without self-attention mechanisms, in addition to removing the classification token, hence, performing the final prediction over the pooled image tokens. Thus, the tokens are processed in parallel through the MLPs without influence from nearby tokens. This striking similarity to HiT raises questions about our model's inferior performance and its interpretability contribution. However, IdentityFormer uses 3×3 convolutions as pooling operations, and as in the previous discussion, this mixes nearby token data similar to CNNs. To this end, and similar to the argument on CNNs, this hampers the direct analysis of the input tokens.

Chapter 7

Conclusions and Perspective

This thesis addresses two complementary branches of the XAI community: counterfactual explanations (CE) and interpretable by-design architectures (ID). Regarding the former, we propose several methods for generating CEs using diffusion models. Concerning the latter, we design a novel Interpretable By-Design (ID) architecture based on transformers, which attempts to bridge the gap between transformers and ID models.

7.1 Counterfactual Explanations

Counterfactual explanations serve as valuable tools for shedding light on the inner workings of neural networks. These explanations employ an interventionist approach, making minimal changes to the input image to alter the model's original prediction. Beyond their main objective, these explanations should possess three key properties: realism, proximity, and diversity. By examining the various possible alterations, we can identify the features used by a model for a given prediction.

To generate CEs for vision-related tasks, researchers employ generative models to constrain the output to plausible instances. In this work, we have pushed the knowledge frontier by incorporating diffusion models [64] into the CE generation process. Our first method, DiME (chapter 3), served as a promising proof of concept for this approach. DiME demonstrated a notable advantage, even compared to contemporary methods: its ability to generate diverse explanations. However, DiME's reliance on generating a clean image to construct guiding gradients [36] results in a computationally intensive process.

Subsequently, we proposed ACE (chapter 4), a CE generation approach based on adversarial attacks [54]. This method aims to link previous studies on adversarial attacks and counterfactual explanations, applying these concepts to vision classification tasks. Like DiME, ACE utilizes diffusion models, but as a noise regularization constraint for the attack rather than for direct generation. In practice, ACE does not exhibit the same level of diversity as DiME. However, it demonstrates clear advantages in terms of realism, and proximity. Notably, this chapter presents the most significant finding of this thesis: we applied real live changes to a picture to change its

classification, thus, confirming that our counterfactuals exhibit the features learned by the classifier, thus allowing us to perform real perturbations to reverse a classifier decision.

Finally, chapter 5 introduces TIME, an approach for generating CEs that leverages foundational models for text-to-image generation. TIME steps away from the traditional optimization-based paradigm, generating explanations without requiring access to the model's internal structure. Instead, it employs textual inversion [49] to learn the concepts used by the model and perfect inversion techniques [183] to generate the explanations. While TIME's CEs are not flawless, they operate in a completely black-box environment, where only the model's input and output are accessible. Moreover, in our assessment, TIME demonstrates the capability to create more complex modifications to alter the classifier's predictions. This alludes to the discussion in chapter 2, the concept of proximity in counterfactuals is an ill-posed property.

From these works, we learned that CEs are a promising tool for explainability due to their simplicity, human-friendly nature, and interventionist approach. DiME (chapter 3) and ACE (chapter 4) were appropriate approaches for simple tasks; however, their explanations tend to remain very local, without the capacity to input new objects into the scene. This limitation restricts their ability to be employed in more complex scenarios since they cannot show all possible ways to change a model's output. Still, this does not mean that both methods will be useless in the future; rather, they are designed for specific types of modifications. In contrast, TIME (chapter 6) was able to modify images by inpainting different kinds of objects. Nonetheless, some parts of the images, like the background, were modified, raising the question of whether these are important for classification or not.

Future Work. There are still many research directions to complement this task, and we highlight two lines of work: the evaluation and the generation techniques. First, evaluating CE is still an open problem. The realism metric relies on current assessments for generative methods, and more specifically the FID. This assessment technique relies on models trained on other classes and thus may react differently to different textures. For proximity, we think that using object detection and scene decomposition techniques could be beneficial as it could count how many objects or features were added, regardless of the size or realism. Even, it might point out towards solving the proximity example Figure 2.3.

Regarding potential algorithmic research directions, we believe that using inpainting techniques as a post-processing step in ACE is a promising approach. Inpainting counterfactual objects or features in an image offers a significant advantage by not modifying the complete structure, as seen in some approaches in the literature. However, this approach introduces a new challenge: automatically selecting where the counterfactual edit will be applied, as it is typically done manually by a

user. Another possible research direction involves extending the locality of CE to analyze a group of counterfactuals, generating a global view of the variables used by the model. In line with this objective, we could extract text-based explanations or rules to enhance the model's interpretability. Essentially, this direction can be conceptualized as jointly exploring concept attribution and counterfactuals.

7.2 Transformer-based Interpretable Architectures

Interpretable-by-design architectures aim to incorporate layers that provide insights into the model's reasoning process, making them an appealing alternative. The literature has primarily focused on creating such layers for CNNs, likely because of the inherent local properties of convolutional operations. Unlike CNNs, transformers lack this inductive bias due to their core operation, the multi-headed attention (MHA) mechanism, which hampers the integration of interpretable layers. To address this challenge, in chapter 6, we propose HiT, an approach that unravels the output from the MHA. By constraining the interaction within patches and simplifying the refinement of the classification token by the MLP blocks, HiT disentangles the classification token as a sum of tokens. Thus, we can decompose the classification tokens in ViTs into patch-level contributions, thereby enabling our goal of creating interpretable transformer architectures.

Unfortunately, our proposed network has several shortcomings. HiT presents a large gap in classification accuracy in the tested datasets due to our strong restriction on patch interaction. In addition, although our model can be considered as an ID architecture, it presents a low degree of interpretability by generating saliency maps. However, we believe that this approach is still a good first approach to token-level interpretability. Although these shortcomings can be considered important, it opens a path for future research to improve these weak points.

Future Work. Apart from the aforementioned, ID architectures still have a long way to go. A main focus in research is on foundational models. Many of these models are based on transformers, such as SAM [91], highlighting the importance of continuing research on interpretable transformers. In a general context, there is still much to be done in the world of interpretability and explainability. Interpretability architectures and explainability methods should work in unison. While the interpretable architecture gives cues about its reasoning process, explainability methods should confirm its behavior. Turning our attention back to counterfactual explanations, if an interpretability method behaves as expected, the counterfactual should input the expected changes learned by the interpretable architecture to change the classification. In parallel, if we assume a correct interpretable performance from the ID architecture, this architecture could provide ground truth for *Post-Hoc* methods.

7.3 Closing Thoughts

As neural networks have become larger, more complex, and better at virtually all tasks, they have become the standard approach in computer vision. However, these highly parameterized and nonlinear architectures come at the cost of interpretability. While high performance is desirable, critical applications require an understanding of the inner mechanisms of these models to ensure fairness and correct behavior. To address this, this thesis explores the field of Explainable AI (XAI), specifically through the lens of counterfactual explanations and interpretable transformers.

The emergence of *foundational* models raises many questions about the future of XAI approaches. These new models have redefined standard paradigms of simple tasks such as classification or object detection [105]. This paradigm shift challenges standard practices in XAI techniques and questions whether they would be applicable for next-generation architectures. In parallel, foundation models are becoming multimodal, and explanations would need to cover all of these representations together. Although we could restrict the explanation to a particular data type, a complete view of the phenomena leading to a specific outcome should include all possible scenarios. However, generating explanations for all possible data types require having explanation algorithms made for each specific data type. In addition, when dealing with complex descriptions, there is a fine line between exhaustive descriptions and useful explanations. Although we don't present solutions or answers to these problems here, research first needs to focus on simple tasks. If the XAI community cannot manage to understand classifiers in simple scenarios, we cannot expect them to work for complex foundational models. Finally, we hope that future XAI research will draw inspiration from the progress we have made and continue to advance efforts to open *black box models*.

Bibliography

[1] Samira Abnar and Willem Zuidema. "Quantifying Attention Flow in Transformers". In: *Annual Meeting of the Association for Computational Linguistics*. 2020. URL: https://api.semanticscholar.org/CorpusID:218487351.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

[3] Naveed Akhtar, Mohammad Jalwana, Mohammed Bennamoun, and Ajmal S Mian. "Attack to fool and explain deep networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[4] Arjun Akula, Shuai Wang, and Song-Chun Zhu. "CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.03 (2020), pp. 2594–2601.

[5] Guillaume Alain and Yoshua Bengio. "Understanding intermediate layers using linear classifier probes". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=HJ4-rAVtl.

[6] Stephan Alaniz, Diego Marcos, Bernt Schiele, and Zeynep Akata. "Learning Decision Trees Recurrently Through Communication". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 13518–13527.

[7] David Alvarez-Melis and Tommi S. Jaakkola. "Towards Robust Interpretability with Self-Explaining Neural Networks". In: *Advances in neural information processing systems (NeurIPS)*. 2018.

[8] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. "Square Attack: a query-efficient black-box adversarial attack via random search". In: (2020).

[9] Sanjeev Arora, Andrej Risteski, and Yi Zhang. "Do GANs learn the distribution? Some Theory and Empirics". In: *International Conference on Learning Representations*. 2018.

[10] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. "Diffusion Visual Counterfactual Explanations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 364–377. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/025f7165a452e7d0b57f1397fed3b0fd-Paper-Conference.pdf.

[11] Maximilian Augustin, Yannic Neuhaus, and Matthias Hein. "DiG-IN: Diffusion Guidance for Investigating Networks - Uncovering Classifier Differences Neuron Visualisations and Visual Counterfactual Explanations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 11093–11103.

[12] Omri Avrahami, Dani Lischinski, and Ohad Fried. "Blended Diffusion for Text-Driven Editing of Natural Images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 18208–18218.

[13] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". In: *CoRR* abs/1607.06450 (2016). arXiv: 1607.06450. URL: http://arxiv.org/abs/1607.06450.

[14] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLoS ONE* 10 (2015). URL: https://api.semanticscholar.org/CorpusID:9327892.

[15] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. "Label-Efficient Semantic Segmentation with Diffusion Models". In: *International Conference on Learning Representations*. 2022.

[16] Moritz Böhle, Mario Fritz, and Bernt Schiele. "B-cos Networks: Alignment is All We Need for Interpretability". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 10319–10328. DOI: 10.1109/CVPR52688.2022.01008. URL: https://doi.org/10.1109/CVPR52688.2022.01008.

[17] Moritz Böhle, Mario Fritz, and Bernt Schiele. "B-Cos Networks: Alignment Is All We Need for Interpretability". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10329–10338.

[18] Moritz Bohle, Mario Fritz, and Bernt Schiele. "Convolutional Dynamic Alignment Networks for Interpretable Classifications". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 10029–10038.

[19] Andrea Bontempelli, Stefano Teso, Katya Tentori, Fausto Giunchiglia, and Andrea Passerini. "Concept-level Debugging of Part-Prototype Networks". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=oiwXWPDTyNk.

[20] Valentyn Boreiko, Maximilian Augustin, Francesco Croce, Philipp Berens, and Matthias Hein. "Sparse visual counterfactual explanations in image space". In: *DAGM German Conference on Pattern Recognition*. Springer. 2022, pp. 133–148.

[21] Wieland Brendel and Matthias Bethge. "Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet". In: *International Conference on Learning Representations*. 2019.

[22] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[23] Moritz Böhle, Navdeeppal Singh, Mario Fritz, and Bernt Schiele. "B-cos Alignment for Inherently Interpretable CNNs and Vision Transformers". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024), pp. 1–15. DOI: 10.1109/TPAMI.2024.3355155.

[24] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. "VGGFace2: A Dataset for Recognising Faces across Pose and Age". In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 2018, pp. 67–74. DOI: 10.1109/FG.2018.00020.

[25] Nicholas Carlini and David Wagner. "Towards evaluating the robustness of neural networks". In: *2017 ieee symposium on security and privacy (sp)*. Ieee. 2017, pp. 39–57.

[26] Zachariah Carmichael, Suhas Lohit, Anoop Cherian, Michael J Jones, and Walter J Scheirer. "Pixel-Grounded Prototypical Part Networks". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 4768–4779.

[27] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018, pp. 839–847. DOI: 10.1109/WACV.2018.00097.

[28] Hila Chefer, Shir Gur, and Lior Wolf. "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 397–406.

[29] Hila Chefer, Shir Gur, and Lior Wolf. "Transformer Interpretability Beyond Attention Visualization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 782–791.

[30] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. "This Looks Like That: Deep Learning for Interpretable Image Recognition". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[31] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15750–15758.

[32] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. "ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 14367–14376.

[33] Francesco Croce and Matthias Hein. "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks". In: *International conference on machine learning*. PMLR. 2020, pp. 2206–2216.

[34] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. "Deformable Convolutional Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[36] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[37] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. "Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[38] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. "Prompt tuning inversion for text-driven image editing using diffusion models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7430–7440.

[39] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. "Deformable ProtoPNet: An Interpretable Image Classifier Using Deformable Prototypes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10265–10275.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[41] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A. Efros. "BlobGAN: Spatially Disentangled Scene Representations". In: *European Conference on Computer Vision (ECCV)* (2022).

[42] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. "Scaling Rectified Flow Transformers for High-Resolution Image Synthesis". In: *Forty-first International Conference on Machine Learning*. 2024. URL: https://openreview.net/forum?id=FPnUhsQJ5B.

[43] Patrick Esser, Robin Rombach, and Bjorn Ommer. "Taming Transformers for High-Resolution Image Synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 12873–12883.

[44] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola Schoenlieb. "On the Connection Between Adversarial Robustness and Saliency Map Interpretability". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1823–1832. URL: https://proceedings.mlr.press/v97/etmann19a.html.

[45] Angela Fan, Edouard Grave, and Armand Joulin. "Reducing Transformer Depth on Demand with Structured Dropout". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SylO2yStDr.

[46] Karim Farid, Simon Schrodi, Max Argus, and Thomas Brox. *Latent Diffusion Counterfactual Explanations*. 2023. arXiv: 2310.06668 [cs.LG].

[47] Thomas Fel, Agustin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. "CRAFT: Concept Recursive Activation FacTorization for Explainability". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 2711–2721.

[48] François-Guillaume Fernandez. *TorchCAM: class activation explorer*. https://github.com/frgfm/torch-cam. 2020.

[49] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=NAQvF08TcyG.

[50] Yunhao Ge, Yao Xiao, Zhi Xu, Meng Zheng, Srikrishna Karanam, Terrence Chen, Laurent Itti, and Ziyan Wu. "A Peek Into the Reasoning of Neural Networks: Interpreting With Structural Visual Concepts". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2195–2204.

[51] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bygh9j09KX.

[52] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. "Towards Automatic Concept-based Explanations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[53] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014.

[54] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[55] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6572.

[56] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. "Counterfactual Visual Explanations". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2376–2384. URL: https://proceedings.mlr.press/v97/goyal19a.html.

[57] Andreas Griewank and Andrea Walther. "Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation". In: *ACM Trans. Math. Softw.* 26.1 (2000), 19–45. ISSN: 0098-3500. DOI: 10.1145/347837.347846. URL: https://doi.org/10.1145/347837.347846.

[58] Jing Gu, Yilin Wang, Nanxuan Zhao, Tsu-Jui Fu, Wei Xiong, Qing Liu, Zhifei Zhang, HE Zhang, Jianming Zhang, HyunJoon Jung, and Xin Eric Wang. "PHOTOSWAP: Personalized Subject Swapping in Images". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=qqcIM8NiiB.

[59] Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. "Interpretable image recognition with hierarchical prototypes". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 2019, pp. 32–40.

[60] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[61] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. "Generating Visual Explanations". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max

Welling. Springer International Publishing, 2016, pp. 3–19. ISBN: 978-3-319-46493-0.

[62] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. "Prompt-to-Prompt Image Editing with Cross-Attention Control". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=_CDixzkzeyb.

[63] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[64] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[65] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021. URL: https://openreview.net/forum?id=qw8AKxfYbI.

[66] Jinyung Hong, Keun Hee Park, and Theodore P Pavlic. "Concept-Centric Transformers: Enhancing Model Interpretability Through Object-Centric Concept Learning Within a Shared Global Workspace". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 4880–4891.

[67] Chin-Wei Huang, Jae Hyun Lim, and Aaron Courville. "A Variational Perspective on Diffusion-Based Generative Models and Score Matching". In: *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*. 2021.

[68] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[69] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[70] Zixuan Huang and Yin Li. "Interpretable and Accurate Fine-grained Recognition via Region Grouping". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[71] Frederik Hvilshøj, Alexandros Iosifidis, and Ira Assent. "ECINN: efficient counterfactuals from invertible neural networks". In: *British Machine Vision Conference 2018, BMVC 2021* (2021).

[72] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. "On Relating Explanations and Adversarial Examples". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[73] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. "Adversarial examples are not bugs, they are features". In: *Advances in neural information processing systems* 32 (2019).

[74] Paul Jacob, Éloi Zablocki, Hedi Ben-Younes, Mickaël Chen, Patrick Pérez, and Matthieu Cord. "STEEX: Steering Counterfactual Explanations with Semantics". In: *ECCV*. 2022.

[75] Sarthak Jain and Byron C. Wallace. "Attention is not Explanation". In: *North American Chapter of the Association for Computational Linguistics*. 2019. URL: https://api.semanticscholar.org/CorpusID:67855860.

[76] Mohammad A. A. K. Jalwana, Naveed Akhtar, Mohammed Bennamoun, and Ajmal Mian. "CAMERAS: Enhanced Resolution and Sanity Preserving Class Activation Mapping for Image Saliency". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 16327–16336.

[77] Guillaume Jeanneret, Juan C. Pérez, and Pablo Arbeláez. "A Hierarchical Assessment of Adversarial Severity". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2021, pp. 61–70.

[78] Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Adversarial Counterfactual Visual Explanations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 16425–16435.

[79] Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. "Diffusion Models for Counterfactual Explanations". In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 2022.

[80] Shalmali Joshi, Oluwasanmi Koyejo, Been Kim, and Joydeep Ghosh. "xGEMs: Generating Examplars to Explain Black-Box Models". In: *ArXiv* abs/1806.08867 (2018).

[81] Hyungsik Jung and Youngrock Oh. "Towards Better Explanations of Class Activation Mapping". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 1336–1344.

[82] Steffen Jung and Margret Keuper. "Internalized biases in fréchet inception distance". In: *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*. 2021.

[83] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Training generative adversarial networks with limited data". In: *Advances in neural information processing systems* 33 (2020), pp. 12104–12114.

[84] Eoin M. Kenny and Mark T Keane. "On Generating Plausible Counterfactual and Semi-Factual Explanations for Deep Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.13 (2021), pp. 11575–11585.

[85] Saeed Khorram and Li Fuxin. "Cycle-Consistent Counterfactuals by Latent Transformations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10203–10212.

[86] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. "Novel Dataset for Fine-Grained Image Categorization". In: *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO, 2011.

[87] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2668–2677.

[88] Sangwon Kim, Jaeyeal Nam, and Byoung Chul Ko. "ViT-NeT: Interpretable Vision Transformers with Neural Tree Decoder". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 11162–11172. URL: https://proceedings.mlr.press/v162/kim22g.html.

[89] Sunnie S. Y. Kim, Nicole Meister, Vikram V. Ramaswamy, Ruth Fong, and Olga Russakovsky. "HIVE: Evaluating the Human Interpretability of Visual Explanations". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022.

[90] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014.

[91] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. "Segment Anything". In: *arXiv:2304.02643* (2023).

[92] Robert van der Klis, Stephan Alaniz, Massimiliano Mancini, Cassio F Dantas, Dino Ienco, Zeynep Akata, and Diego Marcos. "PDiscoNet: Semantically consistent part discovery for fine-grained recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 1866–1876.

[93] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. "Concept Bottleneck Models". In: 2020.

[94] Stefan Kolek, Duc Anh Nguyen, Ron Levie, Joan Bruna, and Gitta Kutyniok. "Cartoon explanations of image classifiers". In: (2022).

[95] Zhifeng Kong and Wei Ping. "On Fast Sampling of Diffusion Probabilistic Models". In: *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*. 2021.

[96] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. "3d object representations for fine-grained categorization". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2013, pp. 554–561.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012.

[98] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Improved Precision and Recall Metric for Assessing Generative Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/0234c510bc6d908b28c70ff313743079-Paper.pdf.

[99] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[100] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[101] Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang. "Relevance-CAM: Your Model Already Knows Where To Look". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 14944–14953.

[102] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: *International conference on machine learning*. PMLR. 2023, pp. 19730–19742.

[103] Yuchao Li, Rongrong Ji, Shaohui Lin, Baochang Zhang, Chenqian Yan, Yongjian Wu, Feiyue Huang, and Ling Shao. "Interpretable Neural Network Decoupling". In: *European Conference on Computer Vision*. 2019. URL: https://api.semanticscholar.org/CorpusID:221297616.

[104] Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. "Training Interpretable Convolutional Neural Networks by Differentiating Class-specific Filters". In: *European Conference on Computer Vision*. Springer. 2020, pp. 622–638.

[105] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual Instruction Tuning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=w0H2xGHlkw.

[106] Shusen Liu, Bhavya Kailkhura, Donald Loveland, and Yong Han. "Generative Counterfactual Introspection for Explainable Deep Learning". In: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2019, pp. 1–5.

[107] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[108] Arnaud Van Looveren and Janis Klaise. "Interpretable counterfactual explanations guided by prototypes". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2021, pp. 650–665.

[109] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[110] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Skq89Scxx.

[111] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. "RePaint: Inpainting Using Denoising Diffusion Probabilistic Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11461–11471.

[112] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[113] Jinqi Luo, Zhaoning Wang, Chen Henry Wu, Dong Huang, and Fernando De la Torre. "Zero-shot Model Diagnosis". In: *CVPR*. 2023.

[114] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018.

[115] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2022.

[116] Silvan Mertes, Tobias Huber, Katharina Weitz, Alexander Heimerl, and Elisabeth André. "GANterfactual—Counterfactual Explanations for Medical Non-experts Using Generative Adversarial Learning". In: *Frontiers in artificial intelligence* 5 (2022).

[117] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (2019), pp. 1–38. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2018.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0004370218305988.

[118] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. "Null-text inversion for editing real images using guided diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 6038–6047.

[119] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[120] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. "Universal adversarial perturbations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1765–1773.

[121] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582.

[122] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. "Explaining machine learning classifiers through diverse counterfactual explanations". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (2020).

[123] Franz Motzkus, Christian Hellert, and Ute Schmid. "CoLa-DCE – Concept-guided Latent Diffusion Counterfactual Explanations". In: 2024. URL: https://api.semanticscholar.org/CorpusID:270226253.

[124] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models". In: *arXiv preprint arXiv:2302.08453* (2023).

[125] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. "Intriguing properties of vision transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23296–23308.

[126] Meike Nauta, Ron van Bree, and Christin Seifert. "Neural Prototype Trees for Interpretable Fine-Grained Image Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 14933–14943.

[127] Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. "This looks like that, because... explaining prototypes for interpretable image recognition". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2021, pp. 441–456.

[128] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. "PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification". In: (2023).

[129] Meike Nauta, Ron Van Bree, and Christin Seifert. "Neural prototype trees for interpretable fine-grained image recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14933–14943.

[130] Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta. "CounteRGAN: Generating counterfactuals for real-time recourse and interpretability using residual GANs". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by James Cussens and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, 2022, pp. 1488–1497. URL: https://proceedings.mlr.press/v180/nemirovsky22a.html.

[131] Alexander Quinn Nichol and Prafulla Dhariwal. "Improved Denoising Diffusion Probabilistic Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8162–8171.

[132] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. "Diffusion Models for Adversarial Purification". In: *International Conference on Machine Learning (ICML)*. 2022.

[133] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. "Label-free Concept Bottleneck Models". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=FlCg47MNvBA.

[134] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. "Multimodal Explanations: Justifying Decisions and Pointing to the Evidence". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[135] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. "Zero-shot image-to-image translation". In: *ACM SIGGRAPH 2023 Conference Proceedings*. 2023, pp. 1–11.

[136] DIPANJYOTI PAUL, Arpita Chowdhury, Xinqi Xiong, Feng-Ju Chang, David Edward Carlyn, Samuel Stevens, Kaiya L Provost, Anuj Karpatne, Bryan Carstens, Daniel Rubenstein, Charles Stewart, Tanya Berger-Wolf, Yu Su, and Wei-Lun Chao. "A Simple Interpretable Transformer for Fine-Grained Image Classification and Analysis". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=bkdWThqE6q.

[137] Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. "Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 4574–4594.

[138] Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. "Exploring Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis". In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, 2022, pp. 4574–4594. URL: https://proceedings.mlr.press/v151/pawelczyk22a.html.

[139] William Peebles and Saining Xie. "Scalable Diffusion Models with Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 4195–4205.

[140] Juan C. Pérez, Motasem Alfarra, Guillaume Jeanneret, Laura Rueda, Ali Thabet, Bernard Ghanem, and Pablo Arbeláez. "Enhancing Adversarial Robustness via Test-Time Transformation Ensembling". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2021, pp. 81–91.

[141] Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 151.

[142] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I. Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. "Black-Box Explanation of Object Detectors via Saliency Maps". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 11443–11452.

[143] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. "Generative adversarial perturbations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4422–4431.

[144] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodríguez, Tijl De Bie, and Peter A. Flach. "FACE: Feasible and Actionable Counterfactual Explanations". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (2020).

[145] Viraj Prabhu, Sriram Yenamandra, Prithvijit Chattopadhyay, and Judy Hoffman. "Bridging the Sim2Real gap with CARE: Supervised Detection Adaptation with Conditional Alignment and Reweighting". In: (2023). arXiv: 2302.04832 [cs.CV].

[146] Viraj Uday Prabhu, Sriram Yenamandra, Prithvijit Chattopadhyay, and Judy Hoffman. "LANCE: Stress-testing Visual Models by Generating Language-guided Counterfactual Images". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=BbIxB4xnbq.

[147] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. "Do Vision Transformers See Like Convolutional Neural Networks?" In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021. URL: https://openreview.net/forum?id=Gl8FHfMVTZu.

[148] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778.

[149] Mattia Rigotti, Christoph Miksovic, Ioana Giurgiu, Thomas Gschwind, and Paolo Scotton. "Attention-based Interpretability with Concept Transformers". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=kAa9eDS0RdO.

[150] Pau Rodríguez, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam Laradji, Laurent Charlin, and David Vazquez. "Beyond Trivial Counterfactual Explanations With Diverse Valuable Explanations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 1056–1065.

[151] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10684–10695.

[152] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 22500–22510.

[153] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. "Interpretable image classification with differentiable prototypes assignment". In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2022).

[154] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. "Palette: Image-to-Image Diffusion Models". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021.

[155] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David Fleet, and Mohammad Norouzi. "Image Super-Resolution via Iterative Refinement". In: *ArXiv* abs/2104.07636 (2021).

[156] Pedro Sanchez and Sotirios A. Tsaftaris. "Diffusion Causal Models for Counterfactual Estimation". In: *Proceedings of the First Conference on Causal Learning and Reasoning*. Ed. by Bernhard Schölkopf, Caroline Uhler, and Kun Zhang. Vol. 177. Proceedings of Machine Learning Research. PMLR, 2022, pp. 647–668. URL: https://proceedings.mlr.press/v177/sanchez22a.html.

[157] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. "Image synthesis with a single (robust) classifier". In: *Advances in Neural Information Processing Systems* 32 (2019).

[158] Axel Sauer and Andreas Geiger. "Counterfactual Generative Networks". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[159] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine

Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. "LAION-5B: An open large-scale dataset for training next generation image-text models". In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2022. URL: https://openreview.net/forum?id=M3Y74vmsMcY.

[160] Lisa Schut, Oscar Key, Rory Mc Grath, Luca Costabello, Bogdan Sacaleanu, Medb Corcoran, and Yarin Gal. "Generating Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, 2021, pp. 1756–1764.

[161] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.

[162] Sofia Serrano and Noah A. Smith. "Is Attention Interpretable?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2931–2951. DOI: 10.18653/v1/P19-1282. URL: https://aclanthology.org/P19-1282.

[163] Wen Shen, Zhihua Wei, Shikun Huang, Binbin Zhang, Jiaqi Fan, Ping Zhao, and Quanshi Zhang. "Interpretable Compositional Convolutional Neural Networks". In: *Proceedings of the International Joint Conference on Artificial Intelligence*. 2021.

[164] Sheng-Min Shih, Pin-Ju Tien, and Zohar Karnin. "GANMEX: One-vs-one Attributions Using GAN-based Model Explainability". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 9592–9602.

[165] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[166] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[167] Sumedha Singla, Brian Pollack, Junxiang Chen, and Kayhan Batmanghelich. "Explanation by Progressive Exaggeration". In: *International Conference on Learning Representations*. 2020.

[168] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2021.

[169] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.

[170] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *International Conference on Machine Learning*. 2017. URL: https://api.semanticscholar.org/CorpusID:16747630.

[171] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. *Learning Global Additive Explanations for Neural Nets Using Model Distillation*. 2018. arXiv: 1801.08640 [stat.ML].

[172] Jayaraman J. Thiagarajan, Vivek Narayanaswamy, Deepta Rajan, Jia Liang, Akshay Chaudhari, and Andreas Spanias. "Designing Counterfactual Generators using Deep Model Inversion". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021.

[173] Hugo Touvron, Matthieu Cord, and Herv'e J'egou. "DeiT III: Revenge of the ViT". In: *European Conference on Computer Vision*. 2022.

[174] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. "Robustness May Be at Odds with Accuracy". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=SyxAb30cY7.

[175] Yuki Ukai, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. "This Looks Like It Rather Than That: ProtoKNN For Similarity-Based Classifiers". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=lh-HRYxuoRr.

[176] Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, and Oliver Cobb. "Conditional generative models for counterfactual explanations". In: *arXiv preprint arXiv:2101.10123* (2021).

[177] Simon Vandenhende, Dhruv Mahajan, Filip Radenovic, and Deepti Ghadiyaram. "Making Heads or Tails: Towards Semantically Consistent Visual Counterfactuals". In: *ECCV 2022*. 2022.

[178] Bhavan Vasu and Chengjiang Long. "Iterative and Adaptive Sampling with Spatial Attention for Black-Box Model Explanations". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2020.

[179] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[180] Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. "Dataset Interfaces: Diagnosing Model Failures Using Controllable Counterfactual Generation". In: *ArXiv preprint arXiv:2302.07865*. 2023.

[181] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR". In: *Harvard Journal of Law and Technology* 31.2 (2018), pp. 841–887. ISSN: 1556-5068. DOI: 10.2139/ssrn.3063289.

[182] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[183] Bram Wallace, Akash Gokul, and Nikhil Naik. "Edict: Exact diffusion inversion via coupled transformations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 22532–22541.

[184] Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis J McCarthy, Helen Frazer, and Gustavo Carneiro. "Learning Support and Trivial Prototypes for Interpretable Image Classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 2062–2072.

[185] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. "Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2020.

[186] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. "Interpretable Image Recognition by Constructing Transparent Embedding Space". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 895–904.

[187] Pei Wang, Yijun Li, Krishna Kumar Singh, Jingwan Lu, and Nuno Vasconcelos. "IMAGINE: Image Synthesis by Image-Guided Model Inversion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 3681–3690.

[188] Pei Wang and Nuno Vasconcelos. "Scout: Self-aware discriminant counterfactual explanations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8981–8990.

[189] Xue Wang, Zhibo Wang, Haiqin Weng, Hengchang Guo, Zhifei Zhang, Lu Jin, Tao Wei, and Kui Ren. "Counterfactual-based Saliency Map: Towards Visual Contrastive Explanations for Neural Networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 2042–2051.

[190] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. "Learning to Efficiently Sample from Diffusion Probabilistic Models". In: *CoRR* abs/2106.03802 (2021). arXiv: 2106.03802.

[191] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. "F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 10267–10276. URL: https://api.semanticscholar.org/CorpusID:85502844.

[192] Yiran Xu, Xiaoyin Yang, Lihang Gong, Hsuan-Chu Lin, Tz-Ying Wu, Yunsheng Li, and Nuno Vasconcelos. "Explainable Object-Induced Action Decision for Autonomous Vehicles". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 9520–9529.

[193] Mengqi Xue, Qihan Huang, Haofei Zhang, Lechao Cheng, Jie Song, Minghui Wu, and Mingli Song. "ProtoPFormer: Concentrating on Prototypical Parts in Vision Transformers for Interpretable Image Recognition". In: *ArXiv* abs/2208.10431 (2022). URL: https://api.semanticscholar.org/CorpusID:251718906.

[194] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. "On Completeness-aware Concept-Based Explanations in Deep Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 20554–20565. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/ecb287ff763c169694f682af52c1f309-Paper.pdf.

[195] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. "A-ViT: Adaptive Tokens for Efficient Vision Transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[196] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[197] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. "BDD100K: A Diverse Driving

Dataset for Heterogeneous Multitask Learning". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 2633–2642.

[198] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. "MetaFormer Baselines for Vision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.2 (2024), pp. 896–912. DOI: 10.1109/TPAMI.2023.3329173.

[199] Yoshinori Konishi Yuhki Hatakeyama Hiroki Sakuma and Kohei Suenaga. "Visualizing Color-wise Saliency of Black-Box Image Classification Models". In: *Asian Conference on Computer Vision (ACCV)*. 2020.

[200] Mehdi Zemni, Mickaël Chen, Éloi Zablocki, Hédi Ben-Younes, Patrick Pérez, and Matthieu Cord. "OCTET: Object-aware Counterfactual Explanations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 15062–15071.

[201] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. *Adding Conditional Control to Text-to-Image Diffusion Models*. 2023.

[202] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[203] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[204] Zhengli Zhao, Dheeru Dua, and Sameer Singh. "Generating Natural Adversarial Examples". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[205] Quan Zheng, Ziwei Wang, Jie Zhou, and Jiwen Lu. "Shap-CAM: Visual Explanations for Convolutional Neural Networks based on Shapley Value". In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2022).

[206] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[207] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. "Interpretable Basis Decomposition for Visual Explanation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[208] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. "Transferable adversarial perturbations". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 452–467.

[209] Ligeng Zhu, Zhijian Liu, and Song Han. "Deep Leakage from Gradients". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf.

[210] Yao Zhu, Jiacheng Ma, Jiacheng Sun, Zewei Chen, Rongxin Jiang, Yaowu Chen, and Zhenguo Li. "Towards understanding the generative capability of adversarially robust classifiers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7728–7737.

Appendix A

Supplementary Material: Adversarial Counterfactual Visual Explanations

A.1 Detailed Implementation Details

For each dataset, we used different configurations in architecture and for the generation of the pre-explanation. Yet, we tune all hyperparameters from a empirical perspective[1]. We tuned τ such that the input image and its filtered instance are visually similar. Additionally, the classification between these two images are the same. To adjust the hyperparameter λ_d, we performed a simple visual inspection. Finally, for the threshold, we ablated its values empirically for each dataset. When using the distance loss ℓ_1, we set the distance regularization constant to $\lambda_d = 0.001$ while $\lambda_d = 0.1$ for ℓ_2. For the final refinement, firstly, we normalize the mask by the maximum pixel's difference magnitude. For the dilation step, we set the mask as a square with a width and height of 15 pixels for all datasets. Finally, we used the cross entropy for all experiments as the L_{class} loss. Next, we will show all implementation details for each dataset.

CelebA [107]: We used the same architecture and weights as [79]. Additionally, we set $\tau = 5$ with a total amount of steps as 50. At the refinement stage, we used the same threshold of 0.15 for both ℓ_1 and ℓ_2 experiments for smile and age attributes.

CelebA HQ [99]: Our model follows the same architecture than[36] for ImageNet 256×256 unconditional generation. Since CelebA HQ is far less complex than ImageNet, we reduced the number of channels from 256 to 128. Also, our model generates samples using 500 diffusion steps instead of 1000. For training, we iterated our model for 120.000 iterations with a batch size of 256 on two V100 GPUs following[36]'s code. We set the learning rate to 10^4, a weight decay of 0.05, and no dropout.

[1]Note that all these hyperparameters are not the same as the classically found in machine learning. These variables can be adjusted by the user in an 'online' manner according to his/her expectations. Hence, a global configuration is a mere rough estimate of these parameters and can be accommodated instance-wise.

To generate the pre-explanations, we noise the image until $\tau = 5$ out of 25 re-spaced steps. To binarize the mask, we used a threshold of 0.15 and 0.1 for the smiling attribute with the ℓ_1 and ℓ_2 distance losses, respectively. For the age attribute, we used 0.15 for ℓ_1 and 0.05 for ℓ_2.

BDD100k/OIA [197, 192]: The counterfactual explanation research community opted to use BDD100k in a 512×256 setup. This is highly demanding computationally to create a DDPM. Thus, since we knew *a priori* that we do not need many iterations for ACE to generate counterfactuals, we trained our diffusion model partially in the Markov chain. That is, our DDPM cannot generate images from pure noise. Instead, we trained it to generate images solely from a quarter of the complete chain, requiring an input instance to warm up the generation. So, we trained our model to generate instances with 250 steps out of 1000. This enabled us to use a lighter model. Artitecnologically, our UNet model has four downsampling stages with $128\,s$ channels, where s is the downsampling stage. Finally, we used the attention layer at the deeper layer of the UNet. At the training phase, we used a batch size of 256, a learning rate of 10^4, and a weight decay and dropout of 0.05 for 50.000 iterations.

To generate our explanations, we used 5 out of 100 (re-spaced) diffusion steps. For ℓ_1, we used a threshold of 0.05 and 0.1 for ℓ_2 for both datasets.

ImageNet [35]: For this dataset, we took advantage of previous works. In this case, we utilised Dhariwal and Nichol [36]'s model on ImageNet 256. To generate the explanations, we used 5 steps out of 25 for the pre-explanations and set the threshold to 0.15 to binarize the mask for all cases.

A.2 Overview of ACE

ACE is a two-step method: firstly is the pre-explanation construction – Algorithm 1 – and then the refinement process – Algorithm 2. To generate the pre-explanation, **(1)** we add noise to the input image x using the forward Markov chain until an intermediate step τ, *i.e.* it doesn't begin from random Gaussian noise. Instead, it warms up the generation with the input image through

$$x_t = \sqrt{\bar{\alpha}_t}\, x + \sqrt{1 - \bar{\alpha}_t}\, \epsilon,\ \epsilon \sim \mathcal{N}(0, I).$$

(2) ACE iteratively denoises the noisy image using the DDPM algorithm with

$$x_{t-1} = \mu_t(x_t) + \Sigma_t(x_t)\, \epsilon,\ \epsilon \sim \mathcal{N}(0, I),$$

where μ_t and Σ_t are the output of the diffusion model. **(3)** The scrutinized classifier uses the filtered image to compute loss function. Then, we calculate the gradients with respect to the input image x in step 1, all the way through the τ steps of the

diffusion model. **(4)** ACE applies the gradients as the update step with the attack of choice. It iterates these four steps to create the pre-explanation. For the refinement, it creates the mask m using the difference between the pre-explanation and the original input. Then, it dilates and thresholds it to generate the binary version. Finally, ACE builds on RePaint to keep untouched any region lying outside the mask. The final result is the counterfactual explanation.

Algorithm 1 Pre-explanation generation

Require: Diffusion Model D, Distance loss d and its regularization constant λ_d, classification loss L_{class} comprising the classifier under observation, number of noising steps τ, attack optimization algorithm PGD, number of update iterations n, initial instance x, target label y

function PRE-EXPLANATION(x, y)
  $n \leftarrow 0$
  $x_{orig} \leftarrow x$
  while $n < N$ **do** ▷ Attack iteration steps
    $\epsilon \sim \mathcal{N}(0, I)$
    $x' \leftarrow \sqrt{\bar{\alpha}_\tau} x + \sqrt{1 - \bar{\alpha}_\tau}\epsilon$ ▷ Add noise
    $ts \leftarrow \tau - 1$
    while $ts \geq 0$ **do** ▷ DDPM denoising
      $\mu, \Sigma \leftarrow D(x', ts)$
      $\epsilon \sim \mathcal{N}(0, I)$
      $x' \leftarrow \mu + \epsilon\Sigma$
      $ts \leftarrow ts - 1$
    end while
    $g \leftarrow \nabla_{x'} L_{class}(x'; y') + \lambda_d d(x', x_{orig})$
    $x \leftarrow PGD(x, g)$ ▷ Update with attack
    $n + 1 \leftarrow n$
  end while
  return x' ▷ Pre-explanation
end function

A.3 Qualitative Results

In this section, we show more qualitative results. We will display the input image, its pre-explanation, the mask, and the final counterfactual for both ℓ_1 and ℓ_2 losses on all datasets. Note that we added a small discussion on the caption analyzing the results. In Fig. A.10, we compare a few examples of DiME and ACE.

Algorithm 2 Post-processing

Require: Diffusion Model D, number of noising steps τ, mask dilation size d, threshold u, initial instance x, pre-explanation x'

function POST-PROCESSING(x, x')
  $x_{orig} \leftarrow x$
  $\epsilon \sim \mathcal{N}(0, I)$
  $x' \leftarrow \sqrt{\bar{\alpha}_\tau} x' + \sqrt{1 - \bar{\alpha}_\tau}\epsilon$
  $ts \leftarrow \tau - 1$
  # Mask generation
  $m \leftarrow sum\_over\_channels(abs(x - x'))$
  $m \leftarrow {}^{m}/_{maximum(m)}$
  $m \leftarrow dilation(m, size = d) > u$
  while $ts \geq 0$ **do** ▷ DDPM denoising
    $\epsilon \sim \mathcal{N}(0, I)$
    $x_{ts} \leftarrow \sqrt{\bar{\alpha}_{ts}} x + \sqrt{1 - \bar{\alpha}_{ts}}\epsilon$
    $x' \leftarrow m\, x' + (1 - m)\, x_{ts}$
    $\mu, \Sigma \leftarrow D(x', ts)$
    $\epsilon \sim \mathcal{N}(0, I)$
    $x' \leftarrow \mu + \epsilon\Sigma$
    $ts \leftarrow ts - 1$
  end while
  return x' ▷ Counterfactual explanation
end function

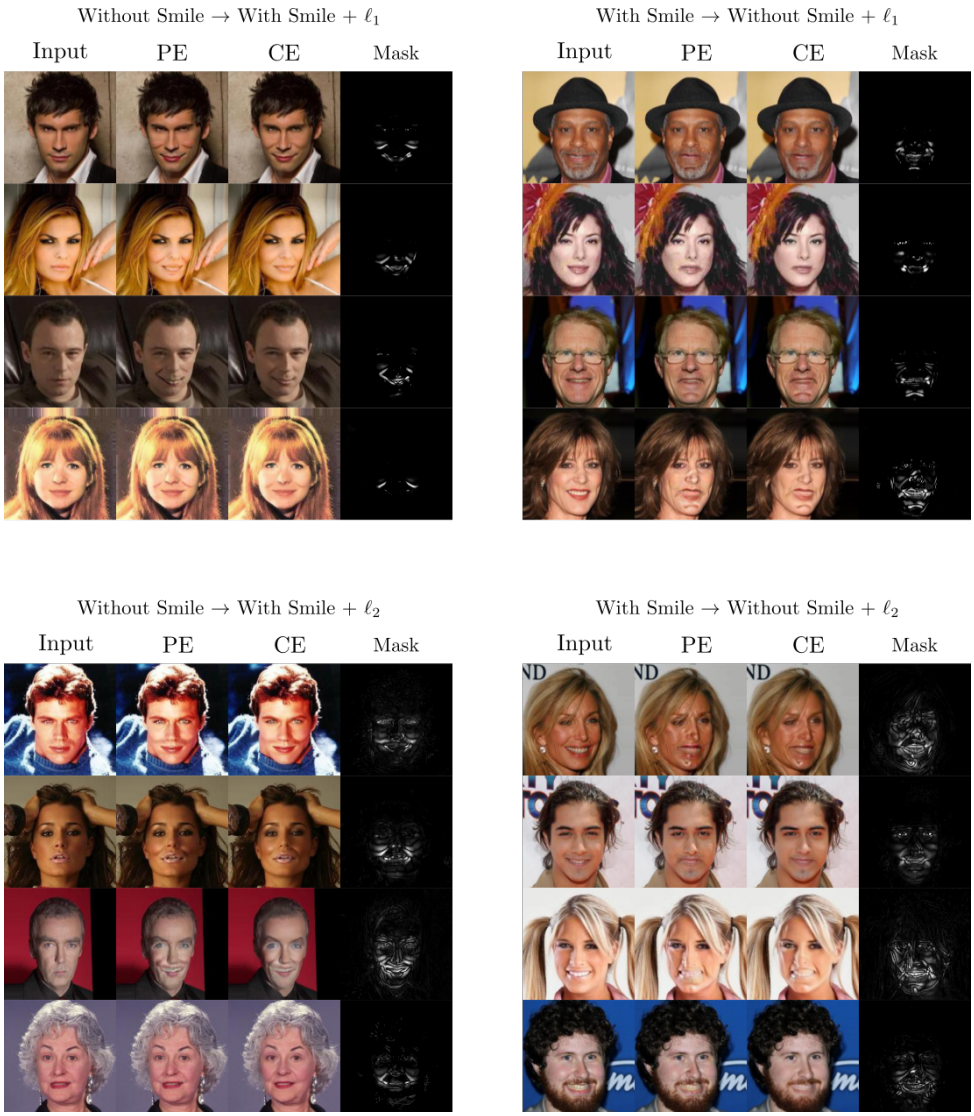


FIGURE A.1: Additional CelebA qualitative results. We show examples for the *Smiling* attribute for both distances losses. From our qualitative experiments, we see that removing the smile attributes is harder than adding them. Additionally, we see that the ℓ_1 loss creates more sparse editings.

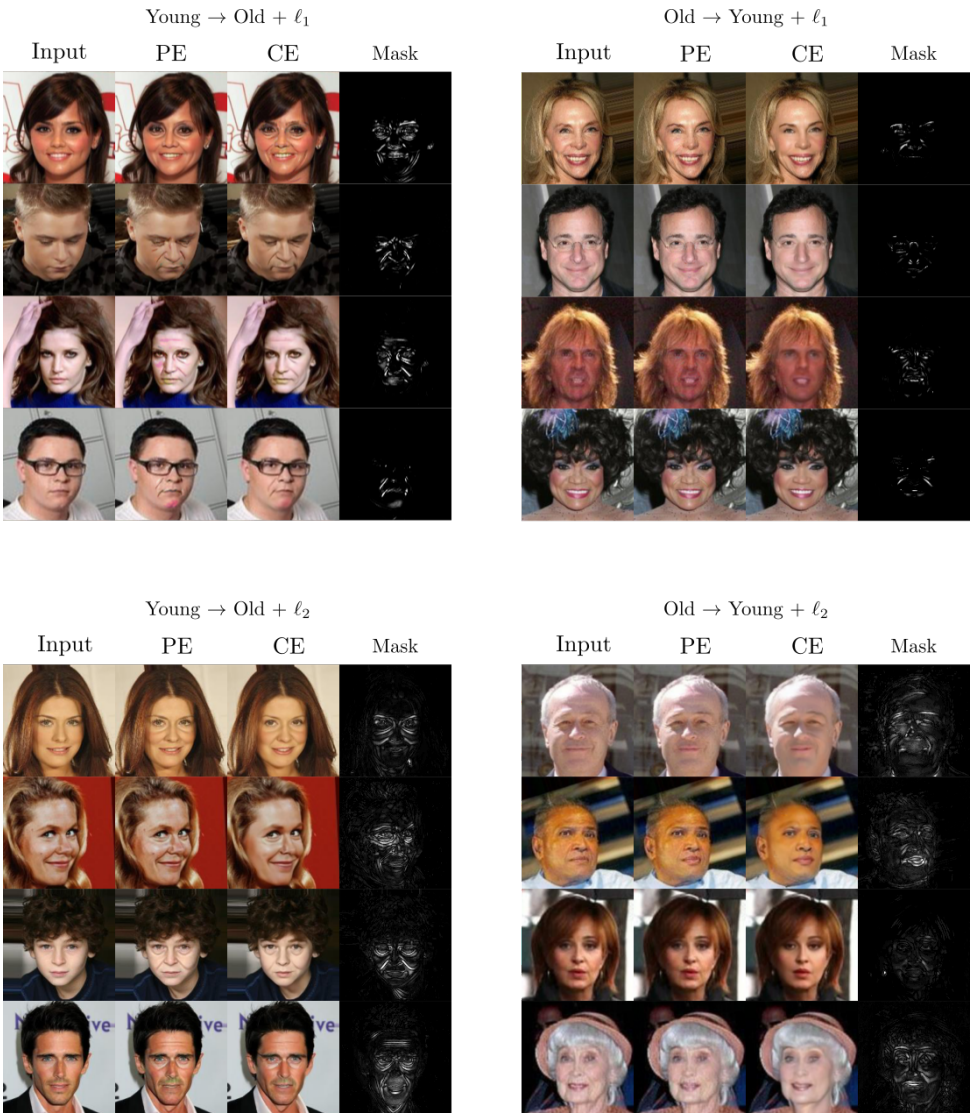


FIGURE A.2: Additional CelebA qualitative results. We show examples for the *Age* attribute for both distances losses. The results show that the ℓ_1 loss creates more out-of-distribution artifacts.

FIGURE A.3: Additional CelebA HQ qualitative results. We show examples for the *Smiling* attribute for both distances losses. We see similar behavior in the CelebA dataset.

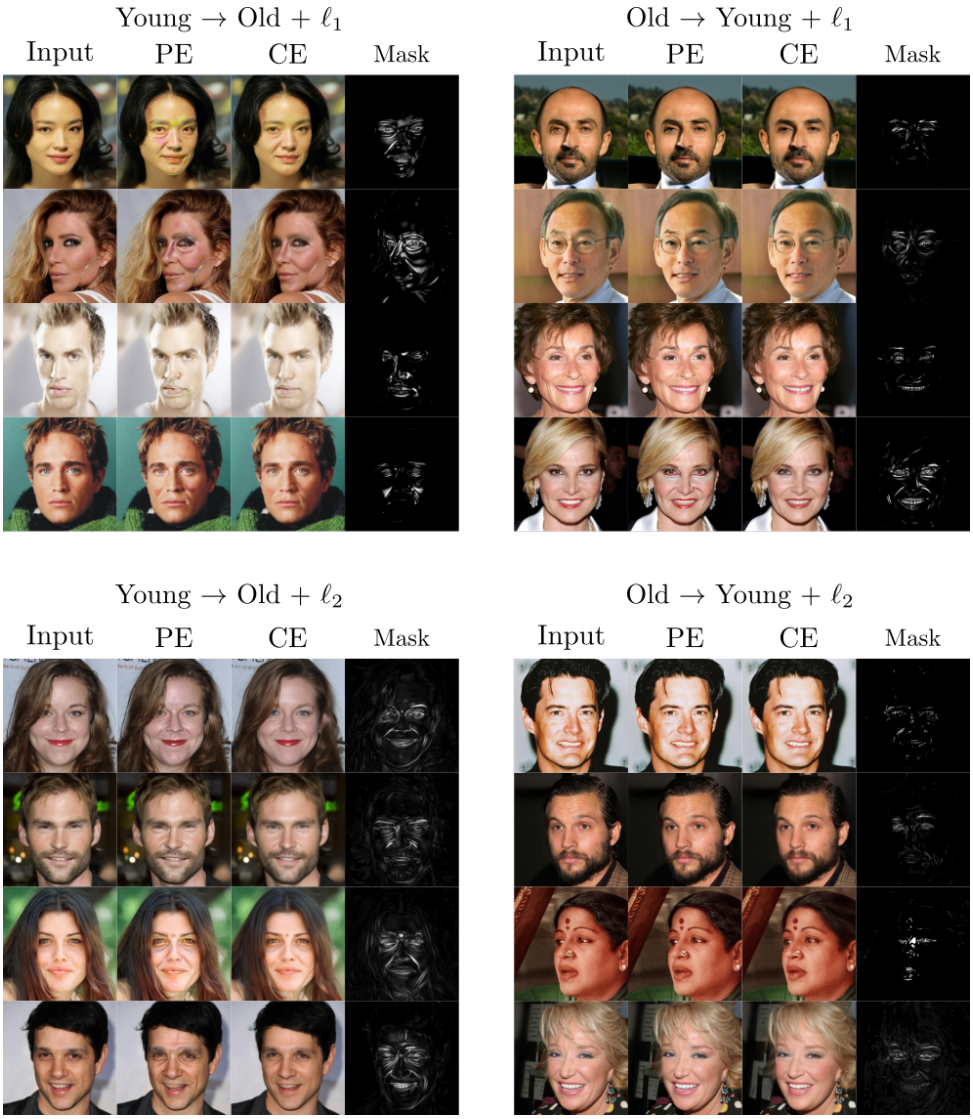


FIGURE A.4: Additional CelebA HQ qualitative results. We show examples for the *Age* attribute for both distances losses. These examples show that transforming *Old* to *Young* is less informative than the other way.

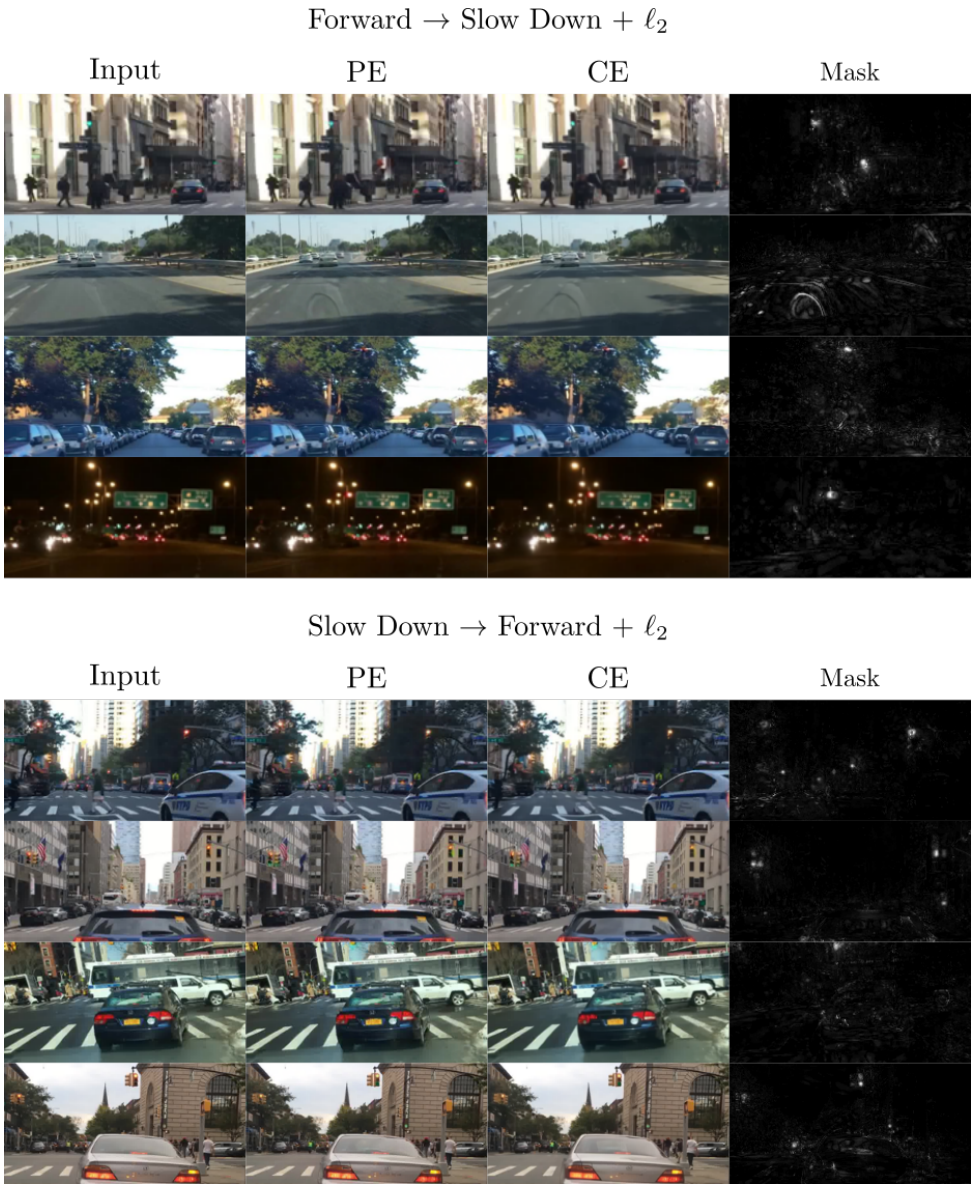


FIGURE A.5: Additional BDD qualitative results. We show examples for the *Forward / Slow Down* binary class for ℓ_2 distance loss. We show a zoom of the changes in the image since the perturbations are sparse. We see that ACE adds traffic light colors in the buildings to change the prediction.

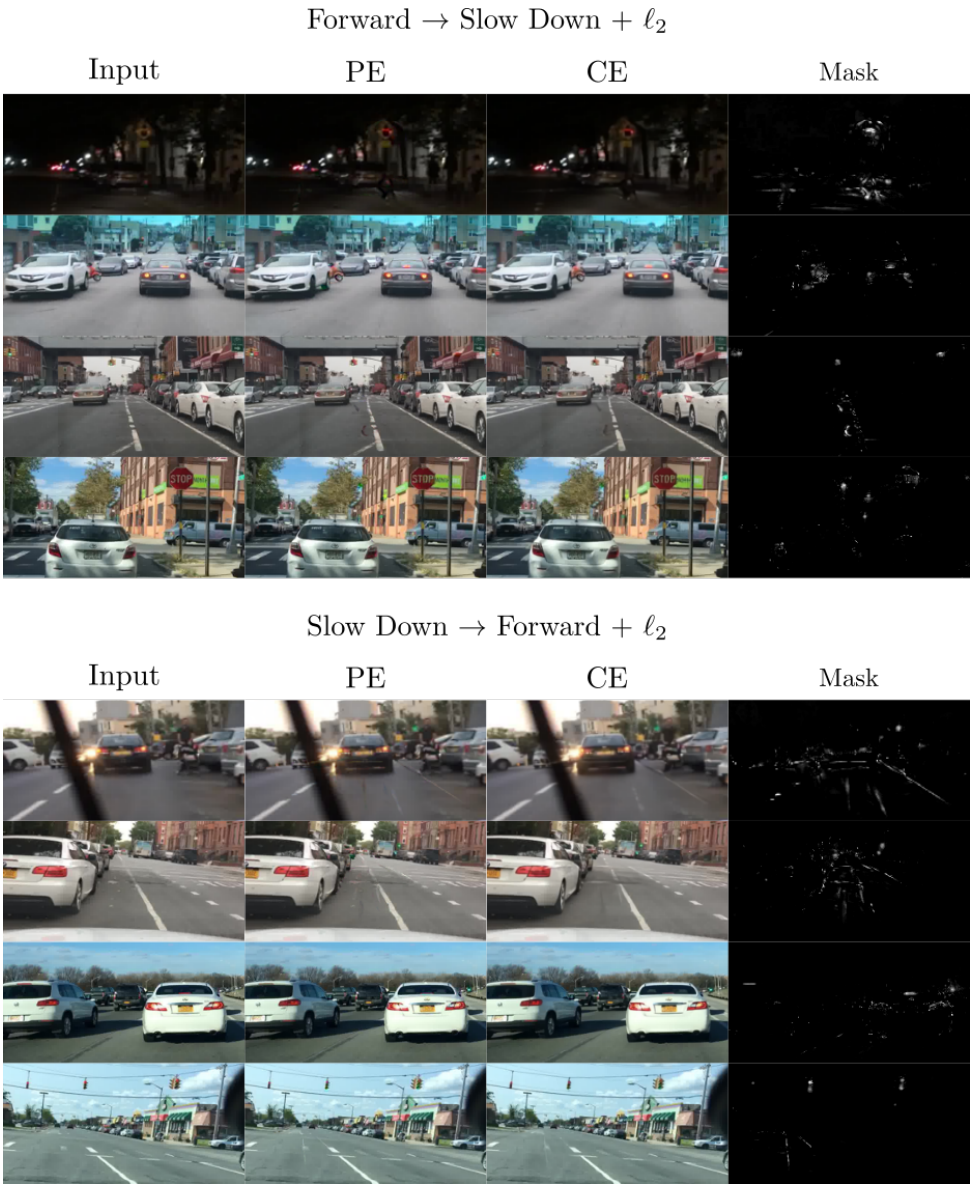


FIGURE A.6: Additional BDD qualitative results. We show examples for the *Forward / Slow Down* binary class for ℓ_1 distance loss. We show a zoom of the changes in the image since the perturbations are sparse. We show a zoom of the changes in the image since the perturbations are sparse. We see that ACE adds traffic light colors in the buildings to change the prediction.

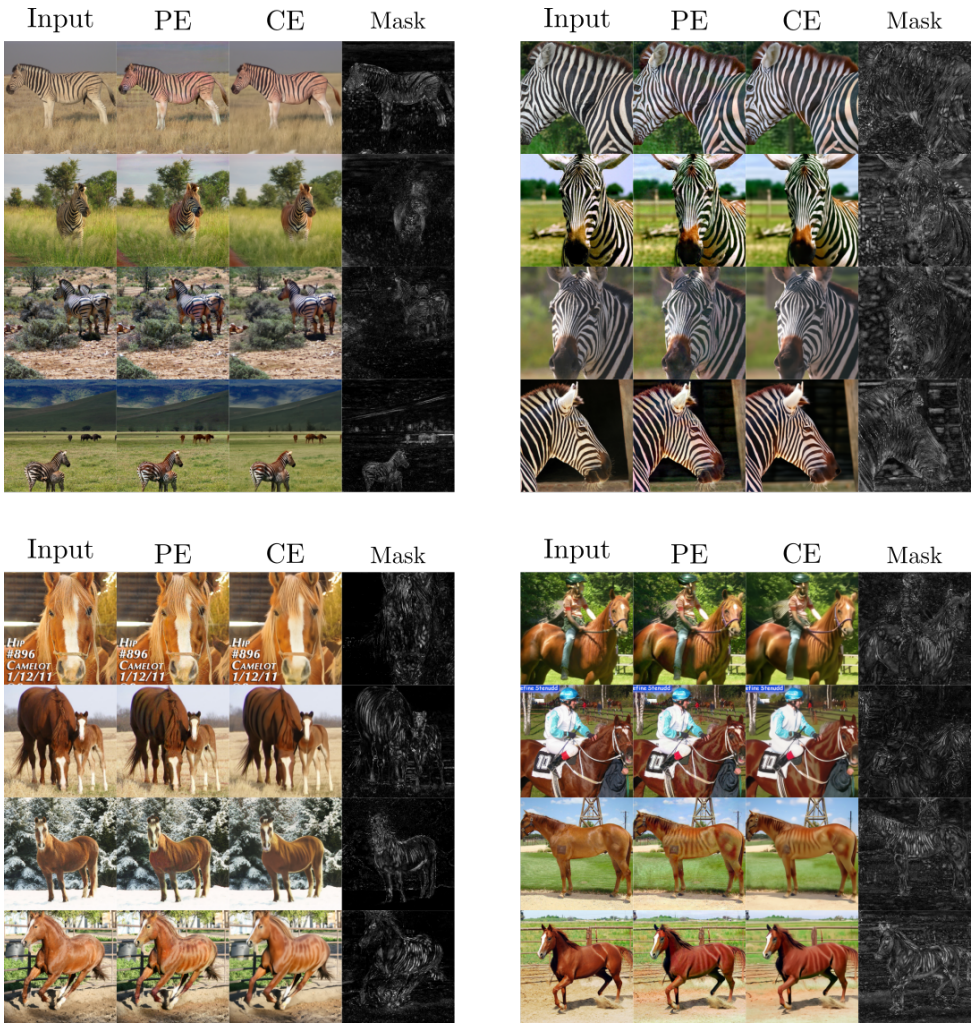


FIGURE A.7: Additional ImageNet qualitative results. We show examples for the *Zebra / Sorrel* categories class. The first column is the ℓ_1 distance loss while the second one is ℓ_2. The initial row is zebra to sorrel and the second one is the inverse. To change from zebras to sorrels, some examples show not only incorporating the brown color sorrel horses but also the context in the background (*e.g.* adding a stable-like background). Vice-versa, to classify a horse as a zebra it is enough to add some strips.

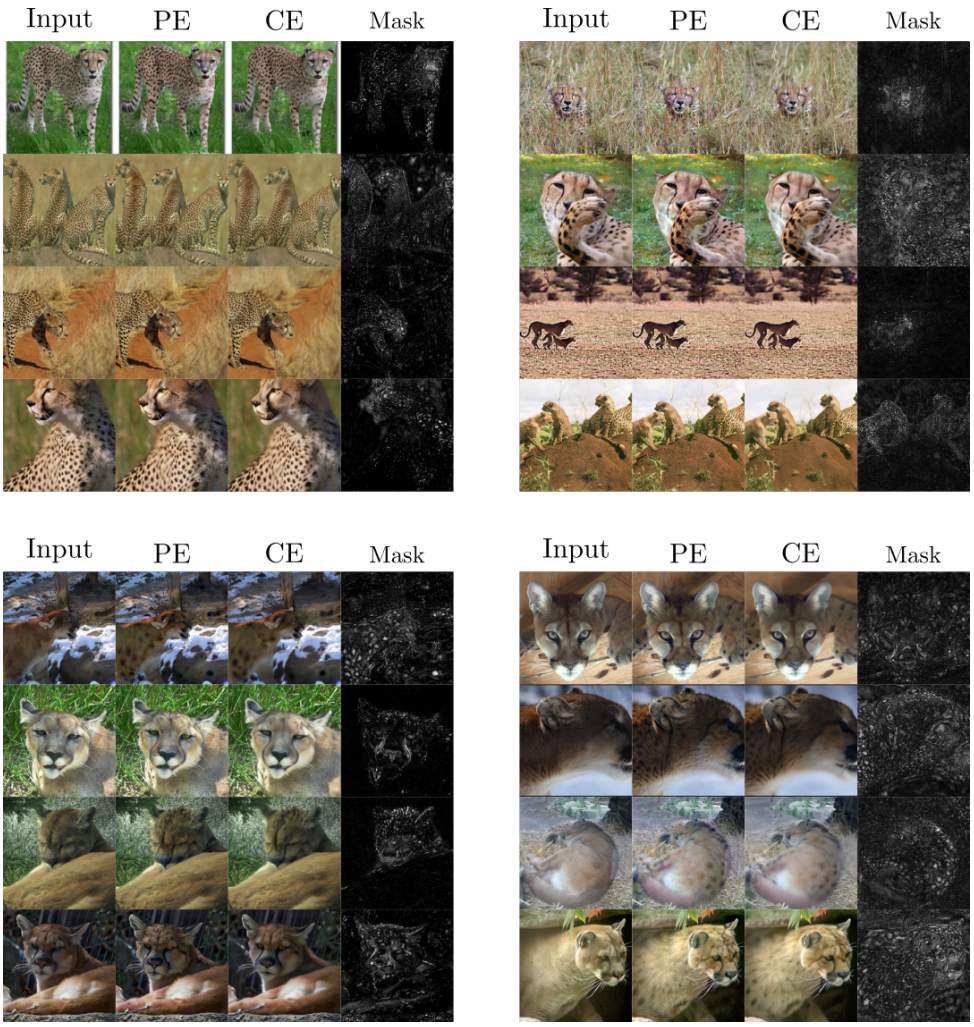


FIGURE A.8: Additional ImageNet qualitative results. We show examples for the *Cheetah / Cougar* categories class. The first column is the ℓ_1 distance loss while the second one is ℓ_2. The first row is cheetah to cougar and the second is the inverse. We mainly see that changing from cheetah to cougar is enough to target the face of the animal. Vice-versa, to classify a cougar as a cheetah, ACE adds spots and characteristic cheetah stripes on the face.

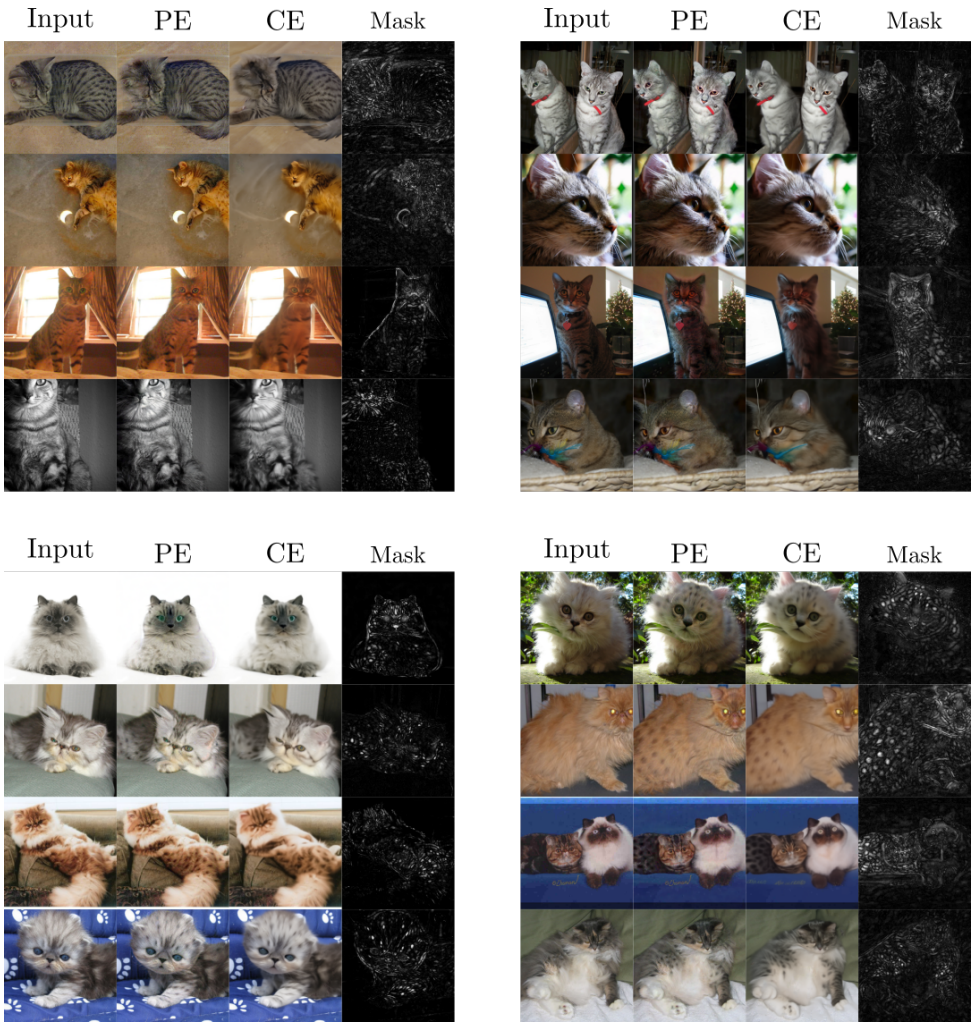


FIGURE A.9: Additional ImageNet qualitative results. We show examples for the *Egyptian / Persian cat* categories class. The first column is the ℓ_1 distance loss while the second one is ℓ_2. The row is Egyptian to Persian cat and the second is the inverse. To change from Egyptian to Persian, we mainly see that ACE adds the Persian cats' fluffy fur. Conversely, from Persian to Egyptian it adds spots.

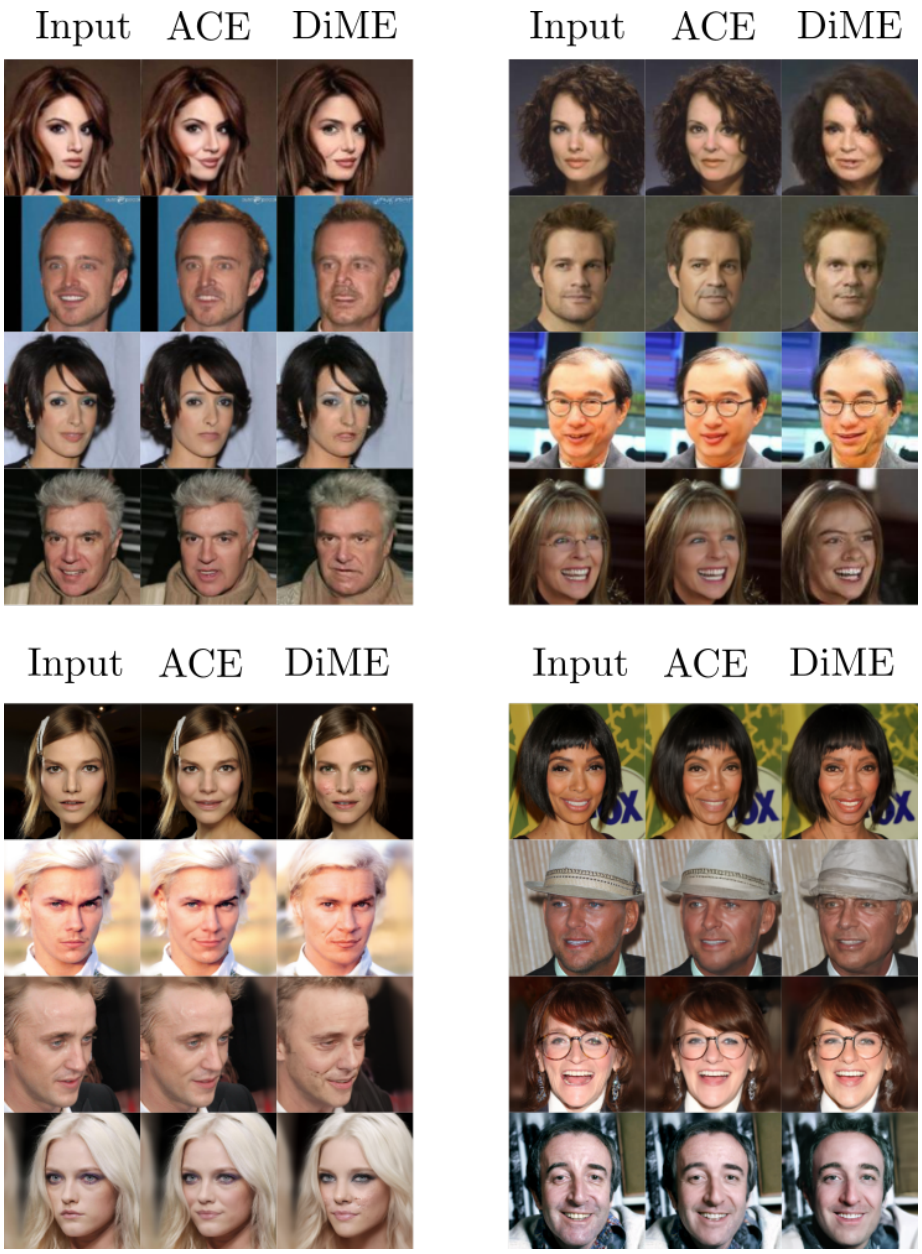


FIGURE A.10: ACE *vs.* DiME. We display some examples showing some differences between DiME counterfactuals and ACE's. In short, ACE is capable of not modifying useless information, such as the background, to generate its counterfactuals. Top row: CelebA. Bottom row: CelebA HQ. Left Column: Smiling attribute. Right Column: Age attribute.

Appendix B

Supplementary Material: Text-to-Image Models for Counterfactual Explanations: a Black-Box Approach

B.1 Evaluation Criteria

Before describing each metric and its formulation, we will thoroughly describe the goals of counterfactual explanations. As we stated in the main manuscript, counterfactual explanations seek to change an instance prediction by modifying the input instance. However, these modifications must be small but perceptually coherent. From the previous statement, we can extract many goals of CEs:

1. CEs must flip the decision of the classifier. In the literature, this feature is called *validity*.

2. The counterfactual changes should be plausible and realistic - simply referred to as *realism*. Visual automated systems are generally brittle to adversarial noise [54]. This noise is designed to fool the classifier, but with the restriction that it is hidden from visual inspection. Since this noise cannot be perceived, it cannot be analyzed to find spurious correlations. Therefore, only realistic and plausible changes are allowed.

3. The algorithm must generate *proximal and sparse* counterfactuals. One could create a valid and realistic explanation by simply replacing the target instance with a new one. This still obeys the realistic and valid goals. However, it does not give any information about the variables. Thus, the modifications must be sparse and close to the image to visually observe which variables have changed.

4. Finally, the algorithm must generate the explanation *efficiently*. This property is required to avoid delays for the user.

Now we will proceed to describe each evaluation metric and link it to its corresponding objective. As for notations, let $M(x, y)$ be the counterfactual algorithm applied to an image $x \in D$ targeting the class y, where D is a dataset. Additionally, let C be the classifier, $\mathbb{1}(condition)$ a function that is one if the condition is true or zero otherwise. Finally, let $a \in A$ be an attribute in a set A, then O^a is an attribute oracle classifier for a. This network predicts if its input has the attribute a. Similarly, let $\mathcal{O}$ be an identity verification network This DNN is trained to give a similarity measure between two images, often computed with the cosine similarity CS.

Success Rate. The success rate (or flip rate) measures the ratio at which counterfactuals have successfully reversed the original classifier's decision. This metric correlates with the validity goal. To measure it, we simply compute the proportion of valid counterfactuals to the size of the dataset, as in

$$SR = \frac{1}{|D|} \sum_{x \in D} \mathbb{1}(C(M(x, y)) = y). \tag{B.1}$$

Realism. To approximate the realism of the counterfactuals, the literature adopts the FID [63] metric from generation research. Furthermore, Jeanneret, Simon, and Jurie [78] extended the metric by computing the FID between the half of the dataset and the counterfactuals of the complement set. This was motivated to reduce the inherent bias in computing the FID, given that the difference between the original images and their CE is a few pixels in the image.

Proximity and Sparsity. To evaluate this goal, previous methods proposed several metrics to quantify the degree of dissimilarity between an instance and its explanation. Initially, most metrics were proposed for face images. Jacob et al. [74] suggested using the mean number of attributes changed (MNAC), computed as follows:

$$MNAC = \frac{1}{|D|} \sum_{x \in D} \sum_{a \in A} \mathbb{1}(O^a(M(x, y)) \neq O^a(x)). \tag{B.2}$$

However, [79] noted that counterfactual methods will change some attributes if they are correlated. Thus, based on the MNAC, the Correlation Difference (CD) [79] measures the correlations produced by M. To further assess the proximity and sparsity in face counterfactuals, Singla et al. [167] suggested using the Face Verification Accuracy (FVA) to compute whether M cannot modify the identity of the person. This metric is calculated as

$$FVA = \frac{1}{|D|} \sum_{x \in D} \mathbb{1}(CS(\mathcal{O}(x), \mathcal{O}(M(x, y))) > 0.5). \tag{B.3}$$

Jeanneret, Simon, and Jurie [78] noted that this metric was already saturated. To measure a more fine-grained metric, they proposed taking the continuous CS and

calling the metric face similarity (FS):

$$FS = \frac{1}{|D|} \sum_{x \in D} CS(\mathcal{O}(x), \mathcal{O}(M(x, y)). \tag{B.4}$$

Finally, the same authors extended this metric for general-purpose images by computing Eq. B.4 using a self-supervised trained model as $\mathcal{O}$. They called this metric S^3. Finally, [85] proposed to compute COUT. This metric computes the probability of the class y using multiple linear interpolations between x and $M(x, y)$.

Efficiency. The literature generally ignores computing an *efficiency* metric. To compute the efficiency of counterfactual models, the widely accepted metric is floating point operations (FLOPs). In addition, it is also recommended to compute the average time per counterfactual. However, this metric is only comparable if all measurements are computed on the under the same circumstances.

B.2 Qualitative Results

In this section, we provide additional qualitative results. For the CelebA HQ [99] dataset, we provide our and ACE [78] counterfactuals to show the differences.

FIGURE B.1: Counterfactual Explanations targeting the Non-Smile attribute.

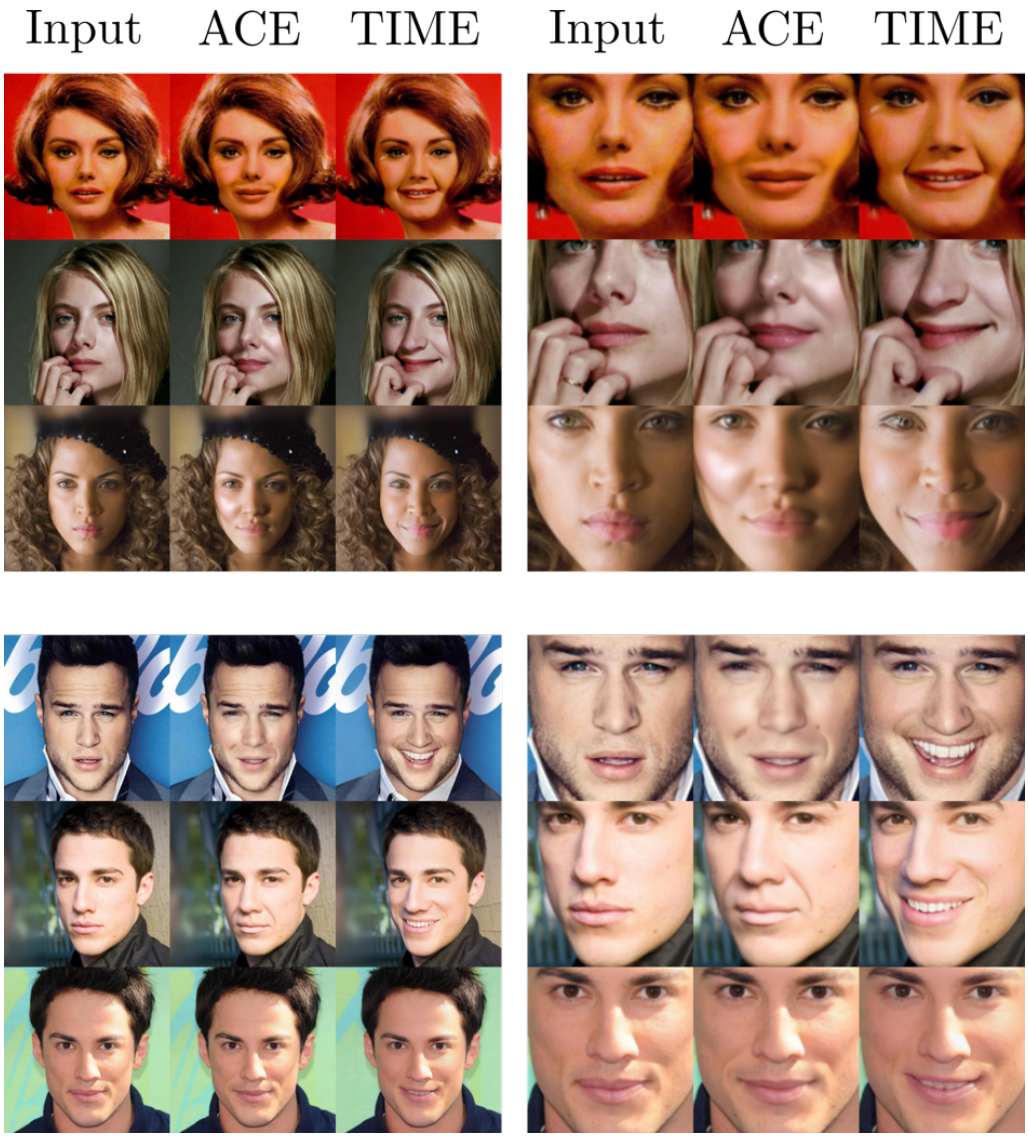


FIGURE B.2: Counterfactual Explanations targeting the Smile attribute.

FIGURE B.3: Counterfactual Explanations targeting the Young attribute.

FIGURE B.4: Counterfactual Explanations targeting the Old attribute.

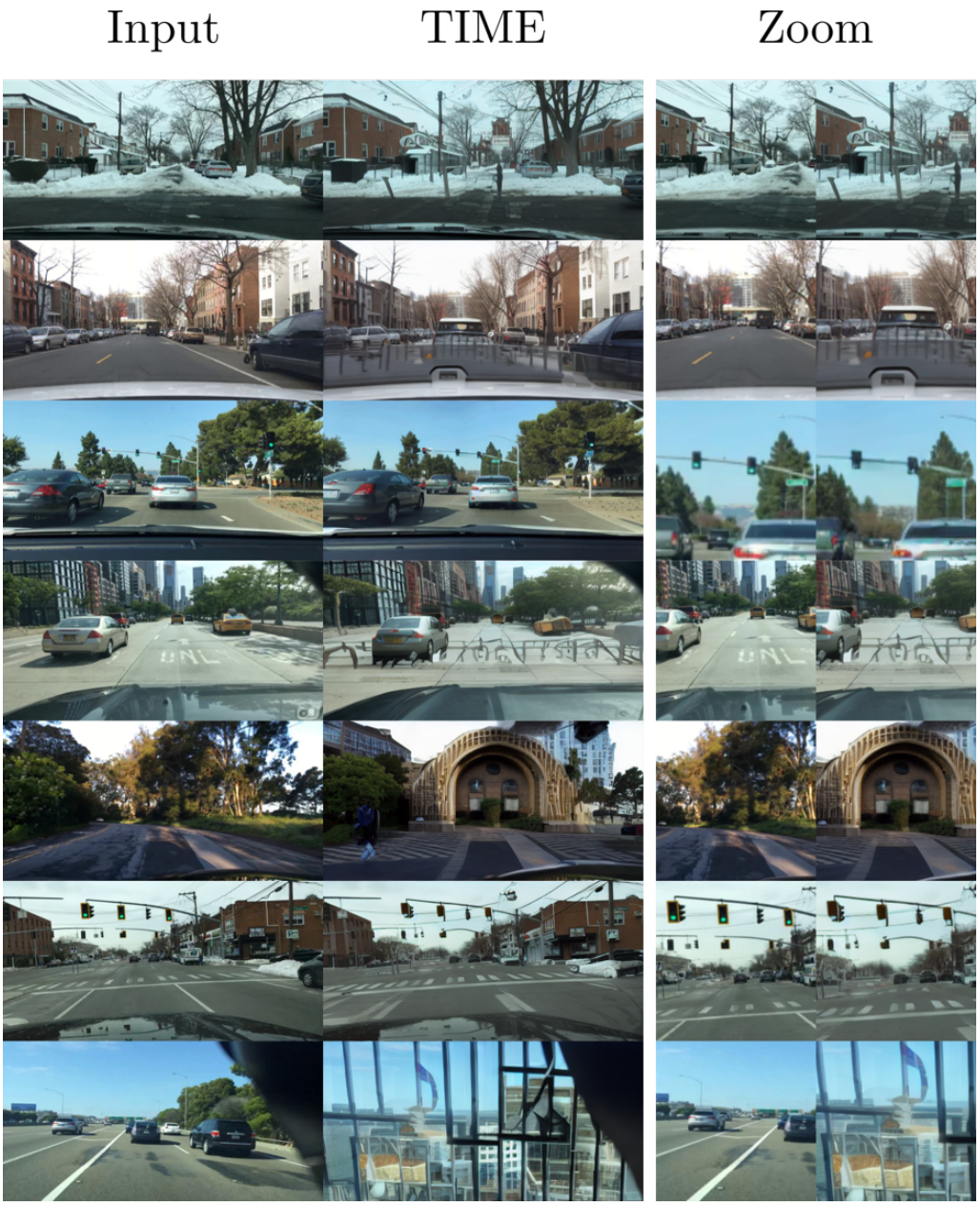


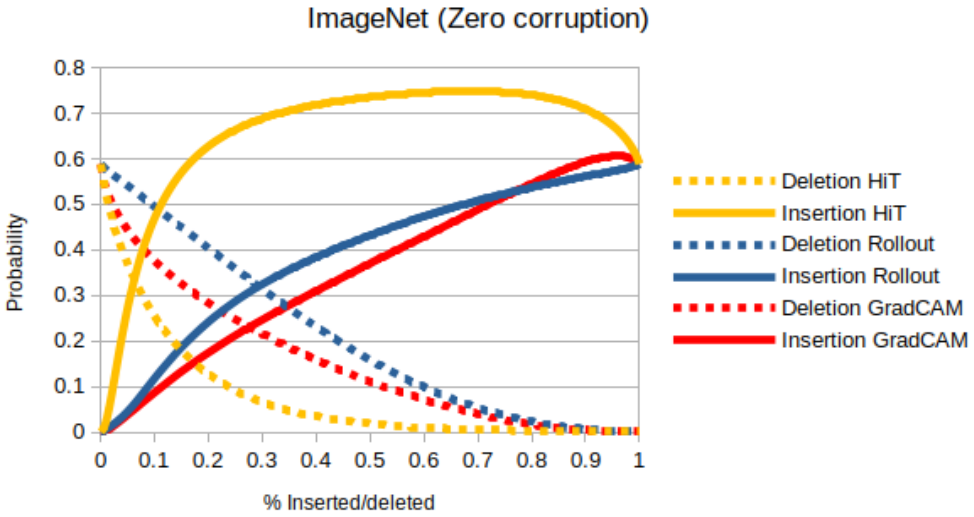
FIGURE B.5: Counterfactual Explanations targeting the Stop action.

FIGURE B.6: Counterfactual Explanations targeting the Forward action.

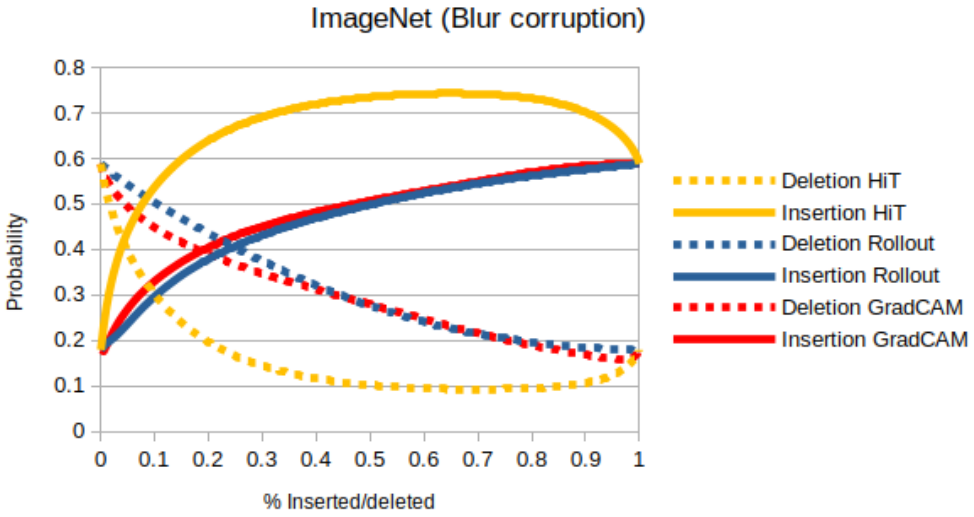
Appendix C

Supplementary Material: Disentangling Visual Transformers: Patch-level Interpretability for Image Classification

In the following figures, we present the insertion-deletion curves for the ImageNet [35], CUB-200-2011 [182], Stanford Cars [96], and Stanford Dogs [86] datasets. We employ both zero-filled and blur-filled corruption strategies. We quantitatively compare our approach to GradCAM [161] and a modified rollout matrix [1]. Since HiT lacks cross-token attention, the rollout is equivalent to averaging attention across all layers. The steep increase and subsequent decrease in probability confirms that HiT maps effectively identify the most important tokens for classification, as the initial inserted tokens significantly increase the probability distribution. In contrast, GradCAM and the rollout matrix fail to achieve comparable performance.

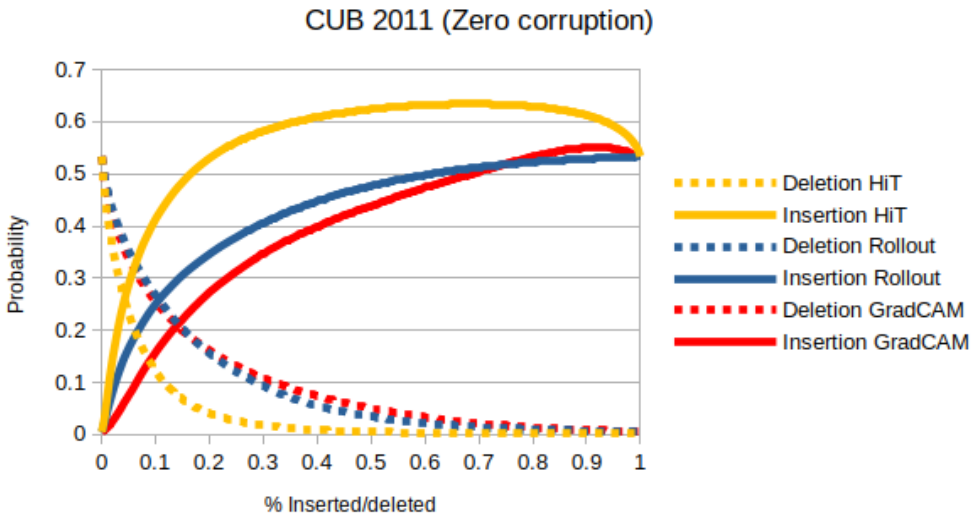


(A) Zero-filled corruption

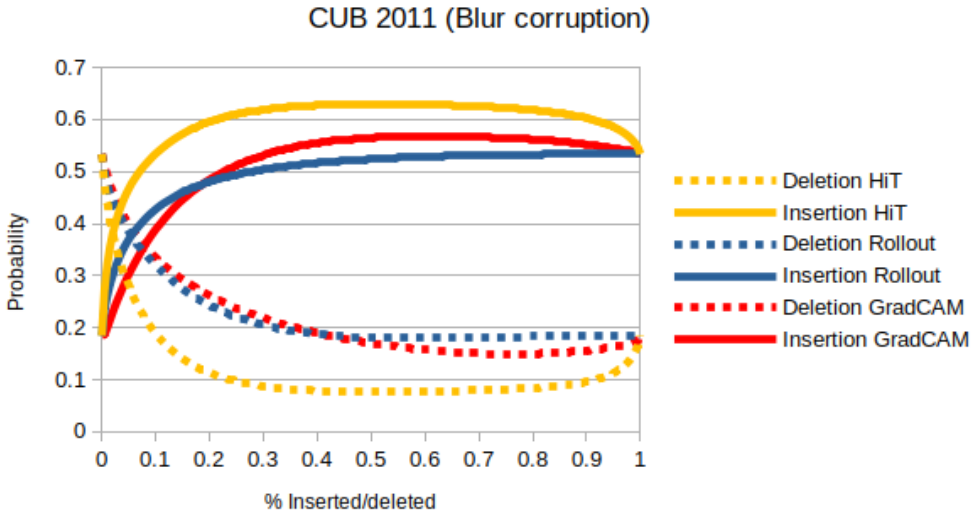


(B) Blur-filled corruption

FIGURE C.1: ImageNet Insertion-Deletion

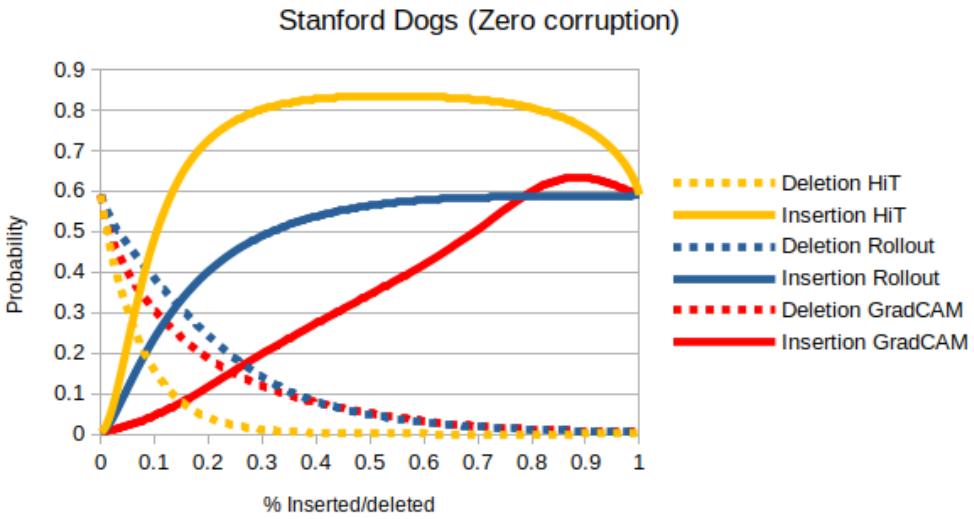


(A) Zero-filled corruption

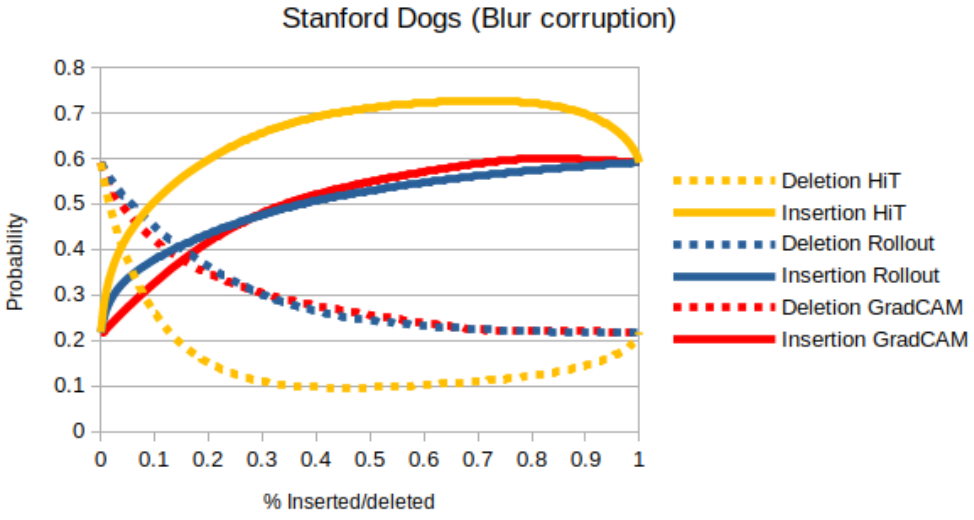


(B) Blur-filled corruption

FIGURE C.2: CUB Insertion-Deletion

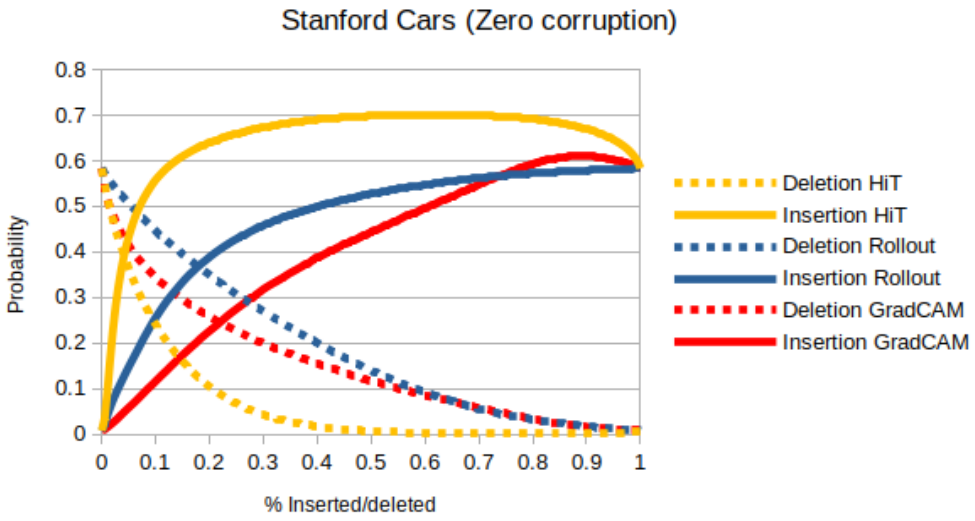


(A) Zero-filled corruption

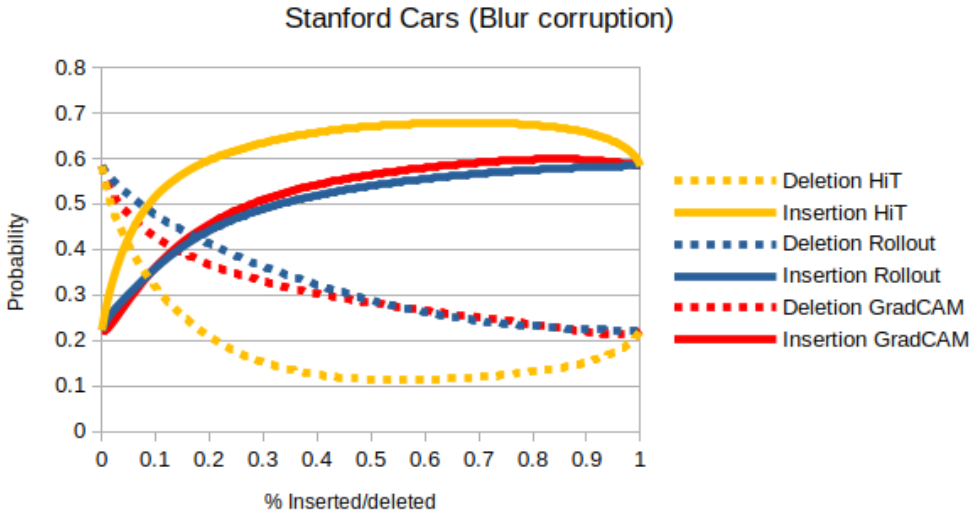


(B) Blur-filled corruption

FIGURE C.3: Stanford Dogs Insertion-Deletion



(A) Zero-filled corruption



(B) Blur-filled corruption

FIGURE C.4: Stanford Cars Insertion-Deletion