Contrôle des modèles génératifs pour l'édition d'images et l'augmentation de données

Perla Doubinsky

▶ **To cite this version:**

Perla Doubinsky. Contrôle des modèles génératifs pour l'édition d'images et l'augmentation de données. Informatique [cs]. HESAM Université, 2024. Français. NNT : 2024HESAC004 . tel-04732212

HAL Id: tel-04732212

https://theses.hal.science/tel-04732212v1

Submitted on 11 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HESAM UNIVERSITÉ

le cnam

ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR
Centre d'études et de recherche en informatique et communications

THÈSE

présentée par : **Perla DOUBINSKY**

soutenue le : **25 mars 2024**

pour obtenir le grade de : **Docteure d'HESAM Université**

préparée au : **Conservatoire National des Arts et Métiers**

Discipline : **Mathématique, Informatique et Systèmes**

Spécialité : **Informatique**

Controllable Generative Models for Image Editing and Data Augmentation

THÈSE dirigée par :
M. CRUCIANU Michel Professeur des universités, Cedric, Cnam

et co-encadrée par :
M. AUDEBERT Nicolas Maître de conférences, Cedric, Cnam
M. LE BORGNE Hervé Ingénieur-chercheur, CEA List

Jury

M^me^ Céline HUDELOT	Professeure, CentraleSupélec	Présidente du jury
M. Andrés ALMANSA	Directeur de recherche, Université Paris Cité	Rapporteur
M. David PICARD	Directeur de recherche, École Nationale des Ponts et Chaussées	Rapporteur
M. Frédéric JURIE	Professeur, Université de Caen	Examinateur
M^me^ Vicky KALOGEITON	Maîtresse de conférences, École Polytechnique	Examinatrice
M. Nicolas AUDEBERT	Maître de conférences, Cnam	Co-encadrant
M. Hervé LE BORGNE	Ingénieur-Chercheur, CEA List	Co-encadrant
M. Michel CRUCIANU	Professeur, Cnam	Directeur de thèse
M. Pierre HELLIER	Professeur, Université de Rennes	Invité

T H È S E

Remerciements

Je tiens à exprimer toute ma gratitude à toutes les personnes qui m'ont apporté leur soutien et qui ont contribué à faire de cette thèse une expérience enrichissante et plaisante.

Tout d'abord, un grand merci à mes encadrants de thèse: Michel Crucianu, Nicolas Audebert et Hervé Le Borgne. Je vous remercie sincèrement pour votre engagement constant, votre attitude positive en toutes circonstances, vos conseils techniques et méthodologiques précieux. Je vous suis très reconnaissante d'avoir toujours été présents tout en m'encourageant à faire mes propres choix. Je n'aurai pu espérer meilleur accompagnement pour cette entrée dans le monde de la recherche.

Un grand merci aux membres du jury : Andrés Almansa, David Picard, Céline Hudelot, Vicky Kalogeiton, Frédéric Jurie et Pierre Hellier, merci d'avoir accepté d'évaluer mon travail, merci pour votre temps et la discussion riche lors de la soutenance.

Je souhaite ensuite remercier tous mes collègues du Cnam : Yannis Karmim, Elias Ramzi, Marc Lafon, Loic Themyr, Wafa Aissa, Laura Calem, Georges Le Bellier, Maxime Mérizette, Léo Géré, Armand Verstraete et celles et ceux qui viennent d'arriver. Pendant trois ans, cela a été un plaisir de venir au bureau à chaque fois et de discuter avec des gens ouverts, intelligents et drôles. Yannis, c'était un réel plaisir de partager un bureau avec toi. Enfin, je suis ravie de constater que les nouvelles arrivées sont principalement féminines, cela me donne de l'espoir pour ce milieu très masculin.

Enfin un immense merci à mes amis et à ma famille. Merci à ma bande de toujours : Sami, Lucie, Clara, Ines, Léa, Maxime pour leur soutien indéfectible et pour toutes les sorties. Merci à ma grande soeur Alice de m'avoir montré l'exemple et à mes parents pour leur soutien à tous les niveaux et les escapades qui m'ont redonné du souffle. Merci de me dire tout haut et souvent que vous êtes fières de mon parcours car cela a toujours été un grand moteur pour moi.

Abstract

Generative modelling applied to visual content has recently made significant progress. Generative models now generate high-resolution, high-quality and diverse images. For various applications such as data augmentation, controlling the properties of the synthesized images is highly desirable to tailor the generated images for a specific downstream task and improve the diversity of training datasets. This thesis aims at defining means to improve this control and exploring how to use it for generating effective and diversified data augmentation.

We first investigate two limitations that arise when controlling the generated images by exploiting the latent space of pre-trained generative models. These models implicitly learn semantic representations of the images that can be manipulated with semantic directions to edit specific semantic attributes. However, the semantic directions commonly affect multiple attributes at once due to biases inherited by the pre-trained models. We demonstrate that this entanglement can be mitigated without post-processing by learning the directions on de-biased datasets. We also explore explicitly-enforced disentanglement with pre-trained attribute classifiers. We expose that the dependence on these models, that are known to lack robustness, can lead to unrealistic edited images. We thus introduce an alternative approach using the optimal transport framework. This framework allows to maintain closeness with real images and the optimality criterion can be leveraged to enforce disentanglement. We demonstrate competitive performances with a classifier-based approach while ensuring more reliability.

Then, we explore the use of large pre-trained models conditioned on text to synthesize training datasets. We demonstrate that text control might be insufficient for tasks that require compositionality. We thus propose to add a task-specific conditioning to adapt these models to generate precise augmentations suitable for supervised learning. We also introduce a strategy to diversify the augmentations by exploiting the dual conditioning - task-specific and text - prompting the generative model with novel but plausible pairs. We apply our method to the task of few-shot object counting

and demonstrate that our augmentations result in enhancing the performances of counting networks.

Keywords: Generative models, Controlled Synthesis, Image Editing, Data Augmentation.

Résumé

La modélisation générative appliquée au contenu visuel a récemment fait des progrès significatifs. Les modèles génératifs génèrent maintenant des images haute résolution, de haute qualité et diversifiées. Pour de nombreuses applications telles que l'augmentation de données, il est souhaitable de contrôler les propriétés des images synthétisées afin de personnaliser les images générées pour une tâche spécifique et d'améliorer la diversité des ensembles de données d'entraînement. Cette thèse vise à définir des moyens d'améliorer ce contrôle et à explorer comment l'utiliser pour générer une augmentation de données efficace et diversifiée.

Nous examinons d'abord deux limitations qui se posent lors du contrôle des images générées en exploitant l'espace latent de modèles génératifs pré-entraînés. Ces modèles apprennent implicitement des représentations sémantiques des images qui peuvent être manipulées avec des directions sémantiques pour éditer des attributs sémantiques spécifiques. Cependant, les directions sémantiques affectent couramment plusieurs attributs à la fois en raison des biais hérités des modèles pré-entraînés. Nous démontrons que cet emmêlement peut être atténué sans post-traitement en apprenant les directions sur des ensembles de données débiaisés. Nous explorons également le désemmêlement explicite à l'aide de classifieurs d'attributs pré-entraînés. Nous montrons que la dépendance à l'égard de ces modèles, connus pour leur manque de robustesse, peut conduire à des images éditées irréalistes. Nous introduisons donc une approche alternative utilisant le cadre du transport optimal. Ce cadre permet de maintenir une proximité avec les images réelles et le critère d'optimalité peut être utilisé pour imposer le désemmêlement. Nous démontrons des performances compétitives par rapport à une approche basée sur des classifieurs, tout en assurant une plus grande fiabilité.

Ensuite, nous explorons l'utilisation de modèles pré-entraînés sur des grandes bases de données et conditionnés par du texte pour synthétiser des ensembles de données d'entraînement. Nous montrons que le contrôle par texte peut être insuffisant pour les tâches nécessitant de la compositionnalité.

Nous proposons donc d'ajouter une condition spécifique à la tâche pour adapter ces modèles afin de générer des augmentations précises adaptées à l'apprentissage supervisé. Nous proposons également une stratégie pour diversifier les augmentations en exploitant la double condition - spécifique à la tâche et texte - guidant le modèle génératif avec des paires nouvelles mais plausibles. Nous appliquons notre méthode à la tâche du comptage d'objets à partir de quelques exemples et démontrons que nos augmentations améliorent les performances des réseaux de comptage.

Mots-clés : Modèles génératifs, Synthèse contrôléee, Édition d'images, Augmentation de données.

Contents

Acknowledgements **v**

Abstract **vii**

Résumé **x**

List of Tables **xvii**

List of Figures **xix**

Chapters

1 Introduction **1**

1.1 Motivation 2

1.2 Research Questions 3

1.2.1 Image Editing 3

1.2.2 Data Augmentation 6

1.3 Contributions 7

2 Controlled Image Synthesis with Generative Models **9**

2.1 Generative Models 10

2.1.1 Generative Adversarial Networks 10

2.1.2 Diffusion Models 15

2.2 Controlled Image Generation 17

2.2.1 Conditional Models 18

2.2.2 Latent Space Manipulation 21

2.2.3 Image Inversion 25
2.3 Generative Data Augmentation 27

3 Disentangled Image Editing with Pre-trained GANs 29
3.1 Introduction 30
3.2 Related work 31
3.3 Balanced Sampling for Disentangled Editing 34
3.3.1 Multi-Attribute Balanced Sampling 34
3.3.2 Semantic Directions Estimation 36
3.4 Experiments 37
3.4.1 Experiments Setup 37
3.4.2 Disentanglement 39
3.4.3 Identity Preservation 45
3.4.4 Orthogonality 46
3.5 Conclusion 46

4 Robust Image Editing with Pre-trained GANs 48
4.1 Introduction 49
4.2 Related work 50
4.3 Wasserstein Guidance for GAN Editing 52
4.3.1 Overview 52
4.3.2 Wasserstein Distance 52
4.3.3 Core Objective 54
4.3.4 Disentanglement Objective 55
4.4 Experiments 55
4.4.1 Experiments Setup 56
4.4.2 Main Results 58
4.4.3 Ablation: Explicit Disentanglement 59
4.4.4 Application to Count Editing 61
4.5 Conclusion 62

5 Data Augmentation with Text-to-Image Diffusion Models 65

5.1 Introduction . . . 66
5.2 Few-shot Object Counting . . . 67
5.3 Semantic Generative Augmentations . . . 70
5.3.1 Text-and-Density Guided Augmentations . . . 70
5.3.2 Diversity-Enhanced Augmentations . . . 71
5.4 Experiments . . . 73
5.4.1 Experiments Setup . . . 73
5.4.2 Few-shot Counting on FSC147 . . . 74
5.4.3 Ablation Study . . . 77
5.4.4 Counting with Fewer Shots . . . 81
5.4.5 Generalization on CARPK . . . 82
5.5 Qualitative results . . . 84
5.6 Conclusion . . . 87

6 Conclusion **88**
6.1 Summary of contributions . . . 88
6.2 Perspectives . . . 90
6.2.1 On-going work . . . 90
6.2.2 Further perspectives . . . 90
6.3 Broader Impacts . . . 92

Bibliography **93**

Resumé en Français **112**

Appendices

A Attribute correlations in CelebA **122**

B Additional experiments on Multi-digit MNIST and CLEVR **124**
B.1 Additional experiments on Multi-digit MNIST . . . 124
B.2 GAN Inversion on CLEVR . . . 128

Acronyms **129**

List of Tables

3.1 Performances of the attribute prediction model for each attribute. 38

3.2 Contingency table for the dataset generated with PGGAN CelebAHQ. 39

3.3 Re-scoring results for PGGAN. For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. We highlight (in **bold** for $\Delta\mathbf{e}$, *underlined* for $\Delta\mathbf{r}$) the best results among the disentangling approaches (IfGAN$^{\perp}$, Ours). 40

3.4 Re-scoring results for StyleGAN models in $\mathcal{Z}$ (top) and $\mathcal{W}$ (bottom). For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. In $\mathcal{Z}$, we highlight the best results among the disentangling approaches (IfGAN$^{\perp}$, Ours). 41

3.5 Re-scoring results for a higher sample size $N_0 = 10K$ for different models: PGGAN, StyleGAN3 in $\mathcal{Z}$ and in $\mathcal{W}$, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect, (*) indicates over-sampling. 44

3.6 Re-scoring results for different boundary calculation methods (given a balanced sample) for different models: PGGAN, StyleGAN3 in $\mathcal{Z}$ and $\mathcal{W}$, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. 44

3.7 ID preservation results (higher the better) for StyleGAN models. 45

3.8 Re-scoring results for PGGAN and StyleGAN3 in $\mathcal{Z}$. For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. 46

3.9 Cosine similarities between directions. 46

4.1 Quantitative results for attributes 'gender' (G), 'age' (A) and 'pale skin' (PS). We compare the classification approach (LT) with our Wasserstein approach (LW). Setting (*) is the "core" method, w/o any regularization. 59

4.2 Quantitative results for the manipulations "adding one digit in an image containing n digits, for $n = 1, 2, 3$" in real images from MultiMNIST [135]. Given a change rate of 100% according to a latent classifier, we report the *actual* change rate as measured by an image classifier. Higher values indicate a lower rate of adversarial samples. 62

5.1 Quantitative results on FSC147. (*) Traditional augmentations include color jitter, random cropping. (†) [158] and [14] are reproduced. 76

5.2 Quantitative results for SAFECount: Test counting accuracy (3-shot) per range of number of objects for SAFECount [158] on FSC147. 78

5.3 Quantitative results: 3-shot and 1-shot evaluation for SAFECount [158] on FSC147. . 82

5.4 Quantitative results: 3-shot and 0-shot evaluation for CounTR [14] on FSC147. Top: 3-shot training. Bottom: [0,3]-shots training. 82

5.5 Counting performance on CARPK with SAFECount. 83

List of Figures

1.1 Art generated with AI 2
1.2 AI tool for clothes designing 3
1.3 Semantic editing in GANs latent space 4
1.4 Limitation 1: Entanglement. 5
1.5 Limitation 2: Unrealistic outputs. 5
1.6 Abilities and limitations of large text-to-image models 6

2.1 Resolution and quality of images generated by GANs from 2014 to 2020. 10
2.2 Architecture of the generator network in DCGAN. 11
2.3 Traditional generator architecture vs. style-based architecture (StyleGAN). 13
2.4 Style mixing with StyleGAN. 14
2.5 Overview of DDPMs 16
2.6 Image-to-image translation. 19
2.7 Text-to-image generation. 20
2.8 Latent space manipulation 21
2.9 Editing results for GANSpace 22
2.10 Disentanglement property 23
2.11 Localized editing with latent space manipulation 25

3.1 Extracting a semantic direction using the decision boundary of a binary classifier. (a) **n** is the vector orthogonal to the decision boundary (dotted line) of a 'glasses' latent classifier. (b) Translating the latent code with **n** leads to adding glasses that become more visible as the magnitude of the translation increases. Figure taken from [127]. . . 31

3.2 Entanglement analysis for PGGAN trained on CelebAHQ. (a) Correlation matrix computed on the GAN training set. (b) Correlation matrix computed on the GAN-generated set. (c) Cosine similarities between attribute directions. Bottom: For each example, the original image is displayed to the left and 3 edited images obtained with translations of successively increased magnitudes are shown to the right. We observe that 'age' is entangled with 'gender' and 'eyeglasses' with 'age'. Figure taken from [127]. 32
3.3 Conditional manipulation. Figure taken from [127]. 33
3.4 Joint distributions for 3 binary facial attributes Age ('O': Old, 'Y': Young), Gender ('M': Male, 'F': Female) and Smile ('S': Smile, 'NS': No Smile). In (a), the positive set contains a majority of *old males* while the negative set contains a majority of *young females*, leading to bias the direction 'glasses' toward the attributes 'age' and 'gender'. 35
3.5 Editing results for PGGAN for attributes Glasses, Gender and Age. 40
3.6 Editing results for StyleGAN2 for attributes Glasses, Gender, Smile and Age. 42
3.7 Results for StyleGAN3 (rare attributes). (a) For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. We highlight the best results among disentangling methods (IfGAN$^{\perp}$, Ours). 'Pale skin' is balanced w.r.t. 'gender' and 'age', 'wavy hair' w.r.t. to 'gender' and 'narrow eyes' w.r.t. 'smile'. 42
3.8 Editing results for StyleGAN3 in $\mathcal{W}$ for attributes Glasses, Gender and Age. 43
3.9 Identity preservation is negatively correlated with the effect of a direction. Above each figure is indicated the moving step α and the identity preservation result (id_p). 45

4.1 Limitations of classifier-based guidance as employed in the method of Yao et al. [157]. Qualitative results for two editing tasks: (a) when attempting to change the gender on FFHQ, the obtained images are unrealistic. (b) when attempting to add one digit in a Multi-digit MNIST image, the number of digits is unchanged in the edited image. . . . 50
4.2 Classifier-based guidance for learning semantic edits. Various approaches such as GuidedStyle [51] employ an attribute classifier (or attribute knowledge network) with a classification loss as supervision to learn latent transformations. Figure taken from [51]. 51

4.3 Method overview. For each semantic attribute a_k (*e.g.* 'glasses') we learn a mapping $\mathbf{H}_k$ that moves the distribution of latent codes lacking the attribute to the distribution of codes having that attribute. We enforce that each latent code is moved near a point that shares similar semantics, thus only changing that attribute. To preserve identity, the resulting distribution does not entirely match the target distribution. 53

4.4 Qualitative results for facial attribute editing. We report the editing results for $\alpha = \pm 2$. We observe that our approach better preserves identity and some facial attributes (*e.g.* expression, absence of makeup) compared to Latent Transformer. 57

4.5 Quantitative results for facial attribute editing. We report the attribute preservation rate (computed on all other attributes indicated here) and the identity preservation rate for different values of α (points of the curves). The x-axis is the ratio of images (among all test images) for which the target attribute is successfully modified. 58

4.6 Qualitative comparison between classifier-based edits (2^{nd} col.) and our Wasserstein-based edits w/o any reg. (3^{nd} col.) and w/ disentanglement reg. (4^{th} col.). 60

4.7 Results for 'gender' editing without the $L2$-regularization on the edited codes for LT. . 61

4.8 Qualitative editing results for MultiMNIST [135]. We show three examples per line. For each example, the first column corresponds to the unedited image. The second column corresponds to the edited image ($\alpha = 1$) with our method. We add one number each time, starting from one (top line), two (middle) or three (bottom) digit(s). 63

5.1 Few-shot Object Couting. FSC networks take as input the target image and few exemplars of the current type of objects to count (in red). Typically, the similarity map between the query image and exemplars features is computed and fed to a convolutional neural network that outputs a density map [114, 128]. This map is a spatial map indicating the probabilities of an object being present, whose sum is the predicted count. The supervision of FSC networks is usually achieved using a MSE loss between the predicted maps and the ground-truth ones. 67

5.2 Overview of SAFECount. Figure taken from [158]. 68

5.3 Overview of CounTR. Figure taken from [14]. 69

5.4 Overview of our approach. We condition a pre-trained diffusion model on both text prompts and density maps and perform swaps with similar captions. The density and original exemplars boxes are used as ground-truth for the generated augmentation. . 72
5.5 Qualitative results for the Baseline vs. Diverse augmentations. At the bottom of each diverse sample we show the caption used to generate the image. Our strategy allows to diversify the type of objects and/or the background. 75
5.6 Qualitative comparison with Real Guidance [45]. Our augmentations preserve the layout while creating more diverse backgrounds. Ground-truth density maps overlap with the generated images (last 2 columns). 77
5.7 Qualitative counting results on FSC147 test images. We compare the model trained with Real Guidance's augmentations (2^{nd} column) vs. our augmentations (3^{rd} column) for images with a high number of objects. Predicted and ground-truth density maps are overlapped with the images. 79
5.8 Distribution of object categories in the set of captions to swap from w.r.t t_c for a sample of category "bread rolls". Lower thresholds result in more diverse augmentations, while objects still belong to similar classes. 80
5.9 Impact of caption similarity threshold t_c (left) and percentage of diverse samples p_c (right) on SAFECount. MAE is reported for the val and test sets of FSC147. 80
5.10 Impact of number of augmentations threshold M (left) and percentage of synthetic data p_0 (right) on SAFECount. MAE is reported for the val and test sets of FSC147. 80
5.11 Caption swap at random (top) vs. similarity-based swap (bottom, $t_c = 0.7$). Random swapping results in a mismatch between the layout and the semantics. 81
5.12 Qualitative results of synthetic augmentations of FSC147. We compare our Baseline vs. Diverse augmentations. 84
5.13 Qualitative results of synthetic augmentations of FSC147. We compare our Baseline vs. Diverse augmentations. 85
5.14 Limitation: our diverse generation strategy can change the size and shape of generated objects, leading to exemplar boxes (in red) that do not fit perfectly. 86

6.1 Schwartz et al. introduce an approach to learn class-specific tokens to generate fine-grained details using a classifier. Figure taken from [125]. 91

A.1 List of attributes in CelebA [83]. 122
A.2 Correlation matrix between CelebA attributes. 123

B.1 Reconstruction of 128 × 128 CLEVR images with e4e [140]. Left: real image, Right: reconstructed image. 128

Chapter 1

Introduction

Contents

1.1 Motivation **2**
1.2 Research Questions **3**
1.2.1 Image Editing 3
1.2.2 Data Augmentation 6
1.3 Contributions **7**

Figure 1.1: Some paintings generated with the help of AI from Refik Anadol's exhibition (Unsupervised - Machine Hallucinations) held at the MoMA in 2023 (source: `refikanadolstudio.com`).

1.1 Motivation

Generative Artificial Intelligence (GenAI) and its application to visual content is a promising tool for artists. A striking example is the recent exhibition held at the MoMA, a globally acclaimed art institution, of a series of artworks synthesized with the help of a generative model (see Fig. 1.1). GenAI may influence various other domains within the creative realm, such as fashion for clothes designing, cinema for VFX, video games for game asset generation. However, GenAI should be a collaborative tool and enable artists to have significant control to precisely align the generated content with their style or intentions. For instance, a clothes designer should be able to control the type of clothes, the style, the color and fabric and to finely edit each property as needed (see Fig. 1.2). Outside the creative industries, the development of generative tools with a range of independent controls to manipulate either a continuous, discrete or structured variable describing an image, could be of interest for various applications. Depending on the nature of the generated images, the control may concern the presence and nature of individual entities in the scene and their visual properties but also their position and orientation, the properties of the scene, the geometric properties of the camera, etc.

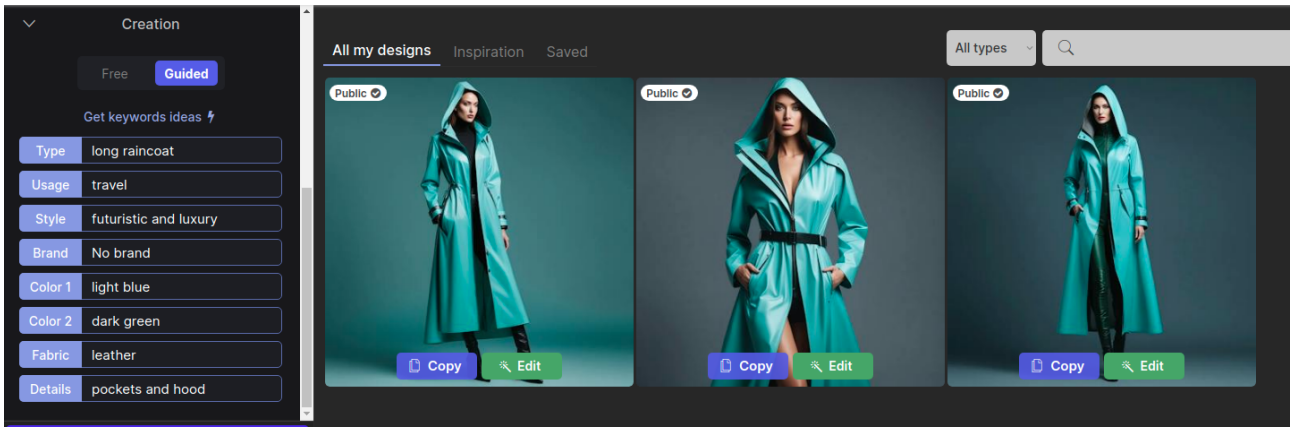


Figure 1.2: GenAI tool for designing clothes that allows the user to control various properties of the clothes: type, style, brand, color, fabric, as shown on the right (source: `thenewblack.ai`).

In the field of Computer Vision (CV), this topic could address an important problem that is an essential part of deep learning-based approaches: improving training datasets with reduced manual effort. Especially, it could improve the following aspects:

- *Labelling* by knowing the properties of generated images *a priori*,
- *Fairness* by generating images with scarcely represented properties,
- *Diversity* by generating images with unseen combinations of factors of variation or new domains,
- *Robustness* by generating images that are difficult to capture in practice (*e.g.* rare events, difficult weather conditions) or hard (*i.e.* where the models fail).

Therefore, in this thesis, we investigate the controllability of deep generative models. We study means to obtain refined control over the properties of generated images. We also explore the use of controllable generative models to generate training datasets with enhanced-diversity.

1.2 Research Questions

1.2.1 Image Editing

The year of 2014 marks a significant milestone for image generation with the introduction of Generative Adversarial Networks (GANs) [41]. Using two competing networks, these generative models showed the ability to generated images with an unprecedented quality but still low-resolution

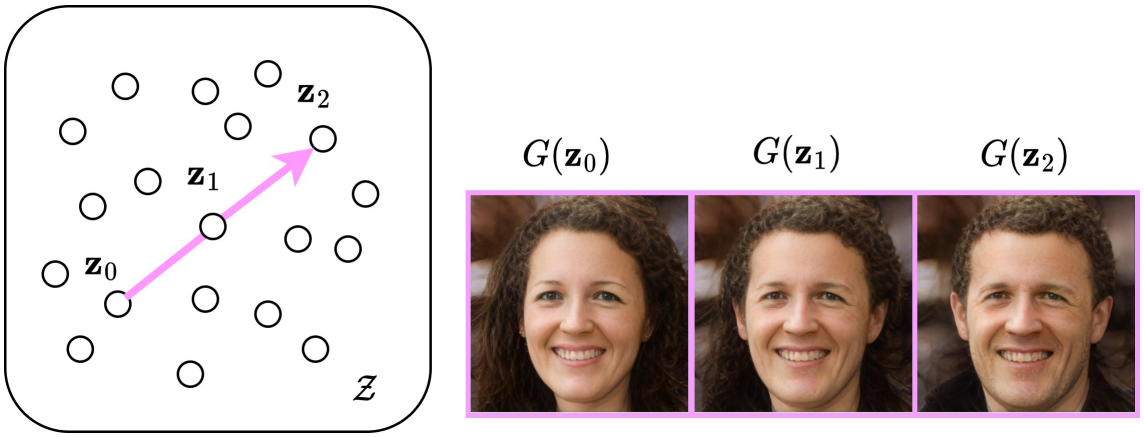


Figure 1.3: GANs generate images from a latent space ($\mathcal{Z}$) where $\mathbf{z} \in \mathcal{Z}$ is a latent code. An interesting property of pre-trained GANs is the existence of trajectories in the latent space (in pink) that correspond to variations of semantic attributes in image space (Female $\rightarrow$ Male here). This property results from GANs implicitly encoding the semantics of the training data during training.

and far from realistic. In 2020, StyleGAN [64] succeeded to generate high-resolution face images almost indistinguishable from real ones. However, controlling the properties of the generated faces *e.g.* synthesizing specifically a face presenting as a woman, is not straightforward as it requires to re-train the GAN with an additional condition as input [90]. Various methods thus focus on exploiting the semantic knowledge encoded in the latent space of pre-trained GANs. Especially, some works identify directions within this latent space corresponding to interpretable variations [42]. For restricted domains such as the face domain, the availability of pre-trained image attribute classifiers allows to learn directions corresponding to specific semantic attributes such as the presence of glasses, gender, age, smile [127]. Moving an existing latent code along one direction results in continuously controlling the attribute in the generated image (cf. Fig. 1.3). This ability thus makes pre-trained GANs natural tools to generate images with a fine control on their semantic properties. However, the learned semantic directions may produce latent codes generating images with undesirable modifications or strong artifacts, as detailed below.

Limitation 1: Entanglement. Pre-trained GANs generally encode various biases from their training data, which leads to learning directions that control various attributes simultaneously instead of solely altering the targeted attribute. For instance, in Fig. 1.4, the direction 'female face to male face' affects the smile (it fades away) in addition to the gender.

Figure 1.4: Limitation 1: Entanglement. A direction encoding a given semantic attribute (gender: Female → Male) may lead to the modification of another attribute (smile: Smile → No Smile). The leftmost image is the unedited image. The two subsequent images correspond to edited images with increased editing strength as we go to the right.

Limitation 2: Unrealistic outputs. To learn latent semantic directions, various supervised approaches rely on a classification framework [171, 157, 51]. In particular, learning the direction 'female face to male face' is formulated as learning the direction that leads to generate faces classified as male faces. However, the guidance of classifiers is unreliable, as, for instance, they might associate high confidence to regions far from the training data [93]. The directions learned with these methods can thus steer edited codes to regions unknown to the generator, leading to unrealistic outputs, as shown in Fig. 1.5.

Figure 1.5: Limitation 2: Unrealistic outputs. A direction encoding a given attribute (gender: Female → Male) produces unnatural images before reaching the desired editing. The leftmost image is the unedited image. The two subsequent images correspond to edited images with increased editing strength as we go to the right.

Addressing the previous limitations is crucial to achieve accurate and robust control. If these properties are ensured, we can use the editing techniques to generate images that can later serve as training data. To create synthetic datasets for various tasks, the generative model should also be able

"A bread roll with a smiley face" "10 bread rolls" "A regular bread roll next to a bread roll with a smiley face"

Figure 1.6: Left: Zero-shot capabilities, Middle: Count. Right: Spatial Understanding. Recent pre-trained text-to-image diffusion models show impressive zero-shot capabilities with the generation of never seen images (Left). However, they struggle to correctly follow prompts that involve count (Middle) or spatial relationship between objects (Right).

to generate a wide range of distributions. Although GANs excel at generating narrow distributions featuring a single type of objects (*e.g.* faces, cars), they struggle to model distributions composed of more diverse images with various objects or complex backgrounds such as ImageNet [22].

1.2.2 Data Augmentation

Denoising Probabilistic Diffusion Models (DDPMs) [131, 133, 50] were introduced after GANs and palliate some of their limitations. The training process of DDPMs is more stable and leads to a better mode coverage of the training distributions. They can thus generate distributions with a high diversity. Various recent research has focused on designing and training DDPMs conditioned on textual inputs [117, 120, 95, 111, 112], leveraging millions of images and texts collected from the internet. With this conditioning, the users can control the generated images in an intuitive and flexible way by formulating their intentions in the form of text. Due to the large-scale training, these models can generate a wide range of images and showcase impressive zero-shot capabilities (cf. left example in Fig. 1.6). Recent works use these text-to-image pre-trained models as off-the-shelf tools to augment various classification datasets, by simply prompting the models with the class labels [45, 141]. Yet, these methods are not directly applicable to tasks beyond classification, in particular tasks involving compositionality such as object detection and counting, or semantic segmentation. Indeed, despite

their expressiveness, these models struggle to follow prompts that include pairwise positional relations or a specific number of objects [99] (cf. middle and right examples in Fig. 1.6).

1.3 Contributions

In this thesis, we first tackle improving the control over the generated images. We focus on editing semantic attributes using the latent space of pre-trained GANs. We address disentanglement (Chapters 3 and 4) and robustness to avoid unrealistic outputs (Chapter 4). Then, we explore how to adapt large pre-trained models controlled by text to generate synthetic datasets with enhanced diversity for tasks that require more complex generation and annotations than classification (Chapter 5).

- Chapter 3: Disentangled Image Editing with Pre-trained GANs

  In this chapter, we explore how to extract disentangled controls from the latent space of GANs. Supervised approaches typically identify the controls by sampling and annotating a collection of latent codes, then training classifiers to extract the directions. Since the GAN-generated data reflects the biases of the original training set, so do the resulting semantic controls. We propose a simple approach that consists in balancing the semantics of the training data before training the classifiers. We demonstrate the effectiveness of this approach by extracting disentangled directions for face manipulation on various GAN architectures and two datasets.

  The work of this chapter led to the following publication:

  - Perla Doubinsky, Nicolas Audebert, Michel Crucianu, Hervé Le Borgne. "Multi-attribute Balanced Sampling for Disentangled GAN Controls". In: *Pattern Recognition Letters*, 2022.

- Chapter 4: Robust Image Editing with Pre-trained GANs

  In this chapter, we investigate an alternative to classifier guidance to learn the semantic controls. Due to their lack of robustness, these models might steer the generation towards out-of-distribution regions of the latent space, leading to unrealistic edited images. We propose to use the optimal transport framework to learn effective edits while enforcing closeness with in-distribution latent codes. We employ the guidance of the Wasserstein loss to minimize the distance between the distribution of edited codes and the distribution of codes having the desired semantic attribute. We also design an optional loss that can be added to enforce disentanglement.

We evaluate our method on two datasets (digits and faces) using a state-of-the-art GAN architecture and show competitive performances with a classifier-based approach.

The work of this chapter led to the following publication:

- Perla Doubinsky, Nicolas Audebert, Michel Crucianu, Hervé Le Borgne. "Wasserstein Loss for Semantic Editing in the Latent Space of GANs". In: *20th International Conference Content-Based Multimedia Indexing (CBMI)*, 2023.

- Chapter 5: Data Augmentation with Text-to-Image Diffusion Models

  In this chapter, we tackle the adaptation of pre-trained text-to-image diffusion models to generate diversified and annotated training datasets for tasks that require precise generation and labels. Since these models struggle with compositionality, we propose finetuning with an additional task-specific condition. To address the diversity of generated images, we take advantage of the double conditioning - text and task-specific. Especially, we vary the textual prompts associated with a given task-specific condition by exploiting a set of captions generated by a captioning model. We apply our approach for the generation of training images for the task of few-shot object counting, using counting density maps as the specific condition. We demonstrate improved performances for two counting networks trained on a mix of real and generated data.

  The work of this chapter led to the following publication:

  - Perla Doubinsky, Nicolas Audebert, Michel Crucianu, Hervé Le Borgne. "Semantic Generative Augmentations for Few-shot Counting". In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

Chapter 2

Controlled Image Synthesis with Generative Models

Contents

2.1 Generative Models **10**
2.1.1 Generative Adversarial Networks 10
2.1.2 Diffusion Models 15
2.2 Controlled Image Generation **17**
2.2.1 Conditional Models 18
2.2.2 Latent Space Manipulation 21
2.2.3 Image Inversion 25
2.3 Generative Data Augmentation **27**

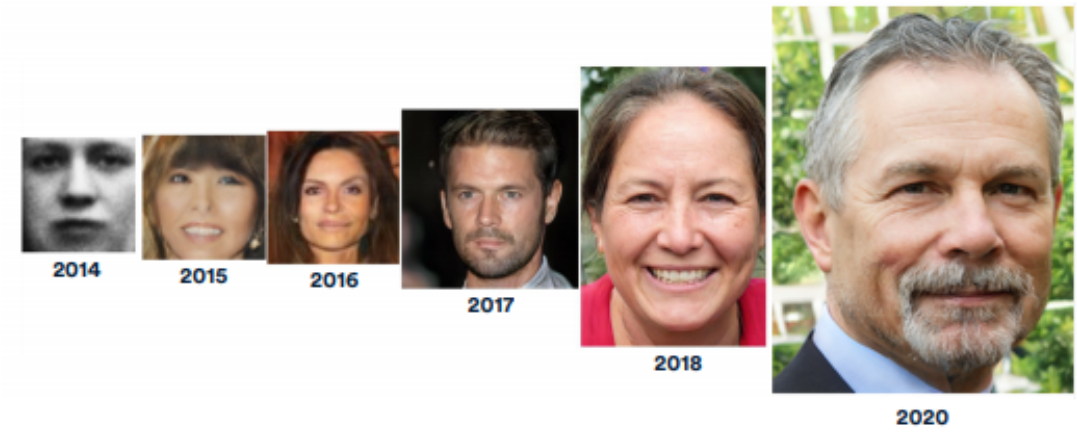


Figure 2.1: Evolution of the resolution and quality of images generated by GANs from 2014 to 2020. Figure taken from Stanford AI Index Report 2021 (`https://dev-hai-aiindex.pantheonsite.io/ai-index-report-2021`).

In this chapter, we present two prominent families of generative models for the task of image generation. Then, we delve into various approaches built upon these models, enabling the synthesis of images with specific semantic properties or the modification of existing properties within a given image. Lastly, we present methods that employ generated images and control methods to enhance and complement real training datasets for different tasks.

2.1 Generative Models

Generative models learn to model a distribution of data samples *e.g.* images, audio samples, then allowing to generate new samples from that distribution. There are various types of generative models such as Variational Auto Encoders (VAEs) [70], Generative Adversarial Networks (GANs) [41], flow-based models [27, 28, 69], auto-regressive models [143, 142, 144, 31], Denoising Probabilistic Diffusion Models (DDPMs) [131, 49, 25]. For image generation, GANs achieve the best image quality and diversity with DDPMs recently rivalling the performances of these models. We thus focus on these two types of models.

2.1.1 Generative Adversarial Networks

GANs were first introduced by Goodfellow et al. [41] in 2014 and have rapidly evolved to synthesize high-resolution and photorealistic images (cf. Fig. 2.1). Their training is built on an adversarial

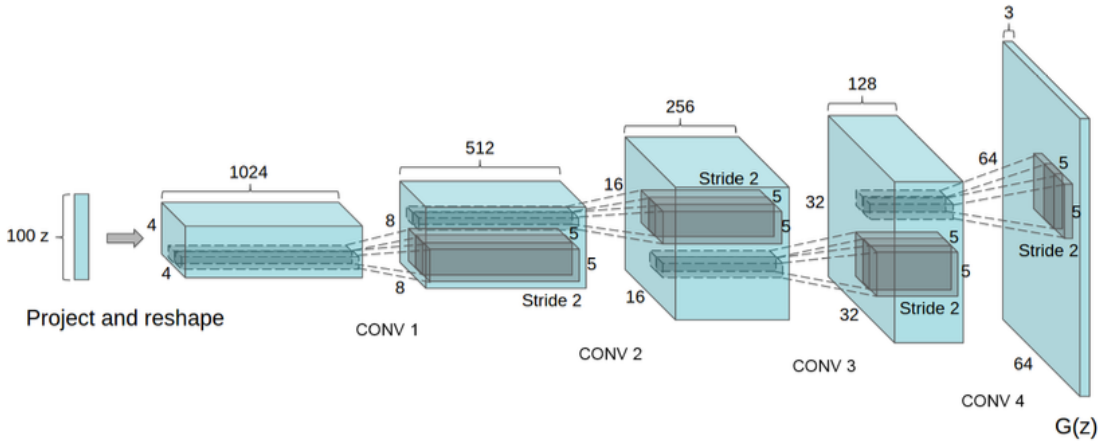


Figure 2.2: Architecture of the generator network in DCGAN. Figure taken from [108].

game between two networks: the discriminator D and the generator G. The generator learns to synthesize data samples from latent codes $\mathbf{z} \sim p(\mathbf{z})$ where the latent space is a d-dimensional space usually modelled by a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I}_d)$. The goal of the discriminator is to differentiate between real samples $\mathbf{x} \sim p(\mathbf{x})$ and the fake samples $G(\mathbf{z})$ synthesized by the generator. The discriminator task thus consists in binary classification. Simultaneously, the goal of the generator is to generate realistic images, hence mis-classified as real by the discriminator. The two antagonistic goals are formulated as the following min-max objective:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[1 - \log D(G(\mathbf{z}))] \tag{2.1}$$

Mirza and Osindero [90] later introduced conditional Generative Adversarial Networks (cGANs) that allow to model conditional distributions to generate images that conform to a specific condition *e.g.* a class label, an image, text. The cGAN formulation is similar to Eq. (2.1) where the input condition $\mathbf{c}$ is fed to both the generator and discriminator, usually via concatenation:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{c})}[\log D(\mathbf{x}, \mathbf{c})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[1 - \log D(G(\mathbf{z}, \mathbf{c}), \mathbf{c})] \tag{2.2}$$

Typical GAN architectures follow DCGAN [108] where both networks are modelled using Convolutional Neural Networks (CNNs). The generator is generally made up of various transposed convolutional layers that progressively upsample 3D-reshaped latent vectors into images (cf. Fig. 2.2).

The discriminator usually mirrors the generator with various convolutional layers that downsample the images into compact representations. The authors of DCGAN also made an interesting discovery regarding the properties of the latent representations. They found that simple arithmetics *e.g.* addition, subtraction lead to the modification of semantic qualities in the generated samples. This study suggested a semantically rich linear structure in GANs latent space that was later exploited by many works to control meaningful characteristics in the generated images, as detailed in Section 2.2.2.

The resolution of images generated by DCGAN is limited (*e.g.* 64×64, 128×128) thus subsequent works focused on improving this architecture to generate higher-resolution images. Progressive Growing GAN (PGGAN) [63] introduces an architecture that progressively grows as the training goes on. The idea is to start generating low-resolution images then gradually increase the resolution and the number of layers accordingly. This breaks the learning into multiple but easier steps thus improving training stability and quality of the generated images. The authors of BigGAN [12] address the resolution issue by increasing the number of layers and units. Later, Karras et al. [64] proposed StyleGAN with a different type of architecture for the generator inspired by style transfer [38]. This architecture has significantly contributed to the success of GANs establishing the state-of-the-art regarding the quality and diversity of generated images and allowing more fine-grained control over the properties of generated images. We present StyleGAN in more detail in the following.

StyleGAN The first novelty is the introduction of an intermediate latent space. In previous GANs, there is a unique latent space $\mathcal{Z}$ modelled by a fixed standard Gaussian distribution. In StyleGAN, this traditional latent space is mapped to a new latent space $\mathcal{W}$ using a Multi-Layer Perceptron (MLP). Each latent vector $\mathbf{w} \in \mathcal{W}$ is then projected to a style code $\mathbf{y} = (\mathbf{y}_s, \mathbf{y}_b)$ via a learnable affine transform. The style code modifies the features maps after each convolutional layer through Adaptive Instance Normalization (AdaIN) layers [54] as follows:

$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \left(\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} \right) + \mathbf{y}_{b,i} \tag{2.3}$$

where $\mathbf{x}_i$ is the i^{th} feature map in the current layer and $dim(\mathbf{y}) = \#$ feature maps in that layer.

AdaIN is an operation that was introduced in the context of style transfer. Huang and Belongie [54] showed that the transfer can be achieved by normalizing the feature map of a content image (as in Eq. (2.3)) with the mean and variance of the feature map of the style image. The style modulation

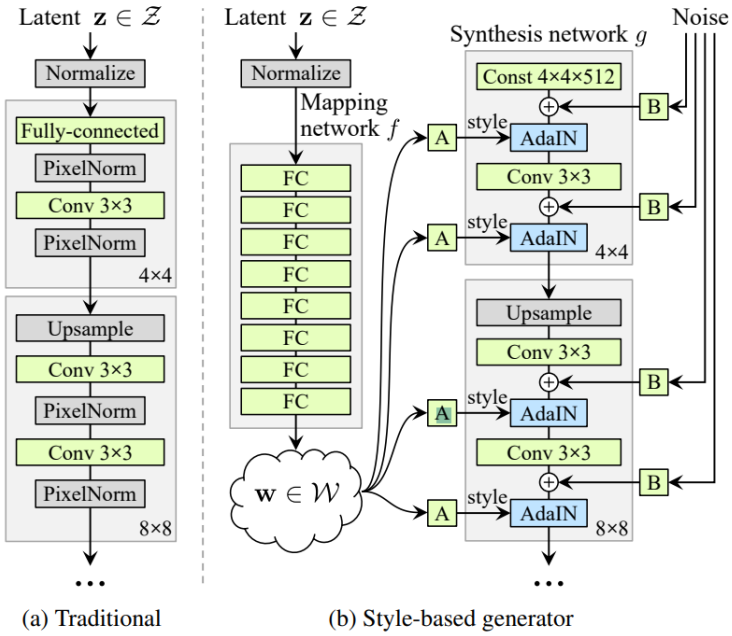


Figure 2.3: Comparison between the traditional generator architecture and the style-based architecture of StyleGAN. (b) StyleGAN introduces an intermediate latent space $\mathcal{W}$ where each code is projected via affine transforms (**A**) to a style code that modulates the feature maps through AdaIN layers. A per-pixel noise (**B**) is also added to the feature maps to generate stochastic variations in the images. Figure taken from [64].

is applied at each layer of the generator (with an independent affine transform) thus the intermediate latent code controls the "style" of generated images at various resolutions. Since earlier layers tend to generate coarse image features while finer details are controlled by later layers, each style code is expected to affect different aspects of the generated images. To further enforce this, Karras et al. [64] introduce a regularization that mixes two intermediate latent codes for some images during training. Fig. 2.4 indeed shows that modifying the styles of the coarse layers leads to altering rough features *e.g.* pose and face shape. For middle layers, it leads to changes in smaller facial features, while for fine layers, it mainly affects the color palette and lighting.

Another interesting property is that the intermediate latent space of StyleGAN has a better semantic organization than $\mathcal{Z}$ space. As first observed in DCGAN, the latter implicitly encodes the semantics of the training data. A common goal of later GANs is thus to have disentangled latent spaces - where semantic factors are linearly organized and well-separated. Karras et al. introduce two metrics to evaluate the disentanglement of the latent spaces: the Perceptual Path Length (PPL)

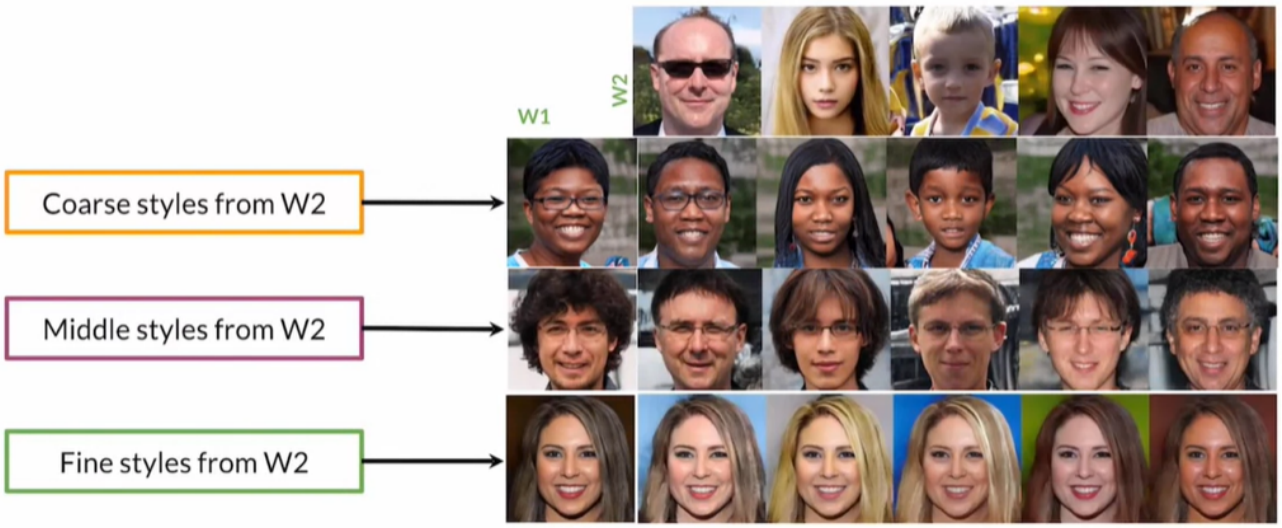


Figure 2.4: Effect of mixing two latent codes from the intermediate latent space of StyleGAN at different layers (source: `https://bit.ly/2TggMR8`).

metric that measures the changes that an image goes through when interpolating in latent space and the linear separability metric that measures how well a binary attribute can be separated in latent space. For both metrics, the intermediate latent space $\mathcal{W}$ outperforms the fixed latent space $\mathcal{Z}$. We discuss disentanglement in more details in Section 2.2.2.

To sum up, in contrast to other GAN architectures, StyleGAN has two latent spaces: the original $\mathcal{Z}$ space and the intermediate $\mathcal{W}$ space that is more disentangled. Abdal et al. also later proposed an extended version of $\mathcal{W}$ referred to as $\mathcal{W}+$ where the latent code $\mathbf{w}$ can be different at each layer of the generator (before the affine projection onto the respective style codes). As we'll see later, this extended latent space allows projecting real images in the latent space leading to more faithful reconstructions [1]. In addition to these latent spaces, there is also the style space S that corresponds to the style codes described previously. In Section 2.2.2, we introduce various methods that exploit these spaces to edit meaningful properties of generated and real images.

Following the first version of StyleGAN, various improvements have been introduced. StyleGAN2 [65] replaces AdaIN layers with weight demodulation to avoid droplet artifacts on the generated images and adds PPL regularization during training. More recently, StyleGAN3 [66] addresses the unnatural transitions observed when interpolating with StyleGAN2, with small architectural changes to make the generator equivariant to translation and rotation.

GAN Limitations Despite their success, GANs have various limitations. First, GANs are known for their training instability inherited from the adversarial formulation. To optimize Eq. (2.1) in practice, the discriminator is trained alternatively with the generator until reaching an equilibrium. Often, the discriminator becomes too strong at the beginning of the training thus preventing the generator from improving, leading to a collapsed training. GANs are also sensitive to 'mode collapse' that refers to the generator only producing one sample or similar-looking samples, ignoring the full diversity of the data. More generally, GANs ignore the rarer modes of the data as these samples might be classified as not realistic by the discriminator. Some approaches such as WGAN [7], that employs the Wasserstein distance between the real data distribution and the generated distribution instead of the original objective, alleviate this issue. However, GANs still struggle to model complex distributions such as ImageNet [22], in particular in the unconditional setting. Most GAN models are limited to modelling narrow distributions *e.g.* faces [83, 64], cars [73], bedrooms [159], which can limit their use in practice.

2.1.2 Diffusion Models

Denoising Probabilistic Diffusion Models (DDPMs) are a class of generative models that was introduced in 2015 [131] with a significant breakthrough in 2020 [50], later beating GANs on the generation of complex multimodal distributions [25]. These models tackle image generation using stochastic processes (cf. Fig. 2.5). Given an image sampled from a real data distribution, $\mathbf{x}_0 \sim q(\mathbf{x})$, the *forward* diffusion process consists in gradually adding Gaussian noise to the image in T steps:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2.4}$$

where the added noise depends on the variance schedule β_t. A nice property of the forward process is that $\mathbf{x}_t$ can be sampled at an arbitrary timestep t in closed form:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad \alpha_t = (1-\beta_t), \quad \bar{\alpha}_t = \Pi_{s=1}^{t}\alpha_s \tag{2.5}$$

When t becomes large, the sampled image follows an isotropic Gaussian distribution, $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. The idea of DDPMs is to learn the *reverse* diffusion process. Given a random noise $\mathbf{x}_T$, the goal is to predict the added noise at step $\mathbf{x}_{t-1}$, successively until retrieving a real image $\mathbf{x}_0$. Each transition is modelled as follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{2.6}$$

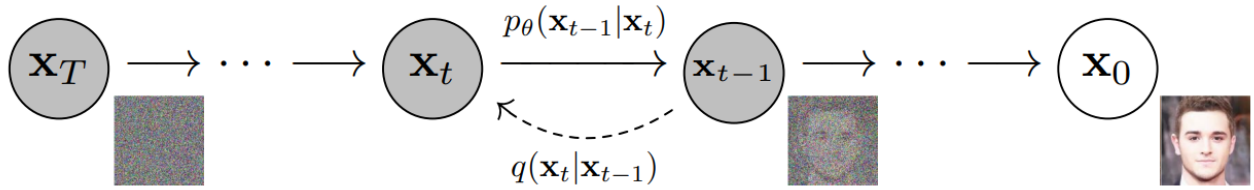


Figure 2.5: Overview of Denoising Diffusion Probabilistic Models (DDPMs). The forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ consists in iteratively adding noise to an input image. The reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_{t-1})$ is learned using a neural network trained to predict various levels of noise added to an image. At inference, it can be used to sample an image by iteratively denoising a Gaussian latent map ($\mathbf{x}_T$). Figure taken from [50].

where the variance parameters are fixed while the mean $\mu_\theta(\mathbf{x}_t, t)$ depends on a model $\epsilon_\theta(,)$ of the diffusion noise whose objective simplifies as the following denoising objective [50]:

$$\mathcal{L}_t = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[||\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)||^2\right], \quad \text{with } \epsilon_t \sim \mathcal{N}(0, \mathbf{I}) \tag{2.7}$$

The denoising network is usually a U-Net [118] that takes as input a noised image $\mathbf{x}_t$ concatenated to the timestep t and outputs the noise in the image.

Sampling To generate a new sample at inference, a latent map is sampled from the standard Gaussian distribution $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively denoised (T to 1) using the learned denoising network. In particular, Ho et al. [50] derive the following sampling equation:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{2.8}$$

Contrary to GANs, the sampling takes several minutes as it requires hundreds of denoising steps [50]. The latter is a significant limitation that might explain why the success arrived later for DDPMs than for GANs. Nichol and Dhariwal [94] address this issue by running a strided sampling schedule *i.e.* taking the sampling update every T/K step instead of every timestep. Song et al. [132] proposed Denoising Diffusion Implicit Models (DDIMs) that includes a deterministic sampling process that generates higher quality samples with fewer steps. More recently, Rombach et al. [117] introduced Latent Diffusion Models (LDMs) that are diffusion models applied in the latent space of generative models such as VQGAN [31]. These models operate in a compressed latent space thus reducing training and inference time. LDMs allows to generate high-quality images in only 20 diffusion steps that corresponds to approximately 3 seconds on a single GPU.

Conditional DDPMs Dhariwal and Nichol [25] use the guidance of an image classifier $p_\phi(\mathbf{c}|\mathbf{x})$ to improve the quality of the generated samples and achieve conditional sampling without retraining DDPMs. The idea is to guide the denoising process towards the desired condition $\mathbf{c}$ using the classifier's gradients $\nabla_{\mathbf{x}_t} \log p_\phi(\mathbf{c}|\mathbf{x}_t)$. This term is introduced in the sampling equation (cf. Eq. (2.8)) and controls the diversity ($\downarrow$) vs. quality ($\uparrow$) trade-off. This method requires to re-train the classifiers to obtain the gradients w.r.t. to noised images, which can be expensive. Ho and Salimans [48] later introduced classifier-free guidance that controls the trade-off without using classifiers but the generative model itself. They jointly train an unconditional and conditional diffusion model. DDPMs can model conditional distributions [119] by giving the condition $\mathbf{c}$ as input to the reverse process function:

$$\mathcal{L}_t = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[||\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})||^2\right], \quad \text{with } \epsilon_t \sim \mathcal{N}(0, \mathbf{I}) \tag{2.9}$$

To train both models jointly, Ho and Salimans randomly replace $\mathbf{c}$ by a non-informative condition *e.g.* an empty string ($\varnothing$) for text conditioning, during training. The sampling is then performed by a linear combination of the conditional and unconditional diffusion noise estimates:

$$\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) = (1 - s) \cdot \epsilon_\theta(\mathbf{x}_t, \varnothing) + s \cdot \epsilon_\theta(\mathbf{x}_t, \mathbf{c}) \tag{2.10}$$

where s is called the guidance scale. The unconditional model is recovered when $s = 0$. For $s = 1$, we have the standard conditional model. When $s > 1$, the diversity of the generated samples decreases but the quality and fitting to the condition increase.

Although DDPM sampling is expensive, these models have some advantages in comparison to GANs. The DDPM framework leads to more stable training and better coverage of the training distribution since it does not rely on an adversarial game. Recently, DDPMs trained on complex distributions such as ImageNet also outperformed GANs regarding the quality of the generated images [25, 48]. As we discuss later, these advantages have led the research community to favor this framework to train models on large multi-modal datasets.

2.2 Controlled Image Generation

We introduced two families of generative models: GANs and DDPMs. While sampling from the unconditional variants of these models yields high-resolution and photo-realistic images, the specific properties exhibited in these images are unknown beforehand. In this section, we delve into controlled

image synthesis and image editing using generative models - two types of approaches to exert control over the generated images. Controlled image synthesis focuses on generating images that adhere to predefined properties, while image editing aims to change specific properties while preserving others within an existing image. Typically, controlled image synthesis is achieved via conditional modelling where a guiding signal is introduced during training. Conversely, image editing is generally carried-out post-generation by exploiting the latent semantic representations implicitly learned by generative models. In Section 2.2.1, we provide a more detailed exploration of generative models conditioned on visual and/or textual cues while in Section 2.2.2, we delve into latent space manipulation methods. In Section 2.2.3, we present image inversion techniques that allow to bridge the gap between generated and real images.

2.2.1 Conditional Models

Image-to-Image models Different approaches condition the generative models on visual data as it constitutes strong control cues providing spatial and structural guidance. These methods are often referred to as image-to-image translations as they aim at transforming an input image from one domain *e.g.* sketches, labels, edges to a target domain *e.g.* natural images. If the pairs of input and corresponding target images are available, Pix2Pix [56] introduces a general framework that employs the cGAN objective with a $L1$ reconstruction loss. The generator has a U-Net [118] architecture with skip connections where the conditioning features are incorporated via concatenation. Wang et al. [152] later proposed an extension of Pix2Pix for the generation of high-resolution images using a coarse-to-fine generator similarly to PGGAN [63] and a multi-scale discriminator. To generate accurate images from semantic maps, SPADE [101] improves the injection of the conditioning maps by introducing a spatial normalization where the parameters depend on the current map while OASIS [136] employs the discriminator as a segmentation network. Other works focus on training image-conditioned models without requiring paired training samples as they are not always available and might be difficult to collect. CycleGAN [168] learns simultaneously two mappings from domain A to B and reciprocally using the typical adversarial loss. To ensure that the input and output images share some properties, they add a cycle-consistency loss which enforces that the obtained image after going through the two mappings successively (A to B to A) match the original image. CycleGAN can be expensive when there are multiple target domains. To enable multi-domain image translation, StarGAN [17] introduces

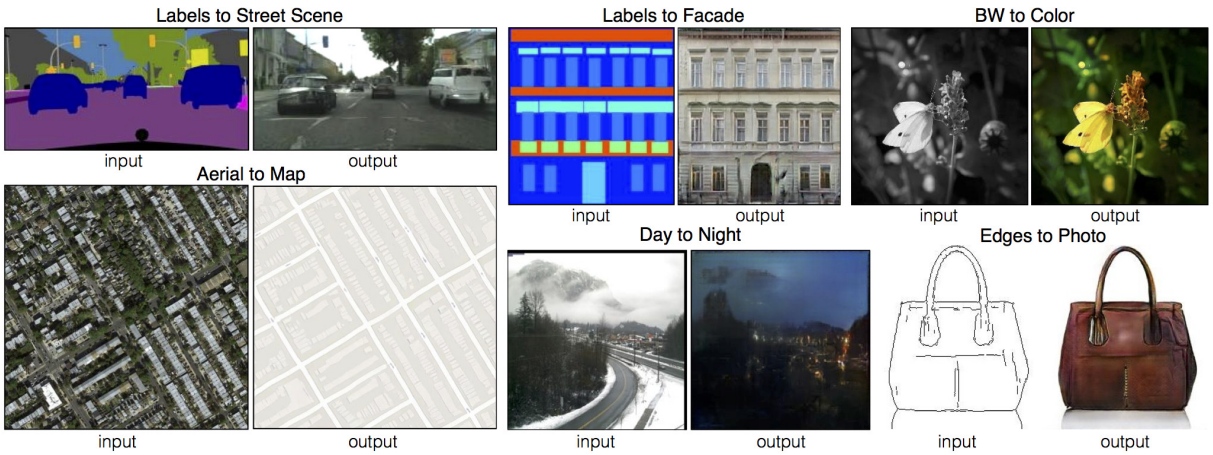


Figure 2.6: Image-to-image translation results. Figure taken from [56].

a generator conditioned on an additional target domain vector. More recently, Richardson et al. [116] introduced a framework that solves various image-to-image translation tasks with a pre-trained StyleGAN by learning an encoder to generate the latent codes of the conditioning images.

Similarly to Pix2Pix, Palette [119] introduces a unified framework to achieve image-to-image translation by training conditional DDPMs on various types of conditioning images. The DDPM framework also allows to achieve image-to-image translation without re-training the models. SDEdit [89] runs the denoising process from the conditioning image perturbed with Gaussian noise. RePaint [86] proposes a similar approach for image inpainting where the known region is taken from the input image and the inpainted part sampled from the denoising process. They introduce resampling steps to ensure semantic consistency between the two parts.

Text-to-image models More recently, various works have tackled text-to-image generation due to the intuitiveness and flexibility of textual cues to control the generation. Ramesh et al. [111] first introduced DALL-E an auto-regressive model *i.e.* that generates an image pixel by pixel conditioning each pixel on the previously generated ones, trained on millions of (text, image) pairs able to achieve impressive zero-shot text-to-image generation. GLIDE [120] re-used DALL-E's training set to learn a diffusion model conditioned on text using classifier-free guidance. Ensuing text-to-image models directly leverage Contrastive Language-Image Pretraining (CLIP) [109], a powerful joint text-image representation learned on 400 million data using contrastive learning and transformer-based architectures [144] to encode the inputs. DALL-E 2 [112] generates CLIP image embeddings to condition a diffusion-

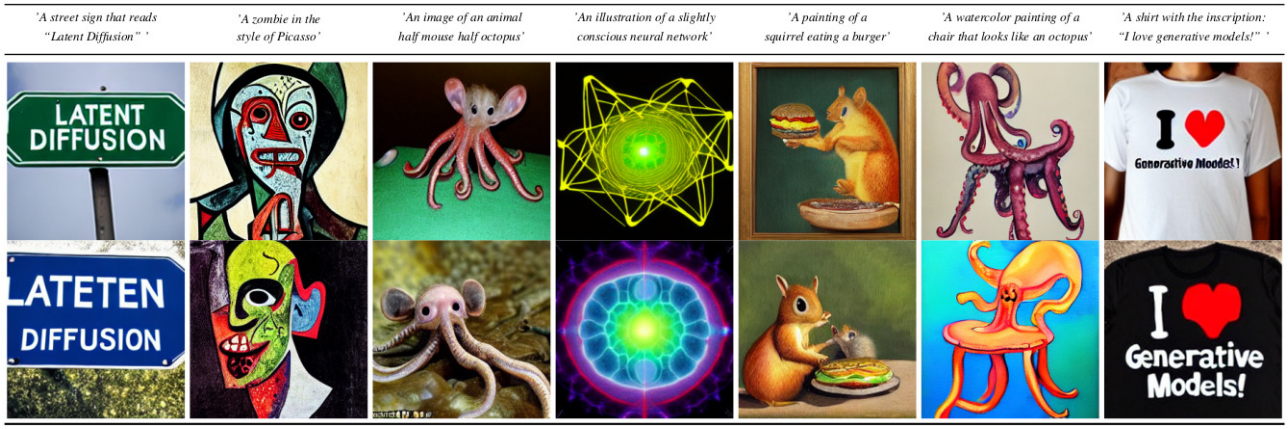


Figure 2.7: Text-to-image generation results. Figure taken from [117]

based decoder for image generation. Stable Diffusion [117] is a LDM conditioned on CLIP's text representations injected via cross-attention in the intermediate layers of the U-Net. Other diffusion models such as Imagen [120] exploits Large Language Models (LLMs) such as T5 [110] to process the textual input. Schuhmann et al. [124] later introduced the LAION dataset that is larger than any previous multi-modal datasets with 2 billion training samples. Re-training on this dataset leads to enhanced zero-shot ability and expressiveness. Most of the previous models are trained using the DDPM framework for training stability and mode coverage reasons. However, two GAN-based architectures were recently introduced and achieve competitive performances [61, 123].

Although textual control is intuitive and flexible, it can be ambiguous *e.g.* the prompt "an elephant and a bird flying" might lead to generate a flying elephant [88]. It also heavily relies on the capabilities of the underlying text encoder. Typically, CLIP is known to poorly integrate compositional concepts such as counting [99], causing to rarely generate the correct number of objects. Various works thus focus on adding additional visual control to these models while retaining their initial capacities. ControlNet [161] creates a trainable copy of a pre-trained LDM with an additional control input such as a semantic, edge or depth map. This trainable copy is linked to a frozen copy using 1×1 convolution layers initialized with zeros. Their method allows to augment the pre-trained network using a small amount of data. T2I-Adapter [91] proposes a similar approach using lighter networks instead of a trainable copy. More recently, ZestGuide [20] introduces zero-shot semantic map conditioning by leveraging implicit segmentations from the cross-attention layers of Stable Diffusion [117].

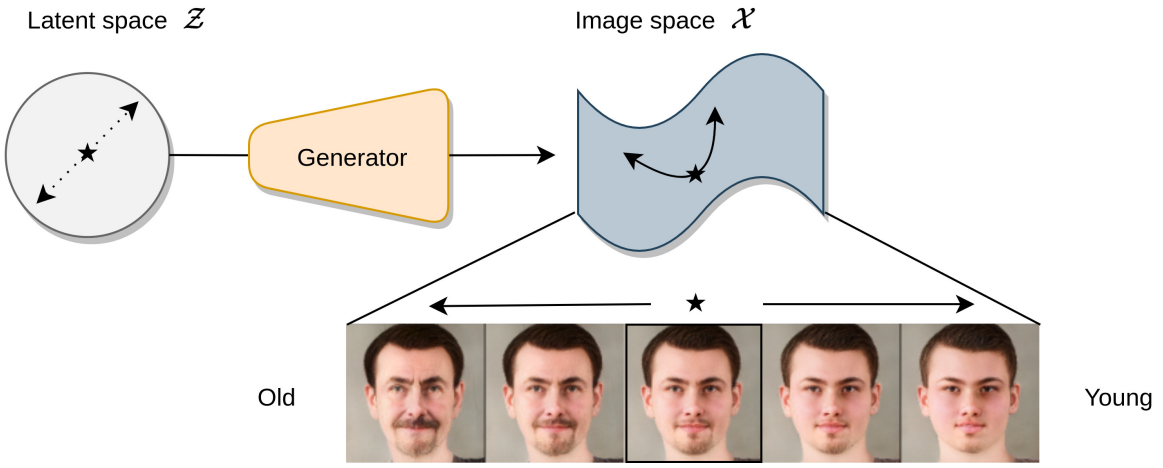


Figure 2.8: The key idea of latent space manipulation methods is to look for trajectories in the latent space ($\mathcal{Z}$) *e.g.* linear directions that produce interpretable changes in the image space ($\mathcal{X}$). For instance, the direction here modulates the age of the face.

Although we focus on controlled synthesis, various image-to-image and text-to-image models can be used or re-purposed for editing tasks [17, 170, 165, 47]. For image-to-image models *e.g.* [17, 170], the editing is achieved by conditioning on the image to be edited. Prompt-to-Prompt [47] edits images using pre-trained text-to-image DDPM by replacing a word in a prompt and re-injecting the cross-attention maps of the image generated with the initial prompt to maintain the source content.

2.2.2 Latent Space Manipulation

As mentioned in Section 2.1.1, early studies on GANs uncovered some level of semantic structure in the latent space [108]. This has led various works to identify trajectories corresponding to interpretable factors such as geometric transformations [57, 105, 134], memorability [40], facial attributes [127, 157] (see Fig. 2.9) to manipulate these properties in the generated images without requiring to re-train the generative model. The manipulation is typically achieved via a function f that moves the latent code $\mathbf{z}$ of an image along the learned trajectories, that can have various forms *e.g.* linear, non-linear. For instance, it can be global translations $\mathbf{n}$: $f(\mathbf{z}, \alpha) = \mathbf{z} + \alpha \cdot \mathbf{n}$, where $\alpha \in \mathbb{R}$ is a parameter that controls the strength and direction of the move and the semantic change accordingly (cf. Fig. 2.8).

To find the trajectories, there are unsupervised approaches that require no annotations but necessitates to manually interpret each trajectory after discovery. GANSpace [42] applies PCA

Figure 2.9: Editing results of GANSpace [42], that performs PCA in the latent space and discovers meaningful concepts (at the bottom of each image) encoded in the principal components. Figure taken from [42].

in the latent space of different pre-trained GANs and finds meaningful concepts encoded in the principal components. Similarly, [126, 149] use the eigenvectors of the affine transform employed at the beginning of most GAN architectures. Voynov and Babenko [147] find meaningful directions without supervision by jointly learning the set of directions and a network to distinguish the corresponding image transformations. To avoid having to manually inspect the directions, following works employ self-supervised [57, 105] or supervised learning [127, 171, 3, 51, 150]. These methods aim to learn directions that correspond to pre-defined semantic concepts. Jahanian et al. [57] introduce a self-supervised approach that modifies a generated image using automatically applicable transformations *e.g.* geometric or color transformations and learns the directions that minimize the LPIPS or L2 distance between the initial and transformed images. Supervised methods allow to broaden the types of modification by relying on auxiliary attribute image classifiers. In InterfaceGAN, Shen et al. [127] employ a facial attributes predictor to edit various attributes such as the gender, age, smile in face images. The predictor annotates a set of latent codes that are then used to find linear subspaces

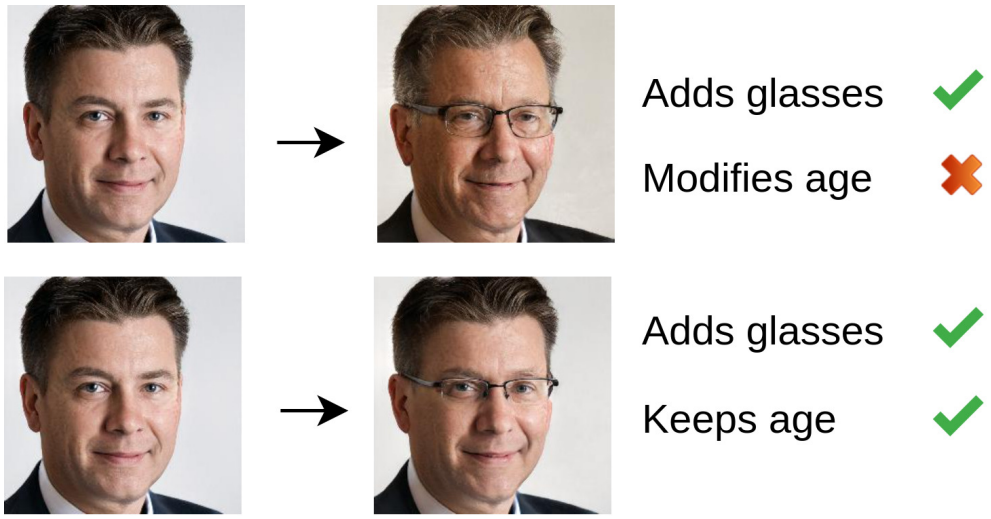


Figure 2.10: Entangled (top) vs. Disentangled (bottom) editing. A desirable property of the discovered latent semantic trajectories is to affect only the targeted attribute (disentanglement).

corresponding to the desired attributes with linear Support Vector Machines (SVMs). This approach will be discussed in more details in Chapter 3.

Most of the previous approaches employ global and linear trajectories. Subsequent works demonstrate improved editing by learning non-linear paths in the latent space specific to the input codes. With StyleFlow [3], Abdal et al. train Continuous Normalizing Flows [100] in latent space conditioned on semantic attributes. The editing is achieved by running inverse inference with the original attributes to recover the latent code followed by forward inference with the modified attributes. HijackGAN [150] updates the latent codes iteratively using the gradients of a classifier trained on a set of annotated codes. Similarly, several methods learn non-linear latent manipulation networks conditioned on the input codes, using the guidance of multi-attribute classifiers [171, 157, 51]. The idea is to learn transformations such that the edited images minimize a classification loss corresponding to the desired editing. These classifier-based approaches will be discussed in more details in Chapter 4.

Disentanglement A common goal of latent space manipulation methods, also considered in Chapter 3 and Chapter 4 of this work, is *disentanglement*. In the context of semantic latent directions, it aims at learning directions that allow the editing of a single semantic property at a time. In practice, a discovered direction such as a direction that adds glasses, tends to have an impact on another or several other properties simultaneously - *e.g.* aging the person as in Fig. 2.10. This phenomenon is caused

by the correlations in GAN training sets *e.g.* older males with glasses might be more represented in the dataset in the previous case; that affects the semantic structure of the latent space [127, 57]. Various approaches tackle this issue. InterfaceGAN [127] proposes a post-processing on the learned directions that consists in enforcing orthogonality between a given direction and the directions of correlated attributes. Yao et al. [157] introduced an explicit disentanglement constraint using a multi-attribute classifier during the learning of the semantic transformations. Other methods also enforce disentanglement through changes in the GAN architecture [64] or learning strategy [130]. Shoshan et al. use contrastive learning to design more disentangled latent spaces. As mentioned previously, the intermediate latent space $\mathcal{W}$ of StyleGAN with a learned latent distribution instead of a fixed one results in more disentanglement. Most of the aforementioned approaches thus operate in this $\mathcal{W}$ space, as well as the extended space $\mathcal{W}+$ [1]. The latter offers even more controllability as subsets of layers were shown to dedicate to specific concepts [156]. For instance, Hou et al. [51] learn semantic edits in $\mathcal{W}+$ jointly with an attention parameter on the different layers.

Local Editing Another goal is to achieve more *localized* editing as previous methods tend to affect the image globally. EditGAN [81] relies on semantic maps to achieve precise and localized editing with latent code manipulation. Following DatasetGAN [163], an additional branch is added in the GAN architecture to generate segmentation maps jointly with the images. To edit an image, a user modifies the initial semantic map and the latent code is optimized to match the changes in the new generated map (cf. Fig. 2.11). Other approaches achieve local image editing without relying on semantic maps but by manipulating the style codes (defined in Section 2.1.1) instead of the latent codes. Collins et al. [18] identify feature channels corresponding to localized semantic concepts and interpolate between the style codes of two images. Similarly, StyleSpace [155] identifies attribute-specific channels and performs editing by increasing or decreasing the value of the style parameter.

Latent space manipulation methods can roughly be divided in two classes: unsupervised approaches that find directions then try to interpret them and classifier-based methods that learn directions corresponding to annotated semantic concepts. More recently, various methods exploit CLIP to edit an image based on an input text. StyleCLIP [102] and StyleMC [72] identify directions in the latent space of pre-trained GANs by minimizing cosine similarity between the CLIP representations of the input text and the edited image. In contrast to GANs, the latent space of DDPMs lacks compactness and

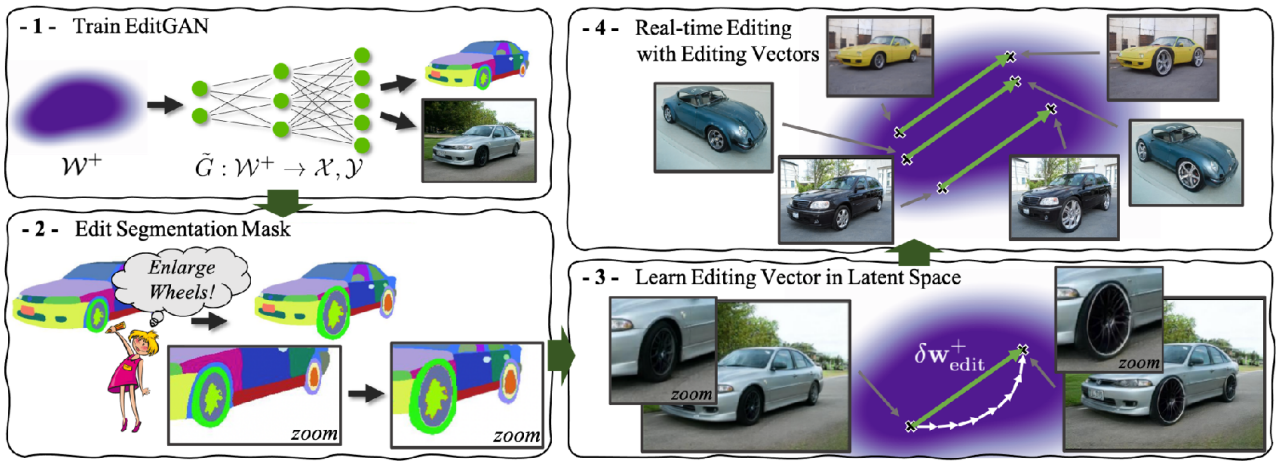


Figure 2.11: Ling et al. learn edits in the latent space corresponding to localized edits in image space by jointly modelling the distribution of images and semantic maps. Figure taken from [81].

semantic meaning [107]. However, classifiers [25] or CLIP [68] trained on noised images can be used to guide the denoising process and control the output. Preechakul et al. [107] also showed that diffusion models can be trained with an additional learnable semantic encoder. The semantic code holds the high-level semantics and is used to condition the diffusion process that generate the stochastic details. After training, GAN latent manipulation methods can be applied to the semantic code.

2.2.3 Image Inversion

Controlled image generation and image editing via latent space navigation in particular is tightly connected to the task of image inversion, that is, projecting a real image to the latent space, as it allows to apply the editing to those images. Inversion methods look for $\mathbf{z}$ such as $G(\mathbf{z}) = \tilde{\mathbf{x}}$ where $\tilde{\mathbf{x}}$ should be as close as possible to the real image $\mathbf{x}$. For GAN-based inversion, existing works thus employ optimization [167, 1], encoders [116, 140, 4] or hybrid approaches [166] with a reconstruction objective computed with L1, L2 or LPIPS [162] distances. With the emergence of StyleGAN, various works focus on the inversion in the $\mathcal{W}$ latent space [1]. In particular, Abdal et al. [2] showed that inversions in the extended space $\mathcal{W}+$ produce superior reconstructions. However, this comes at the expense of the ability to edit the latent codes as it involves moving away from latent codes on which the GAN was trained [140]. Various subsequent works thus focus on improving inversion in this space and the trade-off between reconstruction and editability. Pixel2style2pixel (pSp) [116] trains an encoder that learns each code of the extended space from the feature map of the corresponding resolution.

Encoder4editing (e4e) [140] adds additional regularization to minimize variance between the different codes thus enforcing to stay close to $\mathcal{W}$ and improving editability. ReStyle [4] demonstrates improved inversions by iteratively refining the latent codes. More recently, HyperStyle [5] faithfully reconstruct and edit out-of-domain images by modulating StyleGAN's weights.

In contrast to GANs, DDPMs have a direct way to produce a 2D latent code from a real image (via the forward process). However, this process is inherently stochastic. DDIMs [132] are an alternative to DDPMs that model the forward process as a non-markovian process leading to a deterministic reverse process. The inversion with DDIM leads to near-perfect reconstructions [107] but that can be unstable, producing non-accurate reconstructions in some cases [47]. More recently, EDICT [148] introduced a more stable inversion process inspired from affine coupling layers in normalizing flows [115].

2.3 Generative Data Augmentation

In this section, we focus on the use of generated images to augment or replace real training datasets. Indeed, for diverse tasks, it is critical to have large annotated datasets to train models with good performances and generalization, but such datasets are hard to obtain. Due to the fast improvements in image synthesis, the disparity in quality between generated and real images has significantly diminished. Generative models have thus emerged as cost-effective tools for "acquiring" unlimited training datasets for diverse tasks. Previous works leverage conditional GANs to enhance existing datasets for image classification [6, 33], crowd counting [151] and image segmentation [163, 92]. One advantage of conditional GANs is that the condition can be used as an annotation for the generated data *e.g.* a GAN conditioned on class labels, such as cat and dog, generate pairs of (image, label) as the image represents a cat or a dog. However, these models require to be trained on annotated datasets. For tasks like counting or segmentation with expensive labelling process, image-to-image translation techniques can be employed to translate already annotated data to another domain [151, 92]. More recently, DatasetGAN [163] introduced an approach that leverages the feature space of a pre-trained StyleGAN to generate synthetic datasets with pixel-wise semantic annotations using only a few annotated examples. Some works also explored the use of GAN-generated data only to train classifiers or for representation learning [11, 58]. In particular, Jahanian et al. [58] learn effective representations with contrastive learning, employing close latent codes to generate different views of the same image. Other approaches leverage the latent space of GANs to generate more targeted data, in particular to improve the fairness of facial datasets [138, 26].

Recent works exploit DDPMs that tend to yield higher quality samples and more diverse images due to being trained on large-scale datasets. With the availability of powerful text-conditioned diffusion models [117, 95, 112, 120], various works utilize these models to generate synthetic training datasets for image classification [9, 45, 10, 121, 141]. He et al. [45] synthesize data by prompting a pre-trained model with the class labels and demonstrate improved classification performances on various datasets such as ImageNet [22], EuroSAT [46] in the low-data and zero-shot regimes. Using a similar approach to generate data, Bansal and Grover [10] demonstrate that training ImageNet classifiers on an equal mix of real and generated data improves their robustness to domain shifts. Trabucco et al. [141] propose to synthesize augmentations for more specific classes that might be outside the scope of the

pre-trained model using Textual Inversion [35]. They also introduce local augmentations by leveraging inpainting techniques [86] and object masks. Other works also showcase promising results for using the generated data for representation learning [121, 139]. In contrast to GANs, the pre-trained DDPMs are trained on larger datasets than the datasets that are augmented thus introducing more diversity but also some domain gap that may be harmful. To tackle this, [45, 141] generate images from the real images with some added noise [89], [30] edit real images while [9] finetunes the pre-trained model on the real dataset. To generate better augmentations, some methods also focus on improving the prompts *e.g.* using text-to-sentence model [45], WordNet [121] or LLMs [30]. More recently, various approaches leverage text-to-image models to generate synthetic testing sets by modifying a single element in the prompts *e.g.* the background, co-occurring subjects, data domain, to evaluate classifiers failures [75, 106, 145]. These works mainly focus on error discovery but the resulting knowledge could also be used to produce more targeted data augmentation.

This body of literature consistently demonstrates how generated data, in particular from the large vision-language models, allows to learn more robust representations and improve generalization for image classification. On the other hand, few works employ these models to generate synthetic datasets for more complex tasks such as object counting, detection or semantic segmentation. Compared to image classification, these tasks might involve smaller datasets and require local spatial understanding, as objects can be small and follow complex layouts. The generated data needs a level of compositionality that current generative models, including diffusion models, struggle to achieve. In Chapter 5, we tackle the generation of augmentations for the task of few-shot counting. To bring the power of synthetic data to counting, we propose to condition diffusion models not only on text prompts but also on counting density maps to generate images with the correct number of objects in the desired spatial configuration. We exploit this double control to generate unseen data by prompting the model with novel combinations of the controls.

Chapter 3

Disentangled Image Editing with Pre-trained GANs

Contents

3.1 Introduction **30**
3.2 Related work **31**
3.3 Balanced Sampling for Disentangled Editing **34**
3.3.1 Multi-Attribute Balanced Sampling 34
3.3.2 Semantic Directions Estimation 36
3.4 Experiments **37**
3.4.1 Experiments Setup 37
3.4.2 Disentanglement 39
3.4.3 Identity Preservation 45
3.4.4 Orthogonality 46
3.5 Conclusion **46**

3.1 Introduction

In this chapter, we explore image editing with pre-trained unconditional GANs. Various works have shown that it is possible to leverage the latent space of these models to control the generated images. In particular, different approaches demonstrate that certain *linear* directions can be interpreted as variations of some semantic attributes across the latent space [127, 126, 105, 57, 171, 147]. However, the discovered directions often affect multiple attributes while a desirable property of an editing method is to allow *disentangled* editing - *i.e.* enabling to alter a single attribute at a time.

Learning-based methods commonly rely on a three-stage pipeline that consists of sampling a set of latent codes, then labeling the latent codes from the corresponding images using pre-trained image classifiers and finally, extracting the directions [127, 156, 150, 3]. As GANs learn to approximate the training data distribution that carries different kinds of biases, the sampling stage leads to generating biased datasets that can, in turn, affect the semantic directions. The third stage can be performed by training a linear classifier to separate latent codes corresponding to images with a desired attribute (positive set) from those corresponding to images without the desired attribute (negative set). Assuming that the two sets can be well-separated, the semantic of a given latent code lying on one side of the hyperplane turns into the opposite when moved across the classifier's decision boundary [127]. The vector orthogonal to that boundary can thus constitute a direction for controlling the attribute. Existing correlations among attributes in the generated data may cause the positive and negative sets of a target attribute to be strongly imbalanced with respect to other attributes, thus biasing the direction toward those attributes.

As the correlations are inherited from the GAN training data, reducing entanglement would require access to this dataset, that is not guaranteed, and very expensive GAN re-training. Instead, we propose to reduce the bias in the GAN-generated dataset directly. Specifically, after sampling and labeling the latent codes, we balance the attribute joint distributions and remove correlations. We demonstrate the effectiveness of this approach on the task of facial attributes editing through experiments conducted on different GAN architectures. We show that our directions are naturally disentangled in comparison to InterfaceGAN [127] which requires a post-processing step to reduce entanglement.

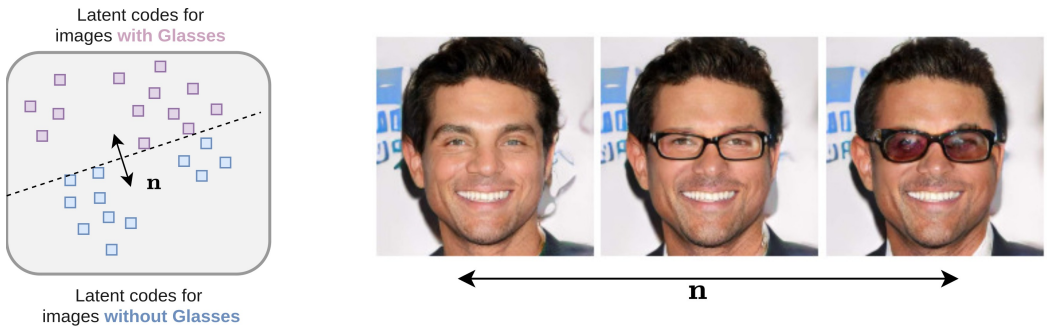


Figure 3.1: Extracting a semantic direction using the decision boundary of a binary classifier. (a) **n** is the vector orthogonal to the decision boundary (dotted line) of a 'glasses' latent classifier. (b) Translating the latent code with **n** leads to adding glasses that become more visible as the magnitude of the translation increases. Figure taken from [127].

3.2 Related work

Exploiting Classifiers' Boundaries Inspired by [67], Denton et al. [24] proposed to create counterfactuals samples by exploiting the vector orthogonal to the decision boundary of a linear classifier trained in the latent space of a pre-trained GAN. InterfaceGAN (IfGAN) [127] uses a similar approach to edit facial attributes in generated images. IfGAN assumes that for a given binary attribute, *e.g.* 'glasses' (presence vs. absence of glasses), there is a hyperplane in the latent space serving as the separation boundary between the positive vs. negative codes w.r.t. this attribute. They thus propose to use the normal vector to the hyperplane, **n**, to move latent codes toward and across the hyperplane (cf. Fig. 3.1 (a)), and show that it allows to vary the semantic score of the attribute accordingly (cf. Fig. 3.1 (b)). To learn the boundaries, they train linear SVMs [19] on a large collection of (latent code, semantic labels) pairs. The labels obtained by sampling latent codes, and then annotating the corresponding synthesized images with semantic labels using pre-trained image classifiers. The vectors orthogonal to the SVM's boundaries are then used to translate the latent codes with a parameter controlling the translation's magnitude. They show that modulating this parameter leads to continuous changes (cf. Fig. 3.1 (b)) even though the problem is formulated as a bi-classification. However, the directions found with this approach also lead to produce images exhibiting undesired modifications of other attributes, as shown in the qualitative results of Fig. 3.2 (bottom).

Disentanglement of semantics Some studies show that the observed entanglement stems from the presence of bias in the datasets used to train GANs [24, 127]. For CelebAHQ dataset [63], Fig. 3.2 (a)

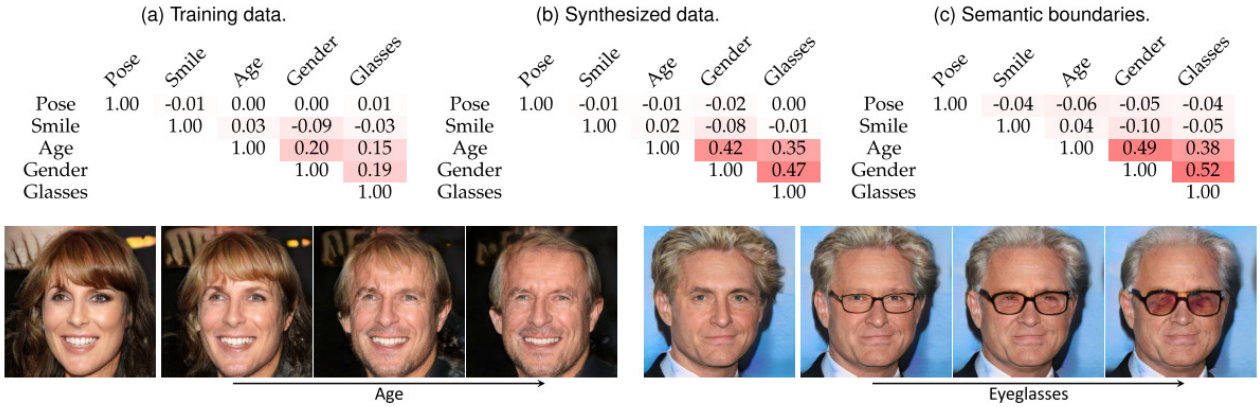


Figure 3.2: Entanglement analysis for PGGAN trained on CelebAHQ. (a) Correlation matrix computed on the GAN training set. (b) Correlation matrix computed on the GAN-generated set. (c) Cosine similarities between attribute directions. Bottom: For each example, the original image is displayed to the left and 3 edited images obtained with translations of successively increased magnitudes are shown to the right. We observe that 'age' is entangled with 'gender' and 'eyeglasses' with 'age'. Figure taken from [127].

reveals spurious correlations among the chosen set of attributes[1], in particular for 'glasses', 'gender' and 'age'. For the data generated by a PGGAN [63] trained on this data, Fig. 3.2 (b) shows these correlations are also present and even reinforced. As mentioned in Section 2.1.1, GANs suffer from mode collapse, that leads to amplifying the bias of their training data [164]. These correlations result in learning semantic directions that encode multiple attributes at once, as reflected in Fig. 3.2 (c) where the directions corresponding to correlated attributes exhibit high cosine similarities.

To reduce this entanglement, *conditional manipulation* is a common post-processing that refines the semantic directions by explicitly enforcing an orthogonality constraint for the new directions [127, 150, 82]. Given two entangled directions $\mathbf{n}_1$ and $\mathbf{n}_2$ associated with attribute a_1 and a_2 respectively, the goal is to obtain a new direction $\mathbf{n}_1'$ that still modifies attribute a_1 but no longer affects a_2. As illustred in Fig. 3.3, the vector $\mathbf{n}_1$ (primal direction) is first projected onto the vector $\mathbf{n}_2$ (conditioning direction), and the resulting vector is then subtracted from the initial direction: $\mathbf{n}_1' = \mathbf{n}_1 - (\mathbf{n}_1^\top \mathbf{n}_2)\mathbf{n}_2$. Vector $\mathbf{n}_1'$ is orthogonal to $\mathbf{n}_2$, so a translation following $\mathbf{n}_1'$ should not produce any translation in the direction of $\mathbf{n}_2$. When there are multiple conditioning vectors, the primal direction has to be projected onto the hyperplane constructed by all conditioning vectors. The method thus assumes that each semantic subspace is independent of others, which may not always be true. More importantly, it

[1]Other attributes in CelebA are also strongly correlated, see the full correlation matrix in Fig. A.2 in Appendix A.

requires an additional ad-hoc step after learning the directions.

Other approaches address disentanglement, for instance, Spingarn-Eliezer et al. [134] introduce more constrained nonlinear paths that are defined as small circles on a sphere. [171, 3] argue that entanglement is reduced if the transformations are learned together. For style-based GAN architectures, Yang et al. [156] propose to manipulate the latent code only for some layers. Different from previous works, we propose a simple and general approach that *proactively* tackles entanglement by debiasing the data employed to discover the directions. Hence, we argue that it can be complementary to previous proposals.

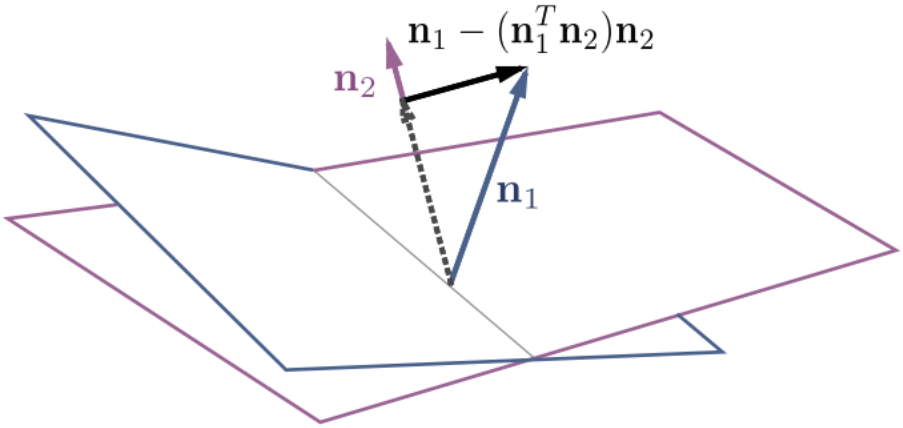


Figure 3.3: Conditional manipulation. Figure taken from [127].

3.3 Balanced Sampling for Disentangled Editing

Our goal is to prevent the propagation of biases by eliminating undesired correlations within the datasets. We are dealing with two datasets: (1) the GAN training data (real data) and (2) the data used to train latent classifiers (generated data). The correlations in (2) are inherited from (1). However, we may not have access to the actual training data as we leverage pre-trained GAN models, and retraining a GAN is resource-intensive. Therefore, we suggest a straightforward approach: remove the correlations in (2) by subsampling the generated data such that the attributes joint distributions are balanced for a given group of attributes.

3.3.1 Multi-Attribute Balanced Sampling

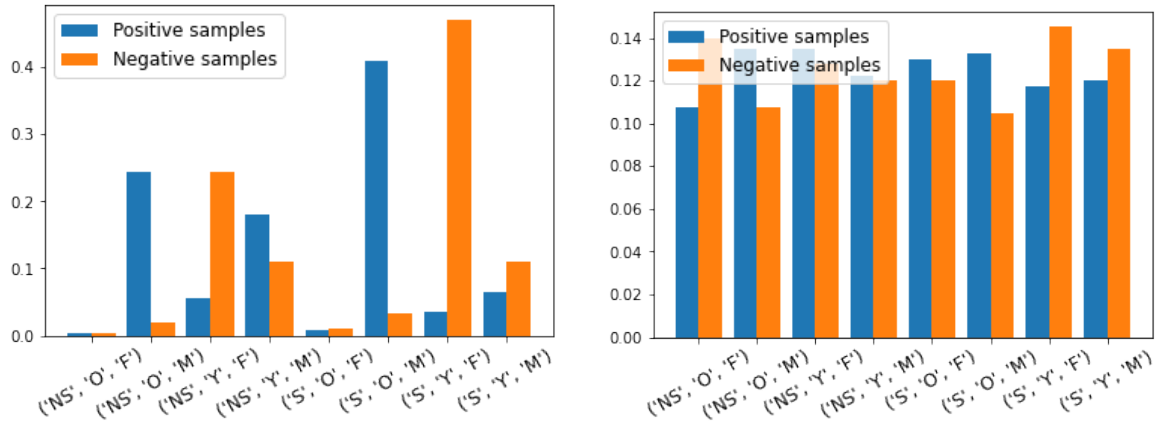
Let G be a pre-trained generator that maps a latent code $\mathbf{z}$ sampled from a d-dimensional latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ to an image $\mathbf{I} = G(\mathbf{z})$. Let $\mathbf{I} \in \mathcal{I} \subseteq \mathbb{R}^{H \times W \times C}$ be described by a set of binary attributes[2] $\mathcal{A} = \{a_k, 1 \leq j \leq m\}$. For each attribute a_k, we aim to find a linear direction in the latent space defined by unit vector $\mathbf{u}_k \in \mathbb{R}^d$:

$$\mathbf{z}' = \mathbf{z} + \alpha \mathbf{u}_k, \quad \alpha \in \mathbb{R} \tag{3.1}$$

such that *only* the intensity of attribute a_k differs in the resulting image $\mathbf{I}' = G(\mathbf{z}')$, where α is a scalar that controls the strength of the change. To constitute pairs of (latent code, labels) as in [127], the first step is to randomly sample N latent codes and generate the corresponding images: $\{(\mathbf{z}^{(i)}, G(\mathbf{z}^{(i)}))_{i=1}^N\}$. Then, every image is labelled with a pre-trained image attribute classifier $F_{\mathcal{I}}$ and the labels are associated to the latent codes to produce $\mathcal{S} = \{(\mathbf{z}^{(i)}, F_{\mathcal{I}}(G(\mathbf{z}^{(i)})))_{i=1}^N\}$.

The distribution of the binary attributes for a set of data can be represented in an m-dimensional contingency table (one dimension per attribute) where each of the 2^m cells contains the number of samples that have the corresponding combination of values for the m attributes. If there are strong correlations between attributes in the GAN training data then the contingency table for that data is strongly imbalanced. The data in $\mathcal{S}$, generated by the trained GAN, is expected to show similar or worse correlations. For an attribute a_j, the sets $\mathcal{S}_j^+$ and $\mathcal{S}_j^-$ employed for training a classifier in the latent space mirror the imbalance in $\mathcal{S}$. If we consider the attribute 'glasses' in CelebAHQ, Fig. 3.4 (a) shows how imbalanced the associated $\mathcal{S}_j^+$ and $\mathcal{S}_j^-$ sets are with respect to the attributes 'age', 'gender'

[2] or continuous attributes that are binarized.



(a) Random sampling.

(b) Our sampling.

Figure 3.4: Joint distributions for 3 binary facial attributes Age ('O': Old, 'Y': Young), Gender ('M': Male, 'F': Female) and Smile ('S': Smile, 'NS': No Smile). In (a), the positive set contains a majority of *old males* while the negative set contains a majority of *young females*, leading to bias the direction 'glasses' toward the attributes 'age' and 'gender'.

and 'smile'. It is natural to expect that a classifier Ψ_j trained on such imbalanced data is influenced by the strong correlations. And, consequently, the unit vector $\mathbf{u}_j$ that is orthogonal to its decision boundary entangles the control of the target attribute with the most correlated attributes.

Sampling procedure The idea of our method is to subsample the data in $\mathcal{S}$ so as to obtain approximately the same number of samples in each cell of the contingency table. We build a multi-attribute balanced sample $\mathcal{B} \subset \mathcal{S}$ by iteratively selecting data from $\mathcal{S}$ until we reach the total number of samples $N_0 \leq N$ we aim to obtain. At each iteration, we first uniformly sample one combination of attribute values (one cell of the contingency table), then we uniformly sample without replacement one data point $(\mathbf{z}, F_{\mathcal{I}}(G(\mathbf{z})))$ with that combination. In this way, at the end of the sampling procedure, we expect to have a balanced contingency table for $\mathcal{B}$ where each of the 2^m cells contains approximately $\frac{N_0}{2^m}$ data points, as shown in Fig. 3.4 (b). The procedure is outlined in Algorithm 1.

The subsampling procedure works well if there is enough data in $\mathcal{S}$ for each combination of attribute values. For strongly imbalanced data, we may have to address the case where there is no more data in $\mathcal{S}$ for one or more combinations before reaching the desired total number of samples N_0. Note that, as we show in Section 3.4.2, good results can be obtained with moderate values for N_0. The ideal solution for having a balanced $\mathcal{B}$ is to expand $\mathcal{S}$ by generating more images with G. But this can be

very expensive since the imbalance of $\mathcal{S}$ reflects the imbalance of the training dataset. Hence, we may require the generation of a very large number of images to obtain one more image with a rare combination of attribute values. Instead, our solution consists in simply skipping the current iteration if no more data is available for that combination. For high values of N_0, the resulting $\mathcal{B}$ is no longer so well-balanced but, as we show in Section 3.4.2, this only causes a slight decay in performance. An alternative is to oversample the already generated data corresponding to the rarest combinations of attribute values, *i.e.* random sample *with* replacement for a combination if its cell in the contingency table of $\mathcal{S}$ has much less than $\frac{N_0}{2^m}$ data points. As shown in Section 3.4.2, this allows to maintain good performances for high values of N_0.

Algorithm 1: Multi-attribute balanced sampling.

Data: $\mathcal{S}$ a list of N labeled latent codes, $\mathcal{A}$ the corresponding multi-attribute labels, N_0 target number of samples

Result: N_0 latent codes balanced over $\mathcal{A}$

for *every attribute combination* $(a_1, a_2, \ldots, a_m) \in \mathcal{A}$ **do**
  $\mathcal{C}[a_1, a_2, \ldots, a_m] \leftarrow$ latent codes of $\mathcal{S}$ labeled with this set of attributes;
end for

$\mathcal{B} \leftarrow [\,]$;

for $i \leftarrow 1...N_0$ **do**
  $a_1, a_2, \ldots, a_m \leftarrow$ a random non-empty cell of $\mathcal{C}$;
  $s \leftarrow$ a random latent code from $\mathcal{C}[a_1, a_2, \ldots, a_m]$;
  remove s from $\mathcal{C}$;
  $\mathcal{B} \leftarrow \mathcal{B} \cup s$;
end for

return $\mathcal{B}$

3.3.2 Semantic Directions Estimation

Our sampling procedure leads to a sample $\mathcal{B}$ of size N_0 that is balanced with respect to all attributes. For each attribute a_j, two sets $\mathcal{B}_j^+$ of size $N_j^+ \approx \frac{N_0}{2}$ and $\mathcal{B}_j^-$ of size $N_j^- \approx \frac{N_0}{2}$ can be readily obtained by considering the data having positive and respectively negative labels for attribute a_j. To find the direction $\mathbf{u}_j$ in latent space that allows to control attribute a_j, a good solution is to

train a linear classifier on $\mathcal{B}_j^+ \cup \mathcal{B}_j^-$, then take as $\mathbf{u}_j$ the vector orthogonal to the decision boundary. Preference is usually given (*e.g.* [127]) to linear Support Vector Machines (SVMs) that are fast to train and effective in high dimensions. To improve generalization, the value of the regularization hyperparameter should be selected by cross-validation. But as we show later in Section 3.4.2, when the dataset is balanced, a stronger regularization (larger SVM margin) tends to produce directions that allow more disentangled edits. If the linear SVM has a very large margin, the decision boundary becomes orthogonal to the line connecting the centroids of the two classes. Instead of training SVMs, we thus directly employ this easy-to-compute direction. For attribute a_j, this direction is defined by:

$$\mathbf{u}_j = \frac{1}{N_j^+} \sum_{i=1}^{N_j^+} \mathbf{z}_i^+ - \frac{1}{N_j^-} \sum_{i=1}^{N_j^-} \mathbf{z}_i^-, \quad \mathbf{z}^+ \in B_j^+ \text{ and } \mathbf{z}^- \in B_j^-. \tag{3.2}$$

3.4 Experiments

We conduct experiments on the task of facial attributes editing. In Section 3.4.2, we present our disentanglement results considering the following attributes: 'glasses' (Gl), 'gender' (Ge), 'smile' (S) and 'age' (A). We compare to InterfaceGAN before (IfGAN) and after conditional manipulation (IfGAN$^\perp$). We also explore the influence of (i) the sample size and (ii) the method to estimate the direction. In Section 3.4.3, we also investigate the ability of both methods to preserve the identity of a face. Finally, in Section 3.4.4, we discuss the orthogonality of our directions.

3.4.1 Experiments Setup

Models We conduct experiments with state-of-the-art GAN models trained on two face datasets, PGGAN trained on CelebAHQ [63] and StyleGAN, StyleGAN2 and StyleGAN3 trained on FFHQ [64, 65, 66]. All models generate 1024×1024 images. To label the generated images, we train an auxiliary image classifier on CelebA with a ResNet-50 [43] using multi-task learning to predict the attributes simultaneously. For each attribute, the task is a bi-classification problem with a softmax cross-entropy loss. In Table 3.1, we report the performances of the classifier for the attributes considered in the following experiments. There is a high accuracy on average for all four attributes ($\approx 95\%$).

Attribute	Precision (%)	Recall (%)	Accuracy (%)
Glasses	98.08	95.88	99.58
Gender	98.55	98.66	98.81
Smile	94.27	91.71	93.30
Age	89.32	95.89	88.37

Table 3.1: Performances of the attribute prediction model for each attribute.

Implementation details We synthesize $N = 1M$ images with PGGAN and $N = 500K$ images with StyleGAN models. We prepare a larger dataset for PGGAN as some combinations of attributes are rarer in CelebAHQ than in FFHQ. We apply the attribute predictors to all the generated images and discard the samples having confidence below 0.9 (for all attributes). For each attribute, we collect $N_0 = 1000$ samples using our multi-attribute balanced sampling. We choose this value depending on the number of samples in the cell with the fewest samples. The contingency table for PGGAN is reported in Table 3.2. Some combinations of attributes are really rare. For instance, the combination (Glasses=1, Smile=0, Gender=0, Age=0) contains only 46 samples for PGGAN. As a consequence, some cells can be empty before reaching the desired sample size, that is why we choose a moderate sample size N_0.

The semantic directions are then obtained by taking the direction defined by the centroids of each class (see Section 3.3.2). For a fair comparison, we reproduce InterFaceGAN results instead of using the provided directions as they were not computed using the same attribute prediction model[3] nor the same number of samples. For InterFaceGAN, we uniformly subsample the generated dataset and then train linear SVMs with $C = 1.0$[4] to obtain the semantic directions given by unit vectors. These vectors are $512d$ (dimension of the latent spaces of PGGAN and StyleGAN).

Metrics We use the *re-scoring* metric [127] to quantify the *desired effect* and *entanglement* associated with a direction. This metric measures how the attribute scores vary after manipulating the latent codes. Intuitively, a good direction should induce an increase in the score corresponding to the target attribute while not affecting other scores. Given a direction $\mathbf{u}_j$ for attribute a_j, the re-scoring for

[3]The model was not made available by the authors.
[4]As in the code provided by the authors: `https://github.com/genforce/interfacegan`.

		Gender=0		Gender=1		Total
		Age=0	Age=1	Age=0	Age=1	
Glasses=0	Smile=0	304	142 523	8722	56 606	208 155
	Smile=1	6087	262 442	22 089	54 202	344 820
Glasses=1	Smile=0	46	503	3387	2516	6452
	Smile=1	220	322	5246	970	6758
						566 185

Table 3.2: Contingency table for the dataset generated with PGGAN CelebAHQ.

attribute a_k is computed as:

$$\Delta \mathbf{s}_k = \frac{1}{n} \sum_{i=1}^{n} [F_{\mathcal{I},k}(G(\mathbf{z}_i)) - F_{\mathcal{I},k}(G(\mathbf{z}_i + \alpha \mathbf{u}_j))] \tag{3.3}$$

The desired *effect* $\Delta \mathbf{r}$ of direction $\mathbf{u}_j$ is given by the re-scoring result for the target attribute a_j. The *entanglement* of direction $\mathbf{u}_j$ with another attribute a_k is given by the re-scoring for that attribute. To obtain the *overall entanglement* $\Delta \mathbf{e}$ associated with a direction, the re-scoring results are averaged over the non-target attributes: $\Delta \mathbf{e} = \frac{1}{|\mathcal{A}|-1} \sum_{i \in \mathcal{A} \setminus a_j} |\Delta \mathbf{s}_i|$.

To quantify how the manipulations affect face identity, we employ a popular face recognition model pre-trained on VGGFace2 [13] and compute the cosine similarity between face embeddings before and after editing, as in [171]. We extract embeddings of dimension 2048.

Each metric is evaluated on $n = 2000$ latent codes with $\alpha = 2.0$ for the editing. The reported results are averaged over 3 experiments.

3.4.2 Disentanglement

PGGAN The results in Table 3.3 show a strong entanglement for IfGAN especially for the attributes 'glasses', 'gender' and 'age', which are the most correlated attributes. The conditional manipulation allows to reduce the entanglement while maintaining the desired effect.

Our approach succeeds in extracting directions allowing disentangled edits without requiring conditional manipulation. It significantly outperforms IfGAN and performs on par with IfGAN$^{\perp}$. Qualitative results are presented in Fig. 3.5.

Table 3.3: Re-scoring results for PGGAN. For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. We highlight (in **bold** for $\Delta\mathbf{e}$, *underlined* for $\Delta\mathbf{r}$) the best results among the disentangling approaches (IfGAN$^{\perp}$, Ours).

		Glasses	Gender	Smile	Age
IfGAN	$\Delta\mathbf{e}\downarrow$	0.205	0.118	0.034	0.125
	$\Delta\mathbf{r}\uparrow$	*0.386*	*0.519*	*0.386*	*0.142*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.055	**0.018**	0.015	**0.055**
	$\Delta\mathbf{r}\uparrow$	*0.231*	*0.420*	*0.381*	*0.115*
Ours	$\Delta\mathbf{e}\downarrow$	**0.038**	0.041	**0.013**	0.072
	$\Delta\mathbf{r}\uparrow$	*0.286*	*0.448*	*0.370*	*0.129*

Figure 3.5: Editing results for PGGAN for attributes Glasses, Gender and Age.

Table 3.4: Re-scoring results for StyleGAN models in $\mathcal{Z}$ (top) and $\mathcal{W}$ (bottom). For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. In $\mathcal{Z}$, we highlight the best results among the disentangling approaches (IfGAN$^{\perp}$, Ours).

(a) StyleGAN $\mathcal{Z}$

		Gl	Ge	S	A
IfGAN	$\Delta\mathbf{e}\downarrow$	0.140	0.161	0.050	0.108
	$\Delta\mathbf{r}\uparrow$	*0.339*	*0.335*	*0.154*	*0.156*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.064	0.061	0.033	0.060
	$\Delta\mathbf{r}\uparrow$	*0.278*	*0.266*	*0.145*	*0.131*
Ours	$\Delta\mathbf{e}\downarrow$	**0.042**	**0.060**	**0.024**	**0.054**
	$\Delta\mathbf{r}\uparrow$	*0.345*	*0.307*	*0.173*	*0.142*

(b) StyleGAN2 $\mathcal{Z}$

		Gl	Ge	S	A
IfGAN	$\Delta\mathbf{e}\downarrow$	0.135	0.106	0.041	0.111
	$\Delta\mathbf{r}\uparrow$	*0.335*	*0.406*	*0.100*	*0.113*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.049	**0.047**	0.026	0.068
	$\Delta\mathbf{r}\uparrow$	*0.232*	*0.346*	*0.099*	*0.091*
Ours	$\Delta\mathbf{e}\downarrow$	**0.038**	0.059	**0.024**	**0.062**
	$\Delta\mathbf{r}\uparrow$	*0.290*	*0.386*	*0.111*	*0.097*

(c) StyleGAN3 $\mathcal{Z}$

		Gl	Ge	S	A
IfGAN	$\Delta\mathbf{e}\downarrow$	0.121	0.122	0.041	0.153
	$\Delta\mathbf{r}\uparrow$	*0.444*	*0.395*	*0.301*	*0.182*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	**0.030**	0.037	0.018	**0.069**
	$\Delta\mathbf{r}\uparrow$	*0.382*	*0.346*	*0.295*	*0.163*
Ours	$\Delta\mathbf{e}\downarrow$	0.044	**0.027**	**0.014**	0.076
	$\Delta\mathbf{r}\uparrow$	*0.398*	*0.377*	*0.305*	*0.172*

(d) StyleGAN $\mathcal{W}$

		Gl	Ge	S	A
IfGAN	$\Delta\mathbf{e}\downarrow$	0.046	0.140	0.073	0.076
	$\Delta\mathbf{r}\uparrow$	*0.480*	*0.370*	*0.237*	*0.167*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	**0.033**	**0.073**	**0.052**	**0.046**
	$\Delta\mathbf{r}\uparrow$	*0.603*	*0.435*	*0.238*	*0.169*
Ours	$\Delta\mathbf{e}\downarrow$	**0.031**	0.031	0.026	0.055
	$\Delta\mathbf{r}\uparrow$	*0.278*	*0.294*	*0.129*	*0.102*

(e) StyleGAN2 $\mathcal{W}$

		Gl	Ge	S	A
IfGAN	$\Delta\mathbf{e}\downarrow$	0.040	**0.030**	**0.025**	**0.021**
	$\Delta\mathbf{r}\uparrow$	*0.207*	*0.245*	*0.132*	*0.092*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.065	0.062	0.024	0.077
	$\Delta\mathbf{r}\uparrow$	*0.417*	*0.229*	*0.248*	*0.145*
Ours	$\Delta\mathbf{e}\downarrow$	**0.036**	**0.037**	**0.009**	**0.051**
	$\Delta\mathbf{r}\uparrow$	*0.383*	*0.303*	*0.287*	*0.146*

(f) StyleGAN3 $\mathcal{W}$

StyleGAN models The results for $\mathcal{Z}$ space, presented in Tables 3.4a to 3.4c exhibit similar trends to those observed in PGGAN. The $\mathcal{W}$ space being less entangled than $\mathcal{Z}$ space, the results of IfGAN improve in that space as shown in Tables 3.4d to 3.4f. The qualitative results in Fig. 3.6 also illustrate enhanced editing, as evident for the attribute 'glasses'. Our method still reaches slightly better results than IfGAN for some attributes, especially for rarer attributes. In Fig. 3.7a, we present quantitative results for 'pale skin', 'wavy hair' and 'narrow eyes' for the StyleGAN3 model. We notice that the attribute 'narrow eyes' is not naturally disentangled from 'smile' in $\mathcal{W}$ space. Our method allows to significantly reduce the entanglement with this attribute as shown in Fig. 3.7b. Fig. 3.8 shows additional qualitative results with intermediate α values for StyleGAN3 in $\mathcal{W}$ space.

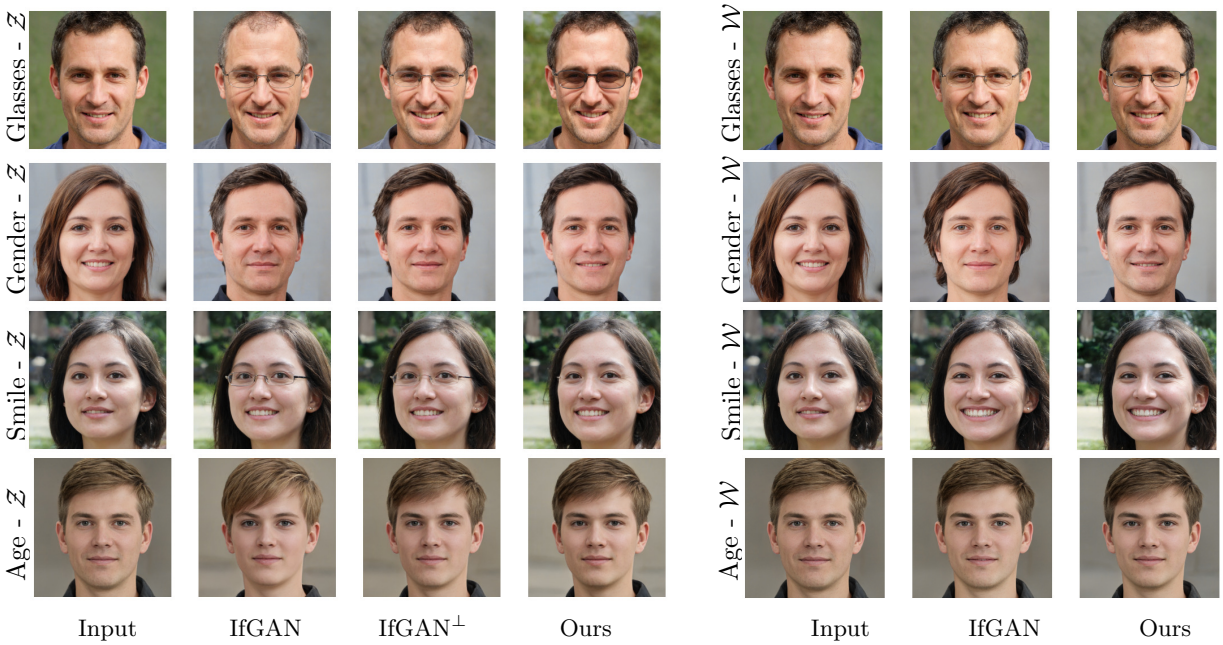
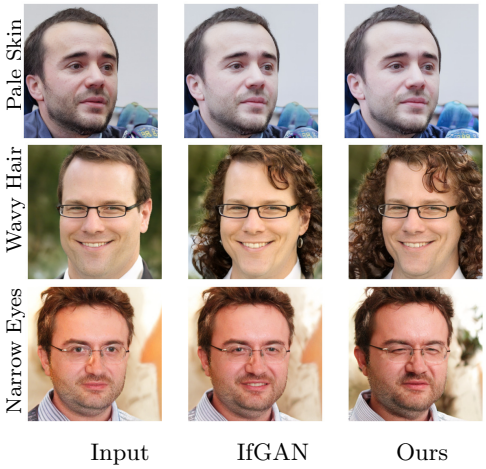


Figure 3.6: Editing results for StyleGAN2 for attributes Glasses, Gender, Smile and Age.

StyleGAN3 $\mathcal{Z}$

		Pale S.	Wavy H.	Narrow E.
IfGAN	$\Delta\mathbf{e}\downarrow$	0.106	0.180	0.165
	$\Delta\mathbf{r}\uparrow$	*0.277*	*0.302*	*0.234*
IfGAN$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.036	**0.068**	0.096
	$\Delta\mathbf{r}\uparrow$	*0.251*	*0.273*	*0.208*
Ours	$\Delta\mathbf{e}\downarrow$	**0.027**	0.083	**0.059**
	$\Delta\mathbf{r}\uparrow$	*0.308*	*0.341*	*0.239*
StyleGAN3 $\mathcal{W}$				
IfGAN	$\Delta\mathbf{e}\downarrow$	0.038	0.059	0.120
	$\Delta\mathbf{r}\uparrow$	*0.256*	*0.153*	*0.196*
Ours	$\Delta\mathbf{e}\downarrow$	**0.035**	**0.032**	**0.055**
	$\Delta\mathbf{r}\uparrow$	*0.252*	*0.260*	*0.245*

(a) Re-scoring results.



(b) Qualitative results.

Figure 3.7: Results for StyleGAN3 (rare attributes). (a) For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect. We highlight the best results among disentangling methods (IfGAN$^{\perp}$, Ours). 'Pale skin' is balanced w.r.t. 'gender' and 'age', 'wavy hair' w.r.t. to 'gender' and 'narrow eyes' w.r.t. 'smile'.

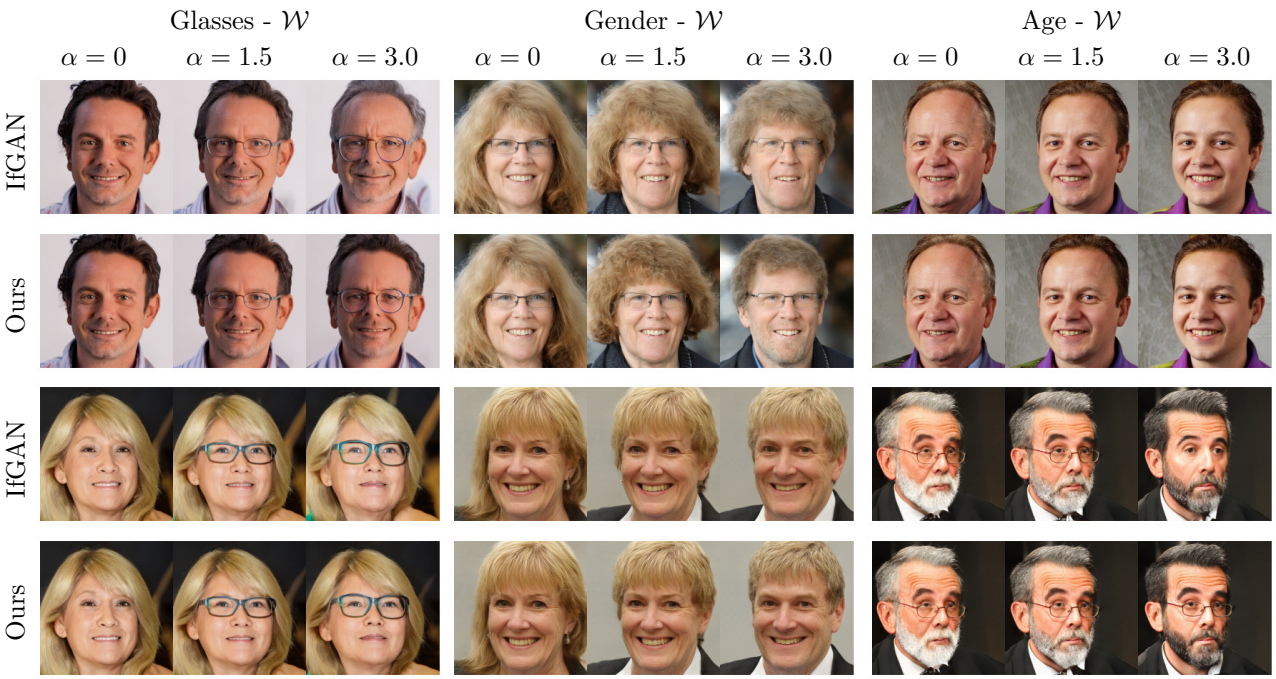


Figure 3.8: Editing results for StyleGAN3 in $\mathcal{W}$ for attributes Glasses, Gender and Age.

Impact of Sample Size We study the influence of using a larger sample size to estimate the directions. The directions are calculated with a sample of size $N_0 = 10000$ (instead of $N_0 = 1000$). For a larger sample size, the distributions are no longer well-balanced as potentially many cells of the contingency tables have been emptied. As shown in Table 3.5, the entanglement for the attribute 'glasses' remains quite low but the entanglement for the attribute 'gender' and 'age' is quite high, almost similar to IfGAN. To mitigate this effect, we propose to oversample the rarest samples as mentioned in Section 3.3.1. The results (Ours* in Table 3.5) show that it allows to decrease the entanglement and to finally reach a similar entanglement as for the directions estimated with a sample of size $N_0 = 1000$. We also find that the directions seem to have more effect when using a larger sample size. Following these observations, we argue that a large sample size (as in [127]) is not necessary to obtain meaningful directions. Nevertheless, oversampling allows to increase sample size for a stronger effect, while keeping a low entanglement.

Table 3.5: Re-scoring results for a higher sample size $N_0 = 10K$ for different models: PGGAN, StyleGAN3 in $\mathcal{Z}$ and in $\mathcal{W}$, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect, (*) indicates oversampling.

	(a) PGGAN				(b) StyleGAN3 $\mathcal{Z}$				(c) StyleGAN3 $\mathcal{W}$			
	Gl	Ge	S	A	Gl	Ge	S	A	Gl	Ge	S	A
IfGAN $\Delta\mathbf{e}\downarrow$	0.238	0.141	0.034	0.137	0.167	0.196	0.043	0.183	0.078	0.087	0.030	0.075
$N_0 = 10^4$ $\Delta\mathbf{r}\uparrow$	*0.488*	*0.560*	*0.411*	*0.149*	*0.570*	*0.464*	*0.346*	*0.192*	*0.527*	*0.305*	*0.292*	*0.152*
Ours $\Delta\mathbf{e}\downarrow$	0.099	0.099	0.017	0.108	**0.040**	0.114	0.042	0.123	0.037	0.066	0.033	0.072
$N_0 = 10^4$ $\Delta\mathbf{r}\uparrow$	*0.415*	*0.543*	*0.407*	*0.140*	*0.523*	*0.444*	*0.341*	*0.184*	*0.414*	*0.321*	*0.299*	*0.152*
Ours* $\Delta\mathbf{e}\downarrow$	**0.055**	**0.035**	**0.003**	**0.078**	0.044	**0.048**	**0.018**	**0.087**	**0.036**	**0.031**	**0.009**	**0.051**
$N_0 = 10^4$ $\Delta\mathbf{r}\uparrow$	*0.367*	*0.505*	*0.400*	*0.136*	*0.515*	*0.419*	*0.335*	*0.182*	*0.413*	*0.309*	*0.296*	*0.150*
Ours $\Delta\mathbf{e}\downarrow$	0.038	0.041	0.013	0.072	0.044	0.027	0.014	0.076	0.036	0.037	0.009	0.051
$N_0 = 10^3$ $\Delta\mathbf{r}\uparrow$	*0.286*	*0.448*	*0.370*	*0.129*	*0.398*	*0.377*	*0.305*	*0.172*	*0.383*	*0.303*	*0.287*	*0.146*

SVM vs. Centroids The calculation of the directions is usually performed using SVMs trained in latent space. We study the influence of the regularization parameter on the extracted directions. Table 3.6 shows that a stronger regularization (lower C) leads to smaller entanglement, while the effect on the target attribute remains almost unchanged. This observation led us to consider the case of a very large SVM margin, when the decision boundary becomes orthogonal to the direction connecting the centroids of the two classes (see Section 3.3.2). This direction gives the best performances.

Table 3.6: Re-scoring results for different boundary calculation methods (given a balanced sample) for different models: PGGAN, StyleGAN3 in $\mathcal{Z}$ and $\mathcal{W}$, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect.

	(a) PGGAN				(b) StyleGAN3 $\mathcal{Z}$				(c) StyleGAN3 $\mathcal{W}$			
	Gl	Ge	S	A	Gl	Ge	S	A	Gl	Ge	S	A
SVM $\Delta\mathbf{e}\downarrow$	0.118	0.080	0.011	0.113	0.069	0.108	0.060	0.114	0.051	0.054	0.021	0.054
$C = 1.0$ $\Delta\mathbf{r}\uparrow$	*0.326*	*0.500*	*0.382*	*0.136*	*0.423*	*0.3901*	*0.296*	*0.177*	*0.404*	*0.232*	*0.261*	*0.144*
SVM $\Delta\mathbf{e}\downarrow$	0.069	0.045	**0.010**	0.091	0.054	0.067	0.029	0.087	0.042	0.044	0.019	0.055
$C = 10^{-3}$ $\Delta\mathbf{r}\uparrow$	*0.323*	*0.471*	*0.379*	*0.133*	*0.431*	*0.391*	*0.310*	*0.176*	*0.409*	*0.296*	*0.291*	*0.155*
$\Delta\mathbf{e}\downarrow$	**0.038**	**0.041**	0.013	**0.072**	**0.044**	**0.027**	**0.014**	**0.076**	**0.036**	**0.037**	**0.009**	**0.051**
centroids $\Delta\mathbf{r}\uparrow$	*0.286*	*0.448*	*0.370*	*0.129*	*0.398*	*0.377*	*0.305*	*0.172*	*0.383*	*0.303*	*0.287*	*0.146*

Table 3.7: ID preservation results (higher the better) for StyleGAN models.

	(a) StyleGAN $\mathcal{Z}$				(b) StyleGAN2 $\mathcal{Z}$				(c) StyleGAN3 $\mathcal{Z}$			
	Gl	Ge	S	A	Gl	Ge	S	A	Gl	Ge	S	A
IfGAN	0.70	0.68	0.87	0.70	0.76	0.72	0.92	0.76	0.71	0.67	0.86	0.63
IfGAN$^{\perp}$	**0.77**	**0.77**	**0.89**	**0.79**	**0.85**	**0.78**	**0.93**	**0.87**	**0.80**	**0.76**	0.86	**0.74**
Ours	0.73	0.73	0.87	0.73	0.82	0.74	0.92	0.85	0.78	0.72	**0.87**	0.72
	(d) StyleGAN $\mathcal{W}$				(e) StyleGAN2 $\mathcal{W}$				(f) StyleGAN3 $\mathcal{W}$			
IfGAN	**0.72**	**0.70**	**0.78**	**0.77**	**0.91**	**0.88**	**0.92**	**0.93**	0.79	**0.85**	**0.92**	**0.82**
Ours	0.63	0.69	**0.78**	0.69	0.87	0.84	**0.92**	0.86	**0.81**	0.80	0.90	0.80

3.4.3 Identity Preservation

In addition to disentanglement, we evaluate the ability of our method to preserve identity. The results presented in Table 3.7 show that IfGAN reaches higher identity preservation (with conditional manipulation for $\mathcal{Z}$ space). Upon further analysis, this is a result of our directions having more effect than IfGAN's (cf. Table 3.4) as we found that identity preservation is negatively correlated with the effect of a direction. Indeed, although we would expect that the features obtained with a face recognition network trained on a dataset such as VGGFace2 [13] (9131 subjects, 362.3 images per subject with variation of pose, age, expression, accessories) would be invariant to the manipulated attributes (glasses, smile, age), this is not the case in practice. As illustrated for 'glasses' in Fig. 3.9, the identity preservation decreases as soon as the glasses appear and keeps decreasing as they become more visible. This metric should therefore be used when the directions have been calibrated to obtain comparable effects for a given α. However, calibrating directions is not straightforward and generally requires manual search [157].

(a) IfGAN (b) Ours

Figure 3.9: Identity preservation is negatively correlated with the effect of a direction. Above each figure is indicated the moving step α and the identity preservation result (id_p).

Table 3.8: Re-scoring results for PGGAN and StyleGAN3 in $\mathcal{Z}$. For each method, $\Delta\mathbf{e}$: overall entanglement, $\Delta\mathbf{r}$: effect.

(a) PGGAN $\mathcal{Z}$.

		Glasses	Gender	Smile	Age
Ours	$\Delta\mathbf{e}\downarrow$	**0.038**	**0.041**	**0.013**	**0.072**
	$\Delta\mathbf{r}$	*0.286*	*0.448*	*0.370*	*0.129*
Ours$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.092	0.068	0.017	0.083
	$\Delta\mathbf{r}$	*0.329*	*0.453*	*0.370*	*0.132*

(b) StyleGAN3 $\mathcal{Z}$.

		Glasses	Gender	Smile	Age
Ours	$\Delta\mathbf{e}\downarrow$	**0.044**	**0.027**	**0.014**	**0.076**
	$\Delta\mathbf{r}$	*0.398*	*0.377*	*0.305*	*0.172*
Ours$^{\perp}$	$\Delta\mathbf{e}\downarrow$	0.056	0.084	0.029	0.110
	$\Delta\mathbf{r}$	*0.411*	*0.383*	*0.305*	*0.179*

3.4.4 Orthogonality

We investigate if conditional manipulation applied on top of our directions can provide additional benefits. As shown in Table 3.8, it does not lead to further improvements. On the contrary, the entanglement slightly increases in comparison to our method alone. It shows that our directions are sufficiently orthogonal. As we show in Table 3.9, they are quasi-orthogonal even without enforcing it explicitly. The requirement of orthogonality did not have an *a priori* justification but our results indicate that orthogonality in latent space could be a necessary condition for independent controls. This can also explain the success of unsupervised methods that look for orthogonal directions in the latent space. For example, GANSpace [42] applies PCA in the $\mathcal{W}$ space and the authors are able to assign semantic concepts to the resulting directions (orthogonal by definition).

(a) PGGAN $\mathcal{Z}$.

Gl	Ge	S	A
1.00	-0.11	0.00	0.14
	1.00	0.04	0.09
		1.00	0.05
			1.00

(b) StyleGAN3 $\mathcal{Z}$.

Gl	Ge	S	A
1.00	-0.07	-0.04	0.05
	1.00	0.1	0.13
		1.00	0.08
			1.00

Table 3.9: Cosine similarities between directions.

3.5 Conclusion

We focused on the identification of directions in the latent space of a GAN to control semantic attributes of the generated images. Our assumption was that the entanglement typically observed in such situations results from strong correlations among attributes in the training data, that are

transferred to the generated data. To address this issue, we proposed a simple and general method that balances the data among the different combinations of values for the attributes. We believe that our subsampling approach can prove beneficial to other works on GAN control that rely on sampling in the latent space.

Three issues could be raised. First, as in most works on finding supervised controls, we use pseudo-labels provided by image classifiers that are assumed reliable. But they can also be affected by bias, with an impact on both the labelling of the training set and the evaluation since re-scoring depends on the classifiers. However, results on FFHQ show that even classifiers trained on smaller datasets like CelebAHQ transfer quite well. Second, using classifiers to find directions assumes that samples can be grouped in classes. This nevertheless works surprisingly well for binarized continuous attributes (*e.g.* 'age') and might not be a problem in practice. Finally, our method only balances known attributes. Entanglements due to representation biases of unlabeled attributes can remain, and in rare occasions, be worsened by the oversampling of rare combinations. This underlines that the set of attributes should be chosen and labeled carefully to achieve fair and unbiased editing.

Several subsequent studies argue against the simplistic global and linear assumptions introduced in IfGAN and followed in this work, suggesting the latent space of GANs might be non-linear in practice [150, 3, 157, 51]. Instead of employing binary and linear latent classifiers, the later approaches leverage non-linear and multi-attribute models such as Normalizing Flows [3] or Neural Networks [157, 51, 150, 171]. While these methods showcase state-of-the-art editing performances, their robustness can be undermined by the inherent limitations of the models they rely on. In the following chapter, we thus aim at proposing a method that does not rely on any external model but still achieves high editing performances.

Chapter 4

Robust Image Editing with Pre-trained GANs

Contents

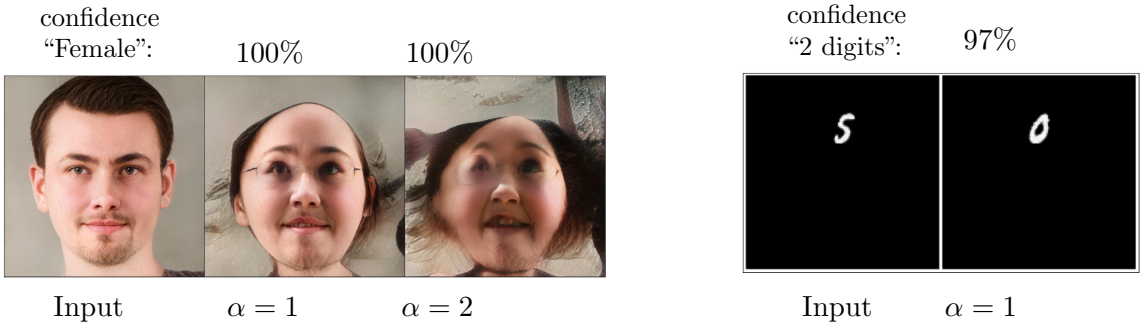
4.1 Introduction **49**
4.2 Related work **50**
4.3 Wasserstein Guidance for GAN Editing **52**
4.3.1 Overview 52
4.3.2 Wasserstein Distance 52
4.3.3 Core Objective 54
4.3.4 Disentanglement Objective 55
4.4 Experiments **55**
4.4.1 Experiments Setup 56
4.4.2 Main Results 58
4.4.3 Ablation: Explicit Disentanglement 59
4.4.4 Application to Count Editing 61
4.5 Conclusion **62**

4.1 Introduction

In this chapter, we investigate approaches that leverage the guidance of non-linear classifiers to learn effective semantic directions and enforce disentanglement. In comparison to the previous chapter, these methods model the latent transformations as local and non-linear directions leading to improved editing. The key idea is that manipulated latent codes (or the images they produce) shift the predictions to match the desired outcome [51, 157, 171]. However, classifiers can easily be fooled [93] *e.g.* they can associate high confidence predictions to out-of-distribution regions. Without regularization, the learned transformations may move latent codes from in-distribution to unknown regions, thus producing unrealistic images (cf. Fig. 4.1a). On the other hand, a regularization to limit the magnitude of the change can lead to create adversarial [137] codes where edited images do not exhibit the desired change although the classification objective is satisfied (cf. Fig. 4.1b). To prevent these issues, we formulate learning semantic edits in latent space as an optimal transport problem [146], leading to a core solution that *does not* rely on classifiers.

Given a distribution of latent codes sharing some semantics, we propose to transport it onto the distribution of latent codes that share the same semantics except for the attribute to be edited. Since the resulting images should not exhibit any other changes than the desired one, the initial points should be transported "close" to points sharing their semantics; that is, the transport should be optimal w.r.t. a cost representing the perceptual similarity. We assume that the Euclidean distance between latent codes reflects this similarity. We thus learn transformations in latent space using the guidance of the Wasserstein loss with a Euclidean cost. To enforce disentanglement explicitly, we show that this loss can be combined with a Wasserstein loss with a cost computed in the attribute space.

We demonstrate the effectiveness of our method on the editing of facial attributes in real face images, using the intermediate latent space of StyleGAN2. We provide quantitative and qualitative results showing that our approach is competitive with the classifier-based approach Latent Transformer (LT) [157] without requiring additional regularization to ensure realistic editing. We also show that our method can be applied to the editing of the number of MNIST digits [76] in real images from Multi-digit MNIST [135].



(a) FFHQ [64], 'Male' → 'Female' (b) Multi-digit MNIST [135], '1 digit' → '2 digits'

Figure 4.1: Limitations of classifier-based guidance as employed in the method of Yao et al. [157]. Qualitative results for two editing tasks: (a) when attempting to change the gender on FFHQ, the obtained images are unrealistic. (b) when attempting to add one digit in a Multi-digit MNIST image, the number of digits is unchanged in the edited image.

4.2 Related work

As seen in the previous chapter, InterfaceGAN [127] learns global and linear directions in the latent space using the decision boundaries of classifiers. Subsequent supervised methods learn non-linear transformations that take as input the initial latent codes, using the guidance of pre-trained multi-attribute classifiers [171, 51], as illustrated in Fig. 4.2. For each attribute a_k to be edited and the associated latent transformation $\mathbf{T}_k$, the edition objective is formulated as a binary cross-entropy:

$$\mathcal{L}_{cls} = y_k \log(\mathbf{p}_k) - (1 - y_k) \log(1 - \mathbf{p}_k), \quad \mathbf{p}_k = \Psi_k(G(\mathbf{T}_k(\mathbf{z}, \alpha))), \quad y_k \in \{0, 1\} \tag{4.1}$$

where Ψ is the pre-trained multi-attribute classifier, y_k the target class and α the scalar editing strength that are set according to the targeted editing. For negative latent codes w.r.t. to a_k, y_k is generally set to 1 with $\alpha = 1$ while for positive latent codes y_k is set to 0 with $\alpha = -1$. The goal is thus to learn the transformation that leads to images being classified as having (if they did not have it initially) or not having (if they did have it initially) the attribute. To reduce the computational cost, the state-of-the-art editing method Latent Transformer (LT) [157] proposed to employ a multi-attribute classifier operating directly in latent space: $\mathbf{p}_k = \Psi_k(\mathbf{T}_k(\mathbf{z}, \alpha))$. In addition, Yao et al. leverage this classifier to introduce an explicit disentanglement constraint. The idea is to enforce that the predictions for non-target attributes remain close to the predictions for the initial latent code by

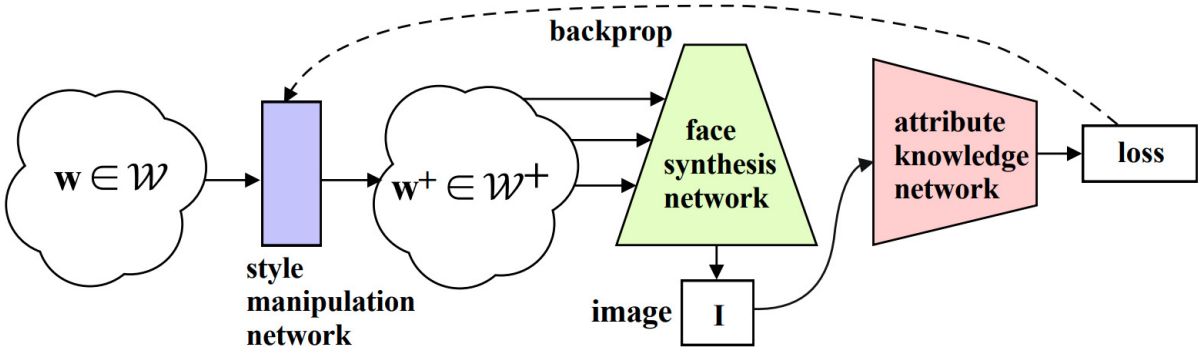


Figure 4.2: Classifier-based guidance for learning semantic edits. Various approaches such as GuidedStyle [51] employ an attribute classifier (or attribute knowledge network) with a classification loss as supervision to learn latent transformations. Figure taken from [51].

minimizing an $L2$ loss between the logits vectors before and after the editing:

$$\mathcal{L}_{\text{attr}} = \sum_{i \neq k} (1 - \gamma_{ik}) \|\mathbf{p}_i - \Psi_i(\mathbf{z})\|_2^2, \quad \mathbf{p}_i = \Psi_i(\mathbf{T}_k(\mathbf{z}, \alpha)) \tag{4.2}$$

where the term γ_{ik} is the absolute correlation between a_i and a_k computed on the training set, and is used to avoid disentangling naturally correlated attributes (*e.g.* 'smile' and 'high cheekbones').

Contrary to previous works relying on simpler models *e.g.* linear SVMs, these methods employ deep classifiers, Ψ is generally composed of various convolutional or fully-connected layers with non-linearities. This type of classifiers have been extensively studied and are known to lack robustness *e.g.* show sensitivity to small perturbations [137] and lack of effective uncertainty estimation [93]. Such drawbacks have a direct impact on editing methods that rely on these models. In particular, we observed that when using the classification objective of Eq. (4.1) together with the disentanglement constraint of Eq. (4.2), the manipulated latent codes produce unrealistic images (cf. Fig. 4.1a), although the latent classifier is highly confident (100% confidence). To address this issue, the authors of LT employ an *ad hoc* $L2$-regularization to minimize the norm of the latent editing:

$$\mathcal{L}_{\text{reg}} = \|\mathbf{T}_k(\mathbf{z}, \alpha) - \mathbf{z}\|_2^2 \tag{4.3}$$

While this fixes out-of-distribution edits, this regularization produces adversarial samples [137] on a dataset outside the face domain *i.e.* Multi-digit MNIST [76, 135]. As shown in Fig. 4.1b, the edited latent codes are correctly classified (with 97% of confidence) but the corresponding images remain unchanged (no digit is added).

To avoid the need for regularization as a consequence of relying on brittle classifiers, we propose an approach based on the optimal transport framework with a core objective that does not rely on classifiers. To the best of our knowledge, this is the first work applying optimal transport for latent space editing.

4.3 Wasserstein Guidance for GAN Editing

4.3.1 Overview

As in Chapter 3, we consider a pre-trained GAN G and a collection of latent codes $\mathcal{S} = \{(\mathbf{z}^{(i)})_{i=1}^{N}\}$, where each code is associated with a set of binary semantic attributes $\mathcal{A} = \{a_1, a_2, .., a_K\} \in \{0, 1\}$. For a given attribute a_k, we aim to learn an affine transform $\mathbf{H}_k$ in $\mathcal{Z}$,

$$\mathbf{z}'_k = \mathbf{z}_k + \alpha \cdot \mathbf{H}_k(\mathbf{z}_k),\ \alpha \in \mathbb{R} \tag{4.4}$$

such that only the intensity of attribute a_k varies in the generated image $G(\mathbf{z}'_k)$.

Let μ_s^k be the distribution of latent codes $\mathbf{z}_k$ that are negative with respect to the binary attribute a_k and μ_t^k the distribution of latent codes $\bar{\mathbf{z}}_k$ positive w.r.t. to a_k (cf. Fig. 4.3). To increase the intensity of the attribute a_k in the generated images, $\mathbf{H}_k$ should transport the distribution of edited latent codes $\mathbf{z}'_k$ denoted by $\mu_s'^k$ close to the distribution μ_t^k. However, the information encoding other attributes or properties should remain unchanged. The theory of optimal transport [146] introduces a framework to transport a distribution to another with a minimal cost. The Wasserstein distance between two distributions represents the minimal value of this cost. Thus, we propose to use this loss as supervision to learn $\mathbf{H}_k$ with a cost in latent space expressing similarity in image space. We present our model denoted by Latent Wasserstein (LW) in the following. First, we give a more formal definition of the Wasserstein distance. Then, we define two objectives based on this distance: the core edition objective and an optional disentanglement objective.

4.3.2 Wasserstein Distance

Let us define two discrete distributions:

$$\mu_s = \sum_{i=1}^{n_s} p_i^s \delta(x_i) \text{ and } \mu_t = \sum_{j=1}^{n_t} p_j^t \delta(y_j) \tag{4.5}$$

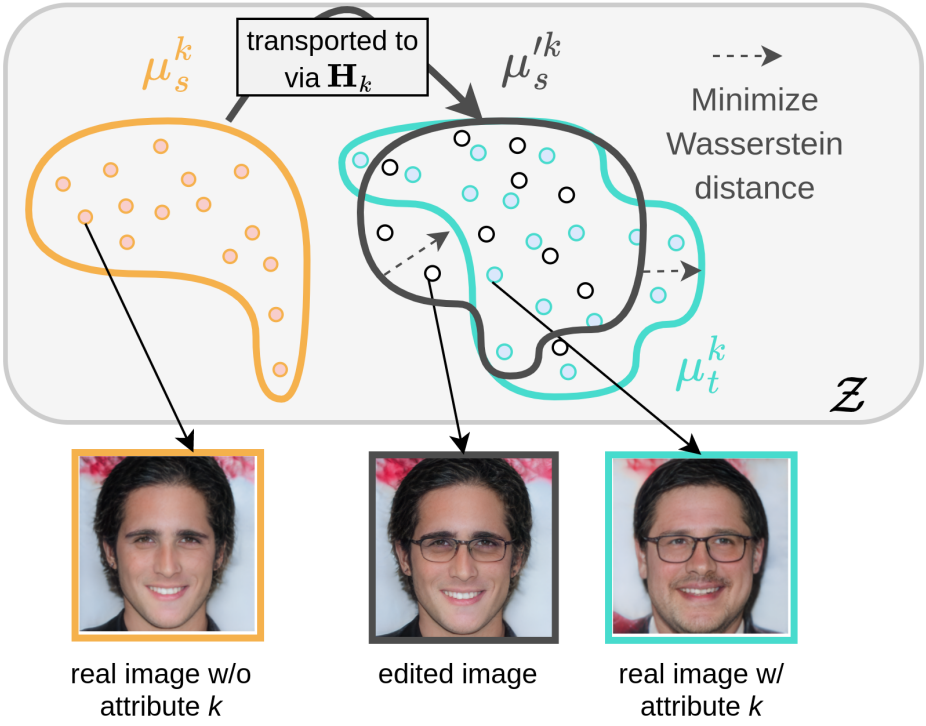


Figure 4.3: Method overview. For each semantic attribute a_k (*e.g.* 'glasses') we learn a mapping $\mathbf{H}_k$ that moves the distribution of latent codes lacking the attribute to the distribution of codes having that attribute. We enforce that each latent code is moved near a point that shares similar semantics, thus only changing that attribute. To preserve identity, the resulting distribution does not entirely match the target distribution.

where $\delta(,)$ is the Dirac function and p_i^s, p_j^t the probability mass associated with each sample.

The Wasserstein distance between μ_s and μ_t is defined as:

$$\begin{aligned} &W(\mu_s, \mu_t) = \min \sum_{i,j} T_{i,j} c_{i,j} \\ &\text{s.t. } T\mathbf{1}_{n_t} = \mu_s, \\ &\quad T^\top \mathbf{1}_{n_s} = \mu_t \quad \text{where } T \text{ is the transport matrix.} \end{aligned} \tag{4.6}$$

$T_{i,j}$ represents how much probability mass must be transported from point x_i to point y_j and $c_{i,j}$ corresponds to the cost of this transport, *e.g.* the Euclidean distance between x_i and y_j.

4.3.3 Core Objective

To obtain latent codes corresponding to images that exhibit the desired attribute a_k, we aim to learn $\mathbf{H}_k$ such that the resulting distribution ($\mu_s'^k$) is close to the distribution of codes having that attribute (μ_t^k). To maintain other attributes, we also seek that each point be close to points that share these attributes. Assuming that the Euclidean distance in latent space is a proxy for semantic distance in image space, we define our main objective as minimizing the Wasserstein distance between $\mu_s'^k$ and μ_t^k with a squared Euclidean cost function:

$$\mathcal{L}_{\text{edit}} = W\left(\mu_s'^k, \mu_t^k\right), \quad x_i = \mathbf{z}'^{(i)}_k \text{ and } y_j = \bar{\mathbf{z}}^{(j)}_k$$
$$c_{i,j} = \frac{1}{2}\|x_i - y_j\|^2 \tag{4.7}$$

In Eq. (4.5), the probability mass of each sample is usually set uniformly across samples and sums to 1, *i.e.*:

$$\mu_s = \sum_{i=1}^{n_s} p_i^s \delta(x_i) = \sum_{i=1}^{n_s} \frac{1}{n_s}\delta(x_i) \text{ and } \mu_t = \sum_{j=1}^{n_s} p_j^t \delta(y_j) = \sum_{j=1}^{n_t} \frac{1}{n_t}\delta(y_j) \tag{4.8}$$

As seen in Chapter 3, the collection of training latent codes often contains various biases that generate an imbalance between the attributes joint distribution of $\mu_s'^k$ and the attributes joint distribution of μ_t^k. If the probability mass of each sample is the same across samples, the total mass of semantically similar samples may vary significantly between the two distributions. This imbalance ultimately leads to transport source samples to target samples with different attributes, thus causing entanglement. To prevent this, we propose a non-uniform weighting scheme for the source samples so that the total mass (or importance) of samples that share the same semantics is equivalent between the two distributions. More formally, we consider semantically similar codes, the ones that share the same combination of attributes *e.g.* $A = (a_1 = 0, ..., a_i = 1, ..., a_K = 1)_{\{i \neq k\}}$. Then, given n_s^A and n_t^A, the number of samples with the combination A in the source and target distribution respectively, we want to have:

$$\sum_{i=1}^{n_s^A} p_i^s = \sum_{j=1}^{n_t^A} \frac{1}{n_t} \tag{4.9}$$

Thus we set:

$$p_i^s = \frac{n_t^A}{n_s^A \times n_t} \tag{4.10}$$

4.3.4 Disentanglement Objective

To ensure that the transported latent codes share the same attributes - except for the target attribute - as the initial ones, we introduce a regularization that enforces to stay close to the original distribution (μ_s^k) in the space of non-target attributes. We thus propose to minimize a preservation loss that is the Wasserstein distance between $\mu_s'^k$ and μ_s^k with a cost computed in the *attribute* space:

$$\begin{aligned} \mathcal{L}_{\text{pres}} &= W\left(\mu_s'^k, \mu_s^k\right), \quad x_i = \mathbf{z'}_k^{(i)} \text{ and } y_j = \mathbf{z}_k^{(j)} \\ c_{i,j} &= \frac{1}{2}\sum_{l \neq k}(1-\gamma_{lk})\|\Psi_l(x_i) - \Psi_l(y_j)\|_2^2 \end{aligned} \tag{4.11}$$

where Ψ is a multi-attribute latent classifier.

The cost function follows the explicit disentanglement constraint introduced in [157] and defined in Eq. (4.2), but our constraint is more relaxed since we operate on the distribution. This second objective involves a classifier, however, it is an optional objective as our core objective already addresses disentanglement since we assume close latent codes in terms of Euclidean distance share semantic properties in image space. We introduce this cost to tackle the case where, even though we are mapping semantically similar points, the distribution of non-target attributes in the target distribution may slightly differ.

The final objective to minimize is then:

$$\mathcal{L} = \mathcal{L}_{\text{edit}} + \lambda \mathcal{L}_{\text{pres}} \tag{4.12}$$

where λ allows to control the strength of the regularization.

4.4 Experiments

We conduct experiments on the task of facial attributes and number of MNIST digits editing in real images. Face editing results are presented in Sections 4.4.2 and 4.4.3: we compare our approach with Latent Transformer (LT) [157] that relies on the guidance of a classifier in the latent space. The comparison is conducted for common ('glasses', 'gender', 'smile', 'age') and rarer attributes ('pale skin', 'bangs', 'blond hair', 'wavy hair'). Attribute and identity preservation results are presented in Section 4.4.2, while the effect of the enforced disentanglement constraint is explored in Section 4.4.3. Finally, in Section 4.4.4 we present the results for editing the number of digits in MNIST-like images.

4.4.1 Experiments Setup

Datasets and models We use two faces datasets: CelebAHQ and FFHQ and, Multi-digit MNIST [135]: a dataset of images with 1 to 4 MNIST digits. We apply the editing in the latent space of StyleGAN2 pretrained on FFHQ respectively Multi-digit MNIST. For the training data, we use real images projected into the $\mathcal{W}+$ latent space using the pSp encoder [116]. We employ respectively the $30k$ labeled 1024×1024 CelebAHQ images for face editing and $25k$ 128×128 Multi-digit MNIST images. The dimension of $\mathcal{W}+$ depends on the resolution of the input images (18×512 for face images and 12×512 for Multi-digit MNIST). Following [157], the latent classifiers Ψ used to guide the editing are 3-layer MLPs pre-trained in that space.

Implementation details To learn a transformation, we use the implementation of the Wasserstein loss provided by the GeomLoss library [32]. Estimating the Wasserstein distance is challenging in practice as it requires to solve the underlying optimal transport. The Wasserstein distance is commonly estimated with the Sinkhorn divergence built on entropic regularization with debiasing terms [32, 21]. We set the batch size as the minimum between the number of samples in the source and target distributions, and drop the last batch if it causes a strong imbalance between the two. We use Adam optimizer with a learning rate of 0.001. To avoid overfitting the target distribution, we perform early stopping on a held-out validation set. As the representation of the attributes is not well-balanced in CelebAHQ, we weight the samples according to Eq. (4.10) considering only the common attributes and use the preservation loss defined in Eq. (4.11) computed on all 39 other attributes of CelebA [83]. We determine the optimal value for λ according to the accuracy on the validation set. We found that 1 is the best value for all considered attributes except for 'glasses' ($\lambda = 15$).

Metrics We compute three metrics: the target change, attribute and identity preservation rates [157]. The *target change* and *attribute preservation* rates respectively relate to the *effect* and *overall entanglement* defined in the previous chapter but are computed in a slightly different manner. As before, the aforementioned metrics are computed by running pretrained attribute image predictors before and after the editing (for a given α) and finding which attributes have changed. In this case, the attributes probabilities are first binarized by considering an attribute present if the probability is greater than 0.5. Then, the metrics are computed as follows:

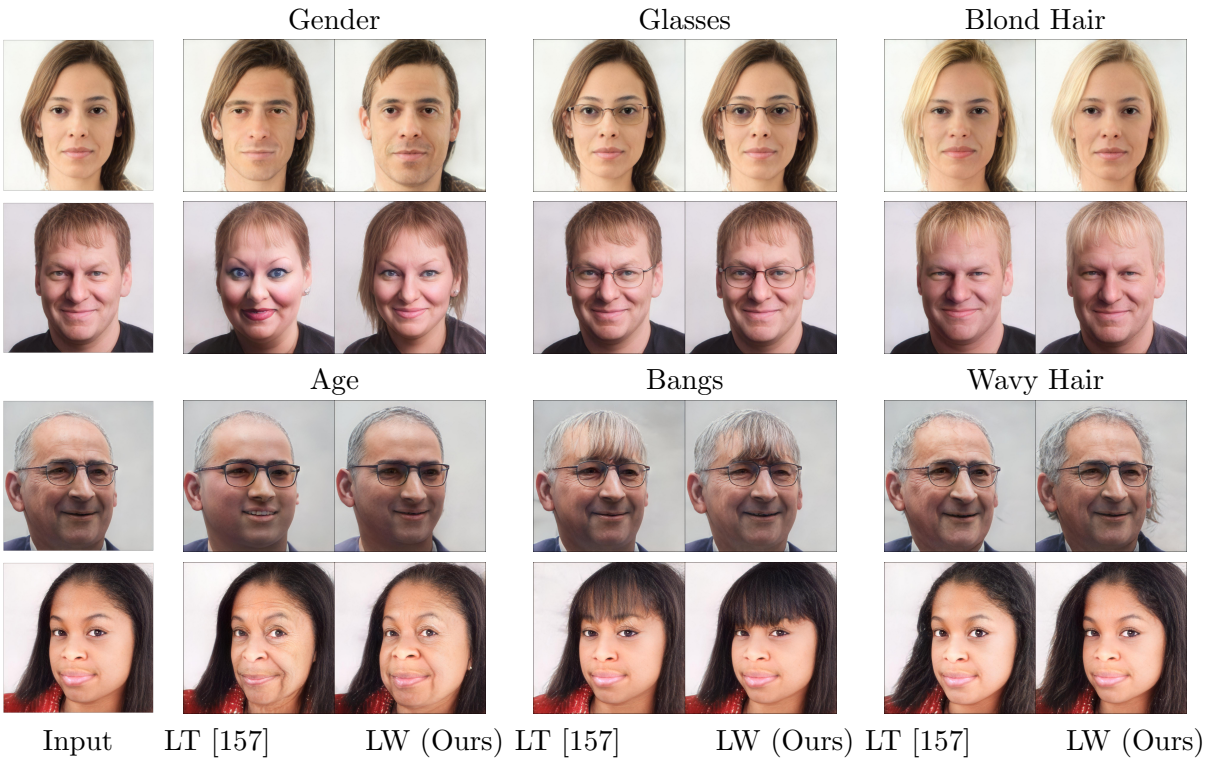


Figure 4.4: Qualitative results for facial attribute editing. We report the editing results for $\alpha = \pm 2$. We observe that our approach better preserves identity and some facial attributes (*e.g.* expression, absence of makeup) compared to Latent Transformer.

- **Target change**: Percentage of images for which the target attribute is indeed modified.
- **Attribute preservation**: Average rate of non-target attributes that are indeed preserved.
- **Identity preservation**: Average cosine similarities between ArcFace [23] features of input and edited images.

For facial attributes editing, all metrics are evaluated on 1000 images from FFHQ. The attribute and identity preservation rates are reported against the target change for 10 values of $\alpha \in [1 \cdot d, 2 \cdot d]$ where d is chosen such that the target change for a given α is comparable between the different methods (using manual search). In tables, we report the mean over all values of α. For Multi-digit MNIST, we measure only the target change (with $\alpha = 1$) using a ResNet-50 trained to predict the number of digits in an image.

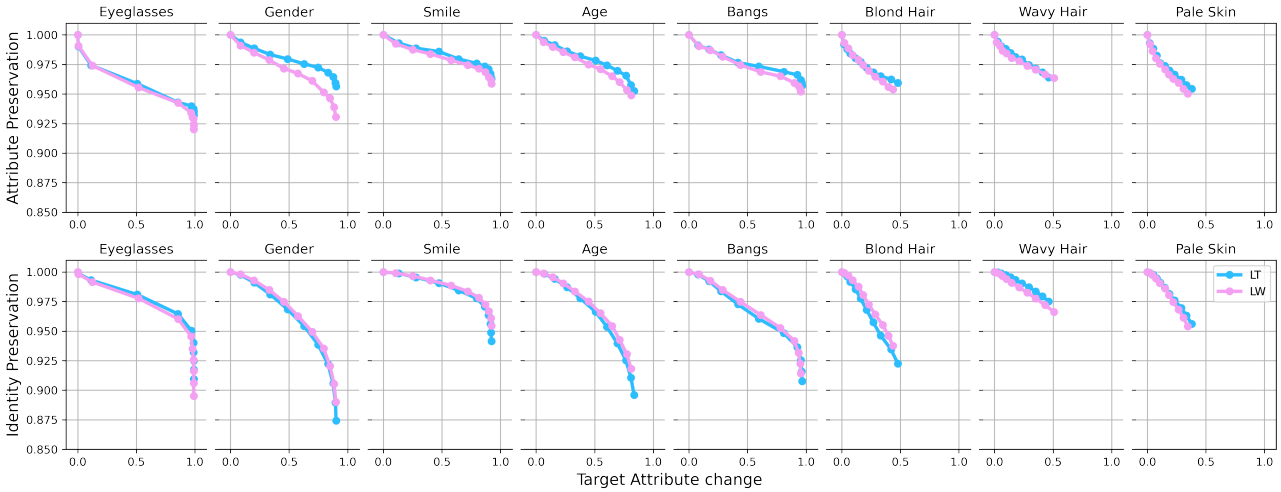


Figure 4.5: Quantitative results for facial attribute editing. We report the attribute preservation rate (computed on all other attributes indicated here) and the identity preservation rate for different values of α (points of the curves). The x-axis is the ratio of images (among all test images) for which the target attribute is successfully modified.

4.4.2 Main Results

Attribute preservation The quantitative results from Fig. 4.5 (top) show that our results are on par with LT with occasionally slightly lower performances (*e.g.* 'gender'). From the qualitative results of Fig. 4.4, it may be due to the non-preservation of hair (*i.e.* our 'gender' editing tends to grow hair for women and remove hair for men) which we attribute to having too few women/men with short/long hair in the target distribution[1]. On the other hand, we observe that LT 'gender' editing is heavily entangled with 'makeup' while LW adds nearly no such entanglement.

Identity preservation The quantitative results from Fig. 4.5 (bottom) show that we achieve slightly higher performances (*e.g.* 'gender', 'age', 'blond hair') for identity preservation. As shown in the qualitative results presented in Fig. 4.4, the nose, lips and eyes shape are much better preserved for 'gender' and 'age' (1-3rd row of first column). This ability to preserve identity is surprising as we do not enforce it explicitly, and the source and target distributions contain different individuals. This is possibly a result of the early stopping, that prevents us from over-fitting our edited codes on the target distribution. It could also be due to the inductive bias of the model, which defines edits as simple affine transformations in the latent space thus acting as a regularization.

[1]The hair length is not annotated in CelebA.

Table 4.1: Quantitative results for attributes 'gender' (G), 'age' (A) and 'pale skin' (PS). We compare the classification approach (LT) with our Wasserstein approach (LW). Setting (*) is the "core" method, w/o any regularization.

	Attribute preservation			Identity preservation		
	Gender	Age	Pale Skin	Gender	Age	Pale Skin
LT$^{(*)}$	0.92	0.95	0.86	**0.94**	0.96	0.96
LW$^{(*)}$	**0.95**	**0.96**	**0.96**	**0.94**	**0.97**	**0.98**
LT	**0.98**	**0.98**	**0.98**	0.95	0.96	**0.98**
LW	0.97	**0.98**	0.97	**0.96**	**0.97**	**0.98**

4.4.3 Ablation: Explicit Disentanglement

Without regularization We evaluate the ability of both methods to achieve disentangled and identity preserving editing without any explicit constraint. We denote by LT$^{(*)}$ the model trained without the disentanglement loss from Eq. (4.2) nor the $L2$-regularization from Eq. (4.3). In Table 4.1 (top), we compare with LW$^{(*)}$ our model trained without the preservation loss (in Eq. (4.12) we set $\lambda = 0$). Our Wasserstein approach outperforms the classifier baseline both regarding disentanglement and identity preservation. As shown in the qualitative results of Fig. 4.6, the latter produces highly entangled edits (*e.g.* with the attribute 'smile') and alters the identity. Without enforcing it explicitly, our method already exhibits a good disentanglement ability and the identity is also well-preserved. A tentative explanation is that the Euclidean cost used in Eq. (4.7) indeed fairly reflects the perceptual distance.

With regularization We study the influence of adding the disentanglement constraint from Eq. (4.11). As shown in Table 4.1, we improve attribute preservation. Qualitatively, the results are also improved as shown in Fig. 4.6. The 'gender' attribute is no longer heavily entangled with 'beard' (1st row) and the slight entanglement with 'smile' is removed. As shown in Fig. 4.7, when the disentanglement constraint is used alone in the classifier-based approach, the edited images are unrealistic. The decision boundaries of classifiers might cover areas that are larger than the area of training samples, hence latent codes which are far away from the training distribution can still minimize the classification loss. The $L2$-regularization in [157], enforcing that the edited latent codes remain close to the initial ones, is thus necessary to circumvent this limitation. Our method does not require any regularization to produce realistic edits, since our main objective enforces closeness to the target distribution.

Figure 4.6: Qualitative comparison between classifier-based edits (2^{nd} col.) and our Wasserstein-based edits w/o any reg. (3^{nd} col.) and w/ disentanglement reg. (4^{th} col.).

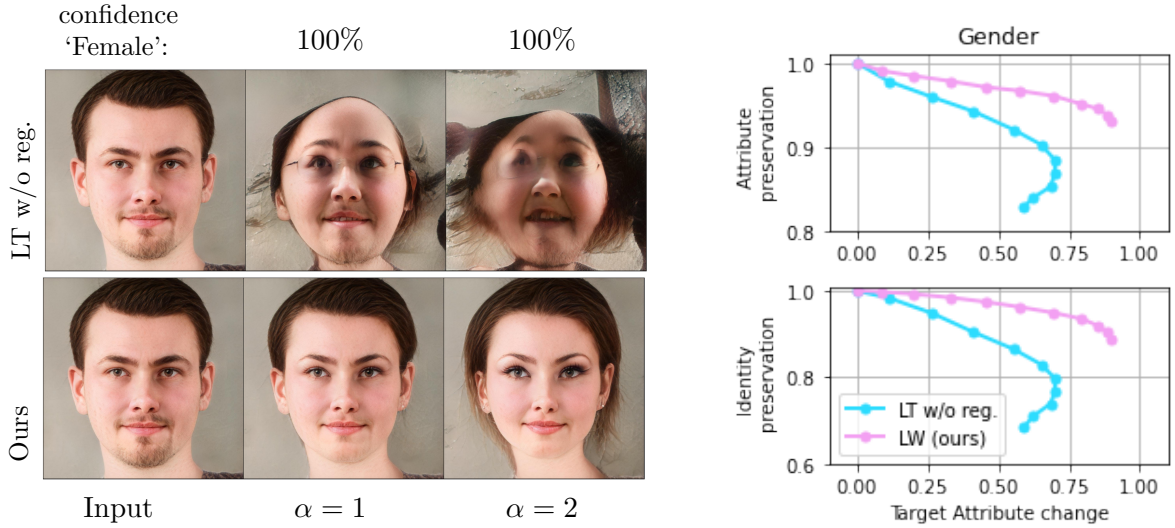


Figure 4.7: Results for 'gender' editing without the $L2$-regularization on the edited codes for LT.

4.4.4 Application to Count Editing

In this section, we apply the editing to data outside the face domain. We compare LT to LW by applying three different changes of the number of digits (1→2, 2→3 and 3→4) in Multi-digit MNIST images. Each of the two methods aims to obtain the required change for all the input images (desired target change rate of 100%). To evaluate the ratio of *successfully* applied changes, we employ the output of a reliable image classifier (actual change rate) applied to the edited images. The quantitative results in Table 4.2 show that for a target change of 100% according to the latent classifier, the image classifier indicates significantly lower target changes for LT. In other words, the latent classifier predicts that the number of digits has increased while it has stayed the same in the image. This can be verified by visualizing the generated images. These observations could indicate that for this type of data, the structure of StyleGAN2's latent space is not as smooth as for face data, leading to more adversarial codes. Our method is still robust in that case with a high editing effect, actually adding the digits in the images. In addition, as shown in the qualitative results from Fig. 4.8, our method naturally preserves the position and the class of the digits initially present in the image.

Table 4.2: Quantitative results for the manipulations "adding one digit in an image containing n digits, for $n = 1, 2, 3$" in real images from MultiMNIST [135]. Given a change rate of 100% according to a latent classifier, we report the *actual* change rate as measured by an image classifier. Higher values indicate a lower rate of adversarial samples.

Method	Actual change rate		
	1→2	2→3	3→ 4
LT	0.32	0.31	0.64
LW (ours)	**0.90**	**0.95**	**0.99**

Discussion. While we observe many adversarial codes for Multi-digit MNIST, we didn't identify many cases for CelebAHQ or FFHQ. These datasets have different degree and type of semantics that might lead to different organization in the latent space. For Multi-digit MNIST, the notion of count seem to be encoded in the latent space but there are few semantics (*e.g.* white digits, plain background) otherwise thus the extended space $\mathcal{W}+$ might be disproportionately large. We found that when working in the native $\mathcal{W}$ space, we succeed to alter the number of objects using the guidance of classifiers (see more details in Appendix B.1) although there is no preservation of the initial digits in contrast to our method and it is thus only applicable to generated images (the inversion of real images is not always faithful in $\mathcal{W}$). We have also conducted experiments on other datasets similar to Multi-digit MNIST with richer semantics: CLEVR [60] and CLEVRTex [62]. For these datasets, we were unable to control the number of objects with either method. In addition, we observed that inversion methods that succeed on face images fail to appropriately inverse such images (cf. Fig. B.1 in Appendix B). This seems to indicate that the notion of count is not encoded at all in the latent space for such datasets. On the other hand, conditional approaches, that are explicitly trained with the count such as [122] successfully generate images based solely on the count.

4.5 Conclusion

We present a new method to learn semantic edits in the latent space of GANs, that proposes to model the problem as an optimal transport problem. We look for transformations that transport a collection of latent codes to the most semantically similar points in the distribution of latent codes with the desired semantic. We use the squared Euclidean distance in latent space as a cost function as it fairly reflects the perceptual distances in image space. This formulation readily produces almost totally disentangled editing whereas classifier-based methods require an explicit disentanglement constraint.

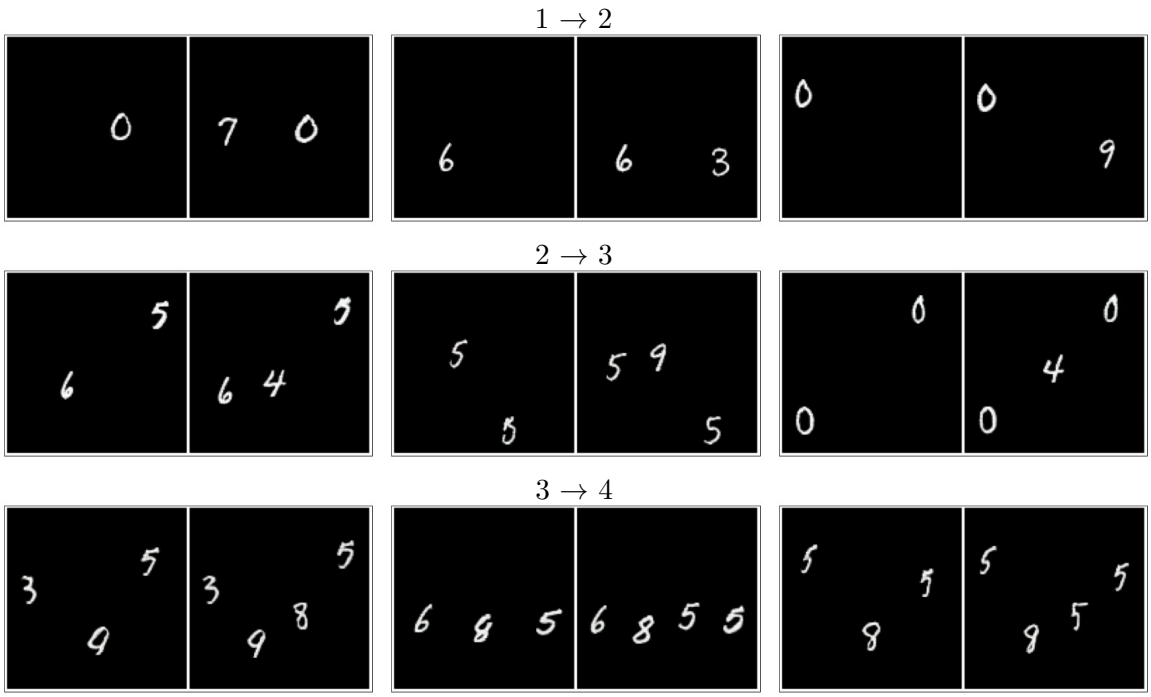


Figure 4.8: Qualitative editing results for MultiMNIST [135]. We show three examples per line. For each example, the first column corresponds to the unedited image. The second column corresponds to the edited image ($\alpha = 1$) with our method. We add one number each time, starting from one (top line), two (middle) or three (bottom) digit(s).

To achieve even more disentangled editing, we introduce an explicit loss enforcing the transported codes to remain close to the distribution of initial codes. This loss is also formulated with optimal transport but using a semantic cost computed in *attribute* space. On the task of facial attribute editing on CelebA/FFHQ, our method is competitive with a state-of-the-art classifier-based method without requiring an additional constraint to ensure that the obtained images are realistic. Our method also alleviates other issues from using classifiers, such as the sensitivity to adversarial examples as we illustrate on the editing of the number of digits in Multi-digit MNIST.

Our approach could benefit from few improvements. While the Wasserstein loss with a latent Euclidean distance results in state-of-the-art editing performances, this cost does not perfectly reflect the perceptual distance in image space. This could explain why some edits are not totally disentangled. We believe the performances could be further improved by using a cost based on the LPIPS metric [162] or an equivalent proxy learned in the latent space to reduce computation time. In addition, partial OT algorithms [15], that allow to transport only a set of points, could be more suited than

regular OT algorithms, in order to account for the unbalanced number of samples between the source and target distributions. Finally, it has been shown that some layers in the $\mathcal{W}+$ space [64, 156] control specific concepts. The transport could thus be applied to a well-chosen subset of layers, allowing to reduce the computational cost and potentially some residual entanglements (*e.g.* the hair).

In the previous chapter, we studied a simple approach that exploits robust linear classifiers but suffers from entanglement, that we mitigated with our balanced sampling approach. In this chapter, we explored a method that achieves state-of-the-art editing by leveraging non-linear classifiers to guide and explicit disentangle the editing. Despite its good results, some edited images end-up unrealistic as such classifiers can be unreliable. We then proposed a method that does not heavily depend on these models thus ensuring more robust editing while maintaining disentanglement. The robustness of the editing process is crucial for enabling the use of edited images in downstream tasks. Data augmentation indeed stands out as a promising application of editing as a means to fill in the gaps of existing databases. However, for this augmentation to be versatile across various tasks, it requires GANs capable of synthesizing a wide range of images and encoding of the desired semantics in the latent space. This may not be the case, as observed for the notion of count for certain datasets. More recently, DDPMs trained on large multi-modal databases have shown to synthesize highly diverse images whose content can be intuitively controlled by text prompts, including in the zero-shot setting. In the following chapter, we explore how these models and their controllability can be exploited for effective data augmentation.

Chapter 5

Data Augmentation with Text-to-Image Diffusion Models

Contents

5.1 Introduction **66**
5.2 Few-shot Object Counting **67**
5.3 Semantic Generative Augmentations **70**
5.3.1 Text-and-Density Guided Augmentations 70
5.3.2 Diversity-Enhanced Augmentations 71
5.4 Experiments **73**
5.4.1 Experiments Setup 73
5.4.2 Few-shot Counting on FSC147 74
5.4.3 Ablation Study 77
5.4.4 Counting with Fewer Shots 81
5.4.5 Generalization on CARPK 82
5.5 Qualitative results **84**
5.6 Conclusion **87**

5.1 Introduction

In the previous chapters, we focused on improving the control over semantic properties of the generated images while in this chapter, we explore how to exploit this control to generate effective and diversified synthetic data to augment real datasets. Recent works show that pre-trained text-to-image diffusion models generate data that can effectively improve classification performances on various datasets [45, 141, 129]. However, these methods fail to produce satisfying images for tasks that require more precise generation such as object detection and counting, semantic segmentation as these models struggle to follow prompts that include compositional concepts *e.g.* number of objects and position [103]. Some works tackle improving the understanding of compositionality in vision-language models [99, 77, 104] but are limited to small numbers of objects. Other works add more control to the pre-trained text-to-image models [161, 53, 91].

To generate accurate and useful augmentations for tasks that require precise generation, we propose to synthesize unseen data with Stable Diffusion conditioned on both a textual prompt and a task-specific condition. The double conditioning, implemented with ControlNet [161], allows us to generate novel synthetic images with a precise control, preserving the ground truth for the downstream task. To increase the diversity of the augmented training set, we swap image descriptions between the n available training samples, leading to $\frac{n(n-1)}{2}$ novel couples, each being the source of several possible synthetic images unseen in the training data. However, we show that many combinations do not make sense and lead to poor quality samples. Therefore, we introduce a strategy based on the descriptions' similarities to only select plausible pairs, resulting in improved augmentation quality.

We evaluate our approach by generating augmentations for the task of Few-shot Object Counting (FSC) [84, 114], that aims at teaching networks to accurately count objects in an image given few exemplars. We create an augmented dataset to train two state-of-the-art counting networks, namely SAFECount [158] and CounTR [14]. We show that our approach significantly improves their performances on the benchmark dataset FSC147 [109] and allows for a better generalization on the car-specific counting dataset, CARPK [52].

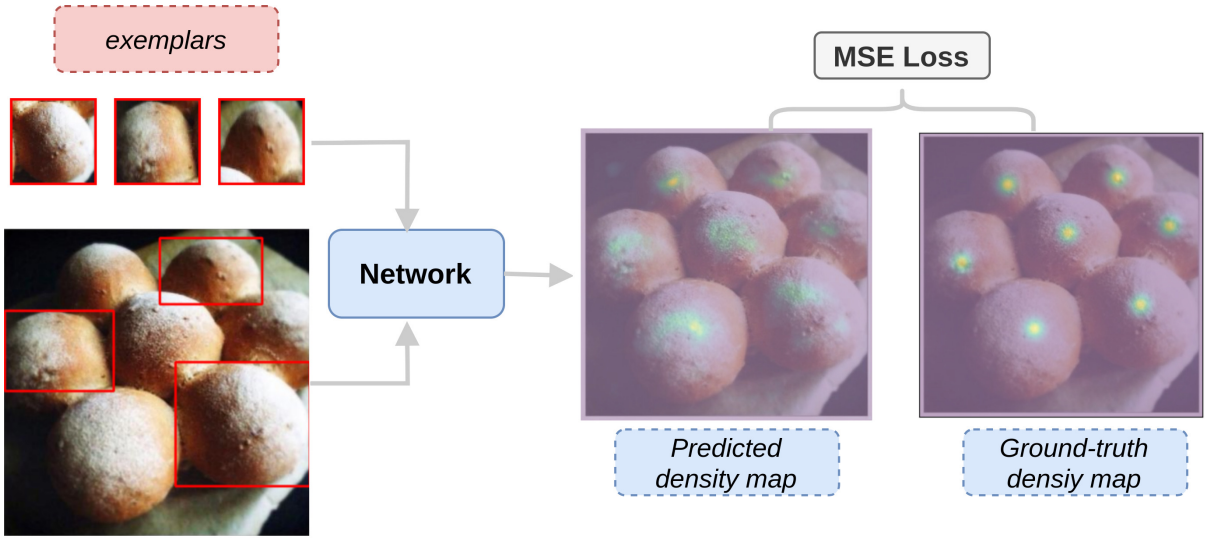


Figure 5.1: Few-shot Object Couting. FSC networks take as input the target image and few exemplars of the current type of objects to count (in red). Typically, the similarity map between the query image and exemplars features is computed and fed to a convolutional neural network that outputs a density map [114, 128]. This map is a spatial map indicating the probabilities of an object being present, whose sum is the predicted count. The supervision of FSC networks is usually achieved using a MSE loss between the predicted maps and the ground-truth ones.

5.2 Few-shot Object Counting

Counting objects is a task with applications in many domains *e.g.* manufacturing, medicine, monitoring, that involve different types of objects. While earlier works focused on learning specialized networks [8, 52, 16, 55], Few-shot Object Counting (FSC) was recently introduced to train models that can count any object, including from categories outside the training data. To achieve this, the query image $x \in \mathbb{R}^{H \times W \times 3}$ is annotated with $n \in \{0, 1, 2, 3...\}$ *exemplar* boxes of coordinates $b \in \mathbb{R}^4$. The counting network takes as input both the query image and the set of n boxes. It predicts a density map $d \in \mathbb{R}^{H \times W}$ of same size as the image. The model is typically trained with an L_2 loss between the predicted and ground-truth densities. As shown in Fig. 5.1 (right), a ground-truth density map is built to have zero values where there are no objects, and a Gaussian kernel of fixed variance at the center of every object. The final count is obtained by summing across all positions of the density map.

Few-shot Object Counting was initially formulated as matching exemplars and image patches features [84]. FSC147 [114] was later put forward as the main dataset for this task, with an open set train and test split to evaluate generalization to unseen object categories. Its authors also introduced

FamNet, a deep net trained to infer density maps from feature similarities. In the same lineage, BMNet [128] refines the similarity map by learning the similarity metric jointly with the counting network. Subsequent approaches include SAFECount [158] and CounTR [14] two state-of-the-art methods[1] used in this work and described in more details in the following.

SAFECount In contrast to previous approaches that predict the density map from the similarity map, SAFECount infers it from the features of the query image, weighted by the similarity map. Two learnable modules are introduced: the Similarity Comparison Module (SCM) and the Feature Enhancement Module (FEM) (cf. Fig. 5.2). The SCM computes the similarity map by convolving the query image features ($\mathbf{f}_q$) using the exemplars features ($\mathbf{f}_s$) as kernels. The FEM integrates the exemplars features into the query image features using the similarity map ($\mathbf{R}$). This ensures the preservation of information related to both the query image and the exemplar-query relationship. This integration is accomplished by convolving the similarity map with the exemplars features, yielding features ($\mathbf{f}_R$) that are then processed through a convolutional layer and added to the query features. Finally, a regression head produces the density map based on these features.

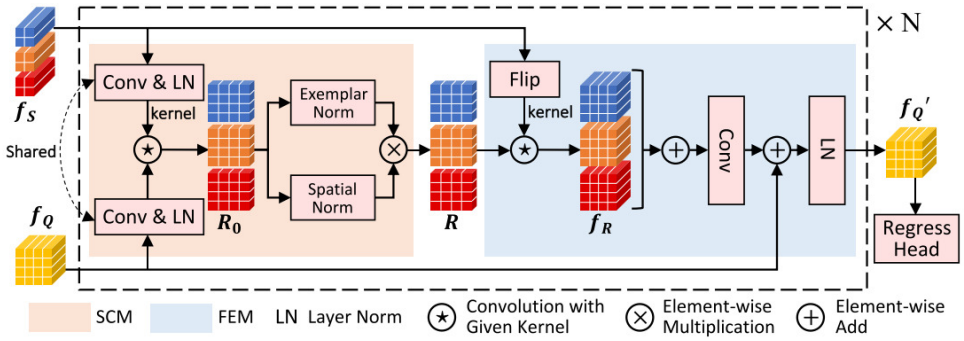


Figure 5.2: Overview of SAFECount. Figure taken from [158].

CounTR While previous approaches that are convolution-based, this method introduces a Feature Interaction Module (FIM) composed of a series of transformer decoder layers to capture the similarity between query image and exemplars features. The query image features are extracted using the ViT [29] encoder and act as the Query while the exemplars features extracted with a CNN act as the Key and Value. The output of the FIM module is then fed to a CNN-based decoder that upsamples

[1]Respectively 4$^{\text{th}}$ and 2$^{\text{nd}}$ on the FSC147 benchmark (`https://paperswithcode.com/sota/object-counting-on-fsc147`)

the features into a density map.

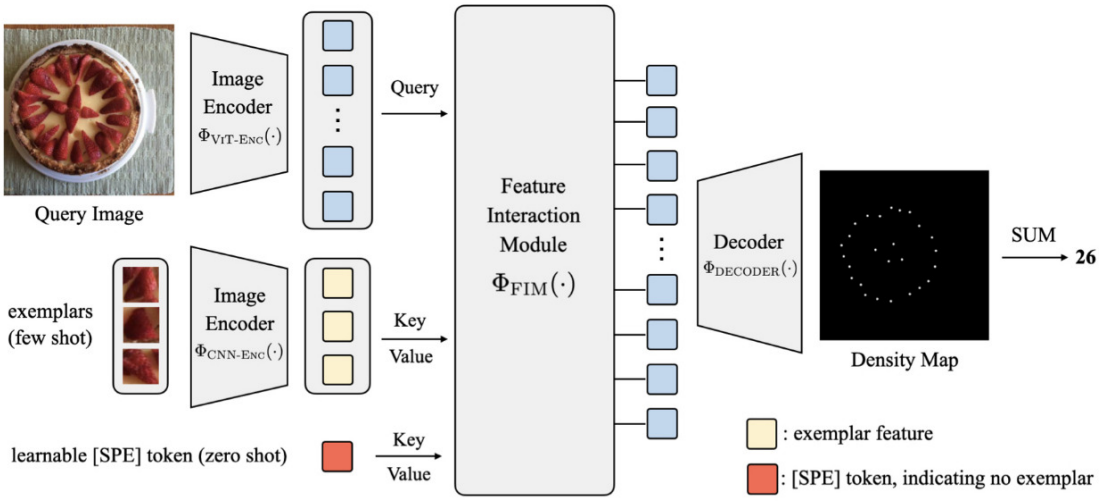


Figure 5.3: Overview of CounTR. Figure taken from [14].

The reference dataset for FSC, namely FSC147 [114], contains a limited amount of data (3659 train images) thus bounding the performances of counting networks [113]. Expanding such a dataset is costly as the annotation process requires pinpointing the center of each object present in a query image, with a potentially high number of occurrences. The closest comparison to our work is the Vicinal Counting Network from Rajan *et al.* [113]. It augments FSC147 with generated data by training a conditional GAN jointly with the counting network, producing augmentations that preserve the image content while modifying its visual appearance. While outperformed by later models, it introduced the idea that well-chosen augmentations can significantly boost counting accuracy. In this work, we leverage large pre-trained text-to-image diffusion models to produce diverse augmentations that not only alter the appearance, but are also able to change the content, to synthesize augmentations with a variety of object semantics.

5.3 Semantic Generative Augmentations

In this section, we introduce our approach to generate augmentations for few-shot counting. First, we give an overview of our generative model. Then, we introduce our strategy to augment the data and enhance the diversity of the augmentations.

5.3.1 Text-and-Density Guided Augmentations

To synthesize new images that can effectively augment a few-shot counting dataset, we need to have control over the number of objects and how they are laid out. Indeed, we need to ensure that we know the density maps of the synthetic samples so that they can be used to train the model. As few-shot counting datasets are generally limited in size, we take advantage of available pre-trained diffusion models to synthesize diversified augmentations of the training samples, reducing overfitting and improving generalization. Large pre-trained generative models such as Stable Diffusion are usually conditioned through textual prompts. To finetune these models, the first step is thus to pair textual captions to the training images.

We obtain diverse and descriptive captions using an off-the-shelf captioning model, *e.g.* BLIP2 [80]. This produces richer captions than plain object categories such as "a photo of {class}" [45]. However, two shortcomings remain. First, generated captions may not contain any information about the number or arrangement of the objects. Second, text-conditioned Latent Diffusion Models [117] poorly respect prompts regarding compositional constraints [103, 99]. Even adding this information in the caption does not guarantee that generated images would follow them. This is especially problematic as the correctness of the layout is a prerequisite to generate images for which we know the ground-truth. Therefore, we further condition the generative model directly on the density maps as an additional input, using the ControlNet [161] fine-tuning strategy. To summarize, our generative model is now conditioned on a text prompt, obtained by an automated captioning of the training image, and on its ground-truth density map to enforce the spatial layout of the objects. This allows us to synthesize new samples that augment the original image, while keeping the ground truth intact, making the augmentation amenable to supervised learning.

5.3.2 Diversity-Enhanced Augmentations

To formalize the augmentation process, let $\mathcal{D}_{\text{train}} = \{x_i, b_i, d_i\}_{i=1}^{N}$ be an annotated counting dataset, with x_i an image, b_i the exemplar bounding boxes for each image, and d_i its ground-truth density map. Let $\mathcal{C} = \{c_i\}_{i=1}^{N}$ be the set of corresponding captions, either from the dataset or obtained by automated captioning. For each image x_i, we aim at generating M augmentations using our text-density conditional generative model $g(d_i, c_i)$.

Baseline We sample augmentations from the LDM by taking advantage of the non-deterministic *reverse* diffusion process and the expressiveness of the pre-trained model. For an image x_i we produce M augmentations $\tilde{x}_i^{(j)}$ that share its caption and density map:

$$\tilde{x}_i^{(j)} = g(d_i, c_i), \quad j = 1, ..., M \tag{5.1}$$

These augmentations preserve both the number and layout of objects – because of the density conditioning – and the semantics *e.g.* object category and type of background – because of the text prompt. This already augments the number of samples available for training.

Diverse We can however go further and *diversify* the augmentations by altering either the text description or the spatial organisation of the objects. To do so, we take advantage of the dual conditioning on both densities and captions. We mix the two sets to create new combinations (density map, caption), producing augmentations that are semantically and geometrically more diverse than the original dataset. Yet, this mixing of the conditionings should be done carefully, to avoid low quality augmentations. Indeed, not all combinations make sense, *e.g.* "a herd of cows" and "a pearl necklace" exhibit very different and even incompatible spatial layouts. To prompt the generative model with realistic (*density*, *text*) pairs, we rely on caption similarity to find new associations between images that share some semantics, *e.g.* "cows" and "bisons".

We swap captions at random between pairs of *compatible* images. Two images are said to be compatible if their captions are more similar than some threshold t_c, *i.e.*:

$$\text{sim}(c_i, c_k) = \frac{\Phi(c_i)^\top \Phi(c_k)}{||\Phi(c_i)||_2 ||\Phi(c_k)||_2} > t_c$$

where Φ is a suitable text encoder *e.g.* BLIP2, CLIP [109] encoders. We then sample new images using the initial density map, but replacing the original caption with the caption $c_k \in \mathcal{C}$ from a compatible

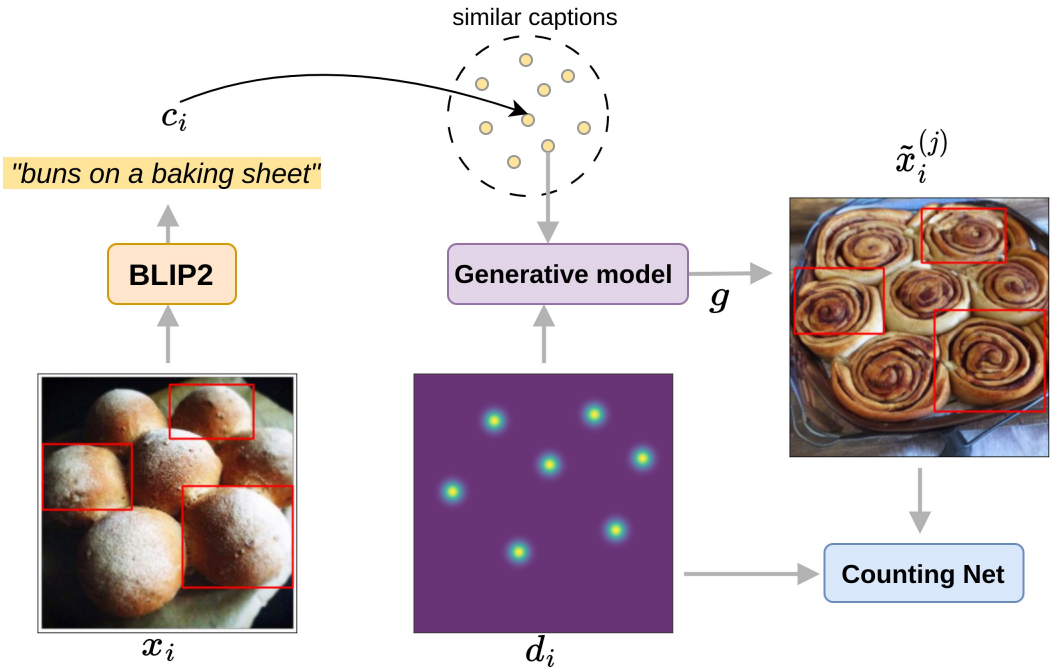


Figure 5.4: Overview of our approach. We condition a pre-trained diffusion model on both text prompts and density maps and perform swaps with similar captions. The density and original exemplars boxes are used as ground-truth for the generated augmentation.

training observation chosen at random:

$$\tilde{x}_i^{(j)} = g(d_i, c_k), \quad j = 1, ..., M \tag{5.2}$$

This process results in more diverse augmentations compared to the baseline and alters more the images than traditional augmentations (color jitter, crops, etc.), as shown in Fig. 5.5.

Synthetic and Diverse Balance We follow the training strategy from Trabucco *et al.* [141], where the synthetic augmentations are used as a regular data augmentation with a probability p_0 when training the counting model. As a way to balance baseline and diversified augmentations, we set a probability p_c that defines the fraction of the M augmentations that use a swapped caption instead of the original one. Typically, $p_c = 0.7$ means that 70% of the generated augmentations employ new (caption, density) combinations and that the remaining 30% use the original (caption, density) pair. For each augmentation, we keep the density used to condition the image generation and the original exemplar boxes as ground truth to train the model.[2]

[2]Note that if the caption changes the object category, bounding boxes for the exemplars might not be accurate anymore (*e.g.* "pens" are narrow and elongated, while "erasers" are closer to squares, see Fig. 5.14).

5.4 Experiments

5.4.1 Experiments Setup

Datasets We evaluate on two different datasets: a dataset with multiple object categories and a single category dataset.

- **FSC147** [114] is a 3-shot counting dataset with 147 object categories. It is the *de facto standard* of class-agnostic counting benchmarking. 89 categories are used for the training set, 29 are included in the validation set the remaining 29 constitute the test set. Note that the categories from the three sets are completely disjoint. In total, the dataset contains 6135 images, from which 3659 are used for training. The number of objects in the images varies from 7 to 3731 with an average of 56. Every image is annotated with 3 exemplar bounding boxes and an object density map.
- **CARPK** [52] is a class-specific dataset for counting cars in parking lots based on overhead imagery. The dataset contains 1448 images from a UAV in 4 parking lots: 3 for training and the last one for testing. There are 5 exemplar objects in total that are randomly extracted from the training set and employed during both training and testing. Following [158, 14], we evaluate on CARPK the generalization ability of our class-agnostic models on a new dataset.

Augmentation We train ControlNet [161] on the training images and density maps from FSC147. Text prompts are obtained by captioning the images with BLIP2 [80]. The underlying pre-trained diffusion model is Stable Diffusion $v1.5$ trained on LAION 2B. We use the default settings and train for 350 epochs. After training, we employ a guidance scale of 2.0 and 20 denoising steps to generate an image. For each augmentation strategy, we generate $M = 10$ augmentations per training sample unless specified otherwise. We swap the original caption with another one with a probability $p_c = 0.5$. Compatible captions to swap with are obtained by extracting caption features with the BLIP2 text encoder and filtering the captions with a similarity higher than $t_c = 0.7$.

Counting networks We demonstrate the effectiveness of our augmentation strategies on two state-of-the-art counting networks: SAFECount [158] and CounTR [14]. SAFECount is a CNN while CounTR is Transformer-based. In CounTR, training is done in two phases. First, the network is pre-trained

using a self-supervised masked auto-encoder [44], then it is fine-tuned in a supervised fashion with the usual L_2 loss on the densities. For CounTR, we employ the pre-trained model released by the authors and only retrain the fine-tuning phase. We use the hyperparameters reported in the original papers to train both networks, except training is 100 epochs longer to account for the higher number of training images. During training, we replace an image x_i with one of its augmentations $\tilde{x}_i^{(j)}$ with probability $p_0 = 0.5$. This balances the ratio of real vs. synthetic data in a single batch. We also employ traditional data augmentation strategies *e.g.* flips, color jitter, random cropping, that are applied to every image, both real and synthetic, as done in the original models.

Metrics We follow the standard evaluation of the counting accuracy through the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). For a ground-truth count y_i and a predicted count $\hat{y}_i$, we define:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \quad \text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{5.3}$$

The networks are trained and evaluated in the 3-shot setting unless specified. All results are averages over two runs.

5.4.2 Few-shot Counting on FSC147

Comparison with Traditional Augmentation We report in Table 5.1 the improvement in counting accuracy on FSC417 with our augmentation strategies when training SAFECount and CounTR. Consistent with the literature on synthetic data augmentation, baseline augmentations improve the results for both networks: MAE decreases by respectively 5% and 10% for SAFECount and CounTR on the val set. Nonetheless, diversifying the augmentations allows us to reduce the MAE even further, by 10% and 11% on the same val set and by 7% (SAFECount) and 13% (CountTR) on the test set. Baseline augmentations may lead to lower improvements as ControlNet seems to overfit the training data due to the small dataset size. The low guidance employed to generate the images (2.0) aims at promoting diversity [121] but, as shown in Fig. 5.5 (Baseline Gen.), the generated images remain close to the original image in terms of visual appearance of the objects and background. However, ControlNet generalizes to different captions. In Fig. 5.5 (Diverse Gen.), we observe that swapping captions allows us to create more diverse data, altering the size and texture of objects and their

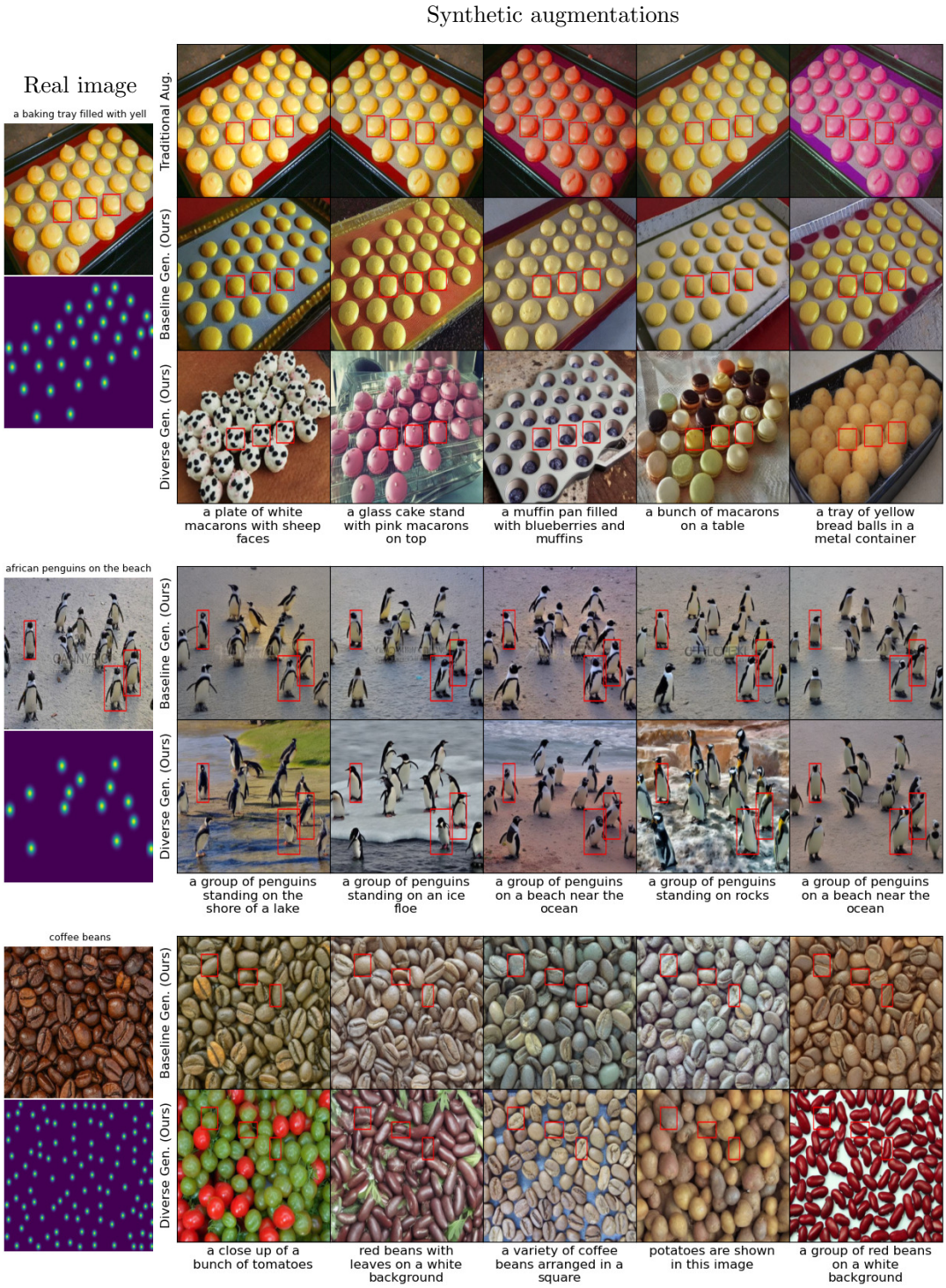


Figure 5.5: Qualitative results for the Baseline vs. Diverse augmentations. At the bottom of each diverse sample we show the caption used to generate the image. Our strategy allows to diversify the type of objects and/or the background.

	(a) SAFECount [158]				(b) CounTR [14]			
	Val		Test		Val		Test	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Traditional Augmentation*,†	13.95	51.73	13.73	91.85	14.25	50.15	13.13	88.21
+ Real Guidance [45]	14.94	53.09	13.48	**80.69**	15.37	49.47	13.37	96.44
+ Baseline Generation (Ours)	13.30	49.38	13.22	92.47	12.60	43.53	11.83	87.97
+ Diverse Generation (Ours)	**12.59**	**44.95**	**12.74**	89.90	**12.31**	**41.65**	**11.32**	**77.50**
Traditional Aug. (reported)	15.28	47.5	14.25	85.54	13.13	49.83	11.95	91.23

Table 5.1: Quantitative results on FSC147. (*) Traditional augmentations include color jitter, random cropping. (†) [158] and [14] are reproduced.

background. Such features cannot be altered with traditional data augmentation. When mixing baseline and diverse augmentations, the performances for both networks improve significantly with respect to the model without synthetic augmentation, or with naive augmentations only.

Comparison with Real Guidance We compare our approach with Real Guidance, an augmentation strategy for image classification by He *et al.* [45]. Augmentations are generated by prompting a pre-trained text-to-image diffusion model with the image classes. To reduce the domain gap, the synthetic images are generated from the real images with added noise as proposed in SDEdit [89]. Table 5.1 shows that our augmentation strategy outperforms Real Guidance, except on test RMSE with SAFECount as we discuss later. Starting from the real image with added noise is generally insufficient to preserve the number of objects and their positions (Fig. 5.6, 2^{nd} column). It shows that the density map conditioning ensures the preservation of object positions and number without requiring to start from the real image, which can limit the diversity of the generated images.

Counting Accuracy per Object Count In Table 5.2, we report the test MAE and RMSE per range of object counts for SAFECount. Our model increases the counting performances for all ranges with respect to the model trained without synthetic data. We also outperform Real Guidance for all ranges except for the range with a very high number of objects ($[301, 3701]$). As shown in Table 5.2, this range of object counts contains few images (they represent 1% of the total number of test images) but they dominate the global RMSE (80.69 for Real Guidance and 89.90 for our model). In particular, there are 2 outlier images with respectively 2560 and 3701 objects. In comparison, the maximum

Figure 5.6: Qualitative comparison with Real Guidance [45]. Our augmentations preserve the layout while creating more diverse backgrounds. Ground-truth density maps overlap with the generated images (last 2 columns).

number of objects in the training set is 1912 objects. Real Guidance performs better on one of these outlier images as shown in the first row of Fig. 5.7. For other images with a high object count, both models are on par (last two rows of Fig. 5.7). The absolute MAE and RMSE, tend to give more importance to some images, and thus potentially not accurately reflect the performances of counting networks. It might be more suitable to compute the errors relative to number of objects in the images.

5.4.3 Ablation Study

We study here the influence of the hyperparameters of our approach: impact of the caption similarity threshold, t_c, influence of introducing diversified augmentations by varying the p_c parameter, the effect of the number of augmentations M, and of the ratio of synthetic samples during training, p_0. All ablations are conducted on SAFECount trained for 200 epochs to reduce training time.

# Objects	# Images	Trad. Aug.		+ Diverse Gen. (Ours)		+ Real Guidance [45]	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
[1, 10]	60	4.76	10.22	3.75	**6.42**	**3.24**	7.47
[11, 20]	268	6.27	47.81	**6.05**	**45.5**	6.22	50.3
[21, 50]	413	6.61	12.22	**5.38**	**10.21**	6.05	11.47
[51, 100]	254	12.14	18.26	**10.29**	**15.84**	12.30	18.39
[101, 300]	172	23.58	35.47	**21.65**	**29.91**	26.08	38.03
[301, 3760]	23	197.18	637.04	205.4	604.42	**186.38**	**538.68**

Table 5.2: Quantitative results for SAFECount: Test counting accuracy (3-shot) per range of number of objects for SAFECount [158] on FSC147.

Caption Similarity Threshold We swap captions based on caption similarity to form novel but plausible (*density*, *text*) combinations. As shown in Fig. 5.11, associating a completely unrelated caption to a given density map results in generated images that do not correspond to the input density map or are of poor quality, as it is harder for the model to generalize. In Fig. 5.9a we evaluate different similarity thresholds to the naive approach where all captions can be swapped freely at random ($t_c = 0.0$). The performances are improved compared to random swaps with all thresholds between 0.5 and 0.9. However, there is a quality-diversity tradeoff shown in Fig. 5.9b. Setting the threshold too high ($t_c = 0.8, 0.9$) swaps captions between images of objects belonging to the same category, thus limiting diversity. With a lower threshold, *e.g.* $t_c = 0.7$, new captions can also belong to objects from different similar categories, *e.g.* swapping "bread rolls" and "macarons".

Rate of Diverse Samples In Fig. 5.8, we vary the rate of diverse augmentations among $M = 10$ augmentations. We compare with SAFECount trained solely with baseline non-diverse augmentations. More diverse samples overall increase the counting accuracy. We further find that adding 70% of diverse samples gives better performances than 50%. This suggests that the diverse augmentations are more beneficial than the baseline ones. The augmentations using the original captions yet remain useful to the model as we observe a slight increase in MAE when more than 90% of the augmentations are obtained on new combinations.

Number of Augmentations In Fig. 5.10a, we vary the number of augmentations generated for each image. With a single augmentation, the performances already improve. For low values of $M = 1, 3, 5$, the performances are comparable, then a stronger increase is observed for $M = 10$. With twice as

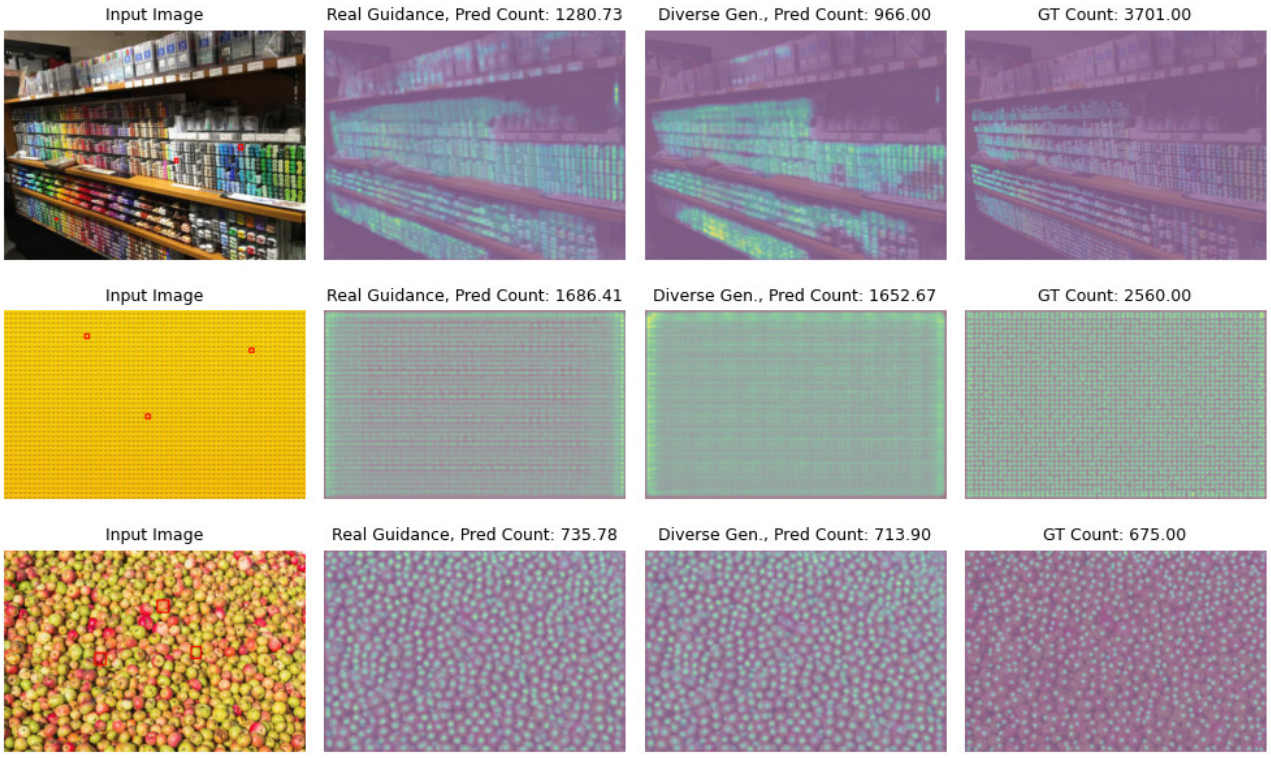


Figure 5.7: Qualitative counting results on FSC147 test images. We compare the model trained with Real Guidance's augmentations (2^{nd} column) vs. our augmentations (3^{rd} column) for images with a high number of objects. Predicted and ground-truth density maps are overlapped with the images.

many augmentations ($M = 20$), performances degrade on the validation set. This might be due to an insufficient convergence, as the model is trained with many more different data points but for the same number of iterations.

Rate of Synthetic Samples Fig. 5.10b shows the counting accuracy w.r.t. ratio p_0 of synthetic samples vs. real samples in a batch. We observe that equally balancing the synthetic and real data gives the best performances, which is consistent with what has been observed in previous works generating synthetic data for image classification [45, 10].

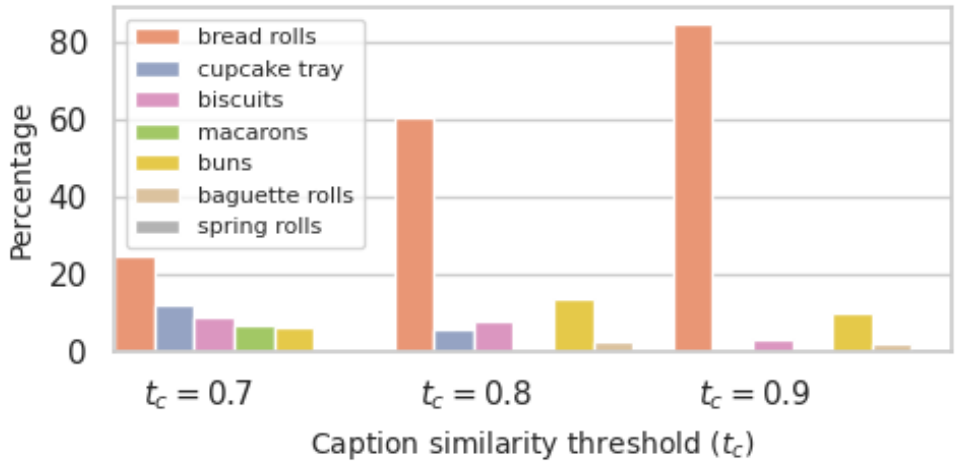
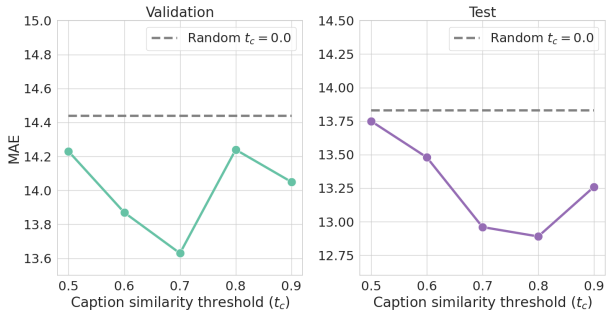
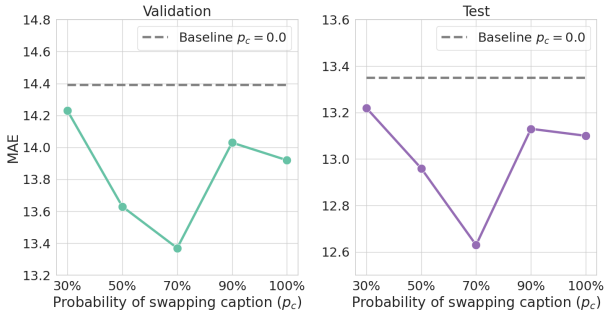


Figure 5.8: Distribution of object categories in the set of captions to swap from w.r.t t_c for a sample of category "bread rolls". Lower thresholds result in more diverse augmentations, while objects still belong to similar classes.

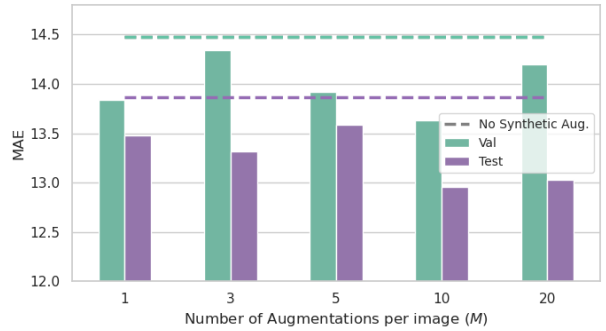


(a) Validation and test MAE on FSC147 w.r.t. t_c.

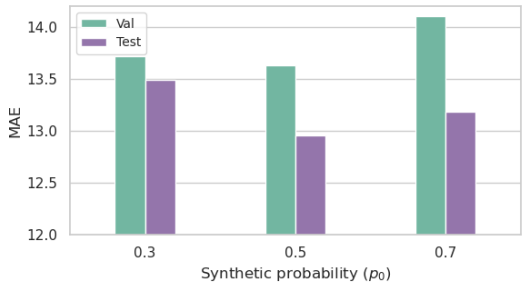


(b) Validation and test MAE on FSC147 w.r.t. p_c.

Figure 5.9: Impact of caption similarity threshold t_c (left) and percentage of diverse samples p_c (right) on SAFECount. MAE is reported for the val and test sets of FSC147.



(a) Validation and test MAE on FSC147 w.r.t. M.



(b) Validation and test MAE on FSC147 w.r.t. p_0.

Figure 5.10: Impact of number of augmentations threshold M (left) and percentage of synthetic data p_0 (right) on SAFECount. MAE is reported for the val and test sets of FSC147.

Figure 5.11: Caption swap at random (top) vs. similarity-based swap (bottom, $t_c = 0.7$). Random swapping results in a mismatch between the layout and the semantics.

5.4.4 Counting with Fewer Shots

SAFECount SAFECount is a 3-shot counting method but it can generalize to 1-shot counting without retraining [158]. We evaluate our model trained on diversified synthetic augmentations in the 1-shot case. As shown in Table 5.3, our model also exhibits good performances in the 1-shot setting, outperforming the traditional augmentation model.

CounTR CounTR is also used for zero-shot counting (counting with no exemplars). We retrained the models with a number of shots randomly chosen between 0 and 3. In Table 5.4, we evaluate the models trained with and without synthetic augmentations in both the 3-shot and 0-shot settings. We observe a small degradation of the performances for both models in the 3-shot case in comparison with the models trained with always 3 shots (first two lines). The performances of our model nevertheless remain higher. In the 0-shot case, we find that both models perform similarly on the validation test, while our model is significantly better on the test set.

	3-shot				*1-shot*			
	Val		Test		Val		Test	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Traditional Augmentation	13.95	51.73	13.73	91.85	19.92	67.63	18.08	**104.32**
+ Diverse Generation (Ours)	**12.59**	**44.95**	**12.74**	**89.90**	**18.55**	**61.22**	**17.60**	106.47

Table 5.3: Quantitative results: 3-shot and 1-shot evaluation for SAFECount [158] on FSC147.

	3-shot				*0-shot*			
	Val		Test		Val		Test	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Traditional Augmentation	14.25	50.15	13.13	88.21	-	-	-	-
+ Diverse Generation (Ours)	12.31	49.47	11.32	77.50	-	-	-	-
Traditional Augmentation	14.61	51.33	13.24	94.01	**18.82**	**69.64**	17.51	122.01
+ Diverse Generation (Ours)	**13.09**	**48.29**	**11.59**	**83.23**	18.84	69.90	**15.70**	**112.25**

Table 5.4: Quantitative results: 3-shot and 0-shot evaluation for CounTR [14] on FSC147. Top: 3-shot training. Bottom: [0,3]-shots training.

5.4.5 Generalization on CARPK

CARPK [52] was introduced to train networks that can count cars in aerial views of parking lots. It is also used to evaluate the ability of class-agnostic models to count in a class-specific setting. Given a model trained on FSC147 (without the examples of the "car" category), the model is first evaluated without any fine-tuning, then with fine-tuning on CARPK. We evaluate our model trained on FSC147 with diverse augmentations in the same setting. Table 5.5 reports improved counting performances in both the pre-trained and fine-tuned settings in comparison to the models trained without synthetic

	Aug.	MAE	RMSE
Pre-trained on FSC147	Trad. Aug.	17.65	23.83
	Div. Gen.	**16.49**	**19.05**
Fine-tuned on CARPK	Trad. Aug.	5.44	6.94
	Div. Gen.	**4.87**	**6.17**
SAFECount [158] (WACV'23)	Trad. Aug.	5.33	7.04
CounTR [14] (BMVC'22)	Trad. Aug.	5.75	7.45
BMNet+ [128] (CVPR'22)	Trad. Aug.	5.76	7.83

Table 5.5: Counting performance on CARPK with SAFECount.

augmentations. In the fine-tuning setting, we reach state-of-the-art counting accuracy (4.87 MAE/6.17 RMSE) on CARPK amongst class-agnostic models.

5.5 Qualitative results

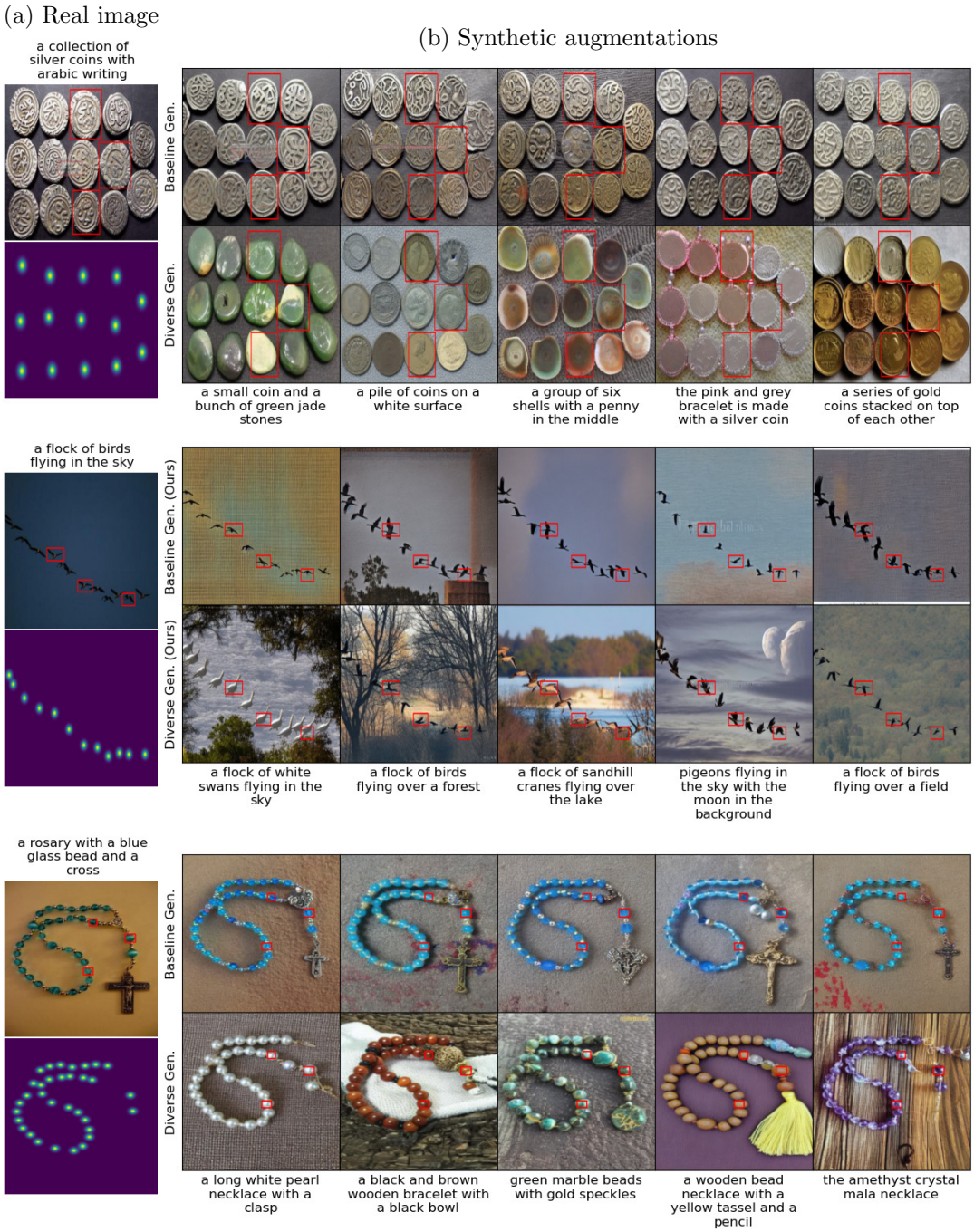


Figure 5.12: Qualitative results of synthetic augmentations of FSC147. We compare our Baseline vs. Diverse augmentations.

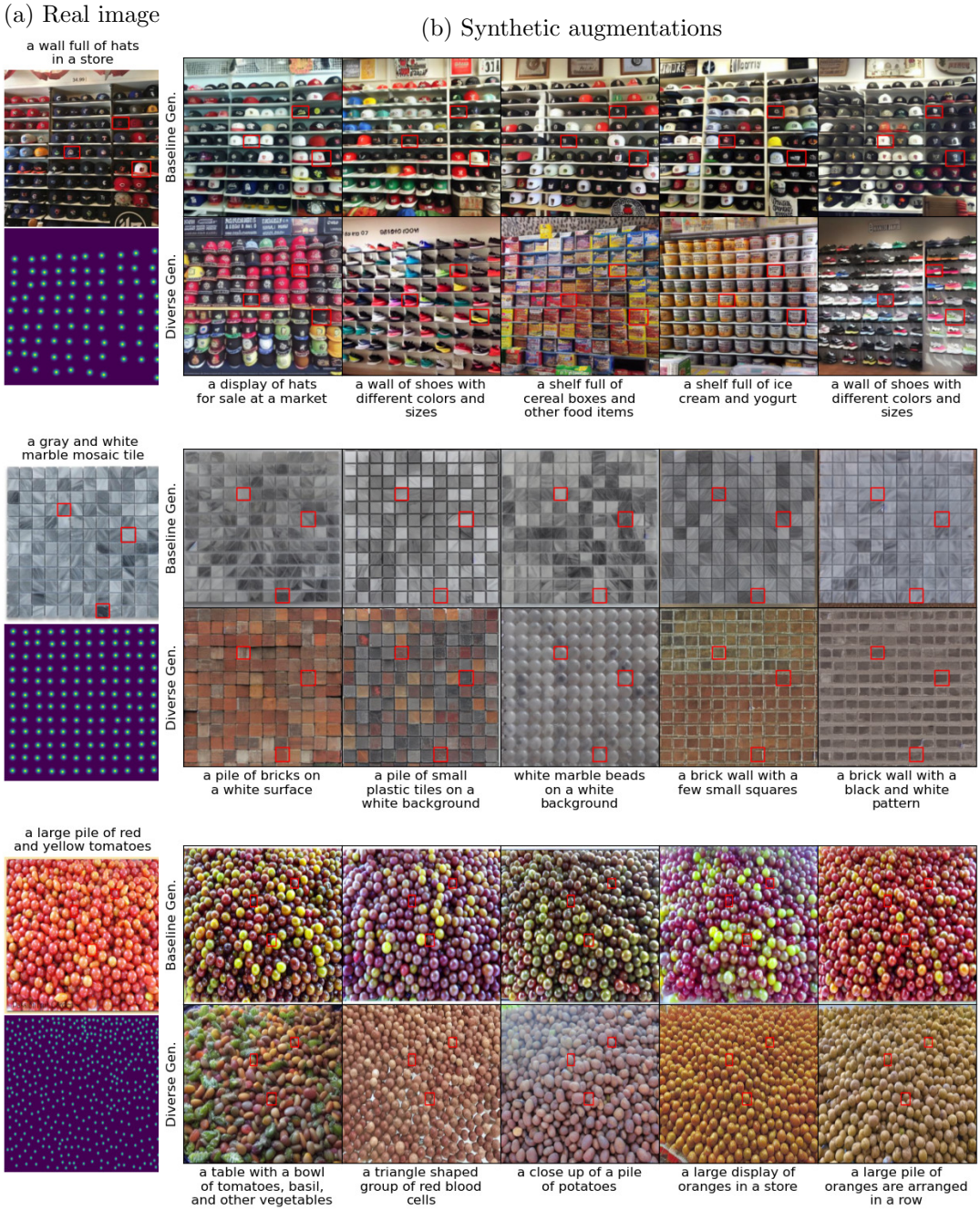


Figure 5.13: Qualitative results of synthetic augmentations of FSC147. We compare our Baseline vs. Diverse augmentations.

Figure 5.14: Limitation: our diverse generation strategy can change the size and shape of generated objects, leading to exemplar boxes (in red) that do not fit perfectly.

5.6 Conclusion

We show that synthetic data generated by diffusion models improves the performances of deep models for tasks that require fine-grained generation such as the few-shot counting task. We adapt a pretrained text-to-image model with a task-specific conditioning *e.g.* density map and we propose a diversification strategy that exploits caption similarities to generate unseen but plausible data that mixes the semantics and the geometry of different training images. We show that selecting compatible images improves synthetic image quality with beneficial effects on model performance. We demonstrate that learning with our diverse synthetic data leads to improved counting accuracy on FSC147 and state of the art generalization on CARPK.

A current limitation of our method is that, in some cases, the diversification strategy may cause the exemplar boxes to not fit the generated objects anymore, as illustrated in Fig. 5.14. Conditioning on densities makes it possible to reuse both the original density and the exemplar bounding boxes. However, changing the caption can affect the object category, and in turn its shape. We explored to what extent refining these boxes could improve our model. We segmented objects using SAM in zero-shot [71] prompted with object centers. However, the preliminary results showed no improvement with box refinement, possibly due to inaccurate segmentation.

Our augmentation strategy could be adapted to other tasks requiring fine-grained compositionality, such as object detection and semantic segmentation. A concurrent work [96] proposes a similar approach to generate synthetic datasets to improve domain generalization in semantic segmentation. They employ a pre-trained diffusion model conditioned on semantic maps and introduce a text prompt generator instead of relying on set of captions generated by a captioning model. Our diversification scheme could also be further extended by swapping both the captions and the task-specific controls, by introducing a suitable similarity metric that operates on the specific condition. A generative model could also be trained to generate the task-specific conditions.

Chapter 6

Conclusion

Contents

6.1 Summary of contributions **88**
6.2 Perspectives **90**
6.2.1 On-going work 90
6.2.2 Further perspectives 90
6.3 Broader Impacts **92**

6.1 Summary of contributions

In this thesis, we focused on improving means for controlling the properties of images synthesized by deep generative models. In particular, we tackled the learning of disentangled and robust controls to modify a semantic attribute independently of others and ensure realistic edited images (Chapters 3 and 4). We also explored how to exploit text-controlled generative models to generate annotated and diversified synthetic training data tailored for a specific task (Chapter 5).

Disentangled and Robust Image Editing In Chapters 3 and 4, we address two limitations associated with semantic directions extracted from the latent space of GANs: entanglement and unrealistic outputs (see Section 1.2.1). In Chapter 3, we show that the correlations of the GAN training set are propagated to the directions via the GAN-generated data used as the training data. We thus propose to mitigate the problem by balancing this data w.r.t. multiple attributes before learning the directions. We apply our method to extract linear directions corresponding to facial attributes with two GAN architectures and demonstrate improved disentanglement in comparison to a popular editing

method that requires post-processing. Although this is a simple and general approach, it may require to generate a lot of data to have a balanced sample of sufficient size, and doesn't allow complete disentanglement. In Chapter 4, we thus explore a method that introduces an explicit disentanglement constraint but relies on the guidance of a pre-trained classifier. Despite state-of-the-art disentanglement, we identify some cases where this guidance results in learning directions that produce latent codes generating unnatural images, that may be attributed to classifiers being highly confident in regions outside the training data. To avoid dependence on brittle models while maintaining disentanglement, we propose an alternative formulation using the Wasserstein loss. This loss allows to learn directions that produce a new distribution of codes with the desired attribute by minimizing the distance with the distribution of training codes having that attribute. This thus satisfies two objectives: achieving the desired semantic edit and remaining in-distribution to avoid unrealistic images. Employed a with a regular Euclidean cost, we demonstrate that this loss leads to implicit disentanglement but we also explore using this loss with an explicit disentanglement cost. We demonstrate competitive performances for face editing in comparison to a state-of-the-art classifier-based approach. However, we struggle to apply our method on data outside the face domain, in particular data with multiple objects and varied semantics. These findings indicate that the latent space of GANs might not always encode the desired semantics. As GANs also fail at generating distributions with many modes, their use to generate controlled synthetic training datasets for a wide range of tasks is not straightforward.

Diversified Data Augmentation with Text-to-Image models In Chapter 5, we thus leverage recent pre-trained diffusion models that have shown to be effective off-the-shelf tools to augment diverse classification datasets, due to their large-scale training and textual control. For tasks that require the generation of multiple objects and spatial understanding however, this control is not sufficient due to the text encoder's limited understanding of compositional concepts *e.g.* position and count. We address this issue by finetuning a model with an additional task-specific spatial condition. Yet, we find that the finetuned model generates images with limited diversity. To generate images with unseen configurations of semantics and spatial layouts, we introduce an approach to prompt our model with novel but plausible combinations of input controls. We generate a set of captions and perform random swaps among semantically similar captions. We apply our approach to the task of few-shot object counting. We train two state-of-the-art counting networks on the real benchmark augmented with our

generated data, leading to improved counting accuracy.

6.2 Perspectives

6.2.1 On-going work

In Chapter 5, we address complementing an existing database by generating data that is more diverse than the original dataset. Although we see improvements on the performances of the models when introducing this data during training, it is possible that not all generated images are useful. In this following work, we aim at reducing the number of generated images by targeting the generation of images where the models make mistakes. However, the question is how to automatically identify the controls that lead to such images. Some recent works focus on this problem [59, 145, 106, 154]. For instance, Jain et al. [59] project validation images in a joint vision-language space and learn the boundary that separates correctly classified from wrongly classified images using SVMs. The idea is then to re-use this boundary to generate hard images. However, this approach allows to only extract the main mode of error. Vendrow et al. [145] manually craft 23 domain shifts *e.g.* "in the grass", "at dusk" added as a suffix in prompts of the type "a photo of an {object}" and evaluate the classifiers on each shift. Prabhu et al. [106] use a similar approach but edit the prompts automatically using an LLM. Inspired from [125] that learns class-specific tokens using a classifier (cf. Fig. 6.1), we are currently working on an approach that aims at learning automatically multiple tokens that lead to wrongly classify a set of images. Specifically, given N tokens as follows: "$S_1^* S_2^* ... S_i^* \{class\} S_{i+1}^* ... S_N^*$", the idea is to optimize the tokens such as the prediction of the classifier differs from $class$. We can set the target class based on errors on a validation set or manually, for instance if $class$ = 'bee', we could set the new target class as 'fly'. In contrast to previous approaches, this approach would allow to discover many modes or errors without relying on a brute force strategy.

6.2.2 Further perspectives

Evaluation of Generated Data Generative data augmentation with T2I models is a promising topic and still faces a lot of challenges. One of the challenges is to develop metrics to evaluate the augmented data. In particular, it would be desirable to evaluate the diversity, the faithfulness to the task-specific input condition (and thus labels) and the usefulness of a generated sample. These metrics would allow

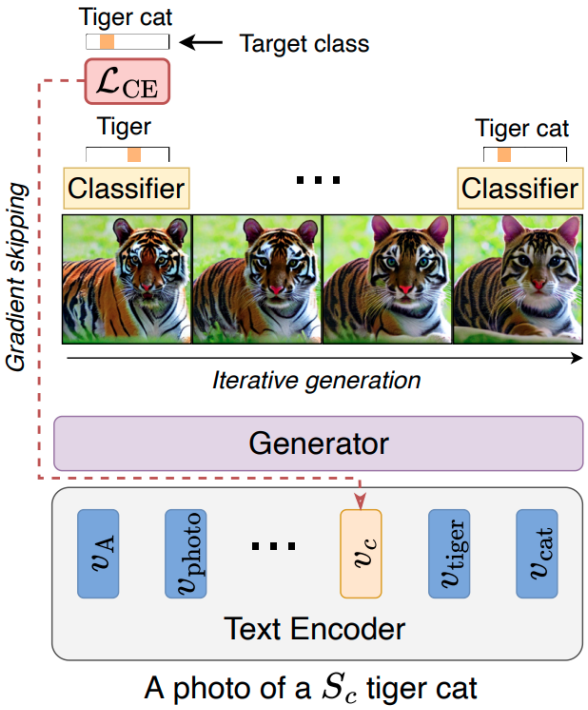


Figure 6.1: Schwartz et al. introduce an approach to learn class-specific tokens to generate fine-grained details using a classifier. Figure taken from [125].

to better filter the augmented data. For the faithfulness, a first lead would be to leverage the internal knowledge of the generative model such as in [20] and its consistency with the original condition. The usefulness might be more tricky to define but could be interpreted as the uncertainty of a given model on the generated sample (cf. on-going work), and is thus related to the criteria used for sample selection in active learning.

Interactive Image Generation and Editing One interesting and long-term goal is to bring more interactivity into the image generation and editing process. Recent works aim at combining LLMs such as GPT-4 [97] with a text-to-image model such as Stable Diffusion [117]. The LLM transforms the question of the user into prompts fed to the generative model and can also edit the prompts based on the interaction with the user [160]. Very recently, DALL·E-3 was introduced in ChatGPT. An interesting question is how to transform a user demand into a spatial condition that can be fed to a model like ControlNet [161]. To facilitate interactivity, the exploration of voice as conditioning in place of text is also an interesting prospect.

6.3 Broader Impacts

In 2013, Ari Folman directed the science-fiction movie "The Congress", that portrays a future world where major Hollywood studios are equipped with advanced technology that can scan the body of actors to build digital replicas that mimic their appearance and emotions. A fictionalized version of the famous Hollywood actress Robin Wright accepts a financial deal that has her disappear from the big screen to be replaced by her virtual version. In July of 2023, Hollywood actors started an important strike that marked the first actors strike since 1980. One of their primary concern was the use of Generative Artificial Intelligence (GenAI) to create digital replicas that could be used for any purpose forever without their consent or compensation [87]. In the span of a decade, GenAI is no longer a fantasy or science-fiction, it has become a reality and it is affecting our society. Its impact might be positive in some cases but it also raises a lot questions and concerns in others. For instance, the ownership of intellectual property for the generated content remains unclear. Various copyrighted artworks were also used to train large models without the artists' knowledge [39]. This allows random users to easily replicate the style of these artists. Research works are beginning to tackle this issue by exploring ways to erase concepts from the pre-trained models given the name of the style [36, 37]. Another issue identified by various studies is the significant biases within generative models, particularly against women and people of color [85, 78]. In Chapters 3 and 4, we mitigate this issue with the introduced disentanglement strategies. However, we focus on a limited set of attributes and a pre-trained model restricted on the face domain. [34, 98] propose ways to automatically adjust the prompts to generate fairer outputs with pre-trained text-to-image models. Finally, these tools are frequently employed to propagate fake news and participate in blackmail or humiliation with the generation of pornographic deepfakes. The detection of generated images has been widely explored but the generalization to new generative models is still a challenge [153]. There are also potential unknown implications of GenAI but one thing is for sure, these issues should be discussed among stakeholders but also with the general public, to implement regulations that prevent misuses while still leaving room for innovation. On Friday the 2$^{\text{nd}}$ of February 2024, just as I write these final words, an agreement was reached between all 27 European countries to sign the AI Act that marks the first rules for AI in the world [74].

Bibliography

[1] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020.

[3] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021. ISSN 0730-0301. doi: 10.1145/3447648. URL `https://doi.org/10.1145/3447648`.

[4] Y. Alaluf, O. Patashnik, and D. Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021.

[5] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18511–18521, 2022.

[6] A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[7] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[8] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the wild. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 483–498, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46478-7.

[9] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

[10] H. Bansal and A. Grover. Leaving reality to imagination: Robust classification via generated datasets. In *ICLR Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

[11] V. Besnier, H. Jain, A. Bursuc, M. Cord, and P. Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020.

[12] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[13] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. 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.

[14] L. Chang, Z. Yujie, Z. Andrew, and X. Weidi. Countr: Transformer-based generalised visual counting. In *British Machine Vision Conference (BMVC)*, 2022.

[15] L. Chapel, M. Z. Alaya, and G. Gasso. Partial optimal tranport with applications on positive-unlabeled learning. *Advances in Neural Information Processing Systems*, 33:2903–2913, 2020.

[16] J. Chen, W. Su, and Z. Wang. Crowd counting with crowd attention convolutional neural network. *Neurocomputing*, 382:210–220, 2020.

[17] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[18] E. Collins, R. Bala, B. Price, and S. Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020.

[19] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[20] G. Couairon, M. Careil, M. Cord, S. Lathuilière, and J. Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2174–2183, 2023.

[21] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[24] E. Denton, B. Hutchinson, M. Mitchell, T. Gebru, and A. Zaldivar. Image counterfactual sensitivity analysis for detecting unintended bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[25] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[26] M. D'Incà, C. Tzelepis, I. Patras, and N. Sebe. Improving fairness using vision-language driven image augmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4695–4704, 2024.

[27] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[28] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. 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`.

[30] L. Dunlap, A. Umino, H. Zhang, J. Yang, J. E. Gonzalez, and T. Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=9wrYfqdrwk`.

[31] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12868–12878, 2021. doi: 10.1109/CVPR46437.2021.01268.

[32] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré. Interpolating between optimal transport and MMD using sinkhorn divergences. In *AISTATS*, 2019.

[33] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.09.013. URL `https://www.sciencedirect.com/science/article/pii/S0925231218310749`.

[34] F. Friedrich, M. Brack, L. Struppek, D. Hintersdorf, P. Schramowski, S. Luccioni, and K. Kersting. Fair diffusion: Instructing text-to-image generation models on fairness. *arXiv preprint at arXiv:2302.10893*, 2023.

[35] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[36] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2426–2436, October 2023.

[37] R. Gandikota, H. Orgad, Y. Belinkov, J. Materzyńska, and D. Bau. Unified concept editing

in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5111–5120, January 2024.

[38] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[39] A. Gil, J. Neelbauer, and D. A. Schweidel. Generative ai has an intellectual property problem, 2024. URL `https://hbr.org/2023/04/generative-ai-has-an-intellectual-property-problem`. Harvard Business Review, Last accessed 2 February 2024.

[40] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[42] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. Ganspace: Discovering interpretable gan controls. In *Advances in Neural Information Processing Systems*, volume 33, pages 9841–9850, 2020.

[43] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016.

[44] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[45] R. He, S. Sun, X. Yu, C. Xue, W. Zhang, P. Torr, S. Bai, and X. Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

[46] P. Helber, B. Bischke, A. Dengel, and D. Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018 - 2018*

IEEE International Geoscience and Remote Sensing Symposium, pages 204–207, 2018. doi: 10.1109/IGARSS.2018.8519248.

[47] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[48] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[49] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf`.

[50] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[51] X. Hou, X. Zhang, H. Liang, L. Shen, Z. Lai, and J. Wan. Guidedstyle: Attribute knowledge guided style manipulation for semantic face editing. *Neural Networks*, 145:209–220, 2022.

[52] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE international conference on computer vision*, pages 4145–4153, 2017.

[53] L. Huang, D. Chen, Y. Liu, Y. Shen, D. Zhao, and J. Zhou. Composer: Creative and controllable image synthesis with composable conditions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 13753–13773. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/huang23b.html`.

[54] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1501–1510, 2017.

[55] Z. Huang, Y. Ding, G. Song, L. Wang, R. Geng, H. He, S. Du, X. Liu, Y. Tian, Y. Liang, S. K. Zhou, and J. Chen. Bcdata: A large-scale dataset and benchmark for cell detection and counting. In A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou,

D. Racoceanu, and L. Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 289–298, Cham, 2020. Springer International Publishing.

[56] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[57] A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020.

[58] A. Jahanian, X. Puig, Y. Tian, and P. Isola. Generative models as a data source for multiview representation. In *International Conference on Learning Representations*, 2022.

[59] S. Jain, H. Lawrence, A. Moitra, and A. Madry. Distilling model failures as directions in latent space. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=99RpBVpLiX`.

[60] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[61] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[62] L. Karazija, I. Laina, and C. Rupprecht. ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[63] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[64] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[65] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[66] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2021.

[67] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018.

[68] G. Kim, T. Kwon, and J. C. Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022.

[69] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[70] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[71] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[72] U. Kocasarı, A. Dirik, M. Tiftikci, and P. Yanardag. Stylemc: Multi-channel based fast text-guided image generation and manipulation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3441–3450, 2022.

[73] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

[74] C. Kroet. Eu countries approve technical details of ai act, 2024. URL `https://www.euronews.com/my-europe/2024/02/02/eu-countries-approve-technical-details-of-ai-act`. Last accessed 2 February 2024.

[75] T. Le, V. Lal, and P. Howard. Coco-counterfactuals: Automatically constructed counterfactual examples for image-text pairs. In *Advances in Neural Information Processing Systems*, 2023.

[76] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[77] K. Lee, H. Liu, M. Ryu, O. Watkins, Y. Du, C. Boutilier, P. Abbeel, M. Ghavamzadeh, and S. S. Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

[78] N. Leonardo and B. Dina. Humans are biased. generative ai is even worse, 2024. URL `https://www.bloomberg.com/graphics/2023-generative-ai-bias/`. Last accessed 2 February 2024.

[79] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[80] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/li23q.html`.

[81] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 2021.

[82] B. Liu, Y. Zhu, Z. Fu, G. De Melo, and A. Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4836–4843, 2020.

[83] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, December 2015.

[84] E. Lu, W. Xie, and A. Zisserman. Class-agnostic counting. In *Asian conference on computer vision*, pages 669–684. Springer, 2018.

[85] S. Luccioni, C. Akiki, M. Mitchell, and Y. Jernite. Stable bias: Evaluating societal representations in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=qVXYU3F017`.

[86] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[87] G. Maddaus. Sag-aftra strike: Ai fears mount for background actors, 2024. URL `https://variety.com/2023/biz/news/sag-aftra-background-actors-artificial-intelligence-1235673432/`. Last accessed 2 February 2024.

[88] N. Mehrabi, P. Goyal, A. Verma, J. Dhamala, V. Kumar, Q. Hu, K.-W. Chang, R. Zemel, A. Galstyan, and R. Gupta. Is the elephant flying? resolving ambiguities in text-to-image generative models. *arXiv preprint arXiv:2211.12503*, 2022.

[89] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. SDEdit: Image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[90] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[91] C. Mou, X. Wang, L. Xie, J. Zhang, Z. Qi, Y. Shan, and X. Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[92] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[93] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[94] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[95] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. Mcgrew, I. Sutskever, and M. Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 17–23 Jul 2022.

[96] J. Niemeijer, M. Schwonberg, J.-A. Termöhlen, N. M. Schmidt, and T. Fingscheidt. Generalization by adaptation: Diffusion-based domain extension for domain-generalized semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2830–2840, 2024.

[97] OpenAI. Gpt-4 technical report. 2023. URL `https://api.semanticscholar.org/CorpusID:257532815`.

[98] H. Orgad, B. Kawar, and Y. Belinkov. Editing implicit assumptions in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7053–7061, October 2023.

[99] R. Paiss, A. Ephrat, O. Tov, S. Zada, I. Mosseri, M. Irani, and T. Dekel. Teaching clip to count to ten. *arXiv preprint arXiv:2302.12066*, 2023.

[100] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[101] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[102] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, October 2021.

[103] V. Petsiuk, A. E. Siemenn, S. Surbehera, Z. Chin, K. Tyser, G. Hunter, A. Raghavan, Y. Hicke, B. A. Plummer, O. Kerret, et al. Human evaluation of text-to-image models on a multi-task benchmark. In *NeurIPS Workshop on Human Evaluation of Generative Models*, 2022.

[104] Q. Phung, S. Ge, and J.-B. Huang. Grounded text-to-image synthesis with attention refocusing. *arXiv preprint arXiv:2306.05427*, 2023.

[105] A. Plumerault, H. L. Borgne, and C. Hudelot. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations*, 2020.

[106] V. Prabhu, S. Yenamandra, P. Chattopadhyay, and J. Hoffman. Lance: Stress-testing visual models by generating language-guided counterfactual images. In *Advances in Neural Information Processing Systems*, 2023.

[107] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022.

[108] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

[109] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 2021.

[110] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[111] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[112] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

[113] V. Ranjan and M. Hoai. Vicinal counting networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4220–4229, 2022. doi: 10.1109/CVPRW56347.2022.00467.

[114] V. Ranjan, U. Sharma, T. Nguyen, and M. Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021.

[115] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[116] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021.

[117] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[118] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[119] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[120] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

[121] M. B. Sariyildiz, K. Alahari, D. Larlus, and Y. Kalantidis. Fake it till you make it: Learningto-count transferable representations from synthetic imagenet clones. In *CVPR 2023–IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[122] A. Saseendran, K. Skubch, and M. Keuper. Multi-class multi-instance count conditioned adversarial image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[123] A. Sauer, T. Karras, S. Laine, A. Geiger, and T. Aila. StyleGAN-T: Unlocking the power of GANs for fast large-scale text-to-image synthesis. In *International Conference on Machine Learning*, volume abs/2301.09515, 2023. URL `https://arxiv.org/abs/2301.09515`.

[124] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25278–25294. Curran Associates, Inc., 2022.

[125] I. Schwartz, V. Snæbjarnarson, H. Chefer, R. Cotterell, S. Belongie, L. Wolf, and S. Benaim. Discriminative class tokens for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[126] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, June 2021.

[127] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[128] M. Shi, H. Lu, C. Feng, C. Liu, and Z. Cao. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9529–9538, 2022.

[129] J. Shipard, A. Wiliem, K. N. Thanh, W. Xiang, and C. Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 769–778, 2023. doi: 10.1109/CVPRW59228.2023.00084.

[130] A. Shoshan, N. Bhonker, I. Kviatkovsky, and G. Medioni. Gan-control: Explicitly controllable gans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14083–14093, October 2021.

[131] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[132] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[133] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[134] N. Spingarn-Eliezer, R. Banner, and T. Michaeli. Gan" steerability" without optimization. In *International Conference on Learning Representations*, 2021.

[135] S.-H. Sun. Multi-digit mnist for few-shot learning, 2019. URL `https://github.com/shaohua0116/MultiDigitMNIST`.

[136] V. Sushko, E. Schönfeld, D. Zhang, J. Gall, B. Schiele, and A. Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning Representations*, 2021.

[137] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[138] S. Tan, Y. Shen, and B. Zhou. Improving the fairness of deep generative models without retraining. *arXiv preprint arXiv:2012.04842*, 2020.

[139] Y. Tian, L. Fan, P. Isola, H. Chang, and D. Krishnan. StableRep: Synthetic images from text-to-image models make strong visual representation learners. *arXiv preprint arXiv:2306.00984*, 2023.

[140] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459838. URL https://doi.org/10.1145/3450626.3459838.

[141] B. Trabucco, K. Doherty, M. Gurinas, and R. Salakhutdinov. Effective data augmentation with diffusion models. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. URL https://openreview.net/forum?id=dcCpG0CVMf.

[142] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[143] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[144] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[145] J. Vendrow, S. Jain, L. Engstrom, and A. Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. In *Workshop at International Conference on Machine Learning*, 2023.

[146] C. Villani. *Optimal Transport: Old and New*. Springer Berlin Heidelberg, 2008.

[147] A. Voynov and A. Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In *International Conference on Machine Learning*, pages 9786–9796, 2020.

[148] B. Wallace, A. Gokul, and N. Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023.

[149] B. Wang and C. R. Ponce. A geometric analysis of deep generative image models and its applications. In *International Conference on Learning Representations*, 2021.

[150] H.-P. Wang, N. Yu, and M. Fritz. Hijack-gan: Unintended-use of pretrained, black-box gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7872–7881, 2021.

[151] Q. Wang, J. Gao, W. Lin, and Y. Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8198–8207, 2019.

[152] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[153] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22445–22455, October 2023.

[154] O. Wiles, I. Albuquerque, and S. Gowal. Discovering bugs in vision models using off-the-shelf image generation and captioning. In *NeurIPS ML Safety Workshop*, 2022. URL `https://openreview.net/forum?id=maBZZ_W0lD`.

[155] Z. Wu, D. Lischinski, and E. Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021.

[156] C. Yang, Y. Shen, and B. Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision*, 2020.

[157] X. Yao, A. Newson, Y. Gousseau, and P. Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[158] Z. You, K. Yang, W. Luo, X. Lu, L. Cui, and X. Le. Few-shot object counting with similarity-aware feature enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6315–6324, January 2023.

[159] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[160] L. Zeqiang, Z. Xizhou, D. Jifeng, Q. Yu, and W. Wenhai. Mini-dalle3: Interactive text to image by prompting large language models. *arXiv preprint arXiv:2310.07653*, 2023.

[161] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

[162] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[163] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021.

[164] S. Zhao, H. Ren, A. Yuan, J. Song, N. Goodman, and S. Ermon. Bias and generalization in deep generative models: An empirical study. volume 31, 2018.

[165] H. Zheng, Z. Lin, J. Lu, S. Cohen, J. Zhang, N. Xu, and J. Luo. Semantic layout manipulation with high-resolution sparse attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3768–3782, 2022.

[166] J. Zhu, Y. Shen, D. Zhao, and B. Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020.

[167] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *Proceedings of the European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[168] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[169] M. Zhu, H. Chen, Q. Yan, X. Huang, G. Lin, W. Li, Z. Tu, H. Hu, J. Hu, and Y. Wang. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in Neural Information Processing Systems*, 36, 2024.

[170] P. Zhu, R. Abdal, Y. Qin, and P. Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020.

[171] P. Zhuang, O. O. Koyejo, and A. Schwing. Enjoy your editing: Controllable GANs for image editing via latent space navigation. In *International Conference on Learning Representations*, 2021.

Résumé en Français

Introduction

L'intelligence artificielle générative appliquée aux contenus visuels est un outil très prometteur à la fois pour la génération et l'édition de contenu dans divers domaines créatifs (*e.g.* mode, jeux vidéos, cinéma) mais également au sein même du domaine de l'intelligence artificielle, en particulier pour la génération de jeux de données servant à l'entraînement de réseaux de neurones profonds. Pour toutes ces applications, un aspect important de ces outils est de permettre un contrôle significatif sur le contenu généré. Ce contrôle peut être varié et dépend de la nature des images considérées. Dans le cas d'images de visages, il peut par exemple concerner le contrôle d'attributs sémantiques tels que la présence de lunettes ou encore l'intensité du sourire. Plus largement dans une scène visuelle quelconque, cela peut-être le type, le nombre ou encore la position des objets les uns par rapport aux autres ou à la caméra. Dans le cadre de l'augmentation de données, ce contrôle peut permettre d'aborder différents défis pour construire de meilleures bases de données: générer des images avec les propriétés les moins représentées pour avoir des bases de données plus "fair", avec des nouvelles combinaisons de facteurs de variation pour améliorer la diversité ou encore des images difficiles à capturer en pratique (*e.g.* évènements rares) pour tester et améliorer la robustesse des modèles.

Pour la génération d'images, deux types de modèles se sont montrés particulièrement performants: les modèles génératifs antagonistes (GANs) [41] introduits en 2014 et les modèles à diffusion (DDPMs) [49] introduits en 2020. Divers travaux se focalisent sur le contrôle de la génération avec ces modèles. En particulier, certains travaux s'intéressent à l'identification de trajectoires dans l'espace latent, en particulier des GANs, correspondant à des propriétés interprétables dans l'espace des images [42, 127, 126, 105, 57]. Ces méthodes offrent un moyen de contrôler certaines propriétés des images générées en déplaçant les codes latent associés selon les trajectoires découvertes, qui peuvent

prendre différentes formes (*e.g.* linéaires ou non-linéaires). Parmi les différentes approches pour identifier ces directions sémantiques, les méthodes non supervisées [126, 147, 42] requiert d'interpréter manuellement chaque direction. Pour éviter l'inspection manuelle, d'autres approches utilisent de l'apprentissage auto-supervisé [57, 105] ou supervisé [127, 171, 3, 51, 150, 157]. En particulier, les approches supervisées s'appuient sur des classifieurs d'attributs *e.g.* faciaux [83], notamment en labellisant les codes latent, pour apprendre des directions correspondant à ces différents attributs. Les trajectoires sémantiques doivent garantir plusieurs propriétés, en plus de contrôler l'attribut d'intérêt: préserver les autres attributs (désemmêlement) et l'apparence globale de l'image (identité dans le cas d'image de visages) et produire des images réalistes. Cependant, il est commun d'observer qu'une direction affecte plusieurs attributs à la fois. On observe également que certaines méthodes conduisent à l'apprentissage de directions produisant des images avec des artefacts importants.

Dans la première partie de cette thèse, nous nous intéressons à améliorer le contrôle sur les images générées en proposant des solutions pour garantir le désemmêlement et la robustesse des directions sémantiques. Dans un second temps, nous étudions comment exploiter le contrôle pour l'augmentation de données pour les tâches nécessitant un contrôle spatial comme le comptage d'objets ou la segmentation. Pour ce type de tâches, nous exploitons les approches conditionnelles qui permettent d'obtenir du contrôle en ajoutant une condition pendant l'entraînement des modèles génératifs et supportent divers types de conditionnement *e.g.* texte, image. En particulier, nous explorons comment adapter des modèles pré-entraînés sur de large bases de données conditionnés sur du texte et une condition spatiale pour générer des augmentations sémantiques diversifiées.

Échantillonage équilibré pour des éditions desemmêlées

Plusieurs travaux exploitent les frontières de décision de classifieurs binaires entraînés dans l'espace latent pour l'extraction des directions sémantiques [127, 24, 156]. En particulier, étant donné un attribut binaire tel que "lunettes" (présence vs. absence de lunettes), les auteurs d'InterfaceGAN (IfGAN) [127] font l'hypothèse que cet attribut peut-être linéairement séparé dans l'espace latent et proposent d'utiliser le vecteur orthogonal à l'hyperplan séparateur pour contrôler l'attribut. Pour trouver l'hyperplan, ils entraînent un SVM, pour chaque attribut à contrôler, sur un ensemble de paires (codes latent, label). Cet ensemble est obtenu en échantillonnant des codes de l'espace latent puis en passant les images correspondantes à travers un classifieur d'attributs. Cette approche conduit

souvent à extraire des directions qui produisent des changements indésirables pour d'autres attributs.

Plusieurs travaux ont montré que l'emmêlement observé reflète les biais présents dans les jeux de données utilisés pour entraîner les modèles génératifs pré-entraînés [127, 24]. Pour réduire cet emmêlement, l'approche de "manipulation conditionnelle" est une approche commune qui consiste à raffiner les directions *a posteriori* en imposant une contrainte d'orthogonalité sur les nouvelles directions [127]. D'autres approches proposent d'apprendre les directions simultanément [171, 3] ou de manipuler seulement certains sous-espaces de l'espace latent [51]. Pour éviter un post-processing, nous proposons une méthode simple et générale qui consiste à ré-équilibrer les jeux de données par rapport à plusieurs attributs, avant l'apprentissage des directions.

Étant donné G un générateur pré-entraîné qui associe à un code latent $\mathbf{z}$, échantillonné depuis un espace latent $\mathcal{Z} \in \mathbb{R}^d$, une image $\mathbf{I} = G(\mathbf{z})$. Supposons que l'image I est décrite par un ensemble d'attributs binaires $\mathcal{A} = \{a_k, 1 \leq j \leq m\}$. Pour chacun des attributs a_k, nous cherchons une direction linéaire dans l'espace latent définie par un vecteur unitaire $\mathbf{u}_k \in \mathbb{R}^d$:

$$\mathbf{z}' = \mathbf{z} + \alpha \mathbf{u}_k \tag{6.1}$$

telle que seulement l'intensité de a_k diffère dans l'image résultante $\mathbf{I}' = G(\mathbf{z}')$, avec $\alpha \in \mathbb{R}$ le paramètre contrôlant cette intensité. Nous notons $\mathcal{S} = \{(\mathbf{z}^{(i)}, F_{\mathcal{I}}(G(\mathbf{z}^{(i)})))_{i=1}^n\}$ un ensemble de codes latents annotés à l'aide d'un classifieur pré-entraîné $F_{\mathcal{I}}$. La distribution des attributs d'un ensemble de données peut-être représentée par une table de contingence de dimension m où chacune des 2^m cellules contient le nombre d'échantillons qui ont la combinaison d'attributs correspondante pour les m attributs. S'il y a des fortes corrélations entre attribut, la table de contingence pour ces données est fortement déséquilibrée. L'ensemble de données $\mathcal{S}$ généré par le GAN pré-entraîné devrait montrer les mêmes corrélations ou pire que le jeu d'apprentissage du modèle [164]. Pour un attribut a_j, les ensembles $\mathcal{S}_j^+$ et $\mathcal{S}_j^-$ reflètent le déséquilibre de $\mathcal{S}$. Il est donc raisonnable de faire l'hypothèse qu'un classifieur latent Ψ_j entraîné sur ces données déséquilibrées est influencé par les fortes corrélations. Pour éviter la propagation des biais sur les directions sémantiques, nous proposons donc de sous-échantillonner les données de $\mathcal{S}$ pour obtenir approximativement le même nombre de données dans chaque cellule de la matrice de contingence. Nous construisons un ensemble de données $\mathcal{B} \subset$ équilibré par rapport à plusieurs attributs en sélectionnant itérativement des données de $\mathcal{S}$ jusqu'à obtenir le nombre de données $N_0 \leq N$ désiré. À chaque itération, nous échantillonnons uniformément une combinaison

d'attribut et une donnée $(\mathbf{z}, F_{\mathcal{I}}(G(\mathbf{z})))$ avec cette combinaison (sans replacement pour la donnée). Notre procédure d'échantillonage permet d'obtenir un échantillon $\mathcal{B}$ de taille N_0 qui est équilibré par rapport à tous les attributs. La direction $\mathbf{u}_j$ est ensuite calculée en prenant la direction connectant les centroïdes de $\mathcal{S}_j^+$ et $\mathcal{S}_j^-$.

Expériences et résultats. Nous évaluons notre approche sur deux jeux de données de visages: CelebAHQ [83] et FFHQ [64] pour l'édition des attributs faciaux suivants: 'lunettes', 'genre', 'sourire' et 'age'. Nous comparons notre approche avec IfGAN sans et avec manipulation conditionnelle pour différentes architectures de GANs: PGGAN [63] et les modèles StyleGAN [64, 65, 66]. Les métriques utilisés pour évaluer les différentes méthodes sont issue de la métrique de re-scoring introduit dans [127], qui mesure les attributs affectés par l'édition (pour un α fixé) à l'aide de classifieurs images. La métrique d'effet mesure les changements pour l'attribut d'intérêt tandis que la métrique d'emmêlement mesure les changements pour les autres attributs. Nous montrons que notre méthode permet de réduire l'emmêlement significativement par rapport à IfGAN et a des performances similaires à IfGAN avec la manipulation conditionnelle. Cependant, notre méthode a l'avantage de produire des directions qui ont légèrement plus d'effet et ne requiert pas de post-processing.

Distance de Wasserstein pour des éditions robustes

Différentes approches introduites à la suite d'IfGAN recherchent des directions sémantiques plus précises, en particulier, qui dépendent du code latent initial [171, 51, 157]. Par rapport à Eq. (6.1), l'idée est d'apprendre une fonction $\mathbf{H}$, qui peut-être modélisée par un MLP [51] ou une transformation affine [157], comme suit :

$$\mathbf{z}' = \mathbf{z} + \alpha \cdot \mathbf{H}(\mathbf{z}), \quad \alpha \in \mathbb{R} \tag{6.2}$$

Contrairement aux approches précédentes qui exploitent des classifieurs binaires et linéaires, ces méthodes s'appuient sur la supervision indirecte de classifieurs non-linéaires multi-attribut pour l'apprentissage de la fonction $\mathbf{H}$ via un objectif de classification. En particulier, l'approche état-de-l'art LatentTransformer (LT) [157] propose d'exploiter un classifier latent Ψ pour réduire le coût de calcul et introduit également une contrainte explicite de désemmêlement. Cette contrainte consiste à imposer que les prédictions de Ψ pour les codes édités dévient le moins possible des prédictions pour le code

initial, en ce qui concerne les attributs non-cibles :

$$\mathcal{L}_{\text{attr}} = \sum_{i \neq k} ||\Psi_i(\mathbf{z}') - \Psi_i(\mathbf{z})||_2^2 \tag{6.3}$$

Cependant, les classifieurs non-linéaires sont des modèles connus pour être peu fiables. En particulier, certains travaux ont montré leur manque de robustesse face à des petites perturbations [137] ou leur incapacité à correctement estimer l'incertitude [93]. Ces limitations peuvent impacter les méthodes d'édition s'appuyant sur ces modèles. En particulier, à travers des expériences préliminaires, nous montrons que l'objectif de classification employé avec la contrainte définie dans Eq. (6.3) conduit à apprendre des directions sémantiques produisant des images peu réalistes bien que l'objectif de classification soit satisfait. Il est possible d'ajouter une régularisation supplémentaire qui minimise la norme d'édition pour s'assurer de rester proche du code initial. Cependant, cette régularisation n'est pas toujours efficace et peut produire dans certains cas des codes édités adverses. Nous proposons donc une méthode alternative basé sur le transport optimal qui ne reposent pas entièrement sur des classifieurs ou d'autres modèles auxiliaires.

Distance de Wasserstein. La théorie du transport optimal introduit un framework permettant de transporter une distribution source (μ_s) vers une distribution cible (μ_t) de la manière la moins coûteuse possible. Le coût du plan de transport optimal définie une distance entre les deux distributions appelée distance de Wasserstein. Plus formellement, étant donné deux distributions discrètes: $\mu_s = \sum_{i=1}^{n_s} p_i^s \delta(x_i)$ et $\mu_t = \sum_{j=1}^{n_t} p_j^t \delta(y_j)$ où δ est la fonction Dirac et p_i^s, p_i^t sont les masses associées à chaque point. La distance de Wasserstein entre μ_s et μ_t est définie comme suit:

$$W(\mu_s, \mu_t) = \min \sum_{i,j} T_{i,j} c_{i,j}, \quad \text{s.t. } T\mathbf{1}_{n_t} = \mu_s \text{ and } T^\top \mathbf{1}_{n_s} = \mu_t \tag{6.4}$$

où T est la matrice de transport et représente combien de masse doit être transportée du point x_i au point y_j et $c_{i,j}$ représente le coût du transport, par exemple, la distance Euclidienne entre x_i et y_j.

Comme précédemment, nous considérons un GAN pré-entraîné G et un ensemble de codes latent $\mathcal{S} = \{(\mathbf{z}^{(i)})_{i=1}^N\}$ où chaque code est annoté pour un ensemble d'attributs $\mathcal{A} = \{a_1, a_2, ..., a_k\} \in \{0, 1\}$. Pour chaque attribut a_k, nous cherchons une transformation affine $\mathbf{H}_k$ comme dans Eq. (6.2), qui permet de faire varier l'intensité de l'attribut a_k dans l'image générée. Considérons μ_s^k, la distribution

des codes latent $\mathbf{z}_k$ négatifs par rapport à l'attribut a_k et μ_t^k la distribution des codes positifs. Pour apprendre des directions qui ont un effet sur l'attribut a_k, nous proposons de minimiser la distance de Wasserstein entre la distribution des codes latent édités dénotée par $\mu_s'^k$ et la distribution μ_t^k. Nous utilisons en particulier cette distance afin d'exploiter le coût pour minimiser implicitement l'emmêlement. En particulier, nous supposons que l'emmêlement peut-être évité si les points de la distribution source sont transportés vers des points partageant la même apparence et les mêmes attributs (en dehors de l'attribut cible) et donc par la minimisation d'un coût représentant la similarité perceptuelle. Empiriquement, nous montrons que deux codes proches en terme de distance L2 partagent également des propriétés sémantiques et faisons donc l'hypothèse que ce coût peut être utilisé comme un proxy d'une distance sémantique dans l'espace image. Nous définissons donc l'objectif d'édition comme la distance de Wasserstein entre $\mu_s'^k$ et μ_t^k avec un coût L2:

$$\mathcal{L}_{\text{edit}} = W\left(\mu_s'^k, \mu_t^k\right), \quad x_i = \mathbf{z}'^{(i)}_k \text{ and } y_j = \bar{\mathbf{z}}^{(j)}_k$$
$$c_{i,j} = \frac{1}{2}\|x_i - y_j\|^2 \tag{6.5}$$

Pour s'assurer que les codes édités partagent bien les mêmes attributs que les codes initiaux, nous proposons d'ajouter une régularisation explicite qui impose de rester proche de la distribution initiale μ_s dans l'espace des attributs. Nous définissons cette régularisation comme la distance de Wasserstein entre $\mu_s'^k$ et μ_s^k avec un coût exprimé à l'aide d'un classifieur latent Ψ similairement à [157]:

$$\mathcal{L}_{\text{pres}} = W\left(\mu_s'^k, \mu_s^k\right), \quad x_i = \mathbf{z}'^{(i)}_k \text{ and } y_j = \mathbf{z}^{(j)}_k$$
$$c_{i,j} = \frac{1}{2}\sum_{l \neq k} \|\Psi_l(x_i) - \Psi_l(y_j)\|_2^2 \tag{6.6}$$

Expérences et résultats. Nous évaluons notre approche sur deux jeux de données de visages (CelebAHQ, FFHQ) pour l'édition d'attributs faciaux et sur le jeu de données Multi-digit MNIST [135] pour le contrôle du nombre de chiffres MNIST dans une image. Nous comparons notre méthode avec LT pour l'édition d'image réelles inversées dans l'espace latent $\mathcal{W}+$ de StyleGAN2 [65] avec l'encoder e4e [140]. Pour les visages, nous utilisons 3 métriques: le taux d'attributs non-cible préservés, le taux de préservation de l'identité et le taux d'images pour lesquelles l'attribut cible est correctement modifié. Ces métriques sont calculées pour différentes valeurs de α. Pour les chiffres MNIST, nous mesurons simplement le taux d'images correctement modifiées pour un α fixé. Avec la régularisation de désemmêlement explicite, les deux approches montrent en moyenne des performances similaires sur

les données visages mais notre méthode ne requiert pas de régularisation sur la norme de l'édition donc a moins d'hyper-paramètres. Sur les données Multi-MNIST, nous montrons aussi que cette contrainte cause une importante baisse des performances car la plupart des images éditées obtenues n'exhibent pas les changements désirés (exemples adverses). Sans la régularisation de désemmêlement explicite, notre méthode surpasse largement les performances de LT. Nous attribuons ces résultats à l'efficacité de notre contrainte qui impose implicitement de transporter des points vers d'autres points sémantiquement similaires.

Contrôle pour des augmentations sémantiques diversifiées

Une des applications prometteuse du contrôle est la génération d'augmentation de données, en particulier la génération d'augmentations annotées et plus diverses que les données initiales.

Les modèles génératifs ont connu un tournant important avec l'introduction des modèles à diffusion [50] entraînés sur des milliards de paires d'images et de texte [120, 117, 95, 112]. Des travaux récents ont montré que ces modèles pré-entraînés qui synthétisent des images à partir d'une requête textuelle peuvent être utilisés sans ré-entraînement pour produire des augmentations de données sémantiques qui améliorent efficacement les modèles de classification [45, 141, 129]. Ces augmentations sont généralement obtenues en fournissant aux modèles des requêtes textuelles incluant les différentes classes à discriminer. Cependant, ces méthodes sont limitées à la classification et ne s'appliquent pas directement à d'autres tâches, en particulier les tâches qui requiert la préservation du nombre objets et leur position spatiale. En effet, le contrôle par le texte échoue souvent lorsque les requêtes textuelles contiennent des concepts de composition [103, 99]. Divers travaux proposent des stratégies pour améliorer la compréhension des encodeurs de textes. D'autres approches se focalisent sur l'ajout de conditions supplémentaires pour avoir un contrôle plus fin sur la génération tout en préservant les capacités initiales des modèles [161, 91]. Nous proposons d'exploiter ces approches pour générer des augmentations pour des tâches qui requiert un contrôle spatial telles que le comptage et détection d'objets, la segmentation, et d'exploiter les différents contrôles pour diversifier les données générées.

Étant donné un modèle génératif pré-entraîné sur une grande base de données et conditionné par du texte, nous proposons d'abord de finetuner le modèle avec une condition spatiale propre à la tâche *e.g.*

cartes de densité pour le comptage d'objet, cartes sémantiques pour la segmentation, avec la stratégie ControlNet [161]. Pour le contrôle textuel, une stratégie commune consiste à construire des requêtes textuelles de la forme "une photo de $\{classe\}$" où *classe* indique le type d'objets à générer [45, 141]. Nous proposons à la place de passer les images à travers un modèle de captioning [79, 80] pour obtenir des requêtes textuelles plus diverses qui seront utilisées lors du finetuning.

Considérons $\mathcal{D}_{\text{train}} = \{x_i, y_i\}_{i=1}^{N}$ un jeu de données annotés avec x_i une image et y_i les annotations spatiales associées. Considérons $\mathcal{C} = \{c_i\}$ l'ensemble des descriptions obtenues avec le modèle de captioning. Pour chaque image x_i, notre but est de générer M augmentations à l'aide de notre modèle génératif finetuné $g(y_i, c_i)$.

Baseline. Nous échantillonnons des augmentations en profitant du processus de génération non-déterministe des modèles à diffusion et de l'expressivité du modèle lié à son pré-entraînement. Pour une image x_i, nous générons M augmentations $\tilde{x}_i^{(j)}$ qui partagent la même description et organisation spatiale que x_i: $\tilde{x}_i^{(j)} = g(y_i, c_i), \quad j = 1, ..., M$. Ces augmentations préservent le nombre d'objets et leur disposition spatiale - via le conditionnement spatial et la sémantique - via la requête textuelle. Cette stratégie permet déjà d'augmenter le nombre d'exemples d'apprentissage, cependant nous proposons de diversifier d'avantage les augmentations.

Diverse. Nous proposons de mélanger les deux ensembles de contrôles: les conditions spatiales et les descriptions textuelles, pour produire des augmentations sémantiquement et géométriquement plus diverses que le jeu de données original. Cependant, ce mélange doit être réalisé avec précaution pour éviter de mauvaises générations. En effet, certains combinaisons ne sont pas cohérentes, *e.g.* "un troupeau de vaches" et "un collier de perles" ont des organisations spatiales très différentes et mêmes incompatibles. Pour fournir au modèle génératif des combinaisons (y_i, c_i) réalistes, nous nous appuyons sur la similarité entre les descriptions textuelles pour trouver des nouvelles associations qui partagent une certaine sémantique comme "vaches" et "'bisons". Plus précisément, nous échangeons de manière aléatoire des descriptions textuelles entre images *compatibles*. Deux images sont considérées compatibles si leurs descriptions textuelles sont plus similaires qu'un certain seuil t_c:

$$\text{sim}(c_i, c_k) = \frac{\Phi(c_i)^\top \Phi(c_k)}{||\Phi(c_i)||_2 ||\Phi(c_k)||_2} > t_c \tag{6.7}$$

où Φ est un encodeur de texte approprié *e.g.* encodeur du modèle de captioning, CLIP [109]. Nous échantillonnons des nouvelles images en gardant la condition spécifique à la tâche fixée mais en replaçant

la description textuelle initiale avec la description $c_k \in \mathcal{C}$ d'une autre exemple d'apprentissage compatible choisi de manière aléatoire: $\tilde{x}_i^{(j)} = g(y_i, c_k), \quad j = 1, ..., M$. Cette procédure permet d'obtenir des augmentations plus diversifiées en comparaison à l'approche baseline.

Expériences et résultats. Nous évaluons notre méthode sur l'augmentation du jeu de données FSC-147 [114] utilisé pour le comptage d'objets à partir de quelques exemples [84]. Cette tâche consiste à entraîner des réseaux de comptage d'objets agnostiques à la catégorie des objets. Pour cela, les réseaux prennent en entrée l'image cible et quelques "objets exemples" (1∼3) annotés avec des boîtes englobantes et prédisent une carte de densité indiquant à chaque pixel la probabilité de la présence d'un objet. Nous finetunons Stable Diffusion [117] avec les cartes de densités vérité terrain de FSC147 et des descriptions textuelles obtenues avec BLIP2 [80]. Une fois le jeu de données augmenté ($M = 10$), nous entraînons deux réseaux de comptage: SAFECount [158] et CounTR [14]. Nous comparons notre méthode d'augmentation avec celle de He et al. [45] et les modèles entraînés sans augmentations. Nous évaluons aussi l'influence des différents hyper-paramètres, notamment M et le seuil de similarité des descriptions textuelles t_c. Pour les deux méthodes de comptage, les modèles entraînés avec les données synthétiques générées avec notre méthode montrent des performances supérieures aux modèles entraînés sans données synthétiques ou avec les données générées avec la méthode de He et al. [45].

Conclusion et perspectives

Dans cette thèse, nous nous sommes intéressés à l'amélioration du contrôle sur les propriétés des images générées par des modèles génératifs profonds. En particulier, nous avons abordé l'apprentissage de contrôles désemmêlés et robustes correspondant à des attributs sémantiques, en exploitant l'espace latent de modèles pré-entraînés. Pour éviter la propagation des biais de ces modèles sur les contrôles, nous avons proposé une stratégie de ré-équilibrage de codes latent annotés utilisés pour l'apprentissage des contrôles et validé l'efficacité de cette approche en comparaison à une méthode nécessitant un post-processing. Nous avons également proposé une approche plus robuste pour la recherche de ces contrôles, basée sur le transport optimal pour éviter l'utilisation de classifieurs peu fiables. En particulier, nous avons re-formulé le problème comme un transport entre deux distributions - la distribution sans l'attribut d'intérêt vers la distribution avec l'attribut d'intérêt - avec un coût représentant la similarité perceptuelle pour préserver le desemmêlement. Nous avons démontré que cette approche produit

des performances comparables à une méthode basée sur un classifieur tout en étant plus robuste et requérant moins d'hyper-paramètres. Enfin, nous avons également exploité l'utilisation du contrôle et en particulier des modèles conditionnés sur le texte pour générer des jeux de données diversifiés adaptés à une tâche spécifique. Nous avons introduit une stratégie consistant à finetuner ces modèles avec un contrôle spatial supplémentaire spécifique à la tâche et utilisé le double contrôle pour synthétiser des images avec des nouvelles combinaisons de sémantique et de disposition spatiale. Nous avons montré que cette approche permet d'améliorer les performances des modèles de comptage d'objets agnostiques à la catégorie des objets.

Cette dernière contribution tire profit du contrôle pour cibler des augmentations avec plus de diversité. Un autre utilisation pertinente du contrôle est la génération d'images se focalisant sur les faiblesses des modèles. La difficulté est d'identifier automatiquement les contrôles qui conduisent à des images pour lesquelles les modèles sont incertains ou se trompent. Plusieurs méthodes récentes se penchent sur ce problème. Par exemple, Jain et al. [59] projettent des images de validation dans un espace commun image-texte et apprennent les frontière de séparation entre les images correctement et mal classifiées. D'autres approches éditent des images avec différents changements soit déterminés manuellement [145] soit avec des LLMs [106] et évaluent ensuite les modèles sur ces images pour identifier lesquels causent le plus d'erreurs. Pour découvrir plusieurs modes d'erreur en ne s'appuyant pas sur une stratégie exhaustive, une méthode qui apprend automatiquement un ensemble de tokens en maximisant un objectif d'erreur ou d'incertitude sur le modèle, pourrait être envisagée.

L'intelligence artificielle générative permet de nombreuses applications prometteuses mais peut aussi être utilisée à des fins malveillantes. Par exemple, le contenu généré peut être utilisé pour propager des fake news, humilier ou faire chanter des individus. Différents travaux s'intéressent à développer des méthodes pour détecter les contenus générés mais ces méthodes échouent à généraliser correctement sur des nouveaux modèles génératifs [153]. Des travaux récents proposent d'aborder ce problème en créant des bases de données regroupant des images générées par différents modèles [169]. Les méthodes d'édition proposés dans cette thèse pourrait être utilisés pour générer des exemples difficiles avec des deepfakes localisés et entraîner des détecteurs à localiser ces éditions pour leur apprendre à mieux différencier le contenu réel du contenu généré.

Appendix A

Attribute correlations in CelebA

CelebA dataset [83] is a dataset composed of 200K images of celebrities, where each image is annotated with 40 "binary" semantic attributes. The complete list can be found in Fig. A.1. Fig. A.2, shows the correlations between the attributes.

Index	Definition	Index	Definition	Index	Definition	Index	Definition
1	5o'ClockShadow	11	Blurry	21	Male	31	Sideburns
2	ArchedEyebrows	12	BrownHair	22	MouthSlightlyOpen	32	Smiling
3	Attractive	13	BushyEyebrows	23	Mustache	33	StraightHair
4	BagsUnderEyes	14	Chubby	24	NarrowEyes	34	WavyHair
5	Bald	15	DoubleChin	25	NoBeard	35	WearingEarrings
6	Bangs	16	Eyeglasses	26	OvalFace	36	WearingHat
7	BigLips	17	Goatee	27	PaleSkin	37	WearingLipstick
8	BigNose	18	GrayHair	28	PointyNose	38	WearingNecklace
9	BlackHair	19	HeavyMakeup	29	RecedingHairline	39	WearingNecktie
10	BlondHair	20	HighCheekbones	30	RosyCheeks	40	Young

Figure A.1: List of attributes in CelebA [83].

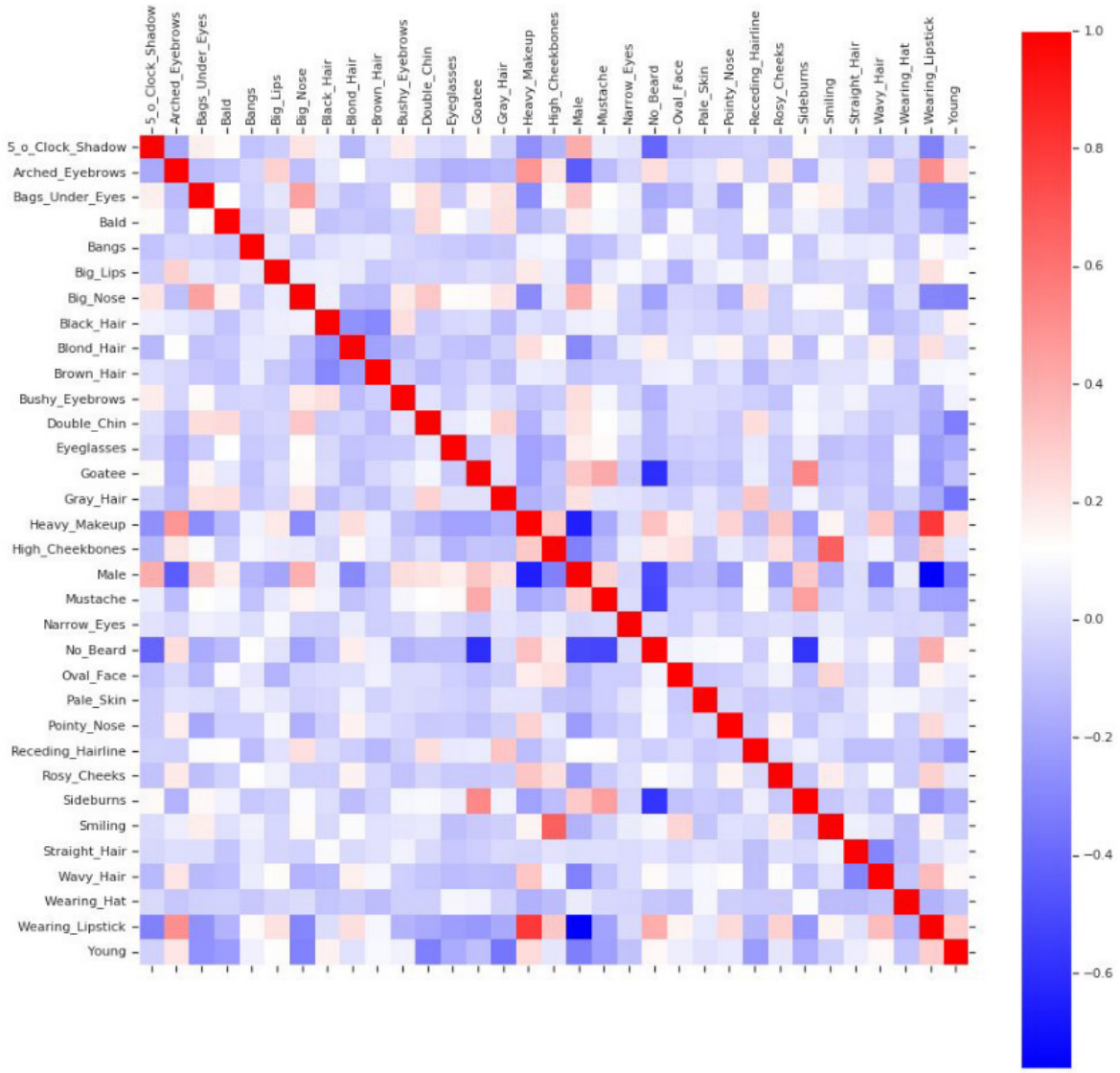


Figure A.2: Correlation matrix between CelebA attributes.

Appendix B

Additional experiments on Multi-digit MNIST and CLEVR

B.1 Additional experiments on Multi-digit MNIST

We present some additional experiments on the control of the number of digits in Multi-digit MNIST images. We explore a multi-class extension of the framework of [157]. The following short-paper summarizes our methodology and findings and was presented at RFIAP in 2022.

Contrôle de la cardinalité par navigation dans l'espace latent des GANs

1 Introduction

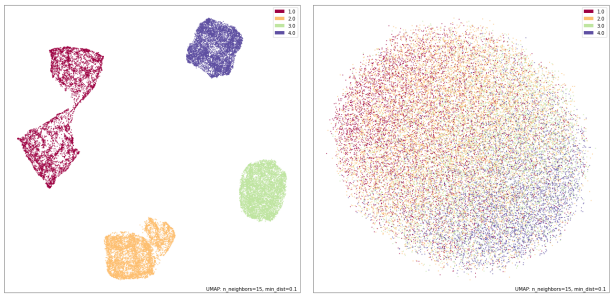
L'intégration de la notion de cardinalité se fait de manière naturelle chez les humains. Il est donc intéressant de se demander si les modèles génératifs comme les GANs intègrent également cette notion. Cela pourrait permettre de contrôler le nombre d'objets présents dans une image générée ou réelle grâce aux méthodes d'inversion. Contrôler le nombre de piétons ou de voitures sur des données représentant des scènes urbaines, par exemple, peut être particulièrement utile pour augmenter de manière ciblée des bases de données existantes. Saseendran et al. proposent une adaptation du GAN conditionnel pour générer des images conditionnées sur le nombre d'objets. L'inconvénient d'utiliser un GAN conditionnel est la nécessité de ré-entrainer le modèle. Nous souhaitons étudier si les modèles pré-entraînés encodent automatiquement la notion de cardinalité. Divers travaux [10, 13, 3, 4, 1, 9] ont étudié l'existence d'une structure reflétant la sémantique des données d'entraînement dans l'espace latent des GANs. Certains travaux ont montré que, pour des GANs entraînés sur des images représentant des visages, il est possible d'identifier des directions permettant de contrôler la présence ou l'intensité d'attributs binaires ou continus comme le genre ou l'âge. Parmi ces méthodes, certaines approches supervisées [13, 10, 3] s'appuient sur des classifieurs binaires entraînés dans l'espace latent pour guider l'apprentissage des directions. Dans ce travail, nous proposons une extension de ces méthodes à la cardinalité, dans un formalisme de régression ou de classification à plusieurs classes. Nous appliquons notre méthode dans l'espace latent $\mathcal{W}$ et la version étendue $\mathcal{W}+$ d'un StyleGAN entrainé sur le jeu de données MultiMNIST. Nos expérimentations sur ce jeu de données montrent que la formalisation multi-classe dans l'espace $\mathcal{W}$ permet de contrôler le nombre de chiffres présents dans une image.

2 Méthode

Étant donné un GAN pré-entraîné que l'on note $\mathbf{G}(\cdot)$ et l'espace latent noté $\mathcal{Z}$ tel que $\mathbf{I} = \mathbf{G}(\mathbf{z})$ avec $\mathbf{z} \in \mathcal{Z}$, nous cherchons à trouver une direction dans cet espace tel que lorsque l'on se déplace dans cette direction, le nombre d'objets présents dans l'image $\mathbf{I}$ varie. Nous adoptons le formalisme de Yao et al. pour la recherche de directions locales : $\mathbf{z}' = \mathbf{z} + \alpha \cdot \mathbf{H}(\mathbf{z})$ où $\mathbf{H}$ est une transformation linéaire et $\alpha \in \mathbb{N}$. L'image modifiée est alors $\mathbf{I}' = \mathbf{G}(\mathbf{z}')$ et doit contenir α objets de plus que l'image $\mathbf{I}$. Pour contrôler la cardinalité, nous proposons d'utiliser un classifieur noté $\mathbf{C}(\cdot)$ entraîné à prédire le nombre d'objets étant donné un code latent $\mathbf{z}$. Ce classifieur permet de guider l'apprentissage des directions grâce à la fonction de coût suivante : $\mathcal{L}_{count} = \text{loss}(\mathbf{C}(\mathbf{z}'), \mathbf{y}+\alpha))$ où $\mathbf{y} = \mathbf{C}(\mathbf{z})$ est le nombre d'objets avant déplacement. Pour l'apprentissage du classifieur $\mathbf{C}(\cdot)$ puis de la matrice $\mathbf{H}$, nous pouvons modéliser le problème de deux façons : régression (MSE) ou classification multi-classe (entropie croisée) où chaque classe correspond à un nombre d'objets. La fonction de coût $\mathcal{L}_{\text{count}}$ permet de s'assurer que la direction nous emmène dans une région de l'espace latent où les codes latents correspondent au nombre d'objets souhaité du point de vue du classifieur mais il est possible que ces régions soient hors-distribution du point de vue du générateur. Afin de limiter cette éventualité, nous cherchons à contraindre le déplacement grâce à la fonction de coût suivante : $\mathcal{L}_{\text{rec}} = \|\mathbf{z}' - \mathbf{z}\|_2$. La fonction de coût finale est donc $\mathcal{L} = \mathcal{L}_{\text{count}} + \lambda \mathcal{L}_{rec}$. Enfin, afin de faciliter l'apprentissage, nous nous limitons à l'ajout ou au retrait d'un objet pendant l'entraînement, soit $\alpha = 1$ ou $\alpha = -1$ choisi de manière aléatoire.

3 Résultats préliminaires

Pour tester la faisabilité de notre approche, nous avons généré un version de MultiMNIST [11]. La version générée est constituée de $40K$ images de taille 128×128 comportant de 1 à 4 chiffres par image. Nous avons choisi d'utiliser StyleGAN2 [6] pour générer les données (FID : 7,14). La particularité de StyleGAN est, premièrement, de disposer d'un espace latent intermédiaire $\mathcal{W}$ avec des propriétés de démêlement, et deuxièmement d'injecter un code latent $\mathbf{w} \in \mathcal{W}$ au niveau des différentes couches de convolution du générateur. L'ensemble des codes latents des différentes couches constitue l'espace latent $\mathcal{W}$ étendu noté $\mathcal{W}+$. Pour les images générées, ces deux espaces sont équivalents car le code latent $\mathbf{w} \in \mathcal{W}$ est simplement répliqué. En revanche, il est également possible de faire du contrôle sur des images réelles. Cela nécessite d'utiliser une méthode d'inversion, autrement dit une méthode permettant de retrouver un code latent qui génère une image proche de l'image réelle. Différents travaux comme e4e [12] ont montré que les résultats d'inversion des images réelles sont meilleurs dans $\mathcal{W}+$ que dans $\mathcal{W}$. Nous avons testé notre méthode à la fois dans $\mathcal{W}$ de dimension 512 et dans $\mathcal{W}+$ de dimension $12 \times 512 = 6144$ en inversant les images avec e4e. Dans le premier cas, les exemples d'apprentissage sont les codes latents échantillonnés puis



(a) Projection UMAP [7] des codes latents colorés en fonction du nombre d'objets. À gauche : $\mathcal{W}+$. À droite : $\mathcal{W}$.

space	architecture	loss	train	test
$\mathcal{W}+$	[6144,2048,512]	MSE	100%	100%
$\mathcal{W}+$	[6144,2048,512]	CE	100%	100%
$\mathcal{W}$	[512,256]	MSE	89%	69%
$\mathcal{W}$	[512,256]	CE	91%	83%

(b) Performances des classifieurs (précision en %) entraînés dans l'espace latent.

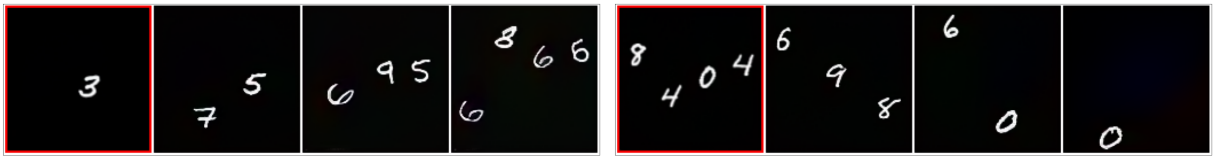


FIGURE 2 – Exemples de contrôle du nombre de chiffres avec notre approche appliqué dans $\mathcal{W}$ pour un classifieur entraîné avec une entropie croisée. L'image encadrée en rouge correspond à une image générée à partir d'un code latent échantillonné. Les images suivantes correspondent aux images obtenues par déplacements successifs d'amplitude $\alpha = \pm 1$.

labellisés grâce à des classifieurs image pré-entraînés appliqués sur les images générés correspondantes. Dans le deuxième cas, les exemples d'apprentissage sont les images réelles inversées. La table 1b présente les performances des différents classifieurs entraînés dans l'espace latent. On observe de bonnes performances dans l'ensemble. On remarque cependant que les performances du classifieur entraîné dans $\mathcal{W}$ avec la MSE sont nettement en dessous de celles du classifieur entraîné avec l'entropie croisée. Ces bonnes performances semblent indiquer une structure reflétant la cardinalité. Ajouté à cela, la figure 1a montre une projection des codes latents où on observe un regroupement en fonction du nombre de chiffres en particulier dans $\mathcal{W}+$. Malgré les bonnes performances des classifieurs dans $\mathcal{W}+$, nous ne sommes pas parvenus à apprendre des directions contrôlant la cardinalité. Notre hypothèse est que les différents clusters vivent sur des variétés trop éloignées les unes des autres. Ceci peut être dû au fait que MultiMNIST est un jeu de données simple et que la dimension de $\mathcal{W}+$ est démesurée pour ce jeu de données. En revanche, avec le modèle de classification multi-classe (entropie croisée) appliqué dans $\mathcal{W}$, nous parvenons à obtenir des directions qui contrôlent le nombre de chiffres présents dans une image comme observé dans la figure 2.

4 Conclusion et perspectives

Nous proposons une méthode permettant de contrôler un nouveau type d'attribut, la cardinalité, dans des modèles GANs pré-entraînés. Nous adaptons les formalismes existants, se concentrant sur des attributs binaires ou continus, au cas d'un attribut discret multi-classe. Nos expérimentations sur MultiMNIST montrent qu'il y a une structure propice au contrôle de la cardinalité et qu'il est possible d'identifier des directions dans l'espace latent $\mathcal{W}$ de StyleGAN avec un classifieur entraîné avec l'entropie croisée. Cependant, le jeu de données MultiMNIST est un jeu de données relativement simple où il y a peu de sémantique pouvant interférer avec la cardinalité. Nous prévoyons de tester sur des jeux de données plus complexes comme CLEVR [5] ou encore Cityscapes [2] pour confirmer ces résultats. Nous souhaitons également étendre la méthode pour pouvoir contrôler l'ajout d'un objet d'une catégorie en particulier (par exemple, les différents chiffres pour MultiMNIST) et assurer la préservation des objets initialement présents. Nous souhaitons également continuer nos expérimentations dans $\mathcal{W}+$ sur d'autres jeux de données et adapter les méthodes d'inversion existantes pour pouvoir mieux contrôler la cardinalité dans les images réelles.

Références

[1] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka. StyleFlow : Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021. ISSN 0730-0301. doi : 10.1145/3447648. URL `https://doi.org/10.1145/3447648`.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The

cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] X. Hou, X. Zhang, H. Liang, L. Shen, Z. Lai, and J. Wan. Guidedstyle : Attribute knowledge guided style manipulation for semantic face editing. *Neural Networks*, 145 :209–220, 2022.

[4] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. GANSpace : Discovering interpretable GAN controls. In *Proc. NeurIPS*, 2020.

[5] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr : A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[6] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[7] L. McInnes, J. Healy, and J. Melville. Umap : Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv :1802.03426*, 2018.

[8] A. Saseendran, K. Skubch, and M. Keuper. Multi-class multi-instance count conditioned adversarial image generation. *arXiv preprint arXiv :2103.16795*, 2021.

[9] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in GANs. In *CVPR*, 2021.

[10] Y. Shen, C. Yang, X. Tang, and B. Zhou. InterFaceGAN : Interpreting the disentangled face representation learned by GANs. *TPAMI*, 2020.

[11] S.-H. Sun. Multi-digit MNIST for few-shot learning, 2019. URL `https://github.com/shaohua0116/MultiDigitMNIST`.

[12] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or. Designing an encoder for StyleGAN image manipulation. *arXiv preprint arXiv :2102.02766*, 2021.

[13] X. Yao, A. Newson, Y. Gousseau, and P. Hellier. A latent transformer for disentangled face editing in images and videos. *2021 International Conference on Computer Vision*, 2021.

B.2 GAN Inversion on CLEVR

Fig. B.1 shows some reconstruction results for real images from the CLEVR [60] dataset obtained with the e4e method [140]. When there are few objects (cf. 1st row), the reconstruction is accurate while for images with more objects (cf. 2nd and 3rd row), some objects are not reconstructed at all.

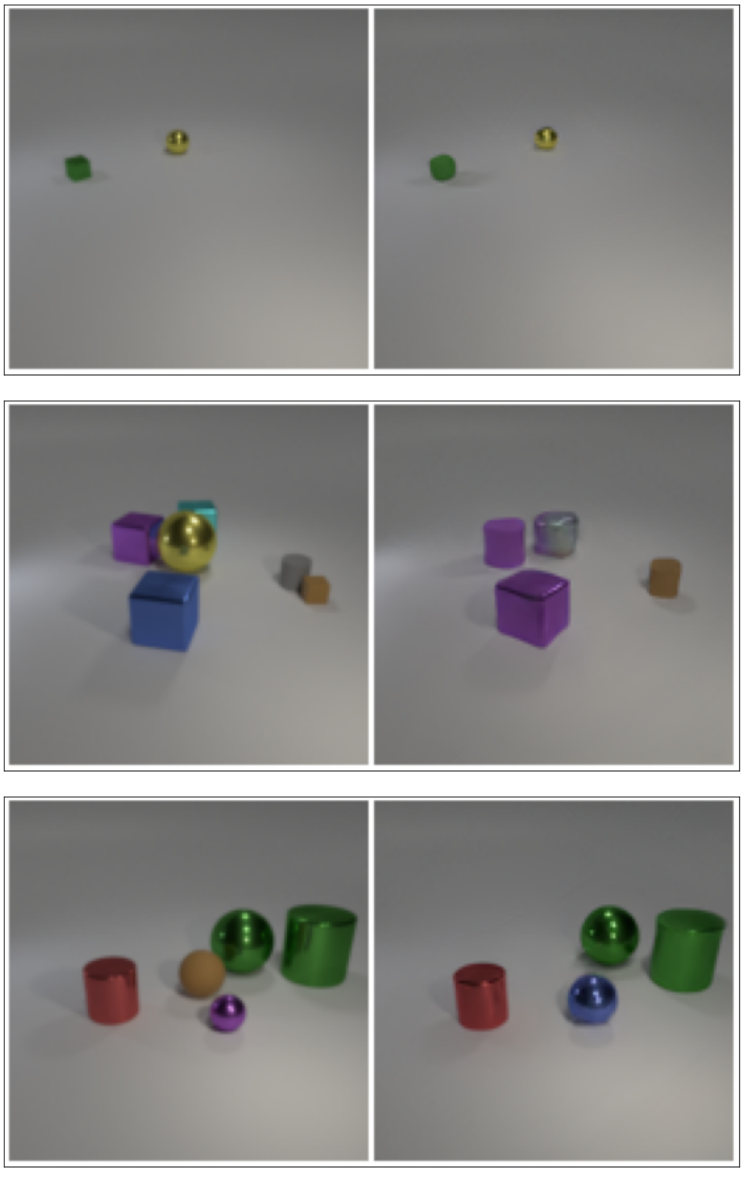


Figure B.1: Reconstruction of 128 × 128 CLEVR images with e4e [140]. Left: real image, Right: reconstructed image.

Acronyms

AdaIN Adaptive Instance Normalization. 12

cGAN conditional Generative Adversarial Network. 11, 18

CLIP Contrastive Language-Image Pretraining. 19, 20, 24

CNN Convolutional Neural Network. 11

CV Computer Vision. 3

DDIM Denoising Diffusion Implicit Model. 16, 26

DDPM Denoising Probabilistic Diffusion Model. 6, 10, 15–17, 19–21, 24, 26, 27, 64

FSC Few-shot Object Counting. 66, 67

GAN Generative Adversarial Network. 3, 4, 6, 10, 12, 15–17, 24, 26, 32, 64

GenAI Generative Artificial Intelligence. 2, 92

LDM Latent Diffusion Model. 16, 20

LLM Large Language Model. 20, 28, 90, 91

MLP Multi-Layer Perceptron. 12

PGGAN Progressive Growing GAN. 12, 18

PPL Perceptual Path Length. 13, 14

SVM Support Vector Machine. 23

VAE Variational Auto Encoder. 10