Improving Quality and Controllability in GAN-based Image Synthesis

A dissertation submitted towards the degree
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Edgar Schönfeld, M.Sc.

Saarbrücken 2022

Day of Colloquium	18th of April, 2023

Dean of the Faculty	Prof. Dr. Jürgen Steimle

Examination Committee

Chair	Prof. Dr. Eddy Ilg
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer	Prof. Dr. Gerard Pons-Moll
Academic Assistant	Dr. Thomas Leimkühler

ABSTRACT

The goal of the field of deep learning-based image generation is to synthesize images that are indistinguishable from real ones, and to precisely control the content of these images. Generative adversarial networks (GANs) have been the most popular image synthesis framework in recent years due to their unrivaled image quality. They consist of a generator and discriminator network, where the discriminator is trained to detect synthetic images, while the generator is trained to outsmart the discriminator by synthesizing more realistic images. Much progress has been made in the development of GANs, but there is still a lot of work to be done to further improve the synthesis quality and control. To this end, this work proposes methods to improve the synthesis quality of GANs and increase the control over the image content.

First, we propose the idea of segmentation-based adversarial losses to increase the quality of synthetic images. In particular, we redesign the GAN discriminator as a segmentation network that classifies image pixels as real or fake. Further, we propose a regularization made possible by the new discriminator design. The new method improves image quality in unconditional and conditional GANs.

Second, we show that segmentation-based adversarial losses are naturally well-suited for semantic image synthesis. Semantic image synthesis is the task of generating images from semantic layouts, which offers precise control over the content. We adapt the approach of a segmentation-based GAN loss to semantic image synthesis and thereby make previously used extra supervision superfluous. In addition, we introduce a noise injection method to increase the synthesis diversity significantly. The effects of the proposed techniques are improved image quality, new possibilities for global and local image editing, better modeling of long-tailed data, the ability to generate images from sparsely-annotated label maps, and a substantial increase in the multi-modality of the synthesized images. In doing so, our model is also conceptually simpler and more parameter-efficient than previous models.

Third, we show that our improvement in multi-modality in semantic image synthesis opens the door for controlling the image content via the latent space of the GAN generator. Therefore, we are the first to introduce a method for finding interpretable directions in the latent space of semantic image synthesis GANs. Consequently, we enable additional control of the image content via discovered latent controls, next to the semantic layouts.

In summary, this work advances the state of the art in image synthesis for several types of GANs, including GANs for semantic image synthesis. We also enable a new form of control over the image content for the latter.

ZUSAMMENFASSUNG

Das Ziel der Deep Learning basierenden Bildgenerierung ist es, Bilder zu synthetisieren, die nicht von echten Bildern zu unterscheiden sind und deren Inhalt genau zu steuern. Generative Adversarial Networks (GANs) waren in den letzten Jahren aufgrund ihrer hohen Bildqualität das beliebteste Framework für die Bildsynthese. GANs setzen sich aus einem Generator- und Diskriminatornetzwerk zusammen, wobei der Diskriminator darauf trainiert wird, synthetische Bilder zu erkennen, während der Generator darauf trainiert wird den Diskriminator zu überlisten indem er realistischere Bilder synthetisiert. Trotz großer Fortschritte in den letzten Jahren ist noch viel Arbeit nötig um die Qualität der Bildsynthese sowie die Kontrolle über den Bildinhalt zu verbessern. Zu diesem Zweck präsentiert diese Arbeit neue Methoden, welche die Qualität und die Kontrolle über den Inhalt von GAN-generierten Bildern verbessern.

Zunächst schlagen wir vor segmentierungsbasierte Zielfunktionen für GANs zu benutzen um die Qualität synthetischer Bilder zu verbessern. Zu diesem Zweck gestalten wir den GAN-Diskriminator als Segmentierungsnetzwerk neu das Pixel als echt oder gefälscht klassifiziert. Weiterhin schlagen wir eine Regularisierung vor die durch das neue Diskriminatordesign ermöglicht wird. Unser Verfahren verbessert die Bildqualität in Klassen-konditionierten und unkonditioniert GANs.

Zweitens zeigen wir, dass segmentierungsbasierte Zielfunktionen sehr gut für die Semantische Bildsynthese geeignet sind, welche Bilder aus semantischen Karten generiert. Wir wenden eine segmentierungsbasierten GAN-Zielfunktion fü die semantische Bildsynthese an und machen dadurch die bisher verwendete zusätzliche Überwachung überflüssig. Darüber hinaus führen wir eine Rauschinjektionsmethode ein welche die Synthesevielfalt erheblich erhöht. Unsere vorgeschlagenen Techniken ermöglichen eine verbesserte Bildqualität, globale und lokalen Bildmanipulation, eine bessere Modellierung von Long-Tail-Daten, die Fähigkeit, Bilder von spärlich annotierten semantischen Karten zu generieren, und eine wesentliche Steigerung der Multimodalität der synthetisierten Bilder. Dabei ist unser Modell auch konzeptionell einfacher und parametereffizienter als bisherige Modelle.

Drittens zeigen wir, dass unsere Verbesserung der Multimodalität in der semantischen Bildsynthese die Steuerung des Bildinhalts über die latente Repräsentation des GAN-Generators ermöglicht. Daher stellen wir als erste eine Methode vor, um interpretierbare Richtungen im latenten Raum von GANs zur Semantischer Bildsynthese zu finden. Folglich ermöglichen wir neben den semantischen Karten eine zusätzliche Kontrolle des Bildinhalts über entdeckte latente Steuerungen.

Zusammenfassend lässt sich sagen, dass diese Arbeit den Stand der Technik in der Bildsynthese für mehrere Arten von GANs voran bringt, einschließlich GANs für die semantische Bildsynthese. Letzteren ermöglichen wir auch eine neue Form der Kontrolle über den Bildinhalt.

ACKNOWLEDGMENTS

I owe sincere gratitude to Prof. Dr. Bernt Schiele and Dr. Anna Khoreva for giving me the opportunity to pursue an industrial Ph.D. at Bosch in collaboration with the Max Planck Institute for Informatics. I want to thank Bernt Schiele for always offering me help, for his valuable feedback on paper writing, and especially for listening very attentively to my project proposals and picking them apart when necessary. I also thank Prof. Dr. Gerard Pons-Moll and Dr. Thomas Leimkühler for agreeing to review my thesis.

I am most thankful to my industry supervisor Dr. Anna Khoreva. Under Anna's guidance, I learned to work with more rigor and diligence and to very critically examine my own ideas. Similarly, I am grateful for her high professional standards regarding paper writing, giving presentations, and writing reviews. I cannot thank her enough for her proactiveness in supervising, providing feedback, and proofreading things. Additionally, Anna helped me change my way of dealing with criticism and build my resilience.

Further, I want to thank Prof. Dr. Zeynep Akata for accepting me as a master's student and guiding me toward a conference paper, which was my first experience in machine learning research. I also want to thank the involved co-authors, Samarth Sinha and Dr. Sayna Ebrahimi, who made me first realize how much fun research can be, reinforcing my choice to pursue a Ph.D.

I sincerely appreciate the company of my fellow Ph.D. students: Vadim Sushko, Nadine Behrmann, Massimo Bini, Yumeng Li, Laura Iacovissi, Nikita Kister, and Haiwen Huang. I especially want to thank Massimo Bini for his fantastic company when we shared an office, Vadim Sushko for being great to work with, and Nadine Behrmann for being a good friend and source of encouragement. I am also thankful to the engineers at Bosch that I had the pleasure of cooperating with, Julio Borges, Mauricio Munoz, and Nicole Finnie. In addition, I want to thank William (Bill) Beluch for being a joy to be around, in and outside the office.

I want to thank my friends Istvan and Francis for being an immense source of encouragement. Finally, I am deeply grateful for my family and their unconditional love, support, advice, and motivation: Thomas, Simone, and my sister Paula.

Contents

1 Introduction **1**

1.1 Contributions of the thesis . . . 3

1.1.1 Unconditional and class-conditional GANs . . . 3

1.1.2 Semantic image synthesis with GANs . . . 5

1.1.3 Discovering GAN controls . . . 7

1.2 Outline of the thesis . . . 8

2 Related Work **11**

2.1 Unconditional and class-conditional GANs . . . 12

2.1.1 General working principle of GANs . . . 12

2.1.2 Improvements in GAN architectures . . . 14

2.1.3 Improvements in training . . . 19

2.2 Semantic image synthesis with GANs . . . 21

2.2.1 Generator architectures . . . 22

2.2.2 Discriminator architectures . . . 23

2.2.3 Perceptual losses . . . 24

2.2.4 Semantic image synthesis models not based on GANs . . . 24

2.3 Discovering GAN controls . . . 25

2.3.1 GAN control discovery . . . 25

2.3.2 Local editing with GANs. . . . 26

2.4 Alternatives to GANs . . . 27

3 A U-Net-Based GAN Discriminator **29**

3.1 Introduction . . . 30

3.2 U-Net GAN model . . . 32

3.2.1 The discriminator baseline. . . . 33

3.2.2 Mix and cut regularizations. . . . 33

3.2.3 U-Net based discriminator . . . 34

3.2.4 Consistency regularization . . . 35

3.2.5 Implementation . . . 38
3.3 Experiments . . . 41
3.3.1 Experimental setup . . . 41
3.3.2 Results . . . 43
3.4 Conclusion . . . 51

4 Semantic Image Synthesis with Only Adversarial Supervision 53
4.1 Introduction . . . 54
4.2 The OASIS model . . . 59
4.2.1 The SPADE baseline . . . 59
4.2.2 The OASIS discriminator . . . 61
4.2.3 The OASIS generator . . . 63
4.2.4 Superfluity of the perceptual loss for OASIS . . . 65
4.3 Experiments . . . 65
4.3.1 Experimental setup . . . 66
4.3.2 Evaluation of the synthesis quality and diversity . . . 68
4.3.3 Synthesis performance on underrepresented classes . . . 71
4.3.4 Image editing with OASIS . . . 74
4.3.5 Synthetic data augmentation . . . 76
4.3.6 Ablations . . . 77
4.4 Conclusion . . . 81

5 Discovering GAN Controls for Semantic Image Synthesis 83
5.1 Introduction . . . 84
5.2 Ctrl-SIS method . . . 86
5.2.1 GAN controls for SIS models . . . 87
5.2.2 Discovery of class-specific GAN controls . . . 88
5.3 Experiments . . . 90
5.3.1 Experimental setup . . . 90
5.3.2 Evaluation of class-specific GAN controls . . . 91
5.3.3 Main results . . . 94
5.4 Conclusion . . . 99

6 Conclusion and Future Perspectives 101
6.1 Discussion of contributions . . . 103
6.1.1 Unconditional and class-conditional image synthesis with GANs . . . 103
6.1.2 Semantic image synthesis with GANs . . . 104
6.1.3 Discovering GAN controls for semantic image synthesis . . . 106

6.2 Future perspectives 107
6.2.1 Unconditional and class-conditional image synthesis with GANs 107
6.2.2 Semantic image synthesis with GANs 110
6.2.3 Discovering GAN controls for semantic image synthesis . . . 112
6.2.4 Broader outlook 113

List of Figures **119**

List of Tables **126**

Bibliogaphy **146**

1 Introduction

Contents

1.1 Contributions of the thesis 3

1.1.1 Unconditional and class-conditional GANs 3

1.1.2 Semantic image synthesis with GANs 5

1.1.3 Discovering GAN controls 7

1.2 Outline of the thesis 8

Humans have the ability to perceive and imagine data. Perception corresponds to encoding and classifying visual, auditory, and other sensory inputs. Imagination is the decoding of ideas into images, sounds, or other types of sensory information. In an effort to imitate humans, the field of deep learning commonly models perception and imagination via neural network-based encoders and decoders. Usually, these two can also be referred to as discriminative and generative models. This thesis focuses on generative models for images.

The applications of image synthesis are manifold. Essentially, it is possible to automate or semi-automate any task that requires the creation of visual imagery. For example, image synthesis models can be a tool for anyone to create artwork or stock footage. Image synthesis models also enable inpainting, super-resolution, artistic style transfer, creating realistic textures for 3D models, or applying filters to faces, such as aging filters. Moreover, synthetic images can serve as training data for other machine learning models. Finally, an important function of an image synthesis model is to provide insight into what an encoder learned. For example, through the visualization capability of models like DALL-E (Ramesh et al., 2021) and follow-up methods, we know that deep learning models can learn the compositional representations required to convincingly combine unrelated concepts, such as an "avocado chair" or "a relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup" (Saharia et al., 2022).

This thesis focuses on generative adversarial networks (GANs), which constitute the most popular image generation paradigm in recent years due to their unparalleled visual quality. GANs consist of two neural networks: a generator and

a discriminator. The generator synthesizes images from noise and some optional conditional information specifying the content. Thereby, the conditional information specifies the image content, while the noise is responsible for the remaining variability. In effect, the generator can sample a diverse set of images. The discriminator classifies images from the dataset as real and synthesized images as fake. The discriminator feedback improves the generator until it produces fake images that the discriminator cannot distinguish from real images anymore. GANs that employ both noise and conditioning information are referred to as conditional GANs, while unconditional GANs only get noise as input.

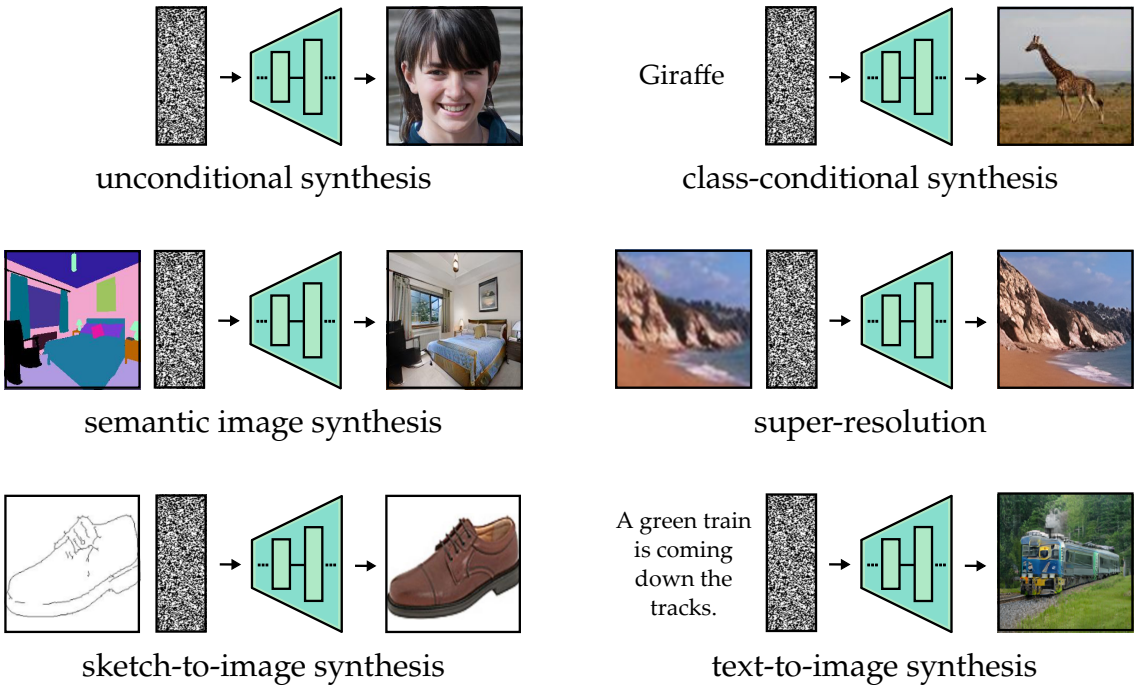

Figure 1.1: Overview of common GAN tasks, using unconditional or conditional generators (green) to generate images from noise.

GANs can be trained for many different conditional generation tasks, such as class-to-image, layout-to-image, image-to-image, or text-to-image. Depending on the task, different conditional GAN architectures are used. Figure 1.1 gives an overview of various GAN tasks. In this thesis, we focus on unconditional GANs, class-conditional GANs, and GANs for semantic image synthesis. Semantic image synthesis is the task of generating images from semantic layouts.

Designing GANs involves several challenges, many of which concern all GANs, whereas some challenges are unique for certain task-specific GANs. An example of a universal problem is mode dropping, which leads the generator to synthesize only a subset of the modes of the training distribution. Also, unstable training is a general problem for GANs, but less for semantic image synthesis GANs. However, there are problems specific to the semantic image synthesis task, such as the GAN generator being less sensitive to input noise and not achieving good image quality

when relying solely on standard GAN optimization. Apart from the challenge of designing GANs that produce diverse high-quality images, there are additional challenges concerning the inspection and evaluation of trained GANs. For example, much research has been devoted to proposing objective quality and diversity metrics (Heusel et al., 2017b; Salimans et al., 2016a; Shmelkov et al., 2018; Sajjadi et al., 2018; Kynkäänniemi et al., 2019), or tools to inspect the latent space and control the synthesis along disentangled image properties (Shen and Zhou, 2021; Härkönen et al., 2020; Peebles et al., 2020).

In this thesis, we provide innovations for improving the synthesis quality and diversity of GAN, but also for the inspection and control of GANs. In particular, we propose architectural changes and regularization for GANs in unconditional, class-conditional, and semantic image synthesis. Additionally, we focus on inspecting the latent space of semantic image synthesis GANs, proposing an algorithm to find semantically meaningful latent space directions to control the appearance of selected classes.

This thesis is structured as follows. In Chapter 2, we provide the background and related work for unconditional GANs, class-conditional GANs, GANs for semantic image synthesis, and methods that find meaningful directions in latent spaces of GAN. In Chapter 3, we introduce a new discriminator architecture and regularization to improve the synthesis quality of unconditional and class-conditional GANs. Next, in Chapter 4, we redesign the architecture of current semantic image synthesis GANs to improve image quality, diversity, and controllability. In Chapter 5, we propose the first method to allow finding latent controls in semantic image synthesis models. Lastly, in Chapter 6 we discuss our results in the context of work that has been done since the corresponding chapters were published and give an outlook on the future of the field.

In the remainder of this chapter, we first discuss the research challenges and contributions of each chapter in Section 1.1. Finally, we provide a detailed outline of this thesis in Section 1.2.

1.1 Contributions of the thesis

This thesis focuses on the three subtasks *unconditional and class-conditional GANs*, *semantic image synthesis with GANs*, and *discovering GAN controls*. In the remainder of this section, we outline the challenges for each task and our contributions to address these challenges.

1.1.1 Unconditional and class-conditional GANs

Unconditional GANs synthesize images from noise only, while class-conditional GANs synthesize images from noise and a one-hot class vector. Improving the

synthesis quality in these two basic GAN tasks is the focus of Chapter 3 and our corresponding publication "A U-Net Based Discriminator for Generative Adversarial Networks" (CVPR 2020) (Schönfeld et al., 2020).

1.1.1.1 Challenges

Training instability. Training instability has been a long-standing issue in GAN training. For example, training collapses in roughly half of all training runs of BigGAN (Brock et al., 2019), which forms the baseline for our work in Chapter 3. The source of instability lies in the sporadic occurrence of very large gradients in the discriminator (Brock et al., 2019), caused by fake batches that strongly perturb the discriminator. That is why training stability can be substantially improved through regularization that limits the gradient magnitude (Miyato et al., 2018; Mescheder et al., 2018). Training also becomes more stable through self-supervision, such as reconstruction losses or consistency regularization, which improves the robustness of the discriminator (Chen et al., 2019; Zhang et al., 2020a).

Mode dropping. Mode dropping occurs when the generator does not cover all modes of the data distribution. In this case, the generator concentrates its modeling power on a set of common appearances in the dataset. The phenomenon can be explained by the discriminator not being able to detect missing modes (Arora et al., 2017). Also, the most commonly used NS-GAN objective assigns much more cost to low image quality than mode dropping (Arjovsky and Bottou, 2017). A GAN generator should ideally be evaluated with respect to precision and recall, where precision measures the visual quality of synthetic images, while recall measures how well the generator covers the full diversity of the training distribution. Metrics of precision and recall are slowly being adopted (Sajjadi et al., 2018; Kynkäänniemi et al., 2019) but are not commonplace.

Long-range and global-local dependencies. Convolutional discriminators can struggle to take long-range interactions in images into account, since only the last layers have a receptive field large enough to connect distant parts of an image. This can become a problem for image datasets that require accurate modeling of complex structures. For example, GANs often produce images of animals with the wrong number of legs. Problems with long-range dependencies have motivated the use of self-attention modules in models like SA-GAN (Zhang et al., 2019) or BigGAN. Further, GAN discriminators also need to take global-local interactions into account. Local image structures build up the global structure, and local details also depend on global information. The convolutional encoder structure of a typical GAN discriminator may not be best suited to integrate information from all scales and positions.

1.1.1.2 Contributions

The first contribution is a U-Net based discriminator. Motivated by the idea of integrating global and local as well as long-range interdependencies, we propose to rethink the typical GAN discriminator as a segmenter. Therefore, we choose the popular U-Net segmenter network structure (Ronneberger et al., 2015). A regular discriminator is a classification network consisting of convolutional blocks with decreasing resolution. In contrast, a U-Net discriminator consists of an encoder and decoder network, connected through skip connections. The real-fake classification is computed on a per-pixel basis. In essence, the task of the discriminator is to segment an image into real and fake parts. Additionally, a scalar discriminator loss is computed from the encoder, corresponding to the regular GAN loss. The U-Net structure naturally integrates information from all scales and locations, such that every per-pixel loss also respects the bigger context. This segmentation discriminator enables a new regularization method.

The second contribution therefore is a spatial consistency regularization. Previous consistency regularizations encouraged invariance to image transformations that do not affect realness. Instead, our consistency regularization encourages equivariance by applying the image transformation also to the discriminator output. This is possible because a segmentation-based discriminator output has the same dimension as the image itself. The new regularization strongly improves synthesis quality.

The third contribution is the resulting improvement over the state-of-the-art BigGAN model, as well as providing the best-recorded performance on the CelebA dataset amongst all published image synthesis methods at the time of publishing our method.

1.1.2 Semantic image synthesis with GANs

In the semantic image synthesis (SIS) setting, the input to the GAN generator is noise and a label map. The label map is a 2D map that specifies a semantic class for each pixel of an image, and is also referred to as segmentation map in other contexts. The GAN is thus trained with pairs of real images and their corresponding label maps. Semantic image synthesis is the focus of Chapter 4 and our corresponding publication "You Only Need Adversarial Supervision for Semantic Image Synthesis" (ICLR 2021) (Schönfeld et al., 2021) and the extended work published in IJCV 2022 (Sushko et al., 2022). Note that GANs for semantic image synthesis suffer from the same challenges as unconditional and class-conditional GANs, such as training instability, mode dropping, and modeling long-range and global-local dependencies. However, SIS GANs suffer from some additional challenges, which are outlined in the following.

1.1.2.1 Challenges

Insensitivity to noise. In unconditional and class-conditional GANs, it is self-evident that resampling the input noise will yield many different images. In fact, training would collapse or end early with major mode collapse if the generator stopped reacting to the noise. On the contrary, semantic image synthesis models have a major problem with noise. In fact, early models like Pix2Pix did not react to noise at all. Training these networks was only possible because label maps act as a source of variation in the input, though not as well as a random number generator. In addition, multiple discriminators at different scales were necessary to avoid collapse. A perceptual additional perceptual loss, first introduced by Chen and Koltun (2017), was crucial to achieving good image quality. As we show in Chapter 4, this perceptual loss is one of several factors reducing the sensitivity to noise.

Necessity of a perceptual loss. A perceptual loss employs a pretrained classifier network to extract deep features from real and fake images. The perceptual loss is defined as the distance between these features and is used to train the generator. As shown in Chapter 4, previous models depend on this loss. Without it, performance is strongly reduced. Unfortunately, this loss suppresses the synthesis diversity and occupies GPU memory. The fact that previous SIS GANs cannot be trained like regular GANs, employing nothing more than a single generator and discriminator network, hints that something is very suboptimal in the training of SIS GANs.

Imbalanced datasets. SIS GANs are trained with semantic segmentation datasets. Class imbalance has been a long-standing problem in semantic segmentation, where models perform best for large or frequently occurring classes at the expense of small or rare classes. Semantic image synthesis is plagued by the same problem, leading to lower diversity and lower quality textures for small and rare classes. In Chapter 4, we give a detailed analysis of performance across semantic classes, taking size and occurrence frequency into account.

1.1.2.2 Contributions

First, we propose redesigning the GAN discriminator to a segmentation-based discriminator, segmenting each pixel into one of the real classes or an additional "fake" class. In effect, we achieve much stronger supervision while using the exact same data as previous methods. With this simple change, both the need for multiple discriminators as well as for a perceptual loss vanishes. Both used to be necessary, because the traditional discriminator design is not powerful enough for the task of SIS.

Second, we introduce LabelMix regularization. LabelMix enforces consistency in the local discriminator predictions when parts of an image are swapped out. To

this end, composite images are formed by cutting and mixing real and fake images of the same label map, respecting the semantic class boundaries. We show that LabelMix improves the performance of our SIS model.

Third, we significantly improve the synthesis diversity. On the one hand, the diversity is increased by not using a perceptual loss. On the other hand, we propose a new way to feed noise into the SIS generator. In particular, the solution is to sample a 3D tensor of noise and use it to modulate the intermediate generator features. A positive side effect of the 3D structure of the noise is that it allows resampling noise for specific image regions, increasing the manipulation control over the image. For example, given a label map of a bedroom, one can resample the whole image or only the bed.

Fourth, our model outperforms the previous state-of-the-art models on the standard benchmark datasets ADE20K, COCO, and Cityscapes. Performance is improved both in terms of image quality and diversity.

Fifth, we propose to use the LVIS dataset (Gupta et al., 2019) to asses the long-tail performance of SIS models. LVIS is strongly imbalanced across more than 1000 classes. Previous SIS models have been trained on at most 184 classes, through the COCO dataset. We observe that previous work performs poorly on underrepresented classes and experiences complete mode collapse on sparsely-annotated label maps. Our model fixes these issues and outperforms previous work by a large margin.

Lastly, we propose to evaluate SIS models through synthetic data augmentation in semantic image synthesis. The measured performance depends on image quality, diversity, and label map alignment and thereby measures all aspects at once that matter in semantic image synthesis.

1.1.3 Discovering GAN controls

The latent space of unconditional and class-conditional GANs encodes diverse semantics of an image. GAN control methods can identify directions in this latent space that correspond to specific semantics. For example, interpolating the latent noise along a direction that corresponds to "zoom" would zoom the image in or out. Another direction may change the weather or the time of the day. In the following, we highlight challenges in this research area and explain why no such methods exist for semantic image synthesis. The contributions of this section correspond to Chapter 5, where we present a GAN control discovery method for semantic image synthesis. The corresponding paper "Discovering Class-Specific GAN Controls for Semantic Image Synthesis" is currently under submission.

1.1.3.1 Challenges

Restricted to unconditional GANs Previously, GAN control methods for semantic image synthesis models did not exist for two reasons: First, the SIS models were insensitive to noise, so no such directions could exist. Second, previous methods cannot easily take the label maps into account. In particular, previous methods establish a correspondence between changes in the whole image and directions in the latent noise. However, the label map has a much greater influence on the image content, while the notion of "directions" does not apply to label maps.

Lack of quantitative evaluation metrics GAN control methods are evaluated mainly by visual inspection. Therefore, it is unclear which method performs better. Further, it is unclear how general the visual results are. A found direction may work well on a few images, but not generalize to all images. For this reason, there is a need for quantitative metrics to have a meaningful comparison between different GAN control discovery methods.

1.1.3.2 Contributions

The first contribution is a method that we term Ctrl-SIS, which is the first to apply GAN control discovery to semantic image synthesis GANs. In contrast to other GAN control discovery methods, ours explicitly uses the label map. Furthermore, our method learns class-specific directions changing the appearance of labels locally, instead of directions that change the image globally. For example, Ctrl-SIS finds different appearances for the class "tree", allowing it to turn green leaves into autumn leaves or to remove them altogether.

The second contribution is the introduction of evaluation metrics for GAN control methods in semantic image synthesis. These metrics evaluate how many unique directions are found, how consistent they are, and to what extent they affect the areas not belonging to the corresponding class. With these metrics, we demonstrate that Ctrl-SIS performs better on the GAN control discovery task than previous methods that were not designed for semantic image synthesis.

1.2 Outline of the thesis

In this section, we briefly describe the content of each chapter. We also list the corresponding publications and the contributions of individual authors when necessary.

Chapter 2: Related work. In this chapter, we describe previous work on which our contributions are built, as well as parallel or later work related to our

research. Among others, we cover the three core areas of our work: generative adversarial networks, semantic image synthesis, and discovering GAN controls.

Chapter 3: A U-Net-based GAN discriminator. This chapter presents a new discriminator architecture for GANs: instead of the typical classification-based discriminator, a discriminator that acts as a segmentation network is proposed. In addition, we propose a new consistency regularization technique that makes use of the new segmentation-based discriminator design. In effect, we improve over the state-of-the-art BigGAN model, which formed the baseline of our architectural modifications, in terms of image quality on several datasets.

The work presented in this chapter was published as the CVPR 2022 paper "A U-Net-based discriminator for generative adversarial networks" (Schönfeld et al., 2020). Edgar Schönfeld was the lead author of the paper.

Chapter 4: Semantic image synthesis with only adversarial supervision. In this chapter, we present a new model for semantic image synthesis — the task of generating a real image from a 2D map of semantic labels. Previous works struggled with two major problems: First, while these models were GAN-based, the adversarial supervision provided by the GAN loss was insufficient to generate images of good quality, making additional networks and losses necessary. Second, while semantic image synthesis is a one-to-many mapping, previous works struggled with producing diverse images for a given label map. Our work addresses these two shortcomings as follows. First, inspired by the work from chapter 3, we redesign the discriminator as a segmentation network. This alone makes the method more data-efficient and alleviates the need for additional losses and networks to achieve good image quality. In addition, the new discriminator significantly improves image quality. Further, we introduce LabelMix regularization, which makes use of the newly introduced segmentation-based discriminator and improves image quality further. Second, we introduce a 3D noise injection scheme, leading to vastly improved diversity between images generated from the same label map, strongly improving the ability of the model to perform a one-to-many mapping.

The presented work is published in the ICLR 2021 paper "You only need adversarial supervision for semantic image synthesis" (Schönfeld et al., 2021). This chapter is based on the extended version published at IJCV 2022 (Sushko et al., 2022). Edgar Schönfeld and Vadim Sushko are joint first authors and contributed equally to all aspects of the paper, including discussion, ablation experiments, final experiments, evaluations, and paper writing.

Chapter 5: Discovering GAN controls for semantic image synthesis. This chapter introduces a method for discovering semantically meaningful directions in the latent space of semantic image synthesis (SIS) models. Inspired by such latent discovery methods for conventional GANs, we present a latent discovery method for SIS GANs. Previously, SIS GANs did not have enough diversity to encode information in the latent noise. However, through our contributions in Chapter 4, the SIS model can acquire sufficient diversity to encode meaningful information in noise. We exploit this fact to develop a latent discovery method for SIS GANs, called Ctrl-SIS. Our method provides a different set of semantically meaningful latent directions for each semantic class, without direct supervision, allowing image editing not only via the label map but also the noise. For example, for the tree class, Ctrl-SIS discovers a latent direction that gives trees autumn leaves and another that removes all leaves altogether. In doing so, we are the first to present such a method.

The work in this chapter is based on the paper "Discovering Class-Specific GAN Controls for Semantic Image Synthesis", currently under submission. Edgar Schönfeld is the lead author. Julio Borges contributed by implementing the related latent discovery methods and to paper writing. Vadim Sushko contributed with training implementations of the related SIS work and paper writing.

Chapter 6: Conclusions and future perspectives. In this chapter, we discuss the results of this thesis. We put the presented work in the context of research published after the work presented in this thesis, and discuss future steps. Lastly, we give a broader outlook on the future of the field.

2 Related Work

Contents

2.1 Unconditional and class-conditional GANs 12
2.1.1 General working principle of GANs 12
2.1.2 Improvements in GAN architectures 14
2.1.3 Improvements in training 19
2.2 Semantic image synthesis with GANs 21
2.2.1 Generator architectures 22
2.2.2 Discriminator architectures 23
2.2.3 Perceptual losses 24
2.2.4 Semantic image synthesis models not based on GANs . . . 24
2.3 Discovering GAN controls 25
2.3.1 GAN control discovery 25
2.3.2 Local editing with GANs. 26
2.4 Alternatives to GANs 27

This chapter gives an overview of the related work for the methods covered in this thesis. We provide background knowledge and discuss the differences and similarities between our proposed methods and the related works.

The general theme of this thesis is image generation with generative adversarial networks (GANs). Therefore, we start by explaining the working principle of GANs in Section 2.1. In addition, we elaborate on the developments in architecture and training responsible for the incredible improvement in synthesis quality since the invention of GANs (see Fig. 2.1).

In doing so, Section 2.1 focuses on unconditional and class-conditional GANs, providing the background knowledge for Chapter 3. Next, Section 2.2 focuses specifically on GANs for semantic image synthesis, which is the main topic of Chapter 4. We highlight problems specific to semantic image synthesis and how they have been addressed in the literature. In Section 2.3 we explain methods

Figure 2.1: 4.5 years of GAN progress on face generation, as tweeted by GAN creator Ian Goodfellow (Goodfellow, 2019).

that give the user better control of the content of GAN-generated images in the context of semantic image synthesis, which is highly relevant for Chapter 5. Lastly, in Section 2.4 we introduce image synthesis models that are alternatives to GANs. Only recently, some of these methods can compete with GANs in terms of synthesis quality. Hence, we compare these methods to GANs regarding their advantages and drawbacks.

2.1 Unconditional and class-conditional GANs

In this section, we first explain the working principle of a GAN. Subsequently, we lay out the most important improvements in architecture design and training that led to the state-of-the-art performance we see today.

2.1.1 General working principle of GANs

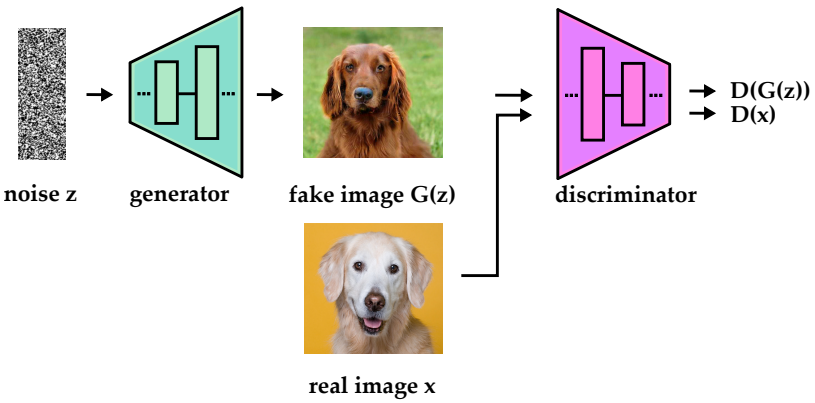

Figure 2.2: Illustration of a generative adversarial network (GAN)

A GAN consists of two separate networks: a generator and a discriminator (see Fig. 2.2). In its most basic form, the generator synthesizes an image from random

noise. At the same time, the discriminator is trained as a classifier to categorize real and synthetic images in the respective "real" or "fake" class. The generator and discriminator are trained alternatingly with one parameter update step each. First, the discriminator is trained on real images and images synthesized by the generator. Second, the generator uses the frozen discriminator as a loss function to improve the realism of the synthetic images. In other words, the generator's goal is to convince the discriminator that the images are real, while the discriminator tries not to be fooled. The solution to this game is known as Nash equilibrium, a stable situation in which neither player G nor D can improve. A GAN is said to have converged if it reaches a local Nash equilibrium. The objective V of this game is expressed as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\Big[\log D(x)\Big] + \mathbb{E}_{z\sim p_z(z)}\Big[\log(1-D(G(z)))\Big] \tag{2.1}$$

Here, G is the generator and D the discriminator network. $D(x)$ represents the probability that x is from the real data distribution. This objective can be split into a discriminator loss $\mathcal{L}_D$ and generator loss $\mathcal{L}_G$ that are minimized alternatingly:

$$\min_D \mathcal{L}_D = -\mathbb{E}_{x\sim p_{data}(x)}\Big[\log D(x)\Big] - \mathbb{E}_{z\sim p_z(z)}\Big[\log(1-D(G(z)))\Big] \tag{2.2}$$

$$\min_G \mathcal{L}_G = \mathbb{E}_{z\sim p_z(z)}\Big[\log(1-D(G(z)))\Big]. \tag{2.3}$$

The loss formulation above is referred to as minimax GAN loss (M-GAN). However, the problem with Eq. 2.3 is that the gradients resulting from $\log(1 - D(G(z)))$ will vanish if the discriminator can confidently distinguish real and fake images. For this reason, $\log D(G(z))$ is maximized instead:

$$\min_G \mathcal{L}_G = -\mathbb{E}_{z\sim p_z(z)}\Big[\log D(G(z))\Big]. \tag{2.4}$$

The standard GAN implementation thus employs the non-saturating (NS-GAN) loss of Eq. 2.4. Originally, for every optimization step of G, k optimization steps of D were performed for D to stay close to its optimal solution (Goodfellow et al., 2014). This approach was replaced by the two time-scale update rule (Heusel et al., 2017b), setting $k = 1$ and instead using a higher learning rate for D. Heusel et al. (2017b) showed that the discriminator converges to a local Nash equilibrium when G and D have separate learning rates, since the generator updates are small enough for the discriminator to react. The training procedure is summarized in Algorithm 1.

Based on this simple setup, many improvements have been proposed, affecting the loss, regularization, and model architecture. To show how current GANs achieve their excellent synthesis ability, we review the most important changes in the following.

Algorithm 1 Training a GAN with minibatch gradient descent.

Input: D: Discriminator network, G: Generator network, N: Batch size,

1 **for** *number of training iterations* **do**

2 · Sample minibatch of noise $\{z_1, ..., z_N\}$

3 · Sample minibatch of data $\{x_1, ..., x_N\}$

4 · Update D with loss

$$-\frac{1}{N}\sum_{n=1}^{N}\Big[\log D(x_n) + \log(1 - D(G(z_n)))\Big]$$

5 · Update G with loss

$$-\frac{1}{N}\sum_{n=1}^{N}\Big[\log D(G(z_n))\Big]$$

6 **end for**

2.1.2 Improvements in GAN architectures

The original GAN paper provided experiments with simple MLP and CNN architectures. These architectures were sufficient to produce gray-scale images from simple datasets of digits or faces (see the leftmost image in Fig. 2.1) but already insufficient for the relatively simple CIFAR-10 dataset with ten object categories. Since the publication of the original GAN, many changes to the discriminator and generator architecture have been proposed until most models converged to a similar setup. In the following, we first outline the anatomy of a typical GAN. We then explain improvements in the basic up- and down-scaling blocks from which GAN architectures are assembled. Next, we discuss improvements in using conditional information in the generator and discriminator networks. Lastly, we describe recently introduced GAN architectures that make use of transformers (Vaswani et al., 2017).

Anatomy of a typical GAN architecture. Both generator and discriminator consist of ResNet blocks. These blocks typically contain two internal convolutional layers with a $3x3$ kernel size and stride 1. The input to a block is added to the output via a residual connection. The output features of each block are modulated via a conditional normalization method, for example, conditional batch normalization. The conditioning information could be a one-hot class vector, a noise vector, or both. It may differ for every generator ResNet block to model image characteristics at different scales separately. No such conditioning takes place in the discriminator. The convolutional blocks do not change the spatial resolution of features, e.g., via strided convolutions. Instead, features are up- or downsampled via interpolation only. In the following, we explain the reasoning behind these basic choices and

which solutions exist to improve them.

Up- and downscaling in GAN blocks. The main difference between generator and discriminator blocks is that the intermediate features are either upscaled or downscaled. The exact way features are up- or downsampled in convolutional architectures has a perhaps surprisingly strong influence on the quality of the synthetic images. For this reason, previous literature designed solutions to problems such as checkerboard artifacts, training stability, and aliasing.

Checkerboard artifacts are periodic fluctuations in color or brightness between pixels (see Fig. 2.3). They occur when upsampling in the generator is implemented using transposed convolutions, especially if the kernel size is not divisible by the stride and therefore produces overlapping patches (Odena et al., 2016). Strided convolutions can also cause checkerboard artifacts in the discriminator. The solution is not to use deconvolution or strided convolutions but to upsample features via interpolation and downsample via average pooling (Odena et al., 2016). This solution is referred to as resize-convolution. The proposed GANs in Chapters 3 and 4 use resize-convolution, in line with the baselines on which we built our models.

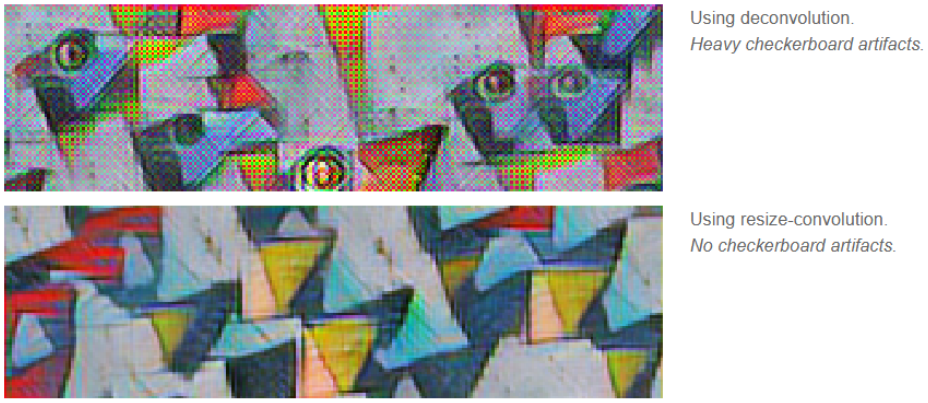

Figure 2.3: An example of heavy checkerboard artifacts[1].

Unfortunately, the higher the resolution, the less stable training becomes since the high resolution makes it easier for the discriminator to distinguish real and fake images. One way to stabilize training is to progressively grow the generator and discriminator by slowly appending higher-resolution ResNet blocks (Karras et al., 2018; Sauer et al., 2022). Progressive growing also speeds up training, since the lower-resolution images are easier to learn and less GPU memory is required. Alternatively, direct skip connections from the generator to discriminator blocks of the same resolution can be established, or variants thereof (Karnewar and Wang, 2020; Karras et al., 2020b). In this case, the generator learns intermediate smaller versions of the image. While recognizable images form early during training in low-resolution layers, they only form in higher-resolution layers later in training.

[1]Figure from Odena et al. (2016) licensed under CC BY 4.0

For comparison with previous work, we neither make use of progressive growing nor skip connections between generator and discriminator in Chapters 3 and 4.

Lastly, correct upsampling is crucial to avoid aliasing and its far-reaching side effects. The fact that GANs process images as *discrete* signals, i.e., pixel values are sampled as a regular 2D grid, causes aliasing. Aliasing causes unintended side effects, such as a lack of rotation equivariance and an imperfect translation equivariance that manifests itself in the phenomenon of texture-sticking (Karras et al., 2021b). Texture-sticking refers to the problem that textures appear to stick to the computer screen when you translate an image. For example, if a generated face is moved around a little by stepping through the latent space, some hairs will stay at their exact pixel coordinates [2]. To define what we mean by equivariance more formally, let f be a network layer and t be a transformation, such as rotation or translation. Equivariance implies that the application of f and t is commutative, i.e., $f \circ t = t \circ f$. This property must hold for all layers f, meaning convolutions, nonlinearities, and upsampling. However, treating the image as a *discrete* signal introduces aliasing, which destroys the commutative property. The solution proposed by Karras et al. (2021b) to achieving this equivariance lies in treating the image approximately as a *continuous* signal and redesigning convolution layers, nonlinearities, and upsampling accordingly.

For example, if we assume that f is a ReLU nonlinearity and t is translation, then $f \circ t = t \circ f$ does not hold in the discrete domain if t translates the image by half a pixel. However, we can approximate the continuous signal by upsampling the image before applying f. In practice, Karras et al. (2021b) find that upsampling by a factor of 2 is sufficient. Consequently, when the default upsampling layer and nonlinearity are fused, a feature map is temporarily upsampled four times.

Thereby, the choice of the upsampling filter itself is essential as well. For example, naive nearest neighbor upsampling creates a faint but unmistakable afterimage, since every pixel at NxN resolution results in a $2x2$ megapixel at $2Nx2N$ resolution. This compromises the commutative relationship between upsampling and small translations, e.g., at subpixel scale. Such forms of aliasing can be suppressed by using a more sophisticated resizing filter, such as a windowed sinc filter with a large Kaiser window (Karras et al., 2021b).

Taken together, these adjustments have been shown to make the generator translation equivariant. Karras et al. (2021b) also proposes to achieve rotation equivariance in the generator via a radially symmetric convolutional kernel, which is satisfied by using only $1x1$ convolutions. For downsampling, average pooling is replaced by a radially symmetric filter. Now that the model allows rotation and translation equivariance, the first input to the generator should be Fourier features that define a spatial map. The alignment of this map is controlled via rotation and translation parameters of a trainable affine transformation layer, which allow to

[2] An example video of texture sticking can be seen here https://tinyurl.com/2p8pda7f

change translation and rotation separately. The design of anti-aliasing GAN blocks was only proposed after our work in Chapters 3 and 4, and their application will likely become standard practice.

In conclusion, a lot of effort has gone into up- and down-sampling GAN blocks to enable highly realistic image synthesis, free of artifacts stemming from the block design. Unlike previous works, we propose discriminator architectures with both downscaling and upscaling blocks in Chapters 3 and 4. In doing so, we show how generator blocks can be repurposed in the discriminator to achieve a segmentation network design. These segmentation-based discriminators improve synthesis performance and allow new regularization methods. We next discuss the incorporation of conditioning information into GAN blocks.

Conditional generators. In the following, we describe design choices in GAN generators to enable conditional image generation. GAN generators take noise and optionally additional information as input. The additional input is typically a one-hot encoding of a class. In unconditional image synthesis, either an identical noise vector or separate layer-specific noise vectors are given to each generator block (Denton et al., 2015; Brock et al., 2018) to allow scale-specific control of the image. For class-conditional image synthesis, one-hot vectors are typically concatenated to the noise (Brock et al., 2018; Karras et al., 2020a). The concatenated vector is fed into each generator block. This setup is also used throughout our experiments on unconditional and class-conditional GANs in Chapter 3. More recently, improved image quality was achieved by replacing the one-hot embedding with a learned class embedding (Sauer et al., 2022). These embeddings were generated by averaging EfficientNet (Tan and Le, 2019) features of all ImageNet images belonging to the same class. Orthogonal to these techniques, feeding the conditioning vector through an MLP before inserting it into the generator layers helps to disentangle the latent space (Karras et al., 2019b, 2020c,a).

Different forms of conditional normalization are the most widespread techniques for injecting noise and additional conditioning information in the generator blocks. In essence, the generator blocks incorporate the injected information by using it to modulate the outputs of the convolutional layers. For example, the conditioning vector can be used to control the scale and shift of a conditional batch normalization layer (Miyato and Koyama, 2018; Miyato et al., 2018; Brock et al., 2018). Analogously, StyleGAN (Karras et al., 2019b) modulates generator features with adaptive instance normalization (AdaIN) with scale and shift learned from the conditioning vector. In StyleGAN-2 (Karras et al., 2020c), AdaIN is replaced by a "modulated convolution", in which the convolutional kernel weights are scaled directly via the conditioning vector. Additionally, StyleGAN-XL (Sauer et al., 2022) proposes to improve class-conditional image synthesis with an external classifier loss. For comparison with previous work, we employ conditional batch normalization for our experiments in Chapter 3.

Inspired by these principles, in Chapter 4 we also propose a conditional normalization technique for the specific task of generating images from label maps, known as semantic image synthesis, greatly enhancing synthesis diversity. In stark contrast to unconditional and class-conditional GANs, previous work in semantic image synthesis suffered from low sensitivity to noise and did not manage to make layer-wise noise injection work. We remove barriers that hampered noise sensitivity and propose to modulate features locally using a 3D noise tensor (i.e., one channel dimension and two spatial dimensions) via a specific form of conditional batch normalization. In effect, we create an effective noise conditioning scheme for semantic image synthesis GANs, greatly enhancing image diversity.

Conditional discriminators. The following describes how GAN discriminators are designed to enable conditional image generation. While conditioning information such as class embeddings is fed to every generator layer, discriminators usually incorporate the class information in the loss alone. The first conditional GAN variants (Mirza and Osindero, 2014) still left the loss untouched. Instead, they simply concatenated the conditioning vector to the discriminator input or an intermediate feature (Reed et al., 2016). However, it is not guaranteed that the discriminator makes use of the conditioning information since it can simply ignore parts of its input. For this reason, it is more effective to use a conditional loss instead. To this end, Odena et al. (2017) proposed to perform class-conditional synthesis by adding a classification head to the final discriminator layer. The currently most widely used conditional loss is the projection loss proposed by Miyato and Koyama (2018), which we also use in Chapter 3. The projection loss allows incorporating any conditioning information in the form of an embedding, i.e., not just categorical information. It computes the inner product between the penultimate layer and the conditional embedding. The inner product is added to the final discriminator output before computing the real-fake classification loss. The projection loss significantly increased the quality of class-conditional image generation with 1000 ImageNet classes (Miyato and Koyama, 2018). Yet, specific GAN architectures still struggled with large-scale class-conditional generation (Sauer et al., 2022). In particular, Sauer et al. (2022) observe that the intra-class diversity can be greatly improved by initializing the class embeddings used for the projection loss with pretrained class embeddings.

In Chapter 4, we propose a conditional discriminator loss in the semantic image synthesis setting. While previously, there was no known way of computing a conditional loss in semantic image synthesis GANs, we enable it by redesigning the discriminator architecture.

Transformer-based GANs. Following the large success of transformers in language modeling (Vaswani et al., 2017; Brown et al., 2020), transformers have been successfully applied to computer vision tasks such as classification (Dosovitskiy et al., 2020), object detection (Carion et al., 2020), or semantic segmentation (Cheng

et al., 2022). Next to these discriminative tasks, transformers have also recently been incorporated into GANs where they can help improve learning long-range dependencies. ViT-GAN (Lee et al., 2021), TransGAN (Jiang et al., 2021), and HiT (Zhao et al., 2021) are transformer-based convolution-free architectures. In these architectures, images are modeled as a sequence of flattened image patches. StyleFormer (Park and Kim, 2022) remodels the StyleGAN2 (Karras et al., 2020b) architecture with transformers. In contrast to purely transformer-based GANs, GANformer 1 and 2 (Hudson and Zitnick, 2021; Arad Hudson and Zitnick, 2021) combine transformer blocks with convolution. In this case, the image is not modeled as a sequence of patches. Instead, the attention mechanism of the transformer is used between latent codes and features in the generator, and also between learned embeddings and features in the discriminator. The use of transformers in GANs is a very recent development and therefore not part of the architectures proposed in this thesis.

2.1.3 Improvements in training

Next to improving GAN architectures, a lot of effort has gone into improving the training procedure. Research on improving GAN training has mainly focused on better objective functions and regularization techniques. In the following, we give an overview of several proposed GAN losses and explain why the original GAN loss is still the most popular objective today. Next, we give a brief overview of the two most effective forms of regularization: Lipschitz regularization and self-supervision.

GAN objectives. The original GAN paper proposed a minimax loss (M-GAN) and a non-saturating loss (NS-GAN). While subsequent literature frequently refers to the original GAN as "standard GAN", we use the more precise M-GAN and NS-GAN in this thesis. The NS-GAN loss alleviates the vanishing-gradient problem of the M-GAN loss and is therefore used by default. Another popular replacement for the M-GAN loss is the Wasserstein GAN (WGAN) loss (Arjovsky et al., 2017). WGAN minimizes the Wasserstein distance between real and synthetic data. In contrast to the Jensen-Shannon Divergence minimized by M-GAN, the Wasserstein distance yields strong and useful gradients even when the real and synthetic data have no meaningful overlap, e.g., at the beginning of training. To estimate the Wasserstein distance, it is necessary to enforce a 1-Lipschitz constraint on the discriminator, meaning that no gradient can be greater than one. Originally, this was achieved by gradient clipping (Arjovsky et al., 2017) and later by a gradient penalty (Gulrajani et al., 2017b) (WGAN-GP). The functional form of the WGAN objective falls into the category of integral probability metrics (IPMs) (Müller, 1997). Other examples of IPM GAN formulations include MMD-GAN (Li et al., 2017), Fisher-GAN (Mroueh and Sercu, 2017), McGAN (Mroueh et al., 2017), Sobolev-GAN (Mroueh et al., 2018), and more. On the other hand, the NS-GAN objective falls under the family of so-

called f-divergences (Nowozin et al., 2016b). Other GANs minimizing f-divergences are f-GAN (Nowozin et al., 2016b), LS-GAN (Mao et al., 2017), EBGAN (Zhao et al., 2017), and more. Today, the original NS-GAN loss is still one of the most-used GAN objectives, for example, in the StyleGAN series of state-of-the-art models (Karras et al., 2019b, 2020b, 2021b; Sauer et al., 2022). Interestingly, the question of which loss function performs best cannot be clearly answered (Mescheder et al., 2018; Lučić et al., 2018; Shannon et al., 2020; Mallasto et al., 2019; Fedus et al., 2018), but it is evident that the original NS-GAN loss compares very favorable to other losses both in theory (Shannon et al., 2020; Fedus et al., 2018) and practice (Lučić et al., 2018; Kurach et al., 2019). In particular, "A Large-Scale Study on Regularization and Normalization in GANs" by Kurach et al. (2019) concludes on an empirical basis that one should use the original NS-GAN loss as the default choice. The GAN models we propose in Chapter 3 and Chapter 4 use the NS formulation as well. Compared to the choice of the loss function, regularization techniques have a substantial impact on performance (Qin et al., 2020; Mescheder et al., 2018) and are discussed next.

Lipschitz regularization. Lipschitz regularization ensures that discriminator gradients do not exceed a certain magnitude. Although Lipschitz regularization is necessary to minimize the Wasserstein distance in WGAN and WGAN-GP, it has proven helpful for all GAN training schemes, regardless if they are necessary from a divergence minimization point of view (Fedus et al., 2018). In fact, Lipschitz regularization leads to convergence for loss functions that otherwise would not converge (Qin et al., 2020; Mescheder et al., 2018). Particularly, Qin et al. (2020) show that with sufficient Lipschitz regularization, even non-standard loss functions, such as a seemingly arbitrary cosine-based GAN loss, yield comparable results to other loss functions. Lipschitz regularization can be implemented via gradient clipping (Arjovsky et al., 2017), gradient penalties (Gulrajani et al., 2017a; Mescheder et al., 2018) or by rescaling the weights of the network (Miyato et al., 2018). The most widely used gradient penalty is R1 regularization (Mescheder et al., 2018). For example, a large scale study by Kurach et al. (2019) recommends the use of R1 regularization in combination with the NS-loss as default, which is also done in the StyleGAN series (Karras et al., 2019b, 2020b, 2021b; Sauer et al., 2022). On the other hand, spectral normalization (Miyato et al., 2018) enforces the Lipschitz constraint by rescaling the network weights, and is for example used in the state-of-the-art models SA-GAN (Zhang et al., 2019) and BigGAN (Brock et al., 2018), as well as the models we use in Chapter 3, 4 and 5.

Self-supervised discriminators. Next to Lipschitz regularization, GAN discriminators can also greatly benefit from self-supervision. Self-supervision techniques for GANs fall into two main categories: First, the discriminator has to solve an auxiliary prediction task via an additional head. Second, the discriminator is trained with consistency regularization to be invariant to image changes that do not affect

realism.

A well-known self-supervised task is training the discriminator with an auxiliary rotation loss (Chen et al., 2019). Real input images are rotated by one of the four angles $\{0°, 90°, 180°, 270°\}$ and the discriminator has to minimize the cross-entropy between the real rotation index and the index predicted by an additional classification head. Interestingly, Chen et al. (2019) find that additional self-supervision not only improves image quality but also strongly reduces the sensitivity of GAN training to the choice of hyperparameters, such as the betas in the Adam optimizer. The regularizing effect can be explained by the fact that self-supervision reduces shortcut learning when discriminating between real and fake images, leading to improved training stability and image quality. Hence, self-supervision and Lipschitz regularization both reduce extreme or abrupt changes in D's parameters. In addition, self-supervision leads to more general and robust features. Another self-supervision task is image-reconstruction: FastGAN (Liu et al., 2021) treats the discriminator as an image encoder and reconstructs the original image from discriminator features with additional decoders. This self-supervision strongly improves image quality, especially in few-shot training scenarios, where the GAN can only learn from approximately 100 images.

The second form of self-supervision used in GANs is consistency regularization (CR). As proposed by Zhang et al. (2020a), the idea is to make the discriminator prediction invariant to transformations T that do not change the realism of an image, such as horizontal flips or translating the image by a few pixels. For this, the discriminator is trained with the additional loss $\|D(x) - D(T(X))\|^2$. Note that this method does not require additional prediction heads. Zhang et al. (2020a) also propose an improved consistency regularization method (ICR), which additionally applies CR to synthetic images and creates extra transformations by perturbing the latent space.

In Chapters 3 and 4, we propose a consistency regularization that encourages equivariance instead of invariance to transformations that do not alter the real-fake classification.

2.2 Semantic image synthesis with GANs

Semantic Image Synthesis (SIS) is the task of synthesizing a realistic-looking image from a label map. A label map is a 2D map that specifies a class index for each pixel in an image. Generating images from label maps requires a specialized GAN architecture that makes efficient use of the provided label maps. Ideally, SIS models should yield the same high image quality and diversity as unconditional GANs. In reality, SIS models lag behind unconditional GANs in both aspects. To understand why, we discuss several generator and discriminator architectures designed for using label maps, and highlight the problems of these models (Sec. 2.2.1&2.2.2).

Importantly, all previous works also make use of a third network, known as the perceptual loss, which we discuss after the generator and discriminator architectures (Sec. 2.2.3). Lastly, we give a brief overview of SIS models not based on GANs (Sec. 2.2.4), some of which are very recent and have caught up with GANs in terms of synthesis quality.

2.2.1 Generator architectures

To enforce the alignment between the generated images and the conditioning label maps, previous methods explored different ways to incorporate the label maps into generator training. In many conventional approaches (Isola et al., 2017; Wang et al., 2018a; Tang et al., 2020c,b; Ntavelis et al., 2020; Richardson et al., 2021), label maps are provided to the generator via an additional encoder network. For example, the Pix2Pix and Pix2PixHD (Isola et al., 2017; Wang et al., 2018a) generator network is a U-Net (Ronneberger et al., 2015), which takes the label map as input and produces an image as output. However, this solution has been shown to be suboptimal at preserving the semantic information until the later stages of image generation. For this reason, SPADE (Park et al., 2019b) introduced a spatially-adaptive denormalization layer (SPADE layer) that directly modulates the label map onto the generator's hidden layer outputs at various scales. In essence, the SPADE layer is a conditional batch normalization layer, except that scale and shift depend on the label map and are therefore learned per pixel. Since the generator is very responsive to the conditional normalization, the SPADE layer makes the U-Net encoder superfluous. However, SPADE still uses an encoder network to predict a latent style vector that is fed as input to the generator. Alternatively, CC-FPSE (Liu et al., 2019) proposed to use spatially-varying convolution kernels conditioned on the label map, which they term conditional convolutions. In this case, an encoder is used to extract a feature pyramid from the label map. From these features, the convolutional kernel weights in the generator are predicted directly. Most recently, SC-GAN (Wang et al., 2021c) utilized label maps as input to generate class-specific semantic vectors at different scales, which are used as conditioning at different layers of the image rendering network. The conditioning is implemented via conditional normalization and a specific form of conditional convolutions. Lastly, CollageGAN (Li et al., 2021b) proposed to extract a label map representation via a feature pyramid encoder and inject it as a spatial style tensor into a StyleGAN2 generator. After generating an initial image, the image is refined with additional class-specific generators.

Our OASIS model proposed in Chapter 4 is based on SPADE and takes over the conditional batch normalization technique to incorporate label information. We show that the main bottleneck for the performance of previous models lies in the discriminator. Hence, despite employing simple conditional batch normalization we outperform CC-FPSE and SC-GAN. SC-GAN constitutes concurrent

work to ours, and Wang et al. (2021c) shows that SC-GAN's performance improves when its discriminator is replaced by our OASIS discriminator. CollageGAN was published after our work and directly compares to OASIS. The comparison is not straightforward, since CollageGAN employs a different baseline for the generator network. However, it is clear from Li et al. (2021b) that the quality improvement comes from the class-specific generators. Unfortunately, such an approach cannot scale to a setting with many classes. In contrast, we use the LVIS dataset with more than 1000 classes to evaluate our model in Chapter 4. Lastly, the comparison to our model in Li et al. (2021b) is limited to only a few datasets and misses important metrics, such as the so-called mean intersection over union.

While improving the quality of generated images, the works published before OASIS (see Chapter 4) struggled to achieve multi-modality through sampling the input noise, as the generator tended to become insensitive to noise or achieved only poor quality, as first observed by (Isola et al., 2017). For example, Pix2Pix (Isola et al., 2017) is completely insensitive to noise and can only achieve negligible diversity via dropout. Thus, the aforementioned approaches resorted to having an image encoder in the generator design to enable multi-modal synthesis. The generator then combines the extracted image style with the label map to reconstruct the original image. One can generate multiple outputs conditioned on the same label map by alternating the style vector. However, using an image encoder is a resource-demanding solution. In Chapter 4, we enable multi-modal synthesis directly through sampling of a 3D noise tensor which is injected at every layer of the network. Different from the structured noise injection of Alharbi and Wonka (2020) and class-specific latent codes of Zhu et al. (2020b), we inject the 3D noise along with label maps and adjust it to image resolution, also enabling resampling of selected semantic segments (see Fig. 4.2).

2.2.2 Discriminator architectures

To provide a powerful guiding signal to the generator, a GAN discriminator for semantic image synthesis should evaluate both the image realism and its alignment to the provided semantic label map. Thus, a fundamental question is to find the most efficient way for the discriminator to utilize the given semantic label maps. To this end, Pix2pix (Isola et al., 2017), Pix2pixHD (Wang et al., 2018a), and SPADE (Park et al., 2019b) rely on concatenating the label maps directly to the input image, which is fed to multiple PatchGAN discriminators at different scales. For this, the image is downscaled 2 and 4 times. As shown in Chapter 4, training becomes unstable when only a single discriminator is used. Alternatively, SESAME (Ntavelis et al., 2020) employed a projection-based discriminator (Miyato and Koyama, 2018), applying an additional branch to process semantic label maps separately from images, and merging the two streams before the last convolutional layer via a pixel-wise multiplication. CC-FPSE (Liu et al., 2019) proposed a feature-

pyramid discriminator, embedding both images and label maps into a joint feature map, and then consecutively upsampling it in order to classify it as real/fake at multiple scales. LGGAN (Tang et al., 2020c) introduced a classification-based feature learning module to learn more discriminative and class-specific features.

In Chapter 4, we propose to use a simple pixel-wise semantic segmentation network as a discriminator instead of multi-scale image classifiers as in the above approaches, and to directly exploit the semantic label maps for its supervision. Segmentation-based discriminators have been shown to improve semantic segmentation (Souly et al., 2017), but have not been explored for semantic image synthesis. Our work is the first to apply an adversarial semantic segmentation loss for this task. The segmentation-based discriminator has since been applied in several subsequent works (Wang et al., 2021c; Lv et al., 2022; Jain et al., 2022; Jeong et al., 2021; Musat et al., 2022; Hao et al., 2021).

2.2.3 Perceptual losses

Gatys et al. (2015), Gatys et al. (2016), Johnson et al. (2016), and Bruna et al. (2016) were pioneers at exploiting perceptual losses to produce high-quality images for super-resolution and style transfer using convolutional networks. Such a loss extracts deep features from real and generated images by an external classification network, and minimizes their L1-distance to bring fake images closer to the real data. For semantic image synthesis, the VGG-based perceptual loss was first introduced by CRN (Chen and Koltun, 2017), and later adopted by Pix2pixHD (Isola et al., 2017). Since then, it has become a default for training the generator (Park et al., 2019b; Liu et al., 2019; Tan et al., 2020; Tang et al., 2020a; Richardson et al., 2021; Wang et al., 2021c; Li et al., 2021b). As the perceptual loss is based on a VGG network pretrained on ImageNet (Deng et al., 2009), methods relying on it are constrained by the ImageNet domain and the representational power of VGG. With the recent progress in GAN training, e.g., by architecture designs and regularization techniques, the actual necessity of the perceptual loss requires a reassessment. In Chapter 4, we experimentally show that such loss imposes unnecessary constraints on the generator, significantly limiting the diversity among samples. We find that without the VGG loss much higher diversity can be achieved in semantic image synthesis, without compromising synthesis quality.

2.2.4 Semantic image synthesis models not based on GANs

Concurrently to the first GAN-based SIS models (Isola et al., 2017), SIS models were proposed that are only trained using the VGG perceptual loss. As such, the cascaded refinement network (CRN) (Chen and Koltun, 2017) trains a generator only via the VGG perceptual loss and a diversity term, but achieves similar quality compared to Pix2Pix. This result yet again highlights the strong dependence of

previous GAN-based models on the perceptual loss. Similarly, SIMS (Qi et al., 2018) combines this approach with a memory bank of real image segments to achieve even higher image quality. Recently, diffusion models emerged as a viable alternative to GANs for unconditional image synthesis (Jolicoeur-Martineau et al., 2020; Song and Ermon, 2020; Ho et al., 2020). SDM applies diffusion models to semantic image synthesis (Wang et al., 2022b), performing similarly to the model we propose in Chapter 4. Moreover, (Wang et al., 2022a) demonstrate a strongly improved performance in one key performance metric, the so-called FID score (Heusel et al., 2017a), by using a *pretrained* diffusion model. Compared to our method (Chapter 4), these more recent diffusion-based works perform similarly or better in terms of FID, but lag behind in terms of the label-map alignment measured by the mIoU score. The two mentioned diffusion-based works directly compare to our method (Chapter 4) in their evaluation.

2.3 Discovering GAN controls

It has been shown that the latent space of GANs frequently exhibits semantically relevant vector space arithmetic (Bau et al., 2019; Goetschalckx et al., 2019; Jahanian et al., 2020; Voynov and Babenko, 2020; Schwettmann et al., 2021). Therefore, works on discovering GAN controls are concerned with automatically identifying directions in the latent space that only change isolated semantic aspects of an image. In this section, we discuss GAN control methods, emphasizing methods that can change images in localized regions without affecting the rest of the image. These related works are highly relevant for Chapter 5, where we present an algorithm to apply GAN control discovery to SIS models, which to the best of our knowledge, has not been done before.

2.3.1 GAN control discovery

Finding steerable directions in the latent space is highly difficult due to the large dimensionality of the latent space and the high diversity of image semantics. For this reason, some works required human supervision (Schwettmann et al., 2021), attribute predictors (Wu et al., 2021; Shen et al., 2020), or predetermined visual transformations such as zooming or rotation (Jahanian et al., 2020; Plumerault et al., 2019) in order to identify interpretable latent directions. This dependence on supervision, however, limits the usability of these methods in practice. Another line of work investigated the unsupervised discovery of GAN controls (Spingarn et al., 2020; Voynov and Babenko, 2020; Tzelepis et al., 2021; Shen and Zhou, 2021; Härkönen et al., 2020; Yüksel et al., 2021). GANSpace (Härkönen et al., 2020) proposed to perform PCA on the intermediate feature space of the generator, discovering useful controls in the latent space resulting from layerwise perturbations along the princi-

pal directions. SeFa (Shen and Zhou, 2021) identified latent controls by performing eigendecomposition of a generator's weights, extracting semantically meaningful directions in closed-form. In contrast, Yüksel et al. (2021) and Tzelepis et al. (2021) relied on gradient-based optimization. WarpedGanSpace (Tzelepis et al., 2021) used a classifier to discriminate among a fixed set of directions in the image space, while LatentCLR (Yüksel et al., 2021) used a contrastive loss optimizing directions to have orthogonal effects on the generator's intermediate features. A common limitation of unsupervised methods is that latent directions obtained this way are left to subjective visual inspection and manual identification of significant controls.

The above work focused mainly on finding latent directions for global image manipulation, considering only unconditional image synthesis GAN models. In contrast, in Chapter 5 we propose a GAN control discovery method for conditional SIS models, where we take advantage of the given semantic label maps to find class-specific latent directions. We pick state-of-the-art GANSpace and SeFa models for comparison with our method due to their conceptual simplicity, the ease with which they are adapted to SIS GAN models, as well as their code availability.

2.3.2 Local editing with GANs.

Recent work enabled local image editing by performing optimization in latent space on specific image regions (Wu et al., 2021; Suzuki et al., 2018; Ling et al., 2021; Pajouheshgar et al., 2021; Zhu et al., 2022, 2021). EditGAN (Ling et al., 2021) jointly modeled images and their segmentations. Users need to modify the segmentation mask, based on which the optimization is performed in the latent space to realize the edit. LELSD (Pajouheshgar et al., 2021) proposed an area loss that, given a binary mask, optimizes changes only within the specified area and minimizes changes outside. However, the found directions are still applied globally. Parallel to our work, ReSeFa (Zhu et al., 2022) proposed to optimize the change of pixel values with respect to the latent code, identifying latent variations corresponding to an image region specified by the user. The main limitation of the above work is that it requires test-time optimization, preventing the user from interactive image editing. In contrast, in Chapter 5 we present a method that is optimized end-to-end once to provide latent directions for interactive editing in the spirit of GANSpace and SeFa.

Unlike existing work, our method is the first method explicitly proposed for SIS GANs (Wang et al., 2021c; Park et al., 2019b; Schönfeld et al., 2021). It takes advantage of the given label maps to find diverse *class-specific* latent directions without requiring further supervision or mask area definitions, enabling the user to perform image editing interactively.

2.4 Alternatives to GANs

In this section, we discuss image synthesis models that are optimized with a traditional maximum likelihood approach instead of adversarial training. The three most widely used paradigms are variational autoencoders (VAEs), transformers, and diffusion models. In the following, we explain these models, including their advantages and disadvantages with respect to GANs.

VAEs. VAEs (Kingma and Welling, 2014) are autoencoders in which latent variables are parametrized by a distribution that we can sample from. The most widely used parametrization is to learn the mean and variance of a latent Gaussian distribution. The training loss is the KL divergence between the latent distribution and a Gaussian prior combined with a reconstruction loss. The best-performing VAEs can achieve high image quality on datasets with a simple structure, e.g., face datasets (Child, 2020; Vahdat and Kautz, 2020; Hazami et al., 2022). However, VAEs struggle to produce satisfying samples on more challenging datasets like ImageNet (Child, 2020). While the current state-of-the-art models work well on resolution 256, training at resolution 1024 does not lead to satisfying visual results (Child, 2020; Hazami et al., 2022). Lastly, VAEs tend to produce blurry images since the euclidean reconstruction loss minimizes the mean distance between all plausible outputs for a given Gaussian latent.

Diffusion models. Diffusion models (Ho et al., 2020) consist of a so-called forward and backward process. The forward process gradually adds Gaussian noise to an image. The backward process gradually restores the unperturbed image, employing a U-Net to train to predict the added noise at various noise levels. During inference, the backward process is applied to pure noise to generate an image via denoising iteratively. Unlike GANs, diffusion models do not suffer from training instabilities and can achieve similar or better image quality (Dhariwal and Nichol, 2021). In addition, diffusion models have better mode coverage of the training distribution compared to GANs, which are well known for suffering from a mode-dropping problem. However, the sampling time is slower due to the iterative generation procedure, which may require thousands of network evaluations. Diffusion models can be used more efficiently by applying them only in an image generator's latent space, referred to as a latent diffusion model (Rombach et al., 2022). In latent diffusion, an autoencoder is trained in combination with a GAN loss. Afterward, the autoencoder is fixed and a diffusion model is trained to generate the latents. In essence, the diffusion model learns the image's high-level structure, while the autoencoder decoder models the low-level details.

Transformers. Transformers are powerful sequence-to-sequence models that can be used to model images in the pixel space (Chen et al., 2020a) or latent space of an autoencoder (Esser et al., 2021). In the latter case, the autoencoder is trained with a

GAN loss to produce sharp images and increase the realism (Esser et al., 2021). The advantages are that each location attends to all others, which contributes to image realism, and that training is stable and therefore highly scalable (Yu et al., 2022). Further, the use of transformers allows seamless integration with language models, benefitting text-to-image synthesis(Yu et al., 2022). The downside of transformers lies in the long iterative sampling time. Hence, just as in the case of diffusion models, modeling the image directly in the pixel space is costly.

3 A U-Net-Based GAN Discriminator

Contents

3.1 Introduction **30**
3.2 U-Net GAN model **32**
3.2.1 The discriminator baseline. 33
3.2.2 Mix and cut regularizations. 33
3.2.3 U-Net based discriminator 34
3.2.4 Consistency regularization 35
3.2.5 Implementation 38
3.3 Experiments **41**
3.3.1 Experimental setup 41
3.3.2 Results 43
3.4 Conclusion **51**

Among the major remaining challenges for generative adversarial networks (GANs) is the capacity to synthesize globally and locally coherent images with object shapes and textures indistinguishable from real images. In this chapter, we address this issue by proposing an alternative U-Net based discriminator architecture, borrowing insights from the segmentation literature. The proposed U-Net based architecture allows to provide detailed per-pixel feedback to the generator while maintaining the global coherence of synthesized images, by providing the global image feedback as well. Empowered by the per-pixel response of the discriminator, we further propose a per-pixel consistency regularization technique based on the CutMix data augmentation, encouraging the U-Net discriminator to focus more on semantic and structural changes between real and fake images. This improves the U-Net discriminator training, further enhancing the quality of generated samples. The novel discriminator improves over the state of the art in terms of the standard distribution and image quality metrics,

enabling the generator to synthesize images with varying structure, appearance, and levels of detail, maintaining global and local realism. Compared to the BigGAN baseline, we achieve an average improvement of 2.7 FID points across FFHQ, CelebA, and the newly introduced COCO-Animals dataset. The code is available at `https://github.com/boschresearch/unetgan`. The work presented in this chapter was published as the CVPR 2022 paper "A U-Net-based discriminator for generative adversarial networks" (Schönfeld et al., 2020).

This chapter takes the first step toward the GAN improvements presented in this thesis. Chapter 4 uses the discriminator architecture proposed in this chapter as starting point to develop a more powerful semantic image synthesis model. Thanks to this architecture change and additional improvements, we strongly increase the synthesis diversity in semantic image synthesis GANs. Based on this increased diversity, in Chapter 5 we propose a method to control the appearance of classes through walks in the latent space of semantic image synthesis GANs

3.1 Introduction

The quality of synthetic images produced by generative adversarial networks (GANs) has seen tremendous improvement recently (Brock et al., 2019; Karras et al., 2019a). The progress is attributed to large-scale training (Menick and Kalchbrenner, 2019; Brock et al., 2019), architectural modifications (Zhang et al., 2019; Karras et al., 2018, 2019a; Lin et al., 2019), and improved training stability via the use of different regularization techniques (Miyato et al., 2018; Zhang et al., 2020a). However, despite the recent advances, learning to synthesize images with global semantic coherence, long-range structure, and the exactness of detail remains challenging.

One source of the problem lies potentially in the discriminator network. The discriminator aims to model the data distribution, acting as a loss function to provide the generator with a learning signal to synthesize realistic image samples. The stronger the discriminator, the better the generator has to become. In the current state-of-the-art GAN models, the discriminator being a classification network learns only a representation that allows to efficiently penalize the generator based on the most discriminative difference between real and synthetic images. Thus, it often focuses on either the global structure or local details. The problem amplifies as the discriminator has to learn in a non-stationary environment: the distribution of synthetic samples shifts as the generator constantly changes through training, and is prone to forgetting previous tasks (Chen et al., 2019) (in the context of the discriminator training, learning semantics, structures, and textures can be considered different tasks). This discriminator is not incentivized to maintain a more powerful data representation, learning both global and local image differences. This often results in the generated images with discontinued and mottled local structures

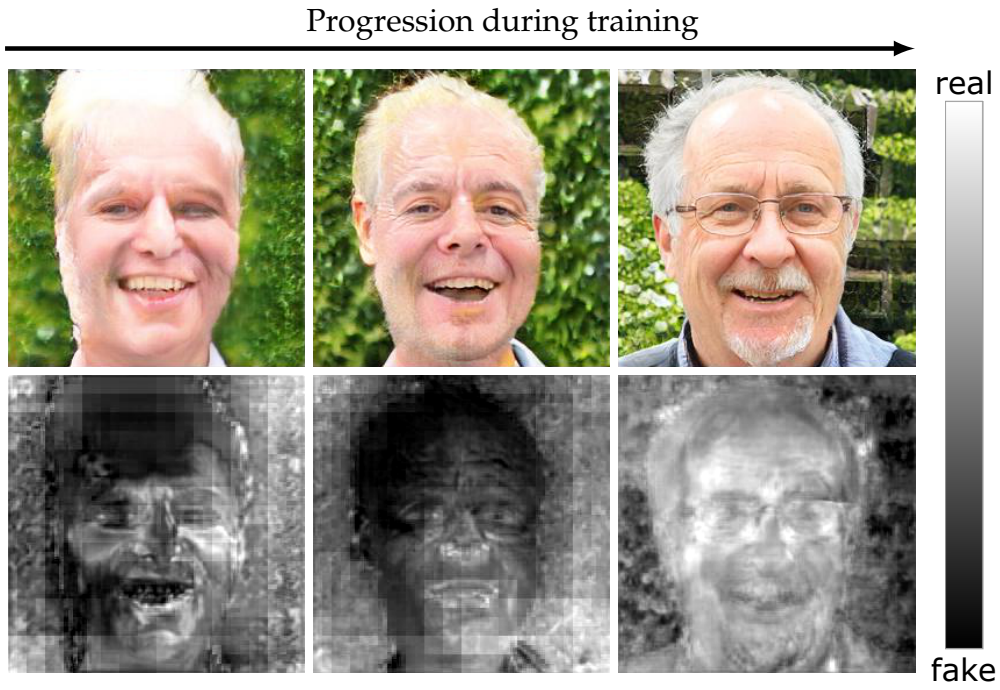

Figure 3.1: Images produced throughout the training by our U-Net GAN model (top row) and their corresponding per-pixel feedback of the *U-Net discriminator* (bottom row). The synthetic image samples are obtained from a fixed noise vector at different training iterations. Brighter colors correspond to the discriminator confidence of pixel being real (and darker of being fake). Note that the U-Net discriminator provides very detailed and spatially coherent response to the generator, enabling it to further improve the image quality, e.g., the unnaturally large man's forehead is recognized as fake by the discriminator and is corrected by the generator throughout the training.

(Lin et al., 2019) or images with incoherent geometric and structural patterns (e.g., asymmetric faces or animals with missing legs) (Zhang et al., 2019).

To mitigate this problem, we propose an alternative discriminator architecture, which simultaneously outputs both global (over the whole image) and local (per-pixel) decision of the image belonging to either the real or fake class, see Figure 3.1. Motivated by the ideas from the segmentation literature, we redesign the discriminator to take a role of both a classifier and segmenter. We change the architecture of the discriminator network to a U-Net (Ronneberger et al., 2015), where the encoder module performs per-image classification, as in the standard GAN setting, and the decoder module outputs per-pixel class decision, providing spatially coherent feedback to the generator, see Figure 3.2. This architectural change leads to a stronger discriminator, which is encouraged to maintain a more powerful data representation, making the generator task of fooling the discriminator more challenging and thus improving the quality of generated samples (as also reflected in the generator and discriminator loss behavior in Figure 3.10). Note that we do

not modify the generator in any way, and our work is orthogonal to the ongoing research on architectural changes of the generator (Karras et al., 2019a; Lin et al., 2019), divergence measures (Li et al., 2017; Arjovsky et al., 2017; Nowozin et al., 2016a), and regularizations (Roth et al., 2017; Gulrajani et al., 2017b; Miyato et al., 2018).

The proposed U-Net based discriminator allows to employ the recently introduced CutMix (Yun et al., 2019) augmentation, which is shown to be effective for classification networks, for consistency regularization in the two-dimensional output space of the decoder. Inspired by Yun et al. (2019), we cut and mix the patches from real and synthetic images together, where the ground truth label maps are spatially combined with respect to the real and fake patch class for the segmenter (U-Net decoder) and the class labels are set to fake for the classifier (U-Net encoder), as globally the CutMix image should be recognized as fake, see Figure 3.3. Empowered by per-pixel feedback of the U-Net discriminator, we further employ these CutMix images for consistency regularization, penalizing per-pixel inconsistent predictions of the discriminator under the CutMix transformations. This fosters the discriminator to focus more on semantic and structural changes between real and fake images and to attend less to domain-preserving perturbations. Moreover, it also helps to improve the localization ability of the decoder. Employing the proposed consistency regularization leads to a stronger generator, which pays more attention to local and global image realism. We call our model U-Net GAN.

We evaluate the proposed U-Net GAN model across several datasets using the state-of-the-art BigGAN model (Brock et al., 2019) as a baseline and observe an improved quality of the generated samples in terms of the FID and IS metrics. For unconditional image synthesis on FFHQ (Karras et al., 2019a) at resolution 256×256, our U-Net GAN model improves 4 FID points over the BigGAN model, synthesizing high-quality human faces (see Figure 3.5). On CelebA (Liu et al., 2015) at resolution 128×128 we achieve 1.6 point FID gain, yielding to the best of our knowledge the lowest known FID score of 2.95. For class-conditional image synthesis on the introduced COCO-Animals dataset (Lin et al., 2014; Kuznetsova et al., 2018) at resolution 128×128 we observe an improvement in FID from 16.37 to 13.73, synthesizing diverse images of different animal classes (see Figure 3.6).

3.2 U-Net GAN model

In this section, we present our U-Net GAN model. The key to our approach is to redesign the "vanilla" GAN discriminator as a U-Net (Ronneberger et al., 2015), as well as a consistency regularization enabled by this new architecture. To this end, we first briefly explain the baseline vanilla discriminator (Sec. 3.2.1), followed by an explanation of recently proposed mix and cut regularization techniques that form the basis of our proposed regularization (Sec. 3.2.2). Next, we present our proposed

U-Net GAN discriminator (Sec. 3.2.3) and consistency regularization (Sec. 3.2.4). Lastly, we present implementation details (Sec. 3.2.5). Note that our method is compatible with most GAN models as it does not modify the generator in any way and leaves the original GAN objective intact.

3.2.1 The discriminator baseline.

A vanilla GAN consists of two networks: a generator G and a discriminator D, trained by minimizing the following competing objectives in an alternating manner:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_x[\log D(x)] - \mathbb{E}_z[\log(1 - D(G(z)))], \\ \mathcal{L}_G &= -\mathbb{E}_z[\log D(G(z))]^1. \end{aligned} \tag{3.1}$$

G aims to map a latent variable $z \sim p(z)$ sampled from a prior distribution to a realistic-looking image, while D aims to distinguish between real x and generated $G(z)$ images. Ordinarily, G and D are modeled as a decoder and an encoder convolutional network, respectively. While there are many variations of the GAN objective function and its network architectures (Kurach et al., 2018; Lučić et al., 2018), in this work we focus on improving the discriminator network. Our chosen baseline is the state-of-the-art model BigGAN (Brock et al., 2019) with an encoder-shaped discriminator. In Section 3.2.3, we propose to alter this D architecture from a standard classification network to an encoder-decoder network – U-Net (Ronneberger et al., 2015), leaving the underlying basic architecture of D – the encoder part – untouched. The proposed discriminator allows to maintain both global and local data representation, providing more informative feedback to the generator.

3.2.2 Mix and cut regularizations.

Recently, a few simple yet effective regularization techniques have been proposed, which are based on augmenting the training data by creating synthetic images via mixing or/and cutting samples from different classes. In MixUp (Zhang et al., 2018b) the input images and their target labels are interpolated using the same randomly chosen factor. Verma et al. (2019a) extends Zhang et al. (2018b) by performing interpolation not only in the input layer but also in the intermediate layers. CutOut (Devries and Taylor, 2017) augments an image by masking a rectangular region to zero. Differently, CutMix (Yun et al., 2019) augments training data by creating synthetic images via cutting and pasting patches from image samples of different classes, marrying the best aspects of MixUp and CutOut. Other works employ the Mix&Cut approaches for consistency regularization (Verma et al., 2019b; Berthelot et al., 2019; Zhang et al., 2020a), to encourage invariance to the MixUp

[1]This formulation is originally proposed as non-saturating (NS) GAN in Goodfellow et al. (2014).

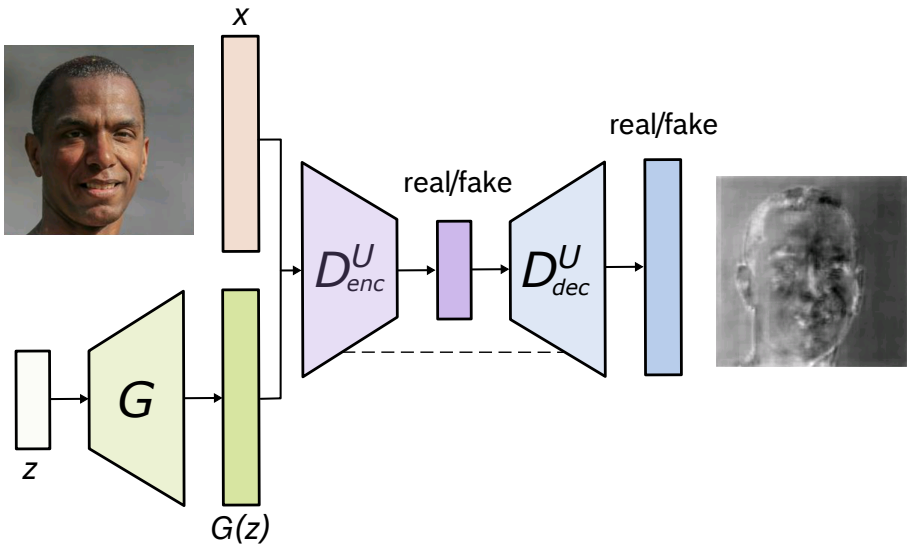

Figure 3.2: U-Net GAN. The proposed U-Net discriminator classifies the input images on a global and local *per-pixel* level. Due to the skip-connections between the encoder and the decoder (dashed line), the channels in the output layer contain both high- and low-level information. Brighter colors in the decoder output correspond to the discriminator confidence of pixels being real (and darker of being fake).

or CutOut transformation. Of the aforementioned previous works, Zhang et al. (2020a) and Zhao et al. (2020b) are the only to apply Mix&Cut approaches to GANs. In our work, we propose the consistency regularization under the CutMix transformation. In contrast to previous work, the CutMix transformation is applied in the *pixel output space* of our U-Net discriminator, mixes real and fake images, and encourages equivariances to the transformation. This helps to improve the localization quality of the U-Net discriminator and induce it to attend more to semantic and structural changes between real and fake samples. We call our model *U-Net GAN*.

3.2.3 U-Net based discriminator

Encoder-decoder networks (Badrinarayanan et al., 2017; Ronneberger et al., 2015) constitute a powerful method for dense prediction. U-Nets (Ronneberger et al., 2015) in particular have demonstrated state-of-the-art performance in many complex image segmentation tasks. In these methods, similarly to image classification networks, the encoder progressively downsamples the input, capturing the global image context. The decoder performs progressive upsampling, matching the output resolution to the input resolution, and thus enabling precise localization. Skip connections route data between the matching resolutions of the two modules, further improving the ability of the network to accurately segment fine details.

Analogously, in this work, we propose to extend a discriminator to form a U-

Net, by reusing building blocks of the original discriminator classification network as an encoder part and building blocks of the generator network as the decoder part. In other words, the discriminator now consists of the original downsampling network and a new upsampling network. The two modules are connected via a bottleneck, as well as skip-connections that copy and concatenate feature maps from the encoder and the decoder modules, following Ronneberger et al. (2015). We will refer to this discriminator as D^U. While the original $D(x)$ classifies the input image x into being real and fake, the U-Net discriminator $D^U(x)$ additionally performs this classification on a *per-pixel* basis, segmenting the image x into real and fake regions, along with the original image classification of x from the encoder, see Figure 3.2. This enables the discriminator to learn both global and local differences between real and fake images.

Hereafter, we refer to the original encoder module of the discriminator as D^U_{enc} and to the introduced decoder module as D^U_{dec}. The new discriminator loss can now be computed by taking the decisions from both D^U_{enc} and D^U_{dec}:

$$\mathcal{L}_{D^U} = \mathcal{L}_{D^U_{enc}} + \mathcal{L}_{D^U_{dec}}, \tag{3.2}$$

where similarly to Eq. 3.1 the loss for the encoder $L_{D^U_{enc}}$ is computed from the scalar output of D^U_{enc}:

$$\mathcal{L}_{D^U_{enc}} = -\mathbb{E}_x[\log D^U_{enc}(x)] - \mathbb{E}_z[\log(1 - D^U_{enc}(G(z)))], \tag{3.3}$$

and the loss for the decoder $L_{D^U_{enc}}$ is computed as the mean decision over all pixels:

$$\mathcal{L}_{D^U_{dec}} = -\mathbb{E}_x\Big[\sum_{i,j} \log[D^U_{dec}(x)]_{i,j}\Big] - \mathbb{E}_z\Big[\sum_{i,j} \log(1 - [D^U_{dec}(G(z))]_{i,j})\Big]. \tag{3.4}$$

Here, $[D^U_{dec}(x)]_{i,j}$ and $[D^U_{dec}(G(z))]_{i,j}$ refer to the discriminator decision at pixel (i, j). These per-pixel outputs of D^U_{dec} are derived based on global information from high-level features, enabled through the process of upsampling from the bottleneck, as well as more local information from low-level features, mediated by the skip connections from the intermediate layers of the encoder network.

Correspondingly, the generator objective becomes:

$$\mathcal{L}_G = -\mathbb{E}_z\Big[\log D^U_{enc}(G(z)) + \sum_{i,j} \log[D^U_{dec}(G(z))]_{i,j}\Big], \tag{3.5}$$

encouraging the generator to focus on both global structures and local details while synthesizing images in order to fool the more powerful discriminator D^U.

3.2.4 Consistency regularization

Here we present the consistency regularization technique for the U-Net based discriminator introduced in the previous section. The per-pixel decision of the well-trained D^U discriminator should be equivariant under any class-domain-altering

transformations of images. However, this property is not explicitly guaranteed. To enable it, the discriminator should be regularized to focus more on semantic and structural changes between real and fake samples and to pay less attention to arbitrary class-domain-preserving perturbations. Therefore, we propose the consistency regularization of the D^U discriminator, explicitly encouraging the decoder module D^U_{dec} to output equivariant predictions under the CutMix transformations (Yun et al., 2019) of real and fake samples. The CutMix augmentation creates synthetic images via cutting and pasting patches from images of different classes. We choose CutMix among other Mix&Cut strategies as it does not alter the real and fake image patches used for mixing, in contrast to Zhang et al. (2018b), preserving their original class domain, and provides a large variety of possible outputs. We visualize the CutMix augmentation strategy and the D^U predictions in Figure 3.3.

Following (Yun et al., 2019), we synthesize a new training sample $\tilde{x}$ for the discriminator D^U by mixing x and $G(z) \in \mathbb{R}^{W \times H \times C}$ with the mask M:

$$\begin{aligned} &\tilde{x} = \text{mix}(x, G(z), \text{M}), \\ &\text{mix}(x, G(z), \text{M}) = \text{M} \odot x + (1 - \text{M}) \odot G(z), \end{aligned} \tag{3.6}$$

where $\text{M} \in \{0,1\}^{W \times H}$ is the binary mask indicating whether the pixel (i, j) comes from the real ($\text{M}_{i,j} = 1$) or fake ($\text{M}_{i,j} = 0$) image, 1 is a binary mask filled with ones, and $\odot$ is an element-wise multiplication. In contrast to (Yun et al., 2019), the class label $c \in \{0,1\}$ for the new CutMix image $\tilde{x}$ is set to be fake, i.e., $c = 0$. Globally, the mixed synthetic image should be recognized as fake by the encoder D^U_{enc}, otherwise the generator can learn to introduce the CutMix augmentation into generated samples, causing undesirable artifacts. Note that for the synthetic sample $\tilde{x}$, $c = 0$ and M are the ground truth for the encoder and decoder modules of the discriminator D^U, respectively.

Given the CutMix operation in Eq. 3.6, we train the discriminator to provide consistent per-pixel predictions, i.e., $D^U_{dec}\big(\text{mix}(x, G(z), \text{M})\big) \approx \text{mix}\big(D^U_{dec}(x), D^U_{dec}(G(z)), \text{M}\big)$, by introducing the consistency regularization loss term in the discriminator objective:

$$\mathcal{L}^{cons}_{D^U_{dec}} = \left\| D^U_{dec}\Big(\text{mix}(x, G(z), \text{M})\Big) - \text{mix}\Big(D^U_{dec}(x), D^U_{dec}(G(z)), \text{M}\Big) \right\|^2, \tag{3.7}$$

where denotes $\|\cdot\|$ the L^2 norm. This consistency loss is then taken between the per-pixel output of D^U_{dec} on the CutMix image and the CutMix between outputs of the D^U_{dec} on real and fake images, penalizing the discriminator for inconsistent predictions.

We add the loss term in Eq. 3.7 to the discriminator objective in Eq. 3.2 with a weighting hyperparameter λ:

$$\mathcal{L}_{D^U} = \mathcal{L}_{D^U_{enc}} + \mathcal{L}_{D^U_{dec}} + \lambda \mathcal{L}^{cons}_{D^U_{dec}}.. \tag{3.8}$$

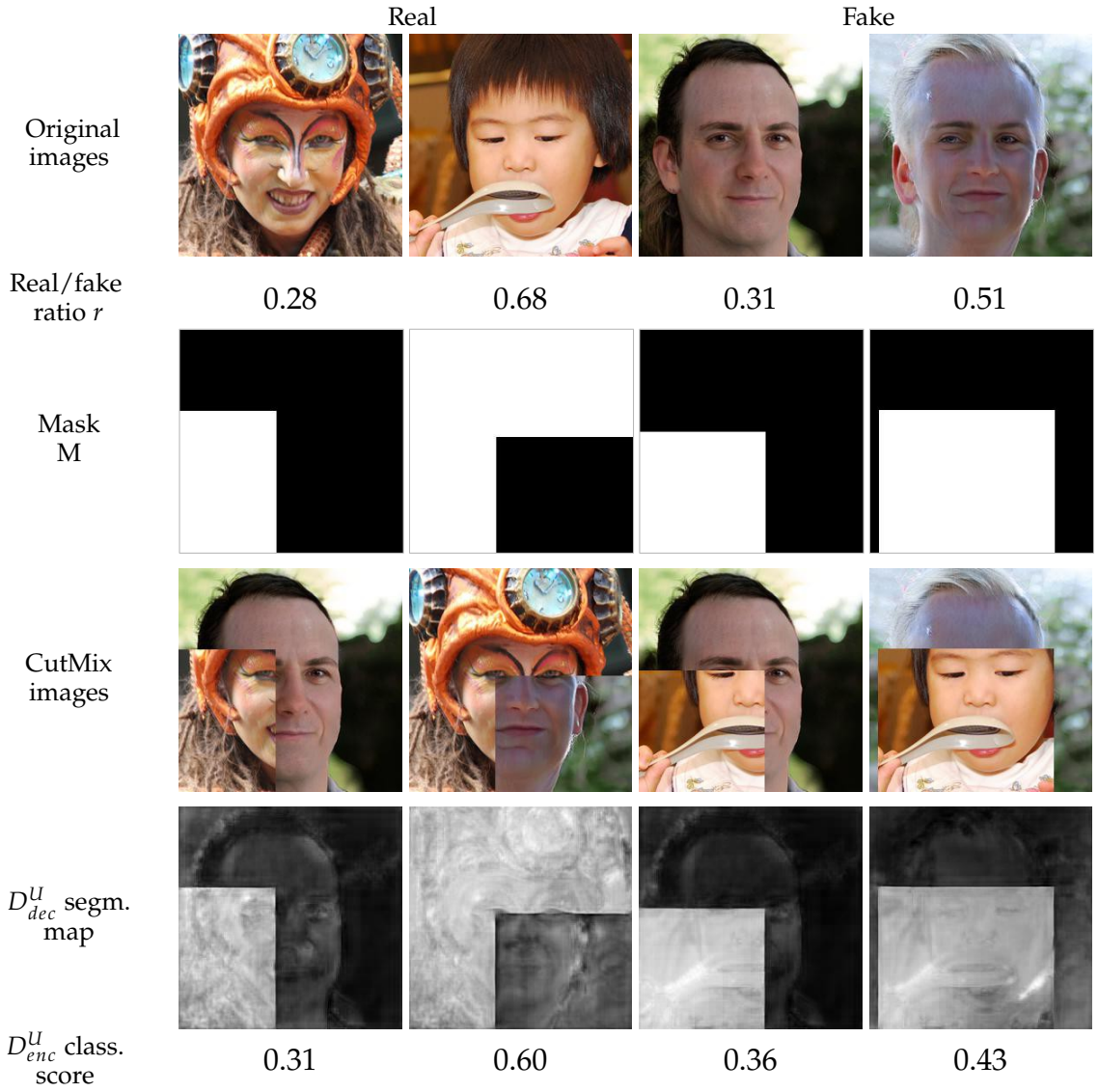

Figure 3.3: Visualization of the CutMix augmentation and the predictions of the U-Net discriminator on CutMix images. 1st row: real and fake samples. 2nd&3rd rows: sampled real/fake CutMix ratio r and corresponding binary masks M (color code: white for real, black for fake). 4th row: generated CutMix images from real and fake samples. 5th&6th row: the corresponding real/fake segmentation maps of D^U with its predicted classification scores.

The generator objective $\mathcal{L}_G$ remains unchanged, see Eq. 3.5.

In addition to the proposed consistency regularization, we also use CutMix samples for training both the encoder and decoder modules of D^U. Note that for the U-Net GAN we use the non-saturating GAN objective formulation (Goodfellow et al., 2014). However, the introduced consistency regularization as well as the U-Net architecture of the discriminator can be combined with any other adversarial losses of the generator and discriminator (Arjovsky et al., 2017; Lim and Ye, 2017; Nowozin et al., 2016a).

3.2.5 Implementation

Here we discuss the implementation details of the U-Net GAN model proposed in Section 3.2.3 and 3.2.4.

U-Net based discriminator. We build upon the recent state-of-the-art BigGAN model (Brock et al., 2019), and extend its discriminator with our proposed changes. The architecture details of the BigGAN model (Brock et al., 2019) and our U-Net discriminator are summarized in Table 3.1, 3.2 and Table 3.3. From these tables it is easy to see that the encoder and decoder of the U-Net discriminator follow the original BigGAN discriminator and generator setups, respectively.

We adopt the BigGAN generator and discriminator architectures for the 256×256 (and 128×128) resolution with a channel multiplier $ch = 64$, as described in detail in (Brock et al., 2019). The original BigGAN discriminator downsamples the input image to a feature map of dimensions $16ch \times 4 \times 4$, on which global sum pooling is applied to derive a $16ch$ dimensional feature vector that is classified into real or fake. In order to turn the discriminator into a U-Net, we copy the generator architecture and append it to the 4×4 output of the discriminator. In effect, the features are successively upsampled via ResNet blocks until the original image resolution ($H \times W$) is reached. To make the U-Net complete, the input to every decoder ResNet block is concatenated with the output features of the encoder blocks that share the same intermediate resolution. In this way, high-level and low-level information are effectively integrated on the way to the output feature map. Hereby, the decoder architecture is almost identical to the generator, with the exception that we change the number of channels of the final output from 3 to ch, append a final block of 1×1 convolutions to produce the $1 \times H \times W$ output map, and do not use class-conditional BatchNorm (de Vries et al., 2017; Dumoulin et al., 2017) in the decoder, nor the encoder. Similarly to (Brock et al., 2019), we provide class information to D^U with projection (Miyato and Koyama, 2018) to the ch-dimensional channel features of the U-Net encoder and decoder output. In contrast to (Brock et al., 2019) and in alignment with (Chen et al., 2018b), we find it beneficial not to use a hierarchical latent space, but to directly feed the same input vector z to BatchNorm at every layer in the generator. Lastly, we also remove the self-attention layer in both encoder and decoder, as in our experiments they did not contribute to the performance but

(a) BigGAN Generator (128×128)

$z \in \mathbb{R}^{120} \sim \mathcal{N}(0, I)$ Embed(y) $\in \mathbb{R}^{128}$
Linear $(20 + 128) \rightarrow 4 \times 4 \times 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 4ch$
ResBlock up $4ch \rightarrow 2ch$
Non-Local Block (64×64)
ResBlock up $2ch \rightarrow ch$
BN, ReLU, 3×3 Conv $ch \rightarrow 3$
Tanh

(b) BigGAN Discriminator (128×128)

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down $ch \rightarrow 2ch$
Non-Local Block (64×64)
ResBlock down $2ch \rightarrow 4ch$
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$
ResBlock down $16ch \rightarrow 16ch$
ReLU, Global sum pooling
Embed(y)$\cdot h$ + (linear$\rightarrow$ 1)

Table 3.1: The BigGAN (Brock et al., 2019) generator and discriminator architectures for *class-conditional* image generation.

led to memory overhead. While the original BigGAN is a class-conditional model, we additionally devise an unconditional version for our experiments. For the unconditional model, we replace class-conditional BatchNorm with self-modulation (Chen et al., 2018b), where the BatchNorm parameters are conditioned only on the latent vector z, and do not use the class projection of (Miyato and Koyama, 2018) in the discriminator.

All these modifications leave us with a two-headed discriminator. While the decoder head is already sufficient to train the network, we find it beneficial to compute the GAN loss at the encoder and decoder head with equal weight. Analogously to BigGAN, we keep the hinge loss (Zhang et al., 2019) in all basic U-Net models, while the models that also employ the consistency regularization in the decoder output space benefit from using the non-saturating loss (Goodfellow et al., 2014). Our implementation builds on top of the original BigGAN PyTorch implementation[2].

Consistency regularization. For each training iteration a mini-batch of CutMix images $(\tilde{x}, c = 0, \mathrm{M})$ is created with probability p_{mix}. This probability is increased linearly from 0 to 0.5 between the first n epochs in order to give the generator time to learn how to synthesize more real looking samples and not to give the discriminator too much power from the start. CutMix images are created from the existing real and fake images in the mini-batch using binary masks M. For sampling M, we use the original CutMix implementation[3]: first sampling the combination ratio r between the real and generated images from the uniform distribution $(0, 1)$ and

[2] https://github.com/ajbrock/BigGAN-PyTorch
[3] https://github.com/clovaai/CutMix-PyTorch

(a) BigGAN Generator (256×256)

$z \in \mathbb{R}^{140} \sim \mathcal{N}(0, I)$
Linear $(20 + 128) \rightarrow 4 \times 4 \times 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 4ch$
ResBlock up $4ch \rightarrow 2ch$
Non-Local Block (128×128)
ResBlock up $2ch \rightarrow ch$
BN, ReLU, 3×3 Conv $ch \rightarrow 3$
Tanh

(b) BigGAN Discriminator (256×256)

RGB image $x \in \mathbb{R}^{256 \times 256 \times 3}$
ResBlock down $ch \rightarrow 2ch$
ResBlock down $2ch \rightarrow 4ch$
Non-Local Block (64×64)
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$
ResBlock down $16ch \rightarrow 16ch$
ReLU, Global sum pooling
linear$\rightarrow 1$

Table 3.2: The BigGAN (Brock et al., 2019) generator and discriminator architectures, modified for *unconditional* image generation.

(a) U-Net GAN Discriminator (256×256, unconditional)

RGB image $x \in \mathbb{R}^{256 \times 256 \times 3}$
ResBlock down $ch \rightarrow 2ch$
ResBlock down $2ch \rightarrow 4ch$
Optional Non-Local Block (64×64)
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$ *(see below)
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $(8 + 8)ch \rightarrow 8ch$
ResBlock up $(8 + 8)ch \rightarrow 4ch$
ResBlock up $(4 + 4)ch \rightarrow 2ch$
ResBlock up $(2 + 2)ch \rightarrow ch$
ResBlock up $(ch + ch) \rightarrow ch$
ResBlock $ch \rightarrow 1$
Sigmoid
* ReLU, Global sum pooling, linear$\rightarrow 1$

(b) U-Net GAN Discriminator (128×128, class-conditional)

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down $ch \rightarrow 2ch$
Optional Non-Local Block (64×64)
ResBlock down $2ch \rightarrow 4ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$ *(see below)
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $(8 + 8)ch \rightarrow 4ch$
ResBlock up $(4 + 4)ch \rightarrow 2ch$
ResBlock up $(2 + 2)ch \rightarrow ch$
ResBlock up $(ch + ch) \rightarrow ch$
Embed(y)$\cdot h$ + (Conv $ch \rightarrow 1$)
Sigmoid
* ReLU, Global sum pooling
Embed(y)$\cdot h$ + (linear$\rightarrow 1$)

Table 3.3: The U-Net GAN discriminator architectures for class-conditional (a) and unconditional (b) tasks of generating images at resolution 128×128 and 256×256, respectively.

then uniformly sample the bounding box coordinates for the cropping regions of x and $G(z)$ to preserve the r ratio, i.e., $r = \frac{|\mathrm{M}|}{W * H}$ (see Figure 3.3). Binary masks M also

denote the target for the decoder D^U_{dec}, while we use *fake*, i.e., $c = 0$, as the target for the encoder D^U_{enc}. We set $\lambda = 1.0$ as it showed empirically to be a good choice. Note that the consistency regularization does not impose much overhead during training. Extra computational cost comes only from feeding additional CutMix images through the discriminator while updating its parameters.

3.3 Experiments

3.3.1 Experimental setup

Datasets. We consider three datasets: FFHQ (Karras et al., 2019a), CelebA (Liu et al., 2015), and the subset of the COCO (Lin et al., 2014) and OpenImages (Kuznetsova et al., 2018) images containing animal classes, which we will further on refer to as COCO-Animals. We use FFHQ and CelebA for unconditional image synthesis and COCO-Animals for class-conditional image synthesis, where the class label is used. We experiment with 256×256 resolution for FFHQ, and 128×128 for CelebA and COCO-Animals.

CelebA is a human face dataset of 200k images, featuring $\sim$ 10k different celebrities with a variety of facial poses and expressions. Similarly, FFHQ is a more recent dataset of human faces, consisting of 70k high-quality images with higher variation in terms of age, ethnicity, accessories, and viewpoints.

The proposed COCO-Animals dataset consists of $\sim$ 38k training images belonging to 10 animal classes, where we choose COCO (Lin et al., 2014) and OpenImages (Kuznetsova et al., 2018) (using the human verified subset with mask annotations) samples in the categories *bird*, *cat*, *dog*, *horse*, *cow*, *sheep*, *giraffe*, *zebra*, *elephant*, and *monkey*. The two datasets have a great overlap in animal classes. We take *all* images from COCO and the aforementioned OpenImages split in the categories *horse*, *cow*, *sheep*, *giraffe*, *zebra*, and *elephant*. The *monkey* images are taken over directly from OpenImages, since this category contained more training samples than the next biggest COCO animal class *bear*. The class *bear* and *monkey* are not shared between COCO and OpenImages. Lastly, the categories *bird*, *cat*, and *dog* contained vastly more samples than all other categories. For this reason, we took over only a subset of the total of all images in these categories. These samples were picked from OpenImages only, for their better visual quality. To ensure good quality of the picked examples, we used the provided bounding boxes to filter out images in which the animal of interest is either too small or too big ($> 80\%$, $< 30\%$ of the image area for cats, $> 70\%$, $< 50\%$ for birds and dogs). The thresholds were chosen such that the number of appropriate images is approximately equal.

With its relatively small size and imbalanced number of images per class as well as due to its variation in poses, shapes, number of objects, and backgrounds, COCO-Animals presents a challenging task for class-conditional image synthesis.

We choose to create this dataset in order to perform conditional image generation in the mid- to high-resolution regime, with a reasonable computational budget and feasible training time. Other datasets in this order of size either have too few examples per class (e.g., AwA (Xian et al., 2018)) or too little inter- and intra-class variability. In contrast, the intra-class variability of COCO-Animals is very high for certain classes, e.g., bird and monkey, which span many subspecies.

Evaluation metrics. For quantitative evaluation we use the Fréchet Inception distance (FID) (Heusel et al., 2017b) as the main metric, and additionally consider the Inception score (IS) (Salimans et al., 2016b). Between the two, FID is a more comprehensive metric, which has been shown to be more consistent with human evaluation in assessing the realism and variation of the generated images (Heusel et al., 2017b), while IS is limited by what the Inception classifier can recognize, which is directly linked to its training data (Barratt and Sharma, 2018). If one learns to generate something not present in the classifier's training data (e.g., human faces) then IS can still be low despite generating high-quality images since that image does not get classified as a distinct class.

In all our experiments, FID and IS are computed using 50k synthetic images, following Karras et al. (2018). By default, all reported numbers correspond to the best or median FID of five independent runs achieved with 400k training iterations for FFHQ and COCO-Animals, and 800k training iterations for CelebA. For evaluation, we employ moving averages of the generator weights following Brock et al. (2019) and Karras et al. (2018), with a decay of 0.9999. Note that we do not use any truncation tricks or rejection sampling for image generation.

Training details. We adopt the original training parameters of Brock et al. (2019) for training U-Net GAN, which are summarized in Table 3.4.

Hyperparameter	Value
Optimizer	Adam ($\beta_1 = 0, \beta_2 = 0.999$)
G's learning rate	1e-4 (256), 5e-5 (128)
D's learning rate	5e-4 (256), 2e-4 (128)
Batch size	20 (256), 80 (128)
Weight Initialization	Orthogonal

Table 3.4: Hyperparameters of U-Net GAN for resolution 256^2 and 128^2.

In particular, we use a uniformly distributed noise vector $z \in [-1, 1]^{140}$ as input to the generator, and the Adam optimizer (Kingma and Ba, 2015) with different learning rates for G and D^U. The number of warmup epochs n for consistency regularization is chosen to be 200 for COCO-Animals, and 20 for FFHQ and CelebA. In contrast to Brock et al. (2019), we operate with considerably smaller mini-batch sizes: 20 for FFHQ, 50 for CelebA, and 80 for COCO-Animals. Regarding the difference between class-conditional and unconditional image generation, it is worth

noting that the CutMix regularization is applied only to samples within the same class. In other words, real and generated samples are mixed only within the class (e.g., real and fake zebras, but not real zebras with fake elephants).

3.3.2 Results

We first test our proposed U-Net discriminator in two settings: unconditional image synthesis on FFHQ and class-conditional image synthesis on COCO-Animals, using the BigGAN model (Brock et al., 2019) as a baseline for comparison. We report our key results in Table 3.5 and Figure 3.4.

Method	FFHQ				COCO-Animals			
	Best		Median		Best		Median	
	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
BigGAN (Brock et al., 2019)	11.48	3.97	12.42	4.02	16.37	11.77	16.55	11.78
U-Net GAN	**7.48**	**4.46**	**7.63**	**4.47**	**13.73**	**12.29**	**13.87**	**12.31**

Table 3.5: Evaluation results on FFHQ and COCO-Animals. We report the best and median FID score across 5 runs and its corresponding IS, see Section 3.3.2 for discussion.

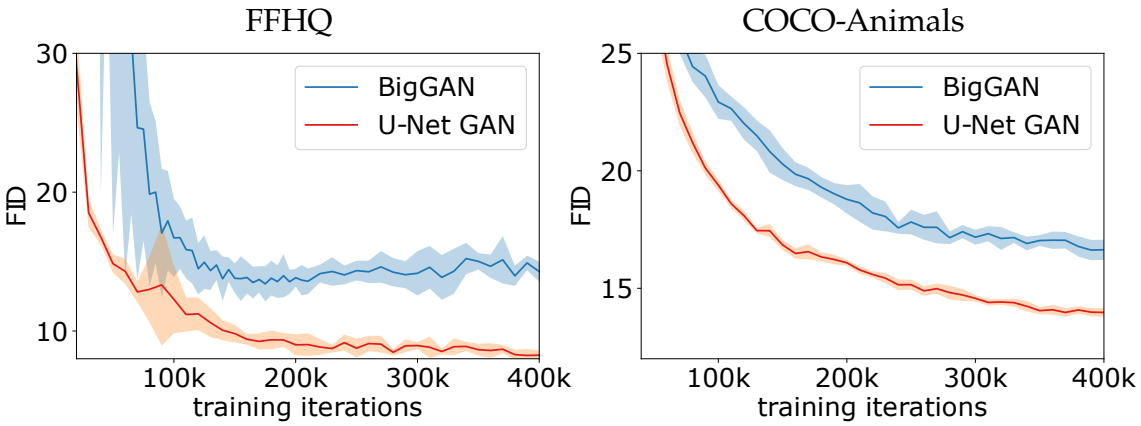

Figure 3.4: FID curves over iterations of the BigGAN model (blue) and the proposed U-Net GAN (red). Depicted are the FID mean and standard deviation across 5 runs per setting.

In the unconditional case, our model achieves the FID score of 7.48, which is an improvement of 4.0 FID points over the canonical BigGAN discriminator (see Table 3.5). In addition, the new U-Net discriminator also improves over the baseline in terms of the IS metric (3.97 vs. 4.46). The same effect is observed for the conditional image generation setting. Here, our U-Net GAN achieves an FID of 13.73, improving 2.64 points over BigGAN, as well as increases the IS score from 11.77 to 12.29. Figure 3.4 visualizes the mean FID behaviour over the training

Method	Dataset	FID			
		Best	Median	Mean	Std
BigGAN	COCO-Animals	16.37	16.55	16.62	0.24
U-Net GAN		**13.73**	**13.87**	**13.88**	**0.11**
BigGAN	FFHQ	11.48	12.42	12.35	0.67
U-Net GAN		**7.48**	**7.63**	**7.73**	**0.56**
BigGAN	CelebA	3.70	3.89	3.94	0.16
U-Net GAN		**2.03**	**2.07**	**2.08**	**0.04**

Table 3.6: Best, median, mean, and std of FID values across 5 runs.

across 5 independent runs. From Figure 3.4 it is evident that the FID score drops for both models at a similar rate, with a constant offset for the U-Net GAN model, as well as the smaller standard deviation of FID. These results showcase the high potential of the new U-Net based discriminator. For a detailed comparison of the FID mean, median, and standard deviation across 5 runs we refer to Table 3.6.

Qualitative results on FFHQ and COCO-Animals are shown in Figure 3.5 and Figure 3.6. Figure 3.5 displays human faces generated by U-Net GAN through linear interpolation in the latent space between two synthetic samples. We observe that the interpolations are semantically smooth between faces, i.e., an open mouth gradually becomes a closed mouth, hair progressively grows in length, beards or glasses smoothly fade or appear, and hair color changes seamlessly. Furthermore, we notice that on several occasions men appear with pink beards. As FFHQ contains a fair share of people with pink hair, we suspect that our generator extrapolates hair color to beards, enabled by the global and local D^U feedback during the training. Figure 3.6 shows generated samples on COCO-Animals. We observe diverse images of high quality. We further notice that employing the class-conditional projection (as used in BigGAN) in the pixel output space of the decoder does not introduce class leakage or influence the class separation in any other way. These observations confirm that our U-Net GAN is effective in both unconditional and class-conditional image generation.

Ablation study. In Table 3.7 we next analyze the individual effect of each of the proposed components of the U-Net GAN model (see Section 3.2 for details) to the baseline architecture of BigGAN on the FFHQ and COCO-Animals datasets, comparing the median FID scores. Note that each of these individual components builds on each other. As shown in Table 3.7, employing the U-Net architecture for the discriminator alone improves the median FID score from 12.42 to 10.86 for FFHQ and 16.55 to 15.86 for COCO-Animals. Adding the CutMix augmentation improves upon these scores even further, achieving an FID of 10.30 for FFHQ and 14.95 for COCO-Animals. Note that we observe a similar improvement if we employ the CutMix augmentation during the BigGAN training as well. Employing the proposed consistency regularization in the segmenter D^U_{dec} output space on the CutMix images enables us to get the most out of the CutMix augmentation

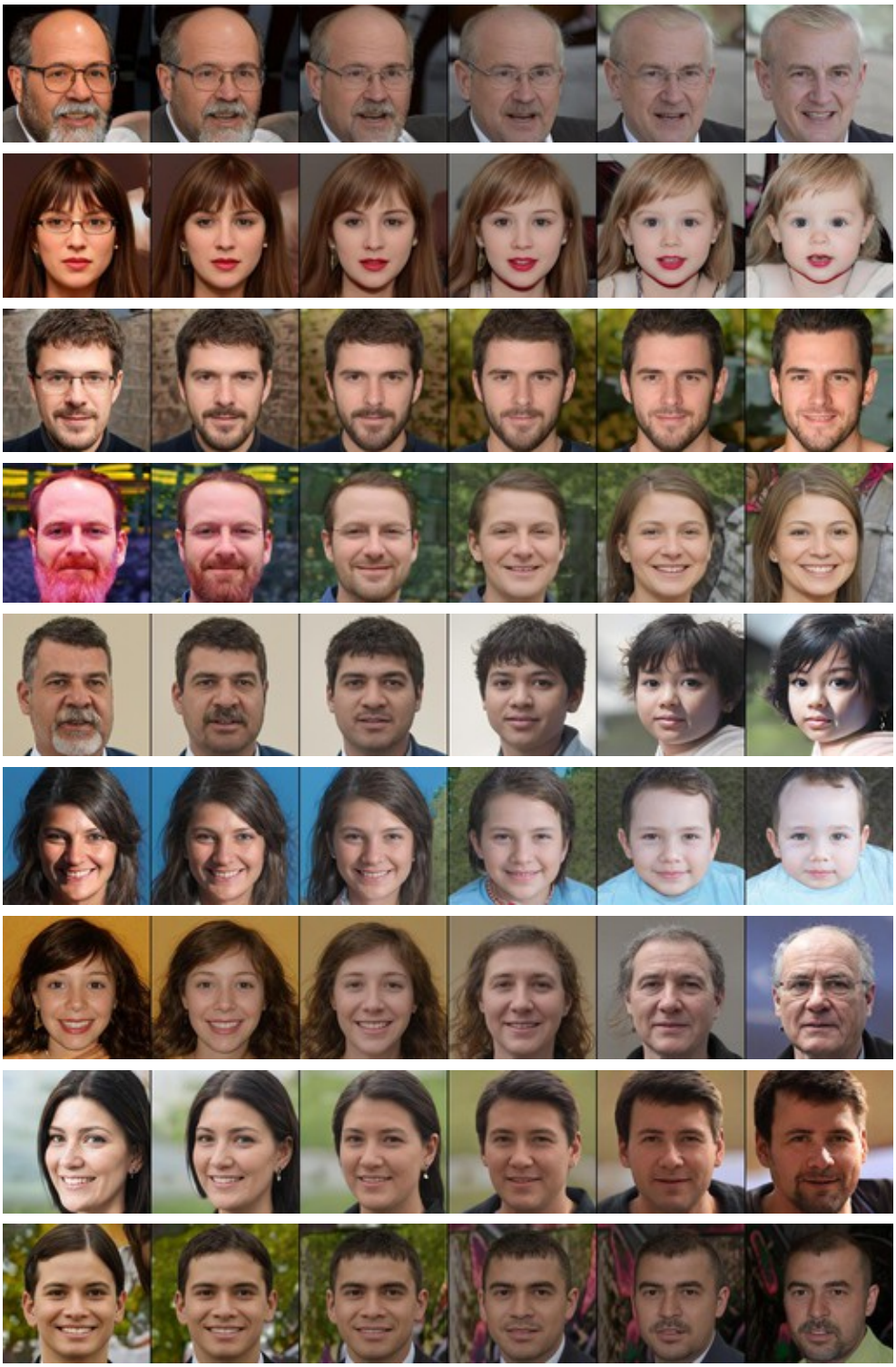

Figure 3.5: Images generated with U-Net GAN trained on FFHQ with resolution 256×256 when interpolating in the latent space between two synthetic samples (left to right). Note the high quality of synthetic samples and very smooth interpolations, maintaining *global* and *local* realism.

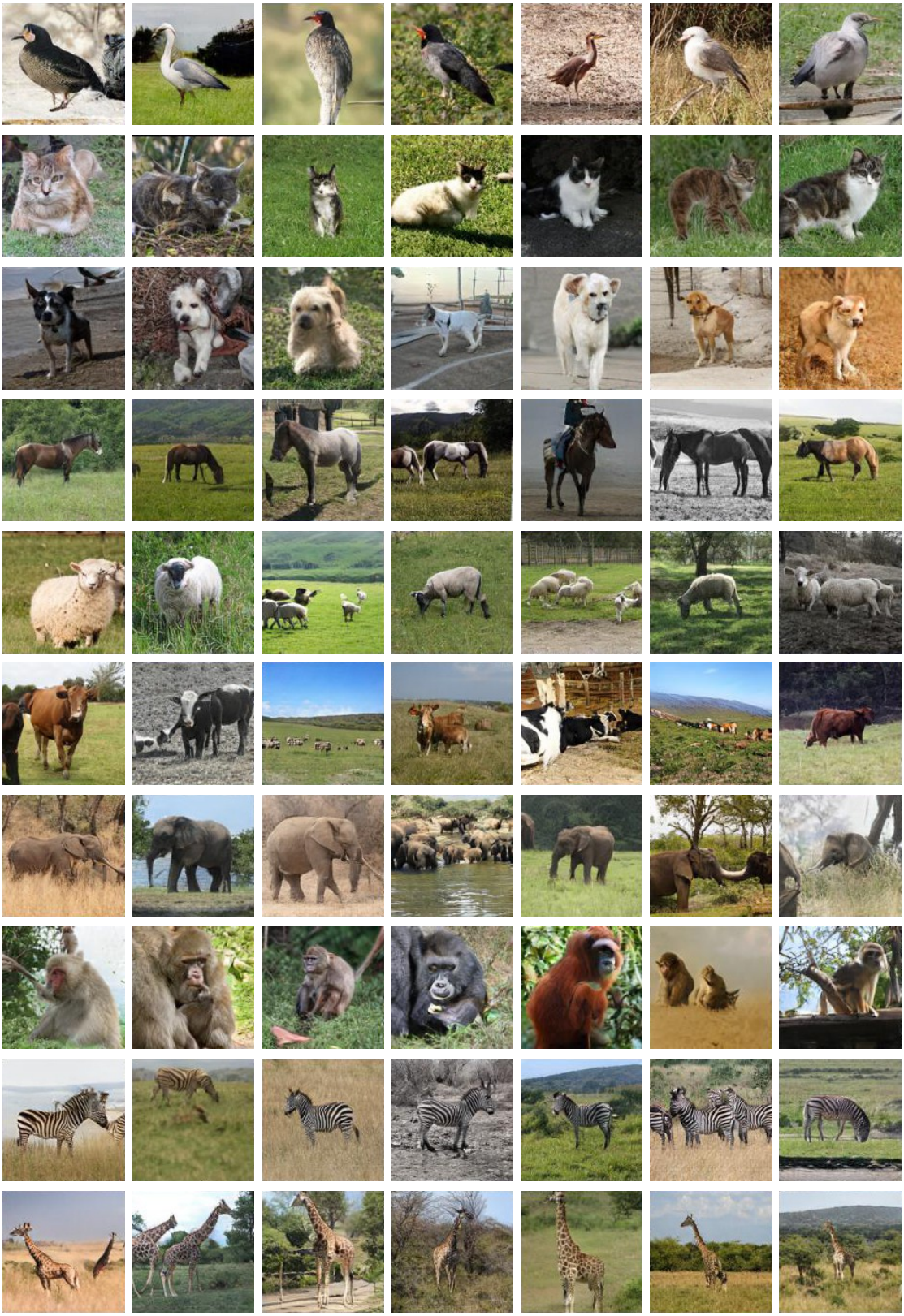

Figure 3.6: Images generated with U-Net GAN trained on COCO-Animals with resolution 128×128.

and allows us to better leverage the per-pixel feedback of the U-Net discriminator, without imposing much computational or memory cost. In effect, the median FID score drops to 7.63 for FFHQ and to 13.87 for COCO-Animals. Overall, we observe that each proposed component of the U-Net GAN model leads to improved performance in terms of FID. Lastly, in Figure 3.7 we present a qualitative comparison of uncurated images generated with the unconditional BigGAN model (Brock et al., 2019) and our U-Net GAN. The images generated by U-Net GAN exhibit finer details and maintain better local realism.

Method	COCO-Animals	FFHQ
BigGAN (Brock et al., 2019)	16.55	12.42
U-Net based discriminator	15.86	10.86
+ CutMix augmentation	14.95	10.30
+ Consistency regularization	**13.87**	**7.63**

Table 3.7: Ablation study of the U-Net GAN model on FFHQ and COCO-Animals. Shown are the median FID scores. The proposed components lead to better performance, on average improving the median FID by 3.7 points over BigGAN.

Comparison with state of the art. Table 3.8 shows that U-Net GAN compares favorably with the state of the art on the CelebA dataset. The BigGAN baseline already outperforms COCO-GAN, the best result reported in the literature to the best of our knowledge, lowering FID from 5.74 to 4.54, whereas U-Net GAN further improves FID to 2.95. FID scores for CelebA were computed with the standard TensorFlow Inception network for comparability, resulting in slightly different numbers compared to Table 3.6. The PyTorch and TensorFlow FIDs for all datasets are presented in Table 3.9. It is worth noting that BigGAN is the representative of just one of the two well-known state-of-the art GAN families, led by BigGAN and StyleGAN, and their respective further improvements (Zhang et al., 2020a; Zhao et al., 2020b; Karras et al., 2020c). While in this work we base our framework on BigGAN, it would be interesting to also explore the application of the U-Net based discriminator for the StyleGAN family.

Method	FID ↓	IS ↑
PG-GAN (Karras et al., 2018)	7.30	–
COCO-GAN (Lin et al., 2019)	5.74	–
BigGAN (Brock et al., 2019)	4.54	3.23
U-Net GAN	**2.95**	**3.43**

Table 3.8: Comparison with the state-of-the-art models on CelebA (128×128).

BigGAN

U-Net GAN

Figure 3.7: Qualitative comparison of uncurated images generated with the unconditional BigGAN model (top) and our U-Net GAN (bottom) on FFHQ with resolution 256×256. Note that the images generated by U-Net GAN exhibit finer details and maintain better local realism.

Dataset	Method	PyTorch FID ↓	PyTorch IS ↑	TensorFlow FID ↓	TensorFlow IS ↑
FFHQ	BigGAN (Brock et al., 2019)	11.48	3.97	14.92	3.96
(256 × 256)	U-Net GAN	**7.48**	**4.46**	**8.88**	**4.50**
COCO-Animals	BigGAN (Brock et al., 2019)	16.37	11.77	16.42	11.34
(128 × 128)	U-Net GAN	**13.73**	**12.29**	**13.96**	**11.77**
	PG-GAN (Karras et al., 2018)	–	–	7.30	–
CelebA	COCO-GAN (Lin et al., 2019)	–	–	5.74	–
(128 × 128)	BigGAN (Brock et al., 2019)	3.70	3.08	4.54	3.23
	U-Net GAN	**2.03**	**3.33**	**2.95**	**3.43**

Table 3.9: Evaluation results on FFHQ, COCO-Animals, and CelebA with PyTorch and TensorFlow FID/IS scores. The difference lies in the choice of framework in which the inception network is implemented, which is used to extract the inception metrics.

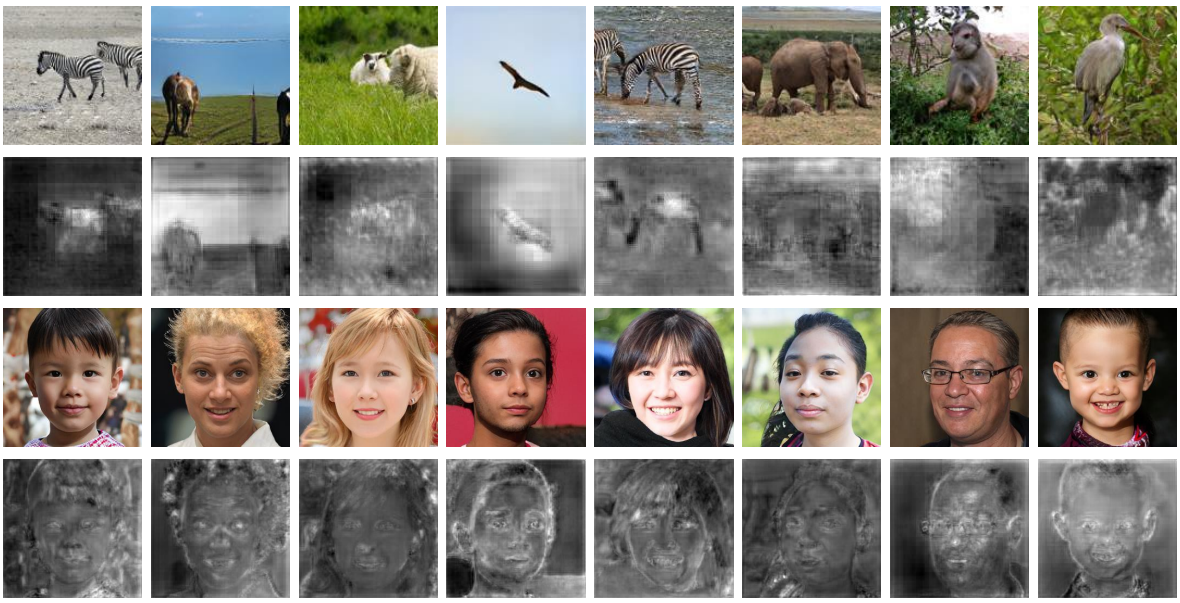

Figure 3.8: Generated samples and the corresponding U-Net decoder predictions for COCO-Animals (row 1 & 2) and FFHQ (row 3 & 4). Brighter areas correspond to the discriminator confidence of pixels being real (and darker of being fake).

Discriminator response visualization. The heterogenous per-pixel predictions of the decoder are visualized in Figure 3.8. For both COCO-Animals and FFHQ it can be seen that differently textured regions evoke different decoder predictions. Yet, the real-fake predictions do not follow texture boundaries strictly, indicating that the decoder predictions are based on learned patterns at different scales. In effect, the U-Net decoder provides a very detailed and spatially coherent response, which enables the generator to further improve the image quality.

Importantly, we observe that encoder (D^U_{enc}) and decoder (D^U_{dec}) often assign different real/fake scores per sample. Figure 3.9 visualizes the per-sample predictions for a complete training batch. Here, the decoder score is computed as the average per-pixel prediction. The scores correlate with each other but have a high variance. The points in the upper left quadrant correspond to samples that are assigned a high probability of being real by the decoder, but a low probability by the encoder. This implies realism on a local level, but not necessarily on a global one. Similarly, the lower right quadrant represents samples that are identified as realistic by the encoder, but contain unrealistic patches which cause a low decoder score. The fact that the encoder and decoder predictions are not tightly coupled further implies that these two components are complementary. In other words, the generator receives more pronounced feedback by the proposed U-Net discriminator than it would get from a standard GAN discriminator.

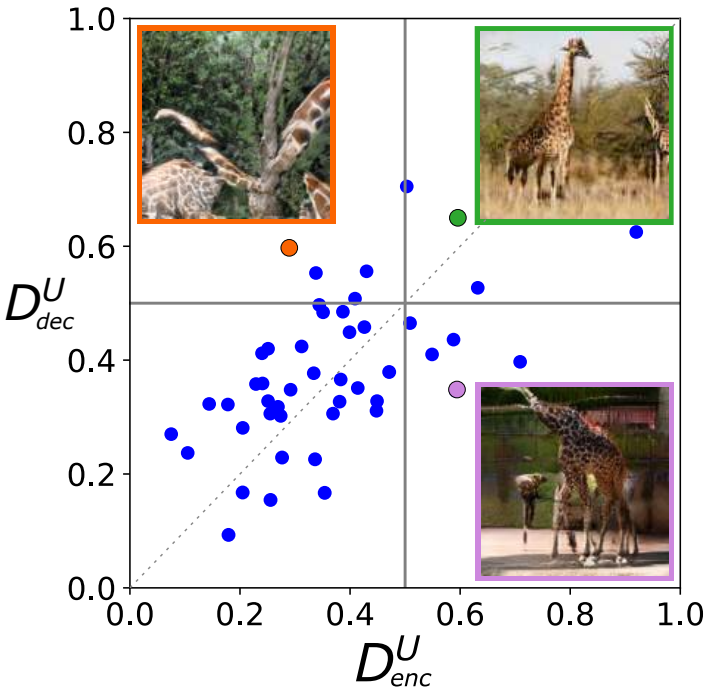

Figure 3.9: Visualization of the predictions of the encoder D^U_{enc} and decoder D^U_{dec} modules during training, within a batch of 50 generated samples. For visualization purposes, the D^U_{dec} score is averaged over all pixels in the output. Note that quite often decisions of D^U_{enc} and D^U_{dec} are not coherent with each other. As judged by the U-Net discriminator, samples in the upper left consist of locally plausible patterns, while not being globally coherent (example in orange), whereas samples in the lower right look globally coherent but have local inconsistencies (example in purple: giraffe with too many legs and vague background).

Characterizing the training dynamics. Both BigGAN and U-Net GAN experience similar stability issues, with ~ 60% of all runs being successful. For U-Net GAN, training collapse occurs generally much earlier (~ 30k iterations) than for BigGAN

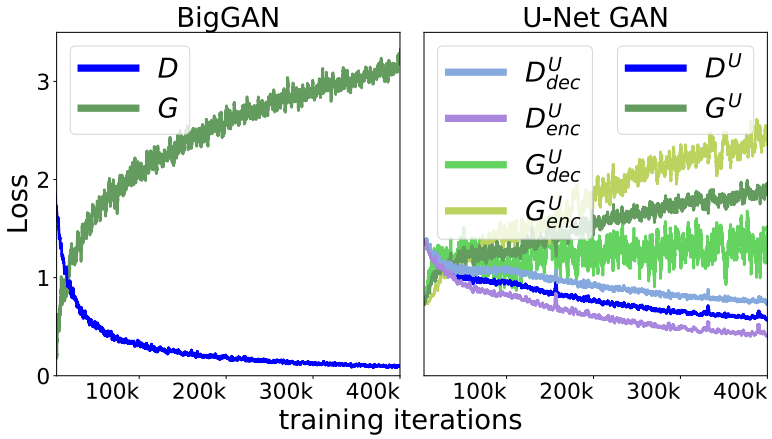

Figure 3.10: Comparison of the generator and discriminator loss behavior over training for U-Net GAN and BigGAN. The generator and discriminator loss of U-Net GAN is additionally split up into its encoder- and decoder components.

(> 200k iterations, as also reported in Brock et al. (2019)), allowing to discard failed runs earlier. Among successful runs for both models, we observe a lower standard deviation in the achieved FID scores, compared to the BigGAN baseline (see Table 3.6). Figure 3.10 depicts the evolution of the generator and discriminator losses (green and blue, respectively) for U-Net GAN and BigGAN over training. For U-Net GAN, the generator and discriminator losses are additionally split into the loss components of the U-Net encoder D^U_{enc} and decoder D^U_{dec}. The U-Net GAN discriminator loss decays slowly, while the BigGAN discriminator loss approaches zero rather quickly, which prevents further learning from the generator. This explains the FID gains of U-Net GAN and shows its potential to improve with longer training. The generator and discriminator loss parts from encoder (image-level) and decoder (pixel-level) show similar trends, i.e., we observe the same decay for D^U_{enc} and D^U_{dec} losses but with different scales. This is expected as D^U_{enc} can easily classify an image as belonging to the real or fake class just by looking at one distinctive trait, while to achieve the same scale D^U_{dec} needs to make a uniform real or fake decision on all image pixels.

3.4 Conclusion

In this work, we propose an alternative U-Net based architecture for the discriminator, which allows to provide both global and local feedback to the generator. In addition, we introduce a consistency regularization technique for the U-Net discriminator based on the CutMix data augmentation. We show that all the proposed changes result in a stronger discriminator, enabling the generator to synthesize images with varying levels of detail, maintaining global and local realism. We demonstrate the improvement over the state-of-the-art BigGAN model (Brock

et al., 2019) in terms of the FID score on three different datasets.

Compared to unconditional and class-conditional image synthesis, other image synthesis tasks can benefit even more from detailed local discriminator feedback. One such task is semantic image synthesis, where a GAN generates images for a given label map. We have already shown that a U-Net discriminator can be class-conditional. However, the decoder loss is computed per pixel, and therefore the losses of different pixels can be conditioned on different classes. This is useful for semantic image synthesis, where different pixels are assigned to different classes. Hence, in Chapter 4 we extend the U-Net discriminator approach for the task of semantic image synthesis and introduce further innovations to improve over previous state-of-the-art models.

4 Semantic Image Synthesis with Only Adversarial Supervision

Contents

4.1 Introduction 54

4.2 The OASIS model 59

4.2.1 The SPADE baseline 59

4.2.2 The OASIS discriminator 61

4.2.3 The OASIS generator 63

4.2.4 Superfluity of the perceptual loss for OASIS 65

4.3 Experiments 65

4.3.1 Experimental setup 66

4.3.2 Evaluation of the synthesis quality and diversity 68

4.3.3 Synthesis performance on underrepresented classes 71

4.3.4 Image editing with OASIS 74

4.3.5 Synthetic data augmentation 76

4.3.6 Ablations 77

4.4 Conclusion 81

In Chapter 3 we demonstrated how a U-Net based discriminator could improve image synthesis in unconditional and class-conditional GANs. In this chapter, we focus on the more complex task of synthesizing images from semantic label maps, known as semantic image synthesis. Despite their recent successes, GANs for semantic image synthesis still suffer from poor image quality when trained with only adversarial supervision. Previously, additionally employing a VGG-based perceptual loss has helped to overcome this issue, significantly improving the synthesis

quality, but at the same time limited the progress of GAN models for semantic image synthesis. In this chapter, we propose a novel, simplified GAN model, which needs only adversarial supervision to achieve high-quality results. Inspired by the approach presented in Chapter 3, we redesign the discriminator as a semantic segmentation network, and directly use the given semantic label maps as the ground truth for training. By providing stronger supervision to the discriminator as well as to the generator through spatially- and semantically-aware discriminator feedback, we are able to synthesize images of higher fidelity and with a better alignment to their input label maps, making the use of the perceptual loss superfluous. Furthermore, we enable high-quality multi-modal image synthesis through global and local sampling of a 3D noise tensor injected into the generator, which allows complete or partial image editing. We show that images synthesized by our model are more diverse and follow the color and texture distributions of real images more closely. We achieve a strong improvement in image synthesis quality over prior state-of-the-art models across the commonly used ADE20K, Cityscapes, and COCO-Stuff datasets using only adversarial supervision. In addition, we investigate semantic image synthesis under severe class imbalance and sparse annotations, which are common aspects in practical applications but were overlooked in prior works. To this end, we evaluate our model on LVIS, a dataset originally introduced for long-tailed object recognition. We thereby demonstrate high performance of our model in the sparse and unbalanced data regimes, achieved by means of the proposed 3D noise and the ability of our discriminator to balance class contributions directly in the loss function. Our code and pretrained models are available at `https://github.com/boschresearch/OASIS`. The content of this chapter corresponds to the ICLR 2021 paper "You only need adversarial supervision for semantic image synthesis" (Schönfeld et al., 2021) and its extended version published at IJCV 2022 (Sushko et al., 2022).

Chapter 5 exploits the increased diversity and the ability to locally manipulate objects, which result from the work in this chapter. Thanks to the increased diversity, a semantic image synthesis GAN can form an interpretable latent space. Moreover, interpretable class-specific latent space directions can emerge due to the ability to manipulate objects locally. In Chapter 5, we present an algorithm to find such directions and use them to edit images in a meaningful way.

4.1 Introduction

Conditional generative adversarial networks (GANs) (Mirza and Osindero, 2014) synthesize images conditioned on class labels (Brock et al., 2019; Casanova et al., 2021), text (Reed et al., 2016; Zhang et al., 2018a, 2021), other images (Isola et al., 2017; Huang et al., 2018; Park et al., 2020), or semantic label maps (Park et al., 2019b; Liu et al., 2019; Wang et al., 2021c). In this work, we focus on the latter,

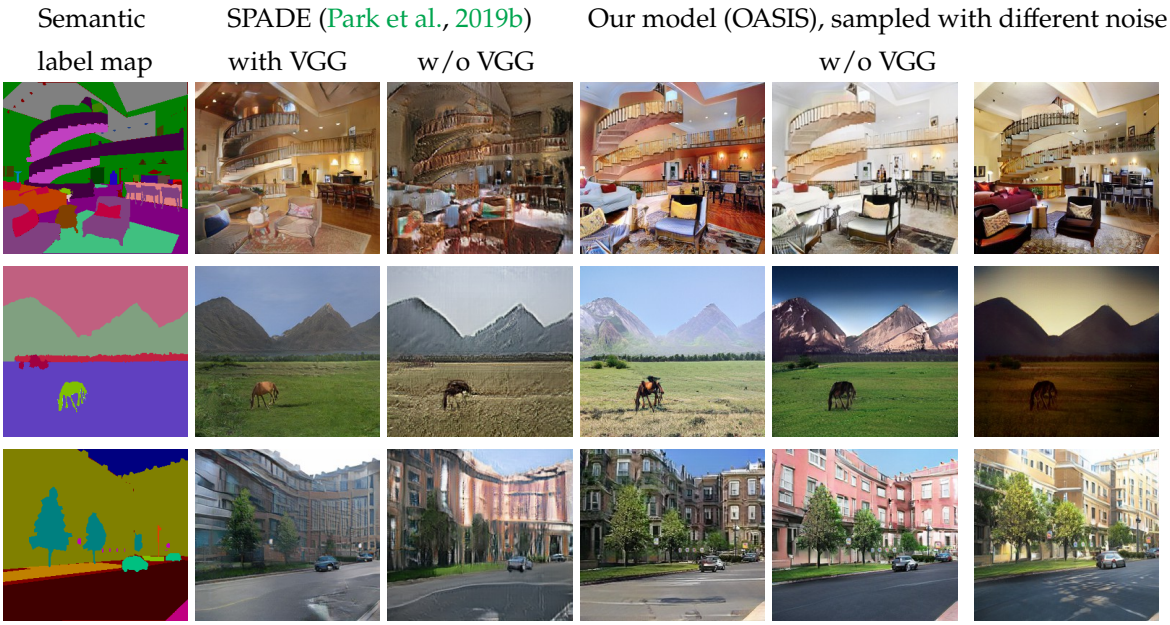

Figure 4.1: Existing semantic image synthesis models heavily rely on the VGG-based perceptual loss to improve the quality of generated images. In contrast, our model (OASIS) can synthesize diverse and high-quality images while only using an adversarial loss, without any external supervision.

addressing semantic image synthesis. Taking pixel-level annotated semantic maps as input, semantic image synthesis enables the rendering of realistic images from user-specified layouts, without the use of an intricate graphics engine. Therefore, its applications range widely from content creation and image editing to producing training data for downstream applications that adhere to specific semantic requirements (Park et al., 2019a; Ntavelis et al., 2020).

Despite the recent progress on stabilizing GANs (Miyato et al., 2018; Zhang and Khoreva, 2019; Karras et al., 2020a; Sauer et al., 2021) and developing their architectures (Karras et al., 2021b, 2019a, 2020c; Brock et al., 2019; Liu et al., 2021), state-of-the-art GAN-based semantic image synthesis models (Park et al., 2019b; Liu et al., 2019; Wang et al., 2021c) still greatly suffer from training instabilities and poor image quality when the generator is only trained to fool the discriminator in an adversarial fashion (see Fig. 4.1). An established practice to overcome this issue is to employ a perceptual loss (Wang et al., 2018a) to train the generator, in addition to the discriminator loss. The perceptual loss aims to match intermediate features of synthetic and real images, that are estimated via an external perception network. A popular choice for such a network is VGG (Simonyan and Zisserman, 2015), pre-trained on ImageNet (Deng et al., 2009). Although the perceptual loss substantially improves the performance of previous methods, it comes with the computational overhead introduced by utilizing an extra network for training. Moreover, as we show in our experiments, it dominates over the adversarial loss during training, as

Figure 4.2: OASIS multi-modal synthesis results. The 3D noise can be sampled globally (first 2 rows), changing the whole scene, or locally (last 2 rows), partially changing the image. For the latter, we sample different noise per region, like the bed segment (in red) or arbitrary areas defined by shapes.

the generator starts to learn mostly through minimizing the VGG loss, which has a negative impact on the diversity and quality of generated images. Therefore, in this work we propose a novel, simplified model that establishes new state-of-the-art results without requiring a perceptual loss.

To achieve semantic image synthesis of high quality, the training signal to the GAN generator should contain feedback on whether the generated images are well aligned to the input label maps. Thus, a fundamental question for GAN-based semantic image synthesis models is how to design the discriminator that would efficiently utilize information from given semantic label maps, in addition to judging the realism of given images. Conventional methods (Park et al., 2019b; Wang et al., 2018a; Liu et al., 2019; Isola et al., 2017; Wang et al., 2021c; Ntavelis et al., 2020) adopt a multi-scale classification network, taking the label map as input along with the image, and making a global image-level real/fake decision. This discriminator has limited representation power, as it is not incentivized to learn high-fidelity pixel-level details of the images and their precise alignment with the input semantic label maps. For example, such a classification-based discriminator can base its decision solely on image realism, without the need of examining the alignment between the image and label map. To mitigate this issue, we propose an alternative architecture for the discriminator, redesigning it as an encoder-decoder

semantic segmentation network (Ronneberger et al., 2015), and directly exploiting the given semantic label maps as ground truth via an (N+1)-class cross-entropy loss. This new discriminator provides semantically-aware pixel-level feedback to the generator, partitioning the image into segments belonging to one of the N real semantic classes or the fake class. With this design, the network cannot ignore the provided label maps, as it has to predict a correct class label for each pixel of an image. Enabled by the discriminator per-pixel response, we further introduce a LabelMix regularization, which fosters the discriminator to focus more on the semantic and structural differences of real and synthetic images. The proposed changes lead to a much stronger discriminator, that maintains a powerful semantic representation of objects, giving more meaningful feedback to the generator, and thus making the perceptual loss supervision superfluous (see Fig. 4.1).

Semantic image synthesis is naturally a one-to-many mapping, where one label map can correspond to many possible real images. Thus, a desirable property of a generator is to generate a diverse set of images from a single label map, only by sampling noise. This property is known as multi-modality. Previously, only using a noise vector as input was not sufficient to achieve multi-modality, because the generator tended to mostly ignore the noise or synthesized images of poor quality (Isola et al., 2017; Wang et al., 2018a). Thus, prior work (Wang et al., 2018a; Park et al., 2019b) resorted to using an image encoder to produce multi-modal outputs. In this work, we enable multi-modal synthesis of the generator via a newly-introduced 3D noise sampling method, without requiring an image encoder and not relying on availability of a reference image to produce new image styles. Empowered by our stronger discriminator, the generator can now effectively synthesize different images by simply resampling a 3D noise tensor, which is used not only as the input, but is also combined with intermediate features via conditional normalization at every layer. This procedure makes the generator spatially sensitive to noise, so we can resample it both globally (channel-wise) and locally (pixel-wise), allowing to change not only the appearance of the whole scene, but also of specific semantic classes or any chosen area (see Fig. 4.2). As shown in our experiments, the proposed 3D noise injection scheme enables a significantly higher diversity of synthesis compared to previous methods.

With the proposed modifications in the discriminator and generator design, we outperform the prior state of the art in synthesis quality across the commonly used datasets ADE20K (Zhou et al., 2017), COCO-Stuff, (Caesar et al., 2018), and Cityscapes (Cordts et al., 2016). Omitting the necessity of the VGG perceptual loss, our model generates samples of higher quality and diversity, and follows the color and texture distributions of real images more closely.

A well-known challenge for semantic segmentation applications is the problem of class imbalance. In practice, a dataset can contain underrepresented classes (representing a very small fraction of the dataset pixels), which can lead to suboptimal performance of models (Sudre et al., 2017). However, to the best of our knowledge,

this problem has not been studied in the context of semantic image synthesis. For this reason, we propose to extend the evaluation setup used in previous works by using the highly imbalanced LVIS dataset (Gupta et al., 2019). Originally introduced as a dataset for long-tailed object recognition, LVIS contains a large set of 1203 classes, the majority of which appear only in a few images. Moreover, to simplify dataset curation, label maps in LVIS were annotated sparsely, with large image areas being occupied with a generic background label. The above properties make LVIS a very challenging evaluation setting for previous semantic image synthesis models, as we demonstrate by the example of the state-of-the-art SPADE model (Park et al., 2019b). As the classification-based discriminator of SPADE makes a global real/fake decision for each image-label pair, the loss contribution originating from underrepresented classes can be dominated by the loss contribution of well represented classes. In contrast, our proposed discriminator mitigates this issue: with the $(N+1)$-class cross-entropy loss computed for each image pixel, it becomes possible to assign higher weights for the pixels belonging to underrepresented classes. As shown in our experiments, our model successfully deals with both the extreme class imbalance and sparsity in label maps, outperforming SPADE on the LVIS dataset by a large margin.

To extend the evaluation of our model further, we test the efficacy of generated images when applied as synthetic data augmentation for the training of semantic segmentation networks. This way, the performance of semantic image synthesis is assessed through a task that holistically requires high image quality, diversity, and precise image alignment to the label maps. We demonstrate that the synthetic data produced by our model achieves high performance on this test, eliciting a notable increase in downstream segmentation performance. In doing so, our model outperforms a strong baseline SPADE (Park et al., 2019b), indicating its high potential to be applied in segmentation applications. In addition, we also demonstrate how our model for the first time enables the application of a GAN-based semantic image synthesis model to unlabelled images, without requiring external segmentation networks. Thanks to a good segmentation performance of our trained discriminator, we can infer the label map of an image and generate many alternative versions of the same scene by varying the 3D noise. We find these results promising for future utilization of our model in applications.

We call our model OASIS, as it needs **o**nly **a**dversarial supervision for **s**emantic **i**mage **s**ynthesis. In summary, our main contributions include:

- We propose a novel segmentation-based discriminator architecture, that gives more powerful feedback to the generator and eliminates the necessity of the perceptual loss supervision.
- We present a simple 3D noise sampling scheme, notably increasing the diversity of multi-modal synthesis and enabling both complete or partial resampling of a generated image.

- With the OASIS model, we achieve high-quality results on the ADE20K, Cityscapes, and COCO-Stuff datasets, outperforming previous state-of-the-art models while relying only on adversarial supervision. We show that images synthesized by OASIS exhibit much higher diversity and more closely follow the color and texture distributions of real images.

- We propose to use the LVIS dataset (Gupta et al., 2019) to assess image generation in the regime with many underrepresented semantic classes, leading to a severe class imbalance. We show how the OASIS design directly addresses these issues and thereby outperforms the strong baseline SPADE (Park et al., 2019b) by a large margin.

- We test the efficacy of generated images for synthetic data augmentation, as a unified measure that simultaneously depends on image quality, diversity, and label map alignment. The images generated by OASIS elicit a stronger increase in downstream segmentation performance compared to SPADE, suggesting a higher potential of our model for future utilization in applications.

The rest of this chapter is organized as follows: In section 4.2 we explain the OASIS model and its key innovation compared to the previous state-of-the-art baseline. Section 4.3 presents an elaborate qualitative and quantitative analysis of OASIS. Lastly, section 4.4 summarizes this chapter.

4.2 The OASIS model

In this section, we present our OASIS model, which, in contrast to other semantic image synthesis methods, needs only adversarial supervision for training. Using SPADE as a starting point (Sec. 4.2.1), we first propose to redesign the discriminator as a semantic segmentation network, directly using the given semantic label maps as ground truth (Sec. 4.2.2). Empowered by spatially- and semantically-aware feedback of the new discriminator, we next redesign the SPADE generator, enabling its effective multi-modal synthesis via 3D noise sampling (Sec. 4.2.3). Lastly, we illustrate the superfluity of the VGG loss for our model (Sec. 4.2.4).

4.2.1 The SPADE baseline

We choose SPADE as our baseline as it is a state-of-the-art model and a relatively simple representative of conventional semantic image synthesis models. As depicted in Fig. 4.3, the discriminator of SPADE largely follows the PatchGAN multi-scale discriminator (Isola et al., 2017), adopting two image classification networks operating at different resolutions. Both of them take the channel-wise concatenation of the semantic label map and the real/fake image as input, and produce

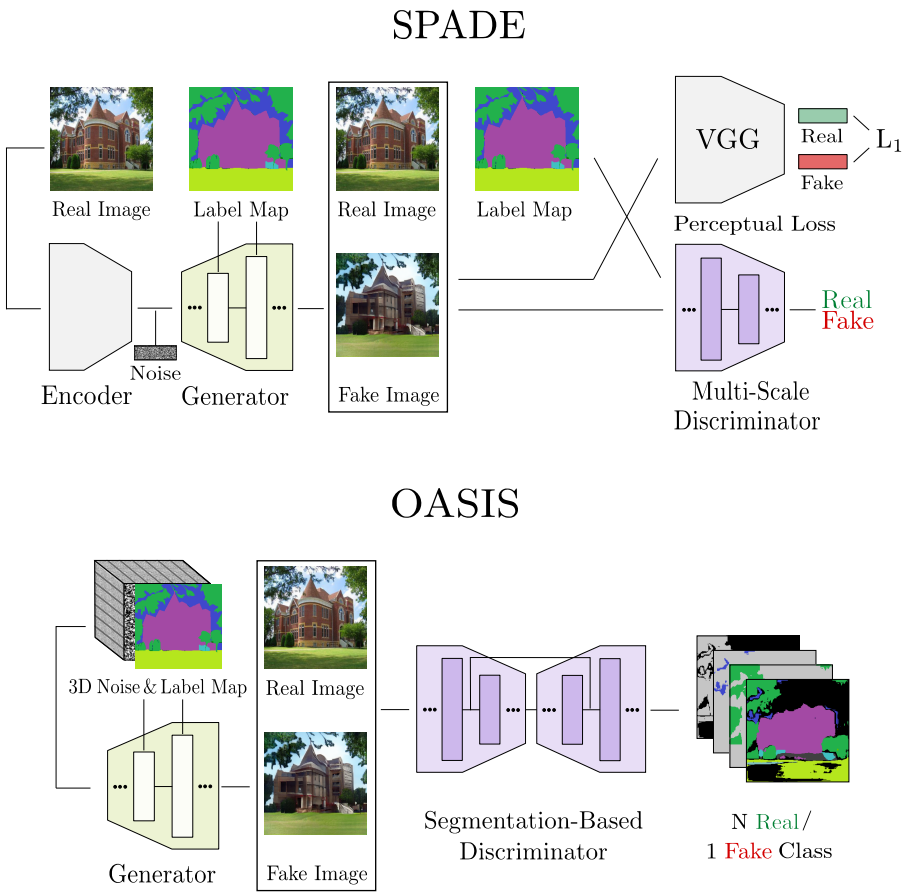

Figure 4.3: SPADE (left) vs. OASIS (right). OASIS outperforms SPADE, while being simpler and lighter: it uses only an adversarial loss as supervision and a single segmentation-based discriminator, without relying on heavy external networks. Furthermore, OASIS learns to synthesize multi-modal outputs by directly resampling the 3D noise tensor, instead of using an image encoder as in SPADE.

real/fake classification scores. On the generator side, SPADE adopts spatially-adaptive normalization layers to effectively integrate the semantic label map into the synthesis process from low to high scales. Additionally, the image encoder is used to extract the style vector from the reference image, which is then combined with a 1D noise vector for multi-modal synthesis. The training loss of SPADE consists of three terms, namely, an adversarial loss, a feature matching loss, and the VGG-based perceptual loss:

$$\mathcal{L} = \max_{G} \min_{D} \mathcal{L}_{\text{adv}} + \lambda_{\text{fm}} \mathcal{L}_{\text{fm}} + \lambda_{\text{vgg}} \mathcal{L}_{\text{vgg}}. \tag{4.1}$$

Overall, SPADE is a resource-demanding model at both training and test time, i.e., with two PatchGAN discriminators, an image encoder in addition to the generator, and the VGG loss. In the following, we revisit its architecture and introduce a simpler and more efficient solution that offers better performance and reduces the model complexity.

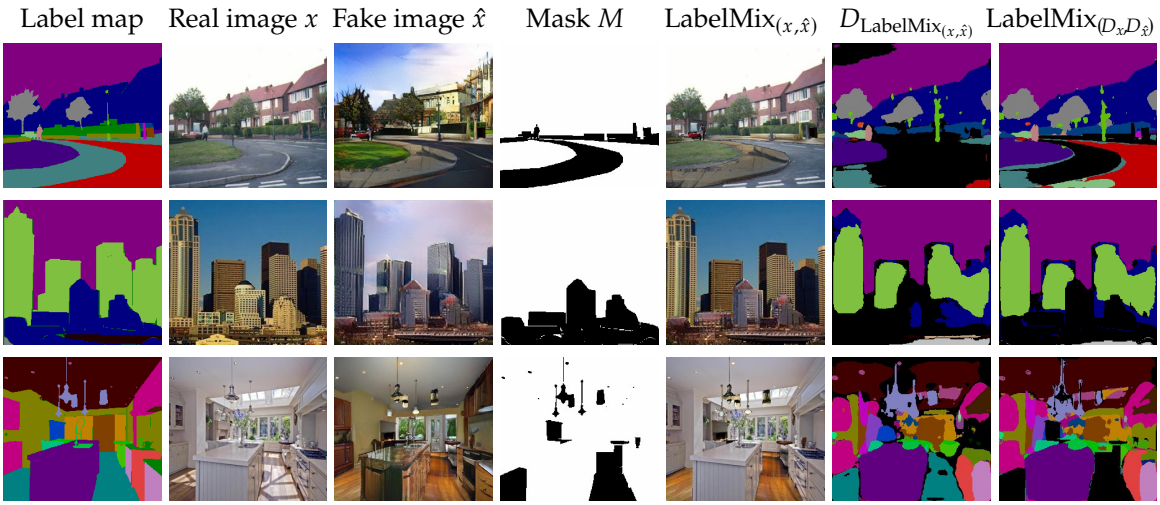

Figure 4.4: LabelMix regularization. Real x and fake $\hat{x}$ images are mixed using a binary mask M, sampled based on the label map, resulting in LabelMix$_{(x,\hat{x})}$. The consistency regularization minimizes the L2 distance between the logits of $D_{\text{LabelMix}_{(x,\hat{x})}}$ and LabelMix$_{(D_x,D_{\hat{x}})}$. In this visualization, **black** corresponds to the fake class in the N+1 segmentation output.

4.2.2 The OASIS discriminator

To train the generator to synthesize high-quality images that are well aligned with the input semantic label maps, we need a powerful discriminator that coherently captures discriminative semantic features at different image scales. While classification-based discriminators, such as PatchGAN, take label maps as input concatenated to images, they can afford to ignore them and make the decision solely on image patch realism. Thus, we propose to cast the discriminator task as a multi-class semantic segmentation problem to directly utilize label maps for supervision, and accordingly alter its architecture to an encoder-decoder segmentation network (see Fig. 4.3). Encoder-decoder networks have been proven to be effective for semantic segmentation (Badrinarayanan et al., 2016; Chen et al., 2018a). Thus, we build our discriminator architecture upon U-Net (Ronneberger et al., 2015), which consists of the encoder and decoder connected by skip connections. This discriminator architecture is multi-scale through its design, integrating information over up- and down-sampling pathways as well as through the encoder-decoder skip connections. The segmentation task of the discriminator is formulated to predict the per-pixel class label of the real images, using the given semantic label maps as ground truth. In addition to the N semantic classes from the label maps, all pixels of fake images are categorized as one extra class. As the formulated semantic segmentation problem has $N + 1$ classes, we propose to use an (N+1)-class cross-entropy loss for training.

In practice, the N semantic classes are often imbalanced, as some of the classes represent significantly less pixels of the dataset compared to others. The loss con-

tribution for such underrepresented classes can be dominated by well-represented classes, which can lead to suboptimal performance. To mitigate this issue, empowered by the pixel-level loss computation of our discriminator, we propose to weight each class by its inverse pixel-wise frequency in a batch, thus giving underrepresented semantic classes more weight. In doing so, the loss contributions of each class are equally balanced, and thus the generator is also encouraged to pay more attention to underrepresented classes. Mathematically, the new discriminator loss is expressed as:

$$\begin{aligned}\mathcal{L}_D = &- \mathbb{E}_{(x,t)}\left[\sum_{c=1}^{N}\alpha_c \sum_{i,j}^{H\times W} t_{i,j,c}\log D(x)_{i,j,c}\right] \\ &- \mathbb{E}_{(z,t)}\left[\sum_{i,j}^{H\times W}\log D(G(z,t))_{i,j,c=N+1}\right],\end{aligned} \tag{4.2}$$

where x denotes the real image; (z,t) is the noise-label map pair used by the generator G to synthesize a fake image; and the discriminator D maps the real or fake image into a per-pixel $(N{+}1)$-class prediction probability. The ground truth label map t has three dimensions, where the first two correspond to the spatial position $(i,j) \in H \times W$, and the third one is a one-hot vector encoding the class $c \in \{1,..,N+1\}$. The class balancing weight α_c is the inverse pixel-wise frequency of a class c per batch:

$$\alpha_c = \frac{H\times W}{\sum_{i,j}^{H\times W} E_t\left[\mathbb{1}[t_{i,j,c}=1]\right]}. \tag{4.3}$$

In effect, improving the synthesis of underrepresented and well-represented classes is equally necessary to minimize the loss. As we show in Sec. 4.3.3, this step helps to improve the synthesis quality of underrepresented classes.

LabelMix regularization. In order to encourage our discriminator to focus on differences in content and structure between the fake and real classes, we propose a LabelMix regularization. Based on the semantic layout, we generate a binary mask M to mix a pair $(x,\hat{x})$ of real and fake images conditioned on the same label map: $\text{LabelMix}(x,\hat{x},M) = M\odot x + (1-M)\odot\hat{x}$, as visualized in Fig. 4.4. Given the mixed image, we further train the discriminator to be equivariant under the LabelMix operation. This is achieved by adding a consistency loss term $\mathcal{L}_{cons}$ to Eq. 4.2:

$$\begin{aligned}\mathcal{L}_{cons} = &\Big\| D_{\text{logits}}\Big(\text{LabelMix}(x,\hat{x},M)\Big) \\ &- \text{LabelMix}\Big(D_{\text{logits}}(x), D_{\text{logits}}(\hat{x}), M\Big)\Big\|^2,\end{aligned} \tag{4.4}$$

where D_{logits} are the logits attained before the last softmax activation layer, and $\|\cdot\|$ is the L_2 norm. This consistency loss compares the output of the discriminator on the LabelMix image with the LabelMix of its outputs, penalizing the discriminator for inconsistent predictions. LabelMix is different to CutMix (Yun et al., 2019), which randomly samples the binary mask M. A random mask will introduce inconsistency between the pixel-level labels and the scene layout provided by the label map. For an object with the class label c, it will contain pixels from both real and fake images, resulting in two labels, i.e., c and $N+1$. To avoid such inconsistency, the mask of LabelMix is generated according to the label map, providing natural borders between semantic regions, see Mask M in Fig. 4.4. Under LabelMix regularization, the generator is encouraged to respect the natural semantic boundaries, improving pixel-level realism while also considering the class segment shapes.

Alternative ways to encode label maps. Besides the proposed (N+1)-class cross-entropy loss, there are other ways to incorporate a label map into the training of a segmentation-based discriminator. One can concatenate the label map to the input image, analogously to SPADE. Another option is to use projection, by taking the inner product between the last linear layer output and the embedded label map, analogous to class-label conditional GANs (Miyato and Koyama, 2018). For both alternatives, the training loss is the pixel-level real/fake binary cross-entropy introduced in Chapter 3 (Schönfeld et al., 2020). As in these two variants the label maps are used as input to the discriminator (concatenated to the input image or fed to the last linear layer), they are propagated *forward* through the network. In contrast, the (N+1)-setting uses label maps only as targets for the loss computation, so they are propagated *backward* through the network via the gradient updates. Backward propagation ensures that the discriminator learns semantic-aware features, in contrast to forward propagation, where the alignment of a generated image to the input label map can be ignored. A comparison between the above label map encodings is provided in the ablations section (see Sec. 4.3.6), in particular Table 4.9.

4.2.3 The OASIS generator

To stay in line with the OASIS discriminator design, the training loss for the generator is changed to

$$\mathcal{L}_G = -\mathbb{E}_{(z,t)}\left[\sum_{c=1}^{N} \alpha_c \sum_{i,j}^{H\times W} t_{i,j,c} \log D(G(z,t))_{i,j,c}\right], \tag{4.5}$$

which is a direct outcome of the non-saturation trick (Goodfellow et al., 2014) to Eq. 4.2. We next redesign the generator to enable multi-modal synthesis through noise sampling. SPADE is deterministic in its default setup, but can be trained with

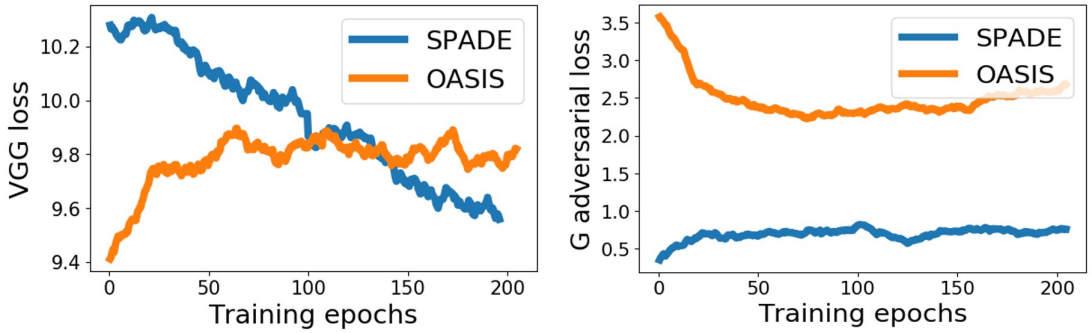

Figure 4.5: VGG and adversarial generator loss functions for SPADE and OASIS trained with VGG loss on ADE20k dataset. The adversarial loss scales are different due to different objectives (binary or (N+1)-class cross entropy loss).

an extra image encoder to generate multi-modal outputs. We introduce a simpler version that enables synthesis of diverse outputs directly from input noise. For this, we construct a noise tensor of size $M \times H \times W$, matching the spatial dimensions of the label map of size $N \times H \times W$, where N is the number of semantic labels and $H \times W$ corresponds to the height and width of the image. Note that for simplicity during training we sample the 3D noise tensor globally, i.e., per-channel, replicating each channel value spatially along the height and width of the tensor. In other words, a M-dimensional latent vector is sampled and then broadcasted to each pixel of an image. We analyze alternative ways of sampling 3D noise during training in the ablation section (see Sec. 4.3.6). After sampling, the noise and the label map are concatenated along the channel dimensions to form a combined noise-label 3D tensor of size $(M+N) \times H \times W$. This combined tensor serves as input to the first generator layer, but also as input to the spatially-adaptive normalization layers in every generator block. This way, all intermediate feature maps are conditioned on both the semantic labels and the noise (see Fig. 4.3), making the noise hard to ignore. As the 3D noise is channel- and pixel-wise sensitive, at test time, one can sample the noise globally, per-channel, and locally, per-segment or per-pixel, for controlled synthesis of the whole scene or of specific semantic objects. For example, when generating a scene of a bedroom, one can resample the noise locally and change the appearance of the bed alone (see Fig. 4.2).

Note that using image styles via an encoder, as in SPADE, is also possible in our setting, as the 3D noise can be simply concatenated to the encoder style features. Lastly, to further reduce the complexity, we remove the first residual block in the generator, reducing the number of parameters from 96M to 72M without a noticeable performance loss (see ablation in Table 4.7).

4.2.4 Superfluity of the perceptual loss for OASIS

In contrast to SPADE, which strongly relies on the perceptual loss during training (see Fig. 4.1), the OASIS generator is trained only with the adversarial loss from the segmentation-based discriminator, according to Eq. 4.5. To illustrate the insignificance of the VGG loss for OASIS, in Fig. 4.5 we compare the curves of the VGG and generator adversarial loss functions of SPADE and OASIS, for comparison additionally trained with the perceptual loss. We see that SPADE focuses on minimizing the VGG loss during training, but keeps the adversarial generator loss constant. Without a rich training signal from its Patch-GAN discriminator, the generator of SPADE resorts to learning mostly from the VGG loss. In contrast, with the stronger discriminator supervision provided by the semantic label maps and the multi-scale U-Net architecture, OASIS achieves a better adversarial balance. Hence, the generator is forced to learn semantically meaningful features that the segmentation-based discriminator judges as real, and the generator loss does not stay constant (see Fig. 4.5).

The advantage of training the generator only with the adversarial loss is threefold. Firstly, the perceptual loss can bias the training signal with the color and texture statistics encoded in the VGG features extracted from ImageNet. As shown in Sec. 4.3.2, the strong adversarial supervision from the OASIS discriminator, without the VGG loss, allows to generate images with color and texture distributions closer to the provided real data. Secondly, the perceptual loss can induce unnecessary constraints on the generator and thus significantly limit the diversity of multi-modal image synthesis. This effect is further demonstrated in Table 4.2. Lastly, removing the perceptual loss eliminates the computational overhead that was introduced by an additional VGG network during training.

4.3 Experiments

We provide an extensive experimental evaluation of our contributions, using the official implementation of SPADE[1] as our baseline. The setup of our experiments is described in detail in Sec. 4.3.1. Firstly, we compare OASIS with prior methods on common semantic image synthesis benchmark datasets, comparing their performance in terms of both image quality and diversity (Sec. 4.3.2). To further highlight the advantages of OASIS over the SPADE baseline, we provide additional discussions on different aspects of semantic image synthesis. In particular, Sec. 4.3.3 is devoted to the performance analysis on the underrepresented classes, extending the comparison of the models to the LVIS dataset (Gupta et al., 2019). Sec. 4.3.4 demonstrates new semantic image editing techniques enabled by OASIS. Sec. 4.3.5 explores the application of generated images as synthetic data augmentation for

[1] github.com/NVlabs/SPADE

the training of semantic segmentation networks. Lastly, we provide an extensive ablation study to verify the effectiveness of the proposed contributions (Sec 4.3.6).

4.3.1 Experimental setup

Datasets. We conduct experiments on several challenging datasets. Firstly, to compare OASIS with prior models, we use the ADE20K (Zhou et al., 2017), COCO-Stuff, (Caesar et al., 2018), and Cityscapes (Cordts et al., 2016), which are the three benchmark datasets commonly used in the semantic image synthesis literature (see Sec. 4.3.2). The image resolution is set to 256x256 for ADE20K and COCO-Stuff, and 256x512 for experiments on Cityscapes. Following Qi et al. (2018), we also evaluate OASIS on ADE20K-outdoors, the subset of ADE20K containing only outdoor scenes.

Secondly, to test the capability of models to learn underrepresented classes, we conduct additional evaluations on the ADE20K and LVIS dataset (Gupta et al., 2019) (see Sec. 4.3.3). We select ADE20K among conventional datasets for its notable class imbalance, as among its 150 classes, more than 86% of the image pixels belong only to the 30 best-represented ones (see Table 4.3). In addition, to test the networks under more extreme class imbalance, we propose to use LVIS, the dataset that has been originally introduced for the task of long-tailed instance segmentation. LVIS employs the same set of training images as COCO-Stuff, but its annotations are different in two important ways. First, LVIS provides a significantly larger set of 1203 annotated classes, following a long-tailed distribution in which some classes are present only in one or a few training samples (see Fig. 4.6). Second, due to a fixed labeling budget, different background types were not considered for annotation in LVIS. Consequently, the images in the LVIS dataset contain large areas belonging to the background class, which sometimes covers more than 90% of the pixels in an image (see gray areas in Fig. 4.10). For the above two reasons, the structure of LVIS poses a new challenge for semantic image synthesis, as models need to account for a much more extreme class imbalance. We conduct experiments on LVIS at the image resolution of 128x128.

Training. We follow the experimental setting of Park et al. (2019b). The Adam (Kingma and Ba, 2015) optimizer was used with momenta $\beta = (0, 0.999)$ and constant learning rates $(0.0001, 0.0004)$ for G and D. We did not use the GAN feature matching loss for OASIS, as we did not observe any improvement with it, and used the VGG loss only for ablations with $\lambda_{\text{VGG}} = 10$. The parameter for LabelMix λ_{LM} was set to 5 for ADE20k and Cityscapes, and to 10 for COCO-Stuff and LVIS. The latent dimension M was set to 64. We did not experience any training instabilities and, thus, did not employ any extra stabilization techniques. All our models use an exponential moving average (EMA) of the generator weights with 0.9999 decay. All the experiments were run on 4 Tesla V100 GPUs, with a batch size of 20 for

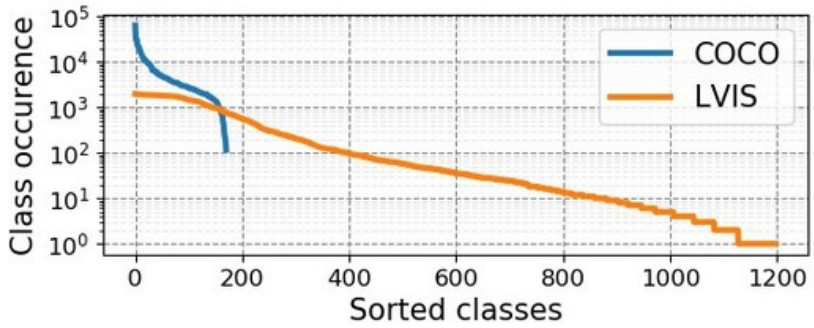

Figure 4.6: Comparison of class distributions of the COCO and LVIS datasets. LVIS has a much larger vocabulary of 1203 classes with a long tail of underrepresented classes.

Cityscapes and 32 for the other datasets. The training epochs are 200 on ADE20K and Cityscapes, and 100 for the larger COCO-Stuff and LVIS datasets. On average, a complete forward-backward pass with batch size 32 on ADE20k takes around 0.95ms per training image.

Evaluation metrics. Following prior work (Park et al., 2019b; Liu et al., 2019), we evaluate the *quality* of semantic image synthesis by computing the FID (Heusel et al., 2017b) and evaluate the *alignment* of the generated images with their semantic label maps via mIoU (mean intersection-over-union) or mAP (mean average precision) on the test set (see Sec. 4.3.2). mIoU evaluates the alignment of generated images with their ground truth label maps, as measured by an external pretrained semantic segmentation network. We use UperNet101 (Xiao et al., 2018) for ADE20K, multi-scale DRN-D-105 (Yu et al., 2017) for Cityscapes, and DeepLabV2 (Chen et al., 2015) for COCO-Stuff. Differently, for the LVIS dataset, the alignment of generated images to ground truth label maps is measured using mAP instead of mIoU, following the official guidelines for evaluating instance segmentation models on this dataset (see Sec. 4.3.3). We compute mAP using a state-of-the-art instance segmentation model from Wang et al. (2021a), pretrained on LVIS.

In addition, to better understand how the perceptual loss influences synthesis performance, we propose to compare the *color and texture statistics* of generated and real images. For this, we compute color histograms in the LAB space and measure the earth mover's distance between the real and generated image sets (Rubner et al., 2000). We also measure the texture similarity to the real data as the χ^2-distance between Local Binary Patterns histograms (Ojala et al., 1996). As different semantic classes have different color and texture distributions, we aggregate the histogram distances separately per class and compute their average.

To measure the *diversity* among synthesized samples in the multi-modal image generation regime, we evaluate MS-SSIM (Wang et al., 2003) and LPIPS (Zhang et al., 2018d) between the images generated from the same label map. For each label map in the test set, we generate 20 images and compute the mean pairwise

Method	# param	VGG	ADE20K		ADE-outd.		Cityscapes		COCO-stuff	
			FID↓	mIoU↑	FID↓	mIoU↑	FID↓	mIoU↑	FID↓	mIoU↑
CRN	84M	✓	73.3	22.4	99.0	16.5	104.7	52.4	70.4	23.7
SIMS	56M	✓	n/a	n/a	67.7	13.1	49.7	47.2	n/a	n/a
Pix2pixHD	183M	✓	81.8	20.3	97.8	17.4	95.0	58.3	111.5	14.6
LGGAN	n/a	✓	31.6	41.6	n/a	n/a	57.7	68.4	n/a	n/a
CC-FPSE	131M	✓	31.7	43.7	n/a	n/a	54.3	65.5	19.2	41.6
SC-GAN	66M	✓	29.3	45.2	n/a	n/a	49.5	66.9	18.1	42.0
SESAME	104M	✓	31.9	**49.0**	n/a	n/a	54.2	66.0	n/a	n/a
SPADE	102M	✓	33.9	38.5	63.3	30.8	71.8	62.3	22.6	37.4
SPADE+	102M	✓	32.9	42.5	51.1	32.1	47.8	64.0	21.7	38.8
		✗	60.7	21.0	65.4	22.7	61.4	47.6	99.1	16.1
OASIS	94M	✗	**28.3**	48.8	**48.6**	**40.4**	**47.7**	**69.3**	**17.0**	**44.1**

Table 4.1: Comparison with other methods across datasets. Bold denotes the best performance.

scores. For the final numbers, the scores are averaged over all label maps.

Lastly, we propose to test the efficacy of generated images when applied as *synthetic data augmentation* for the task of semantic segmentation (see Sec. 4.3.5). For this, we take a DeepLab-V3 segmentation network with a ResNeSt-50 backbone (Zhang et al., 2020b) and train it on ADE20K and Cityscapes. At each training step of DeepLab-V3, we add for each training image its synthetic counterpart to the batch, generated from the same label map. The efficacy of synthetic images is therefore measured by its effect on the downstream mIoU performance of DeepLab-V3.

4.3.2 Evaluation of the synthesis quality and diversity

In this section, we compare OASIS to previous state-of-the-art methods. For a fair comparison to the baseline SPADE, we additionally train this model without the feature matching loss and using EMA (Yaz et al., 2018) at the test phase. We refer to this improved baseline as SPADE+.

Synthesis quality. Table 4.1 compares the synthesis quality achieved by OASIS and previous methods. We report the results of our evaluation for OASIS and SPADE+, and the officially reported numbers for all the other models. As seen from Table 4.1, OASIS outperforms prior state-of-the-art models in FID on all benchmark datasets. Our model also has the highest mIoU scores on three out of four datasets, being almost on par with the highest score on ADE20K achieved by SESAME (Ntavelis et al., 2020) Importantly, OASIS achieves the improvement using only adversarial supervision from its segmentation-based discriminator. On the contrary, in the absence of the VGG loss, the baseline SPADE+ does not produce images of high visual quality (see Fig. 4.1), with two-digit drops in FID scores observed for all

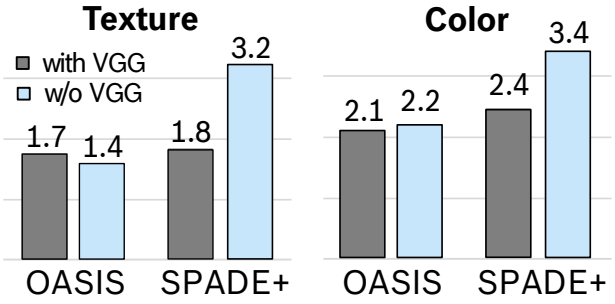

Figure 4.7: Histogram distances to real data on the ADE20K validation set. While SPADE+ relies on the VGG loss to learn colors and textures, OASIS achieves low scores without it.

Method	Multi-mod.	VGG	MS-SSIM↓	LPIPS↑	FID↓	mIoU↑
SPADE+	Encoder	✓	0.85	0.16	33.4	40.2
SPADE+	3D noise	✗	**0.35**	**0.50**	58.4	18.7
		✓	0.53	0.36	34.4	36.2
OASIS	3D noise	✗	0.65	0.35	**28.3**	48.8
		✓	0.88	0.15	31.6	**50.8**

Table 4.2: Multi-modal synthesis evaluation on ADE20K. Bold and red denote the best and the worst performance.

the datasets in Table 4.1. The strong adversarial supervision also allows OASIS to produce images with color and texture distributions closer to the real data, which is demonstrated in Fig. 4.7, where OASIS achieves the lowest color and texture distances to the target distribution. In contrast, SPADE+ needs to compensate for a weaker discriminator signal with the VGG loss, struggling to learn the color and texture distribution of real images without it (see Fig. 4.7).

Fig. 4.8 shows a qualitative comparison of our results to previous models. Our approach noticeably improves image quality, synthesizing finer textures and more natural colors. While the previous methods occasionally produce areas with unnatural checkerboard artifacts, OASIS generates large objects and surfaces with higher photorealism. Notably, the improvement over previous models is especially remarkable for the semantic classes that occupy large areas, e.g., wall (rows 1,4 in Fig. 4.8), road (rows 5,6), or water (row 3).

Synthesis diversity. By resampling the input 3D noise, OASIS can produce diverse images given the same label map (see Fig. 4.2). To measure the diversity of such multi-modal synthesis, we evaluate MS-SSIM (Wang et al., 2003) and LPIPS (Zhang et al., 2018d). The lower the MS-SSIM and the higher the LPIPS scores, the more diverse the generated images are. As seen from Table 4.2, OASIS outperforms SPADE+ in both diversity metrics, improving the MS-SSIM scores from 0.85 to 0.65

and LPIPS from 0.16 to 0.35. To assess the effect of the perceptual loss and the noise sampling on diversity, we train SPADE+ with 3D noise or the image encoder, and with or without the perceptual loss. Table 4.2 shows that OASIS, without the perceptual VGG loss, improves over SPADE+ with the image encoder, both in terms of image diversity (MS-SSIM, LPIPS) and quality (mean FID, mIoU across 20 realizations). Using 3D noise further increases diversity for SPADE+. However, a strong quality-diversity trade-off exists for SPADE+: 3D noise improves diversity at the cost of quality, and the perceptual loss improves quality at the cost of diversity. We conclude that our 3D noise injection strongly improves the synthesis diversity, while the VGG loss decreases it.

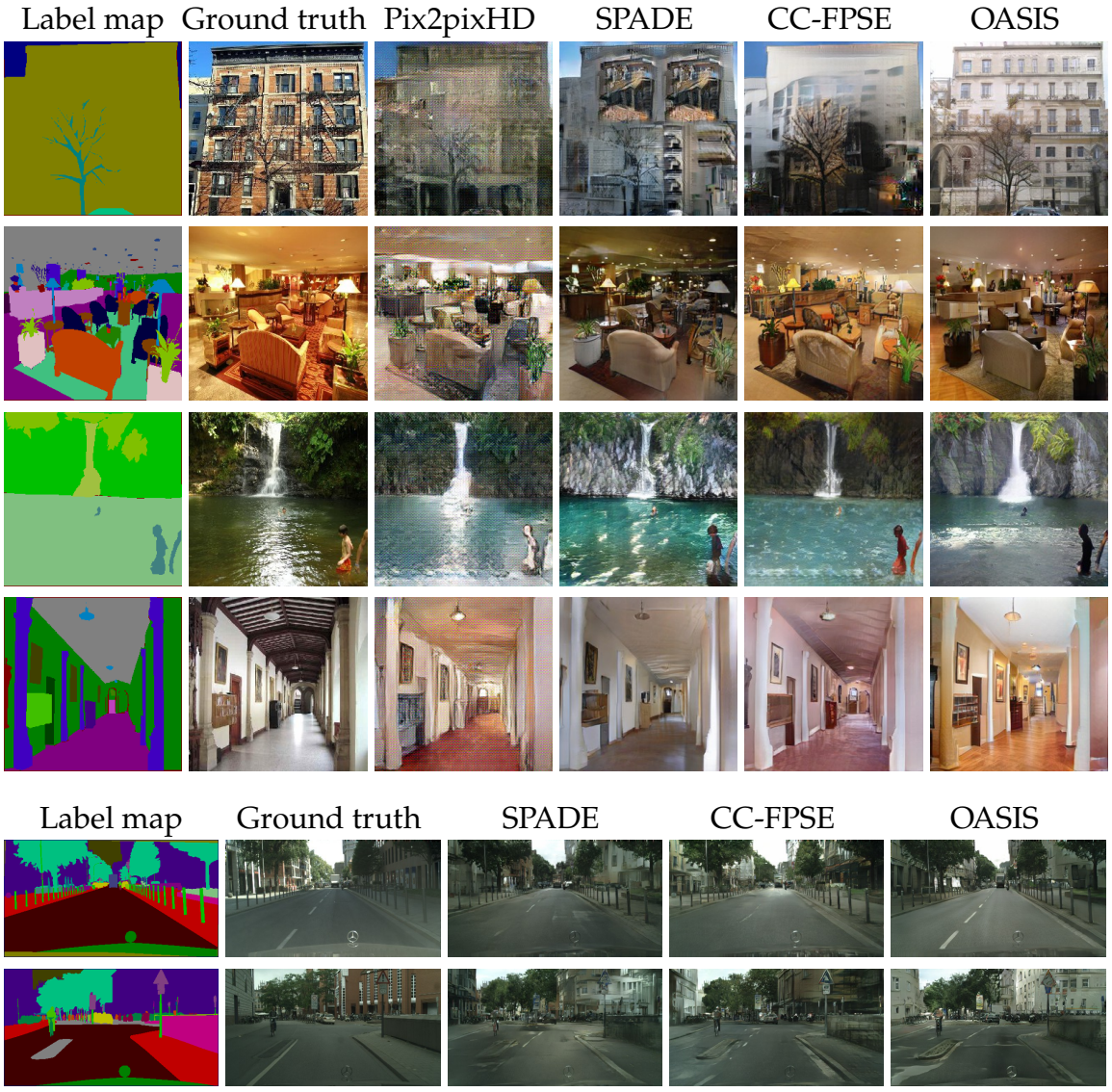

Figure 4.8: Qualitative comparison of OASIS with other methods on ADE20K and Cityscapes. Trained with only adversarial supervision, our model generates images with better perceptual quality and structure.

While the increased diversity is a big advantage, it can also lead to failures in

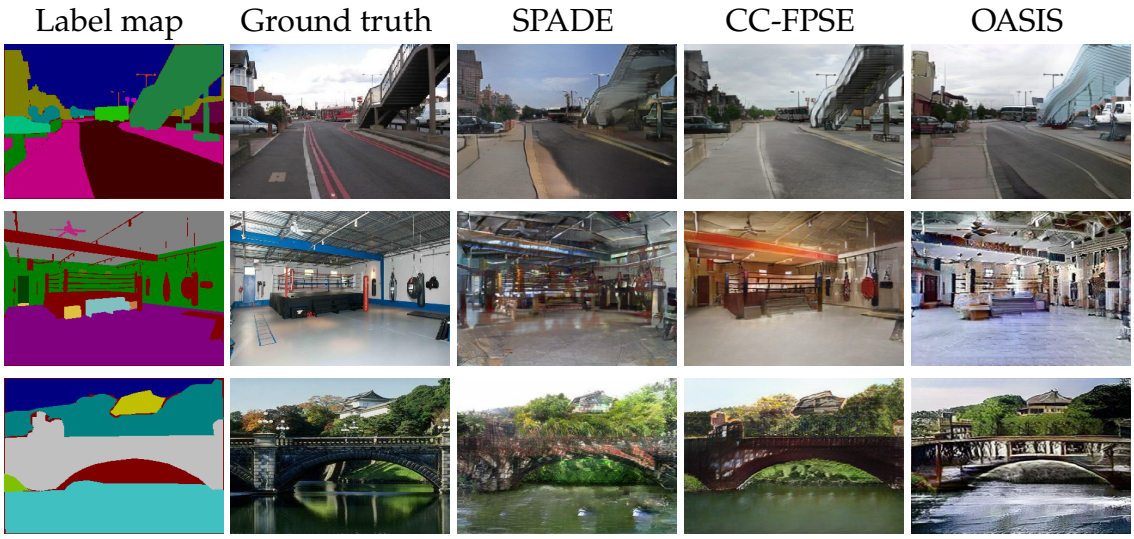

Figure 4.9: Failure mode of OASIS. Without the VGG loss, OASIS has less constraints on the diversity in colors and textures. This helps to achieve higher diversity among the generated samples, but sometimes leads to synthesis of objects with outlier colors and textures which may look less realistic compared to Park et al. (2019b) and Liu et al. (2019).

rare cases: for some samples the colors and textures of objects may lie further from the real distribution and seem unnatural to the human eye (see Fig. 4.9).

4.3.3 Synthesis performance on underrepresented classes

Class imbalance is a well-known challenge in semantic segmentation applications (Sudre et al., 2017). Similarly to semantic segmentation, to ensure good performance in real-life test scenarios, semantic image synthesis models should account for a possible dataset class imbalance, especially considering that GANs are notorious for dropping modes of training data (Arjovsky and Bottou, 2017). However, to the best of our knowledge, this issue was not addressed in prior works. Thus, in what follows, we evaluate the performance of OASIS and SPADE+ on the ADE20K and LVIS datasets, considering their class imbalances. While the class imbalance in ADE20K is notable (e.g., 86.4% of all image pixels belong to the 30 best represented classes), this issue is much more amplified in LVIS, which has a long tail of underrepresented classes (see Fig. 4.6).

Evaluation on ADE20K. OASIS significantly outperforms the SPADE+ baseline in the alignment between generated images and label maps, as measured by mIoU (see Table 4.1). As shown in Table 4.3, the improvement in mIoU on ADE20K comes mainly from the better IoU scores achieved for underrepresented semantic classes. To illustrate this, the semantic classes are sorted by their pixel-wise frequency in the training images, obtained by dividing the number of pixels a class occupies in the dataset by the total number of pixels of all images (2nd column in Table

Classes IDs	Pixel-wise frequency	mIoU		
		SPADE+	OASIS (w/o α_c)	OASIS (w. α_c)
0 - 29	86.4%	63.7	**69.1**	68.8
30 - 59	7.2%	47.4	52.4	**56.6**
60 - 89	3.5%	45.3	47.0	**51.5**
90 - 119	1.8%	29.3	36.2	**41.5**
120 - 149	1.0%	26.2	31.2	**39.7**
0-149 (all classes)	100%	42.4	47.2	**51.6**

Table 4.3: Per-class IoU scores on ADE20k, grouped by pixel-wise frequency (the fraction of all pixels in the datasets belonging to one class). Bold denotes the best performance. Training with per-class loss balancing is denoted by α_c.

Method	FID ↓	mAP, % ↑	classes with AP > 0 ↑
SPADE+	26.8	4.56	439
OASIS	**15.3**	**5.38**	**510**
real data	0	6.70	624

Table 4.4: Comparison of SPADE+ and OASIS on the LVIS dataset with 1203 classes and a long tail of underrepresented classes. Bold denotes the best performance. Last row shows the scores for the LVIS validation set.

4.3). Table 4.3 highlights that the relative gain in mIoU is especially high for the groups of underrepresented semantic classes, that cover less than 3% of all pixels in the dataset. For these classes, the relative gain over the SPADE+ baseline exceeds 40%. Remarkably, the gain for this group mainly comes from the per-class balancing applied in the OASIS loss function (columns "w/o α_c" and "w. α_c"), which draws the attention of the discriminator to underrepresented semantic classes, thus allowing a higher quality of their generation. This class balancing computes a weight α_c for the losses of each class c on a per-batch basis, for which the total number of pixels in a given batch is divided by the number of pixels belonging to the class (see Eq. 4.2 and 4.3). We note that the possibility to introduce the pixel-wise frequency based balancing requires the loss to be computed separately for each image pixel. This is a unique property of the OASIS discriminator, in contrast to conventional classification-based discriminators, which have to evaluate realism with a single score for images containing both well- and underrepresented classes together.

Evaluation on LVIS. A quantitative comparison between the models on the LVIS dataset is shown in Table 4.4. In this more extremely imbalanced data regime,

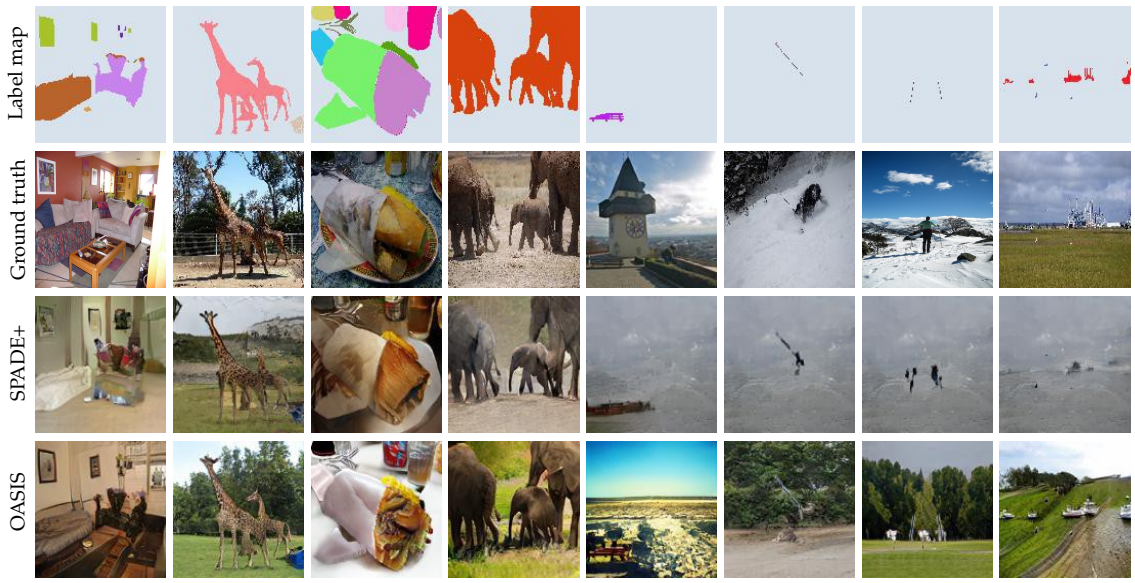

Figure 4.10: Qualitative comparison between OASIS and SPADE+ on the long-tailed LVIS dataset with 1203 classes. OASIS generates higher-quality images with more natural colors and textures. For label maps covered mostly by the background class (four right columns), OASIS hallucinates plausible and diverse images, while SPADE+ suffers from mode collapse.

the gain of our model is pronounced: OASIS outperforms SPADE+ by a large margin, lowering the FID by 43% (from 26.8 to 15.3). Fig. 4.10 shows a qualitative comparison between the models. OASIS produces images of higher visual quality with more natural colors and textures. In Table 4.4 we report the mean Average Precision (mAP) of the instance segmentation network evaluated on the set of generated images. OASIS outperforms SPADE+ in mAP by a notable margin (5.38 vs 4.56), thus producing objects with a more realistic appearance and largely reducing the gap to real data (mAP of 6.70). To evaluate the ability of the models to generate underrepresented classes at the tail of the LVIS data distribution, we count the number of classes for which a non-zero AP score is achieved. Table 4.4 shows that OASIS can model more semantic classes: OASIS achieves a positive AP for 510 semantic classes compared to 439 for SPADE+, thus exhibiting a better capability to synthesize underrepresented classes.

In addition to better handling the class imbalance, OASIS also visually outperforms SPADE+ on the LVIS label maps with a very large proportion of the background class. As seen in Fig. 4.10 (four rightmost columns), from such label maps, SPADE+ fails to produce plausible images and suffers from mode collapse. In contrast, OASIS successfully deals with such kinds of inputs, producing diverse and visually plausible images even for the least annotated label maps, with the highest proportion of the background class.

In conclusion, we consider long-tailed datasets, such as LVIS, an interesting

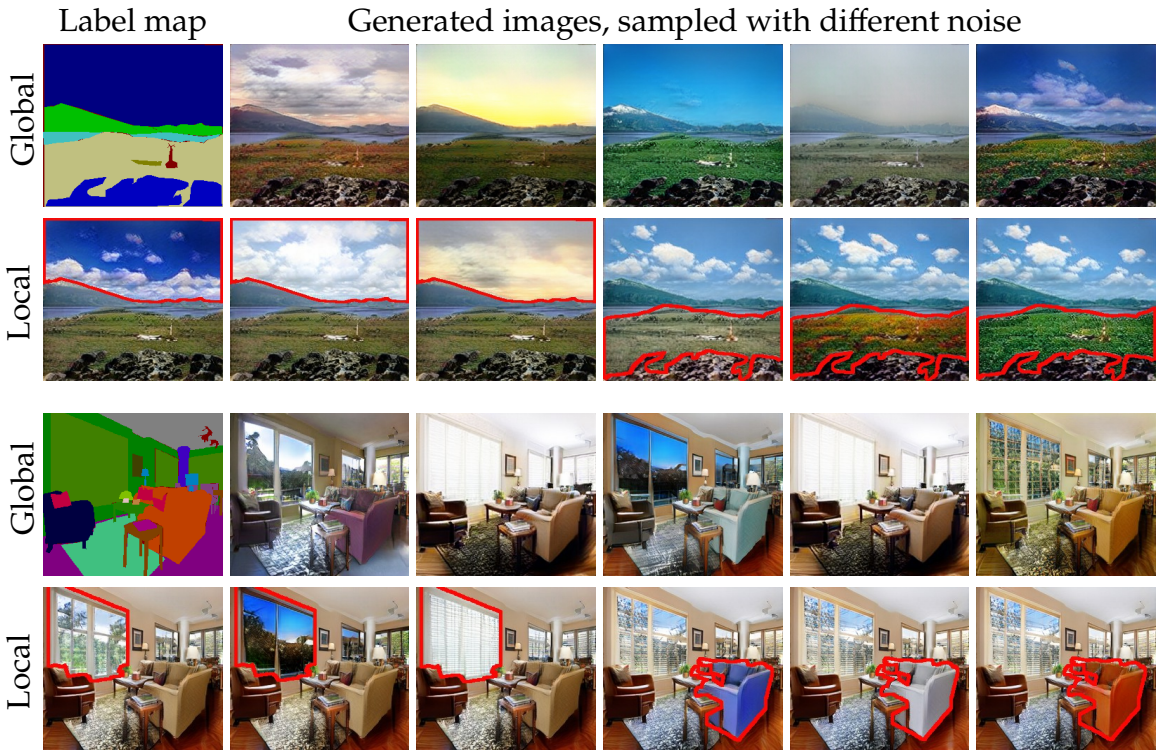

Figure 4.11: Images generated by OASIS on ADE20K with 256×256 resolution using different 3D noise inputs. For both input label maps, the noise is resampled globally (first row) or locally in the areas marked in red (second row).

direction for future work, as the improved synthesis of multiple tail classes under severe imbalance can significantly boost the applicability of semantic image synthesis to real-world applications.

4.3.4 Image editing with OASIS

OASIS can generate diverse images for a single label map by resampling input 3D noise. In the following, we present qualitative multi-modal results and discuss two unique semantic image editing techniques enabled by our model: local resampling of selected semantic classes and diverse resampling of unlabelled images.

Global and local resampling of the 3D noise. The 3D noise of OASIS modulates the activations directly at every generator layer, matching the spatial resolution of features at different generation scales. Therefore, such modulation affects both global and local characteristics of a generated image. At test time, this allows different strategies for noise sampling. For example, the noise can be sampled globally for all pixels, varying the whole image (see Fig. 4.11, first and third rows). Alternatively, a noise vector can be resampled only for specified image regions, resulting in local image editing while preserving the rest of the scene. For example, the local strategy allows to resample only the sky area in a landscape scenery,

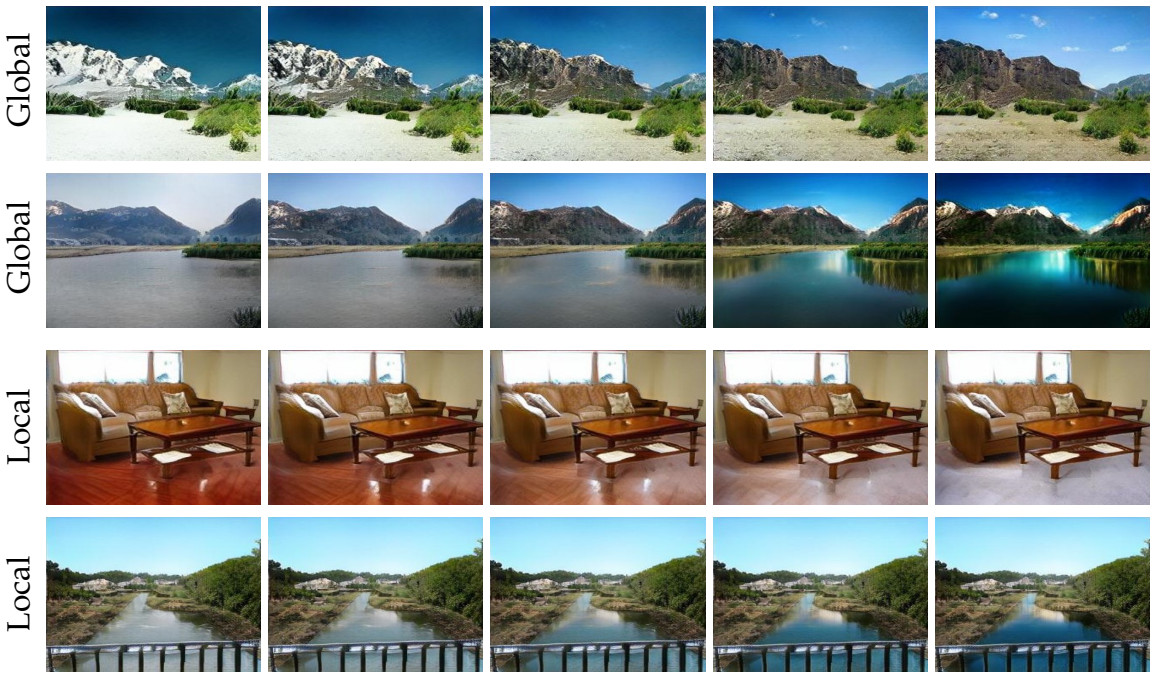

Figure 4.12: Latent space interpolations between images generated by OASIS for the ADE20K dataset at resolution 256×256. The first two rows display *global* interpolations. The second two rows show *local* interpolations of the floor or water only.

or only the window in a scene of a bedroom (see Fig. 4.11, second and fourth rows). Spatial sensitivity of OASIS to 3D noise is further demonstrated in Fig. 4.12, showing interpolations in the latent space. The learned latent space captures well the semantic meaning of objects and allows smooth interpolations not only globally, but also locally for selected objects (see Fig. 4.12, two last rows).

Creating diverse images from unlabelled data. In contrast to previous semantic image synthesis methods, the OASIS discriminator can be reused as a stand-alone image segmenter. To obtain a segmentation prediction for a given image, a user just needs to feed it to our pretrained discriminator and select the highest activation among real classes in its $(N{+}1)$-channel output for each pixel. When tested as an image segmenter on the validation set of ADE20K, the OASIS discriminator reaches a mIoU of 40.0. For comparison, the state-of-the-art model DeepLab-V3 with a ResNeST backbone (Zhang et al., 2020b) achieves an mIoU of 46.91. The good segmentation performance allows OASIS to be applied to unlabelled images: given an unseen image without the ground truth annotation, OASIS can predict a label map via the discriminator. Subsequently feeding this prediction to the generator allows to synthesize a scene with the same layout but different style (see Fig. 4.13). The recreated scenes closely follow the ground truth label map of the original image and vary considerably, due to the high sensitivity of OASIS to the 3D noise. We note that OASIS uniquely reaches this ability using only adversarial training, without the need for an external segmentation network or additional loss

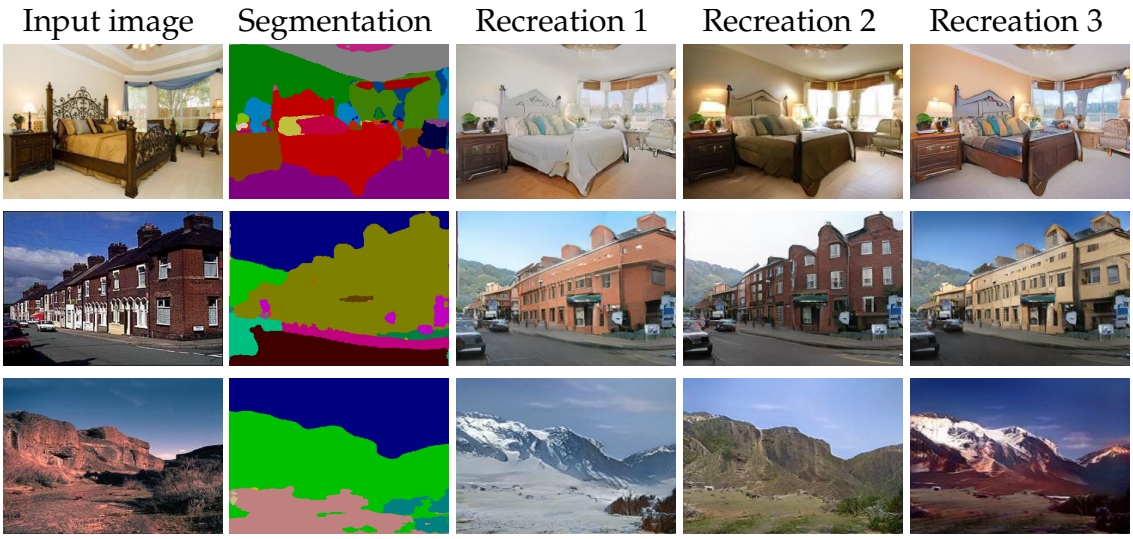

Figure 4.13: After training, the OASIS discriminator can be used to segment images. The first two columns show the real image and the segmentation of the discriminator. Using the predicted label map, the generator can produce multiple versions of the original image by resampling noise (Recreations 1-3). Note that no ground truth maps are required.

functions. We believe that the ability to create multiple versions of one image while retaining the layout, but not requiring the ground truth label map, may provide useful data augmentation for various applications in future research.

4.3.5 Synthetic data augmentation

As an additional evaluation method, we test the efficacy of generated images when applied as synthetic data augmentation for the task of semantic segmentation. Synthetic data augmentation is a task that benefits from both image quality and diversity, as well as the ability to generate semantic classes that are underrepresented in the original data (see Table 4.3). Therefore, the effect of synthetic data augmentation on downstream performance can constitute a more holistic evaluation of semantic image synthesis models. To test the efficiency of OASIS, we train a DeepLab-V3 segmentation network on ADE20K and Cityscapes, at each step augmenting each training image with its synthetic augmentation, produced by OASIS from the same label map.

We compare OASIS against the strong baseline SPADE in Table 4.5. Between the two methods, OASIS elicits a stronger increase in segmentation performance with an improvement of 2.0 mIoU on Cityscapes and 0.8 mIoU on ADE20K, compared to DeepLab-V3 trained without synthetic augmentation. The higher performance improvement of OASIS compared to SPADE is explained by all the previously observed gains in image quality, diversity, and the alignment to input label maps (see Fig. 4.7, Tables 4.1 and 4.2). In addition to that, the segmentation performance

Data augmentation	Cityscapes mIoU↑	ADE20K mIoU↑
no synthetic DA	62.7	41.0
with SPADE	62.6	41.6
with OASIS	**64.7**	**41.8**

Table 4.5: Semantic segmentation performance of ResNeSt-50 with and without synthetic data augmentation (DA). Bold denotes the best performance.

is also improved due to the fact that OASIS tends to synthesize underrepresented classes better than SPADE, which is evident from Table 4.6. This table compares the IoU performance of DeepLab-V3 on the well-represented and underrepresented classes of Cityscapes, as measured by the pixel-wise frequency of the semantic class in the dataset. Examples of well-represented classes are road and building (see the 1st row of Table 4.6), while classes like bicycle or traffic light are the least represented in the dataset (see 4th row in Table 4.6). Note that the IoU comparison in Table 4.6 is different from Table 4.3, where the IoU was measured directly on synthetic data using a pretrained segmenter. It can be seen that the improvement in IoU through OASIS can be mostly attributed to better performance on underrepresented classes, as the gap in performance between OASIS and SPADE becomes larger for the classes which are less represented. Lastly, since the OASIS generator was trained to fool an image segmenter (the OASIS discriminator), it may synthesize harder examples for semantic segmentation than SPADE, thus having higher potential to improve the generalization of segmentation networks to challenging corner cases. We find the above results promising for future utilization of OASIS in various downstream applications. Moreover, for future research, we find it interesting to explore synthetic data augmentation in combination with other data augmentation techniques, e.g., RandAugment (Cubuk et al., 2020), which has the potential to provide further performance gains for downstream applications.

4.3.6 Ablations

We conduct all our ablations on the ADE20K dataset. We choose this dataset as it more challenging (with 150 classes) than Cityscapes (35 classes) and ADE20K-Outdoors (110 classes), and has more reasonable training time (5 days) compared to COCO-Stuff and LVIS (4 weeks). Our main ablation shows the impact of the main technical components of OASIS, including the new discriminator, lighter generator, LabelMix and the 3D noise. Further ablations are concerned with the architecture changes in the discriminator, the label map encoding in the discriminator, different noise sampling strategies, LabelMix, and the GAN feature matching loss.

Main ablation. Table 4.7 shows that SPADE+ achieves low performance on the image quality metrics without the perceptual loss. Replacing the SPADE+ discrim-

Sorted classes	Pixel-wise frequency	None	SPADE		OASIS	
			abs	rel	abs	rel
0 - 4	82.7%	90.6	90.6	+0.0	**90.9**	**+0.3**
5 - 8	12.5%	66.2	66.2	+0.0	**67.4**	**+1.2**
9 - 12	3.3%	50.2	49.1	-1.1	**52.2**	**+2.0**
13 - 18	1.6%	51.9	52.3	+0.4	**55.4**	**+3.5**
all classes	100%	62.7	62.6	-0.1	**64.7**	**+2.0**

Table 4.6: Per-class IoU scores on Cityscapes, obtained without (None) and with synthetic data augmentation using SPADE or OASIS. The classes are sorted and grouped by class pixel-wise frequency, as measured by the total fraction of pixels in the dataset belonging to one class. Bold denotes the best performance. The absolute (abs) and relative (rel) mIoU gain via data augmentation is shown.

G	*D*	VGG	LabelMix	FID↓	mIoU↑
SPADE+	SPADE+	✗	✗	60.7	21.0
SPADE+	OASIS	✗	✗	29.0	**52.1**
OASIS	OASIS	✗	✗	29.3	51.6
		✗	✓	28.4	50.6
OASIS +3D noise	OASIS	✗	✓	**28.3**	48.8
		✓	✓	31.6	50.8

Table 4.7: Main ablation on ADE20K. The OASIS generator is a lighter version of the SPADE+ generator (72M vs 96M parameters). Bold denotes the best performance.

inator with the OASIS discriminator, while keeping the generator fixed, improves FID and mIoU by more than 30 points. Changing the SPADE+ generator to the lighter OASIS generator leads to a negligible degradation of 0.3 in FID and 0.5 in mIoU, but reduces the number of parameters from 96M to 72M. With LabelMix FID improves further by about 1 point. Adding 3D noise improves FID but degrades mIoU, as diversity complicates the task of the pretrained semantic segmentation network used to compute the mIoU score. For OASIS the perceptual loss deteriorates FID by more than 2 points, but improves mIoU. Overall, without the VGG loss the new discriminator is the key to the performance boost over SPADE+.

Ablation on the discriminator architecture. We train the OASIS generator with three alternative discriminators: the original multi-scale PatchGAN consisting of two networks, a single-scale PatchGAN, and a ResNet-based discriminator, corresponding to the encoder of the U-Net shaped OASIS discriminator. Table 4.8 shows that the alternative discriminators only perform well with perceptual supervision, while the OASIS discriminator achieves superior performance independent of it. The single-scale discriminators even collapse without the perceptual loss (red

D architecture	w/o VGG		with VGG	
	FID↓	mIoU↑	FID↓	mIoU↑
MS-PatchGAN (2x)	60.7	21.0	32.9	42.5
PatchGAN	197	0.62	34.2	42.2
ResNet-PatchGAN	147	0.42	32.4	45.1
OASIS	**29.3**	**51.6**	**29.2**	**51.1**

Table 4.8: Ablation on the *D* architecture. Bold denotes the best performance, red highlights collapsed runs.

Label encoding	w/o VGG		with VGG	
	FID↓	mIoU↑	FID↓	mIoU↑
Input concatenation	280	0.02	30.0	43.9
Projection	32.4	44.9	**28.0**	46.9
N+1 loss	**28.3**	47.2	28.6	49.8
Balanced N+1 loss	29.3	**51.6**	29.2	**51.1**

Table 4.9: Ablation on the label map encoding. Bold denotes the best performance, red shows collapsed runs.

colors in Table 4.8).

Ablation on the discriminator label map encoding. We study four different ways to use label maps in the discriminator: the first encoding is input concatenation, as in SPADE. The second option is a pixel-wise projection-based GAN loss (Miyato and Koyama, 2018). Unlike Miyato and Koyama (2018), we condition the GAN loss on the label map instead of a single label. The third and fourth option is to employ the label maps as ground truth for the N+1 segmentation loss, or for the class-balanced N+1 loss (see Sec. 4.2.2). For a fair comparison we use neither 3D noise nor LabelMix. As shown in Table 4.9, input concatenation is not sufficient without additional perceptual loss supervision, leading to training collapse. Without the perceptual loss, the N+1 loss outperforms the input concatenation and the projection in both the FID and mIoU metrics. Finally, the class balancing enables enhanced supervision for underrepresented semantic classes, which noticeably improves mIoU scores. On the other hand, we observed that the FID metric is more sensitive to the synthesis of well-represented classes and not underrepresented classes, which explains the negative effect of the class balancing on FID.

Ablation on noise sampling strategies for training. Our 3D noise can contain the same sampled vector for each pixel, or different vectors for different regions. This allows for different sampling strategies during training. Table 4.10 shows the effect of using different methods of sampling 3D noise for different locations during training: *Image-level* sampling creates one global 1D noise vector and replicates it

Sampling	Cityscapes			ADE20K		
	FID↓	mIoU↑	MS-SSIM↓	FID↓	mIoU↑	MS-SSIM↓
Image-level	47.7	69.3	0.64	**28.3**	**48.8**	0.65
Region-level	48.1	69.7	**0.62**	28.8	48.1	**0.58**
Pixel-level	50.9	65.5	0.84	28.6	34.0	0.68
Mix	**46.4**	**70.9**	0.68	28.5	47.6	0.66

Table 4.10: Different 3D noise sampling strategies during training. Bold denotes the best performance.

along the height and width of the label map to create a 3D noise tensor. *Region-level* sampling relies on generating one 1D noise vector per semantic class, and stacking them in 3D to match the height and width of the semantic label map. *Pixel-level* sampling creates different noise for every spatial position, with no replication taking place. *Mix* switches between image-level and region-level sampling via a coin flip decision at every training step. With no obvious winner in performance, we choose the simplest scheme (image-level) for our experiments. We find a further investigation with more advanced strategies an interesting direction for future work.

Ablation on LabelMix. Consistency regularization for the segmentation output of the discriminator requires a method of generating binary masks. Therefore, we compare the effectiveness of CutMix (Yun et al., 2019) and our proposed LabelMix. Both methods produce binary masks, but only LabelMix respects the boundaries between semantic classes in the label map. Table 4.11 compares the FID and mIoU scores of OASIS trained with both methods on the Cityscapes dataset. As seen from the table, LabelMix improves both FID (51.5 vs. 47.7) and mIoU (66.3 vs. 69.3), in comparison to OASIS without consistency regularization. CutMix-based consistency regularization only improves the mIoU (66.3 vs. 67.4), but not as much as LabelMix (69.3). We suspect that since the images are already partitioned through the label map, an additional partition through CutMix results in a dense patchwork of areas that differ by semantic class and real/fake class identity. This may introduce additional label noise during training for the discriminator. To avoid such inconsistency between semantic classes and real/fake identity, the mask of LabelMix is generated according to the label map, providing natural borders between semantic regions, so that the real and fake objects are placed side-by-side without interfering with each other. Under LabelMix regularization, the generator is encouraged to respect the natural semantic class boundaries, improving pixel-level realism while also considering the class segment shapes.

Ablation on the feature matching loss. We measure the effect of the discriminator feature matching loss (FM) in the absence and presence of the perceptual loss (VGG). The discriminator feature matching loss is used by default in SPADE. Table

Transformation	FID↓	mIoU ↑
No CR	51.5	66.3
CutMix	52.1	67.4
LabelMix	**47.7**	**69.3**

Table 4.11: Ablation study on the impact of LabelMix and CutMix for consistency regularization (CR) in OASIS on Cityscapes. Bold denotes the best performance.

VGG	FM	FID↓	mIoU↑
✗	✗	47.7	69.3
✗	✓	48.5	69.1
✓	✗	**46.1**	**72.0**
✓	✓	46.5	70.9

(a) OASIS on Cityscapes.

VGG	FM	FID↓	mIoU↑
✗	✗	61.4	47.6
✗	✓	57.3	55.8
✓	✗	**47.8**	64.0
✓	✓	48.1	**64.4**

(b) SPADE+ on Cityscapes.

Table 4.12: The effect of the discriminator feature matching loss (FM) in the absence or presence of the perceptual loss (VGG). Bold denotes the best performance.

4.12 presents the results for OASIS and SPADE+ on Cityscapes. For SPADE+, we observe that the feature matching loss affects the metrics notably only when no perceptual loss is used. In this case, the FM loss improves mIoU by 8.2 points. In contrast, the effect of the FM loss on the mIoU is small when the perceptual loss is used (0.4 points). Hence, the role of the FM loss in the training of SPADE+ is to improve performance by stabilizing the training, similar to the perceptual loss. This observation is in line with the general observation that SPADE and other semantic image synthesis models require the help of additional loss functions because the adversarial supervision through the discriminator is not strong enough. In comparison, we did not observe any training collapses in OASIS, despite not using any extra loss functions. For OASIS, the feature matching loss results in a worse FID (by 0.8 points) in the absence of the perceptual loss. We also observe a degradation of 1.1 mIoU points through the FM loss, in the case where the perceptual supervision is present. This indicates that the FM loss negatively affects the strong supervision from the semantic segmentation adversarial loss of OASIS.

4.4 Conclusion

This work studies semantic image synthesis, the task of generating diverse and photorealistic images from semantic label maps. Conventionally, semantic image synthesis GAN models employed a perceptual VGG loss to overcome training instabilities and improve the synthesis quality. In our experiments we demonstrated that the VGG-based perceptual loss imposes unnecessary constraints on the feature

space of the generator, significantly limiting its ability to produce diverse samples from input noise, as well as the ability to produce images with colors and textures closely matching the distribution of real images. Therefore, in this work we propose OASIS, a semantic image synthesis model that needs only adversarial supervision to achieve high-quality results.

The improvement over the prior work in image synthesis quality is achieved via the detailed spatial and semantic-aware supervision from our novel segmentation-based discriminator, which uses semantic label maps as ground truth for training. With this powerful discriminator, OASIS can easily generate diverse outputs from the same semantic label map by resampling 3D noise, eliminating the need for additional image encoders to achieve multi-modality. The proposed 3D noise injection scheme can work both in a global and local regime, allowing to change the appearance of the whole scene and of individual objects. With the proposed modifications, OASIS significantly improves over previous state-of-the-art models in terms of image synthesis quality.

Furthermore, we proposed to use the LVIS dataset to evaluate semantic image synthesis under severe class imbalance and sparse label annotations. Thanks to the class balancing mechanism enabled by its segmentation-based discriminator, OASIS achieves more realistic synthesis of underrepresented classes, achieving pronounced gains on the extremely unbalanced LVIS dataset. Lastly, the design of OASIS can be better suited for image editing applications compared to the SPADE baseline, enabling diverse resampling of scenes from unlabeled images, as well as for synthetic data augmentation, improving the performance of a downstream segmentation network by a larger margin.

The fact that image diversity is substantially increased by using 3D noise and abolishing the perceptual loss raises an interesting question: Does the latent space have a semantic structure, meaning that specific directions in this space correspond to meaningful image transformations, which may even be class-specific? For example, do dedicated directions exist for the type of sheets on the bed or the road's surface? In Chapter 5 we present a method to find such directions and use it to manipulate the appearance of classes in semantic image synthesis models.

5 Discovering GAN Controls for Semantic Image Synthesis

Contents

5.1 Introduction **84**

5.2 Ctrl-SIS method **86**

5.2.1 GAN controls for SIS models 87

5.2.2 Discovery of class-specific GAN controls 88

5.3 Experiments **90**

5.3.1 Experimental setup 90

5.3.2 Evaluation of class-specific GAN controls 91

5.3.3 Main results 94

5.4 Conclusion **99**

In Chapter 4, we proposed changes to make semantic image synthesis (SIS) models significantly more sensitive to input noise, resulting in increased synthesis diversity. In this chapter, we study the diverse latent space that accompanies the increased diversity in SIS GANs. Prior work has extensively studied the latent space structure of GANs for unconditional image synthesis, enabling global editing of generated images by identifying interpretable latent directions. However, the discovery of latent controls for conditional GANs for semantic image synthesis (SIS) has remained unexplored. In this work, we specifically focus on addressing this gap. By making use of inherent semantic label maps in the SIS task, we propose a novel optimization method for finding spatially disentangled class-specific latent controls. We show that the latent directions found by our method can effectively control the local appearance of semantic classes, e.g., changing their internal structure, texture, or color independently from each other. Visual inspection and quantitative evaluation of the discovered GAN controls on various datasets demonstrate that our method discovers a diverse set of unique and semantically meaningful latent directions for class-specific edits. The content of this chapter

corresponds to our paper "Discovering Class-Specific GAN Controls for Semantic Image Synthesis" which is currently under review and not yet published.

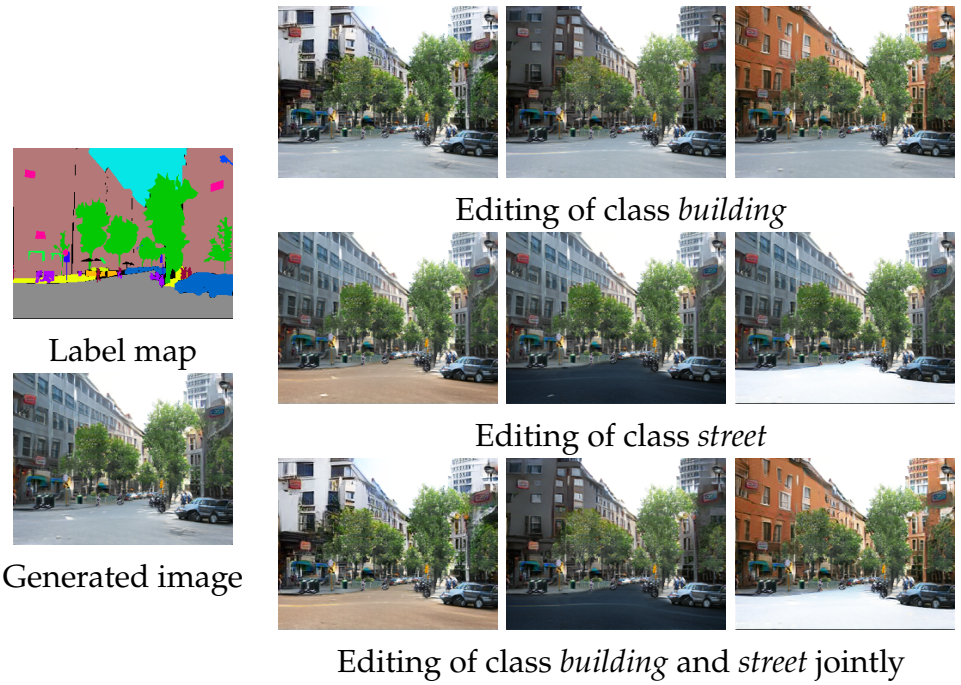

Figure 5.1: Ctrl-SIS learns class-specific directions in the latent space of a SIS model, which can be applied jointly for different classes for local editing of the image.

5.1 Introduction

Semantic image synthesis (SIS) transforms user-specified semantic layouts to realistic images. Its applications range widely from image editing and content creation to synthetic data augmentation, where training data is generated to fulfill specific semantic requirements. For SIS, GANs (Goodfellow et al., 2014) have demonstrated their superiority in terms of the visual quality of synthesized images and their alignment to input semantic label maps (Park et al., 2019b; Schönfeld et al., 2021; Tan et al., 2021; Wang et al., 2021c; Li et al., 2021b). Although some of the GAN-based SIS models allow local appearance editing of single classes or regions in an image – either by style transfer from a reference image (Zhu et al., 2020a; Lee et al., 2020; Tan et al., 2021) or by sampling noise independently for specific image regions (Schönfeld et al., 2021; Zhu et al., 2020b), there is no technique to enable interpretable semantic changes for a specific class without a reference image and user-in-the-loop supervision.

On the other hand, prior work has extensively studied the latent space of unconditional GANs (Goetschalckx et al., 2019; Plumerault et al., 2019; Härkönen et al., 2020; Shen and Zhou, 2021; Tzelepis et al., 2021; Yüksel et al., 2021), finding interpretable latent directions which activate distinctive factors of variations in the

generation process in an unsupervised fashion, without exploiting reference images. Moving latent code(s) along a certain direction can result in domain-agnostic transformations, e.g., rotation or zooming (Voynov and Babenko, 2020; Jahanian et al., 2020; Plumerault et al., 2019), or domain-specific alterations, e.g., age or nose length of a person (Cherepkov et al., 2021; Wu et al., 2021; Ling et al., 2021; Shen et al., 2020; Collins et al., 2020). Despite their recent progress, it remains a challenge to find interpretable latent directions to control interactively the synthesis of specific semantic classes in the image without changing other image regions. Since the above methods were designed specifically for unconditional GANs, they are not well suited to discover class-specific latent directions in the presence of semantic label maps, inherently given for SIS.

In this work, we address this limitation and study the latent space of conditional GANs designed specifically for SIS, which to the best of our knowledge has not been explored previously. In particular, making use of the label maps we devise a method to discover meaningful latent directions that only change a specific semantic class in the image. These directions can, for example, encode different designs of the facade for the building class or surfaces for the street class (Fig. 5.1), enabling the user to perform local semantic edits independently from the rest of the image. Note that in recent state-of-the-art SIS GANs, the generator is already designed to be sensitive to spatial information (Schönfeld et al., 2021) (see Chapter 4). The generator is conditioned on the semantic label map along with the 3D latent code, allowing to modulate the appearance of every single pixel in the image. This results in spatial disentanglement across classes and, thus, better manipulation control for class-specific image regions in comparison to unconditional GANs (Brock et al., 2018; Karras et al., 2020b; Schönfeld et al., 2020; Karras et al., 2021a).

On this basis, we introduce a simple, efficient optimization method to discover class-specific controls in pretrained SIS GANs, which we call *Ctrl-SIS* (see Fig. 5.2). Our optimization objective is designed to ensure that the learned latent directions are 1) diverse and different from each other (*diversity loss*); 2) only affect the image area of the selected class, preserving the appearance of other areas (*disentanglement loss*); and 3) induce the same semantic edits consistently across different initial latent codes and label maps containing the class (*consistency loss*). See Sec. 5.2 for more details. We demonstrate that GAN controls discovered automatically by Ctrl-SIS can effectively manipulate the appearance of the selected semantic class in specific ways, without affecting other classes in the image. For example, we can change the house facade (see Fig. 5.1), remove leaves from trees or cover mountains in snow (see Fig. 5.4). Moreover, we can edit different classes jointly, e.g., alter both the building and the road in the street scene (see Fig. 5.1). Since we use only train-time optimization, instead of exhaustive search as in Wu et al. (2021) or test-time optimization as in Ling et al. (2021), Pajouheshgar et al. (2021), Zhu et al. (2021), or Zhu et al. (2022), our training time stays relatively fast compared to the former, while also allowing interactive image editing compared to the latter.

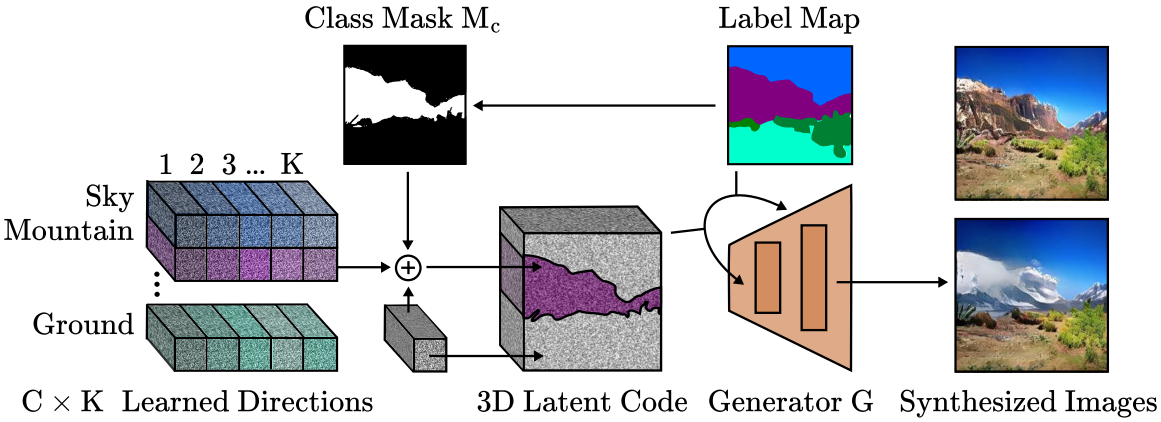

Figure 5.2: Ctrl-SIS provides a set of K class-specific latent directions which control the appearance of C semantic classes. To alter the appearance of class c, a class-specific latent direction is added to the input 3D latent code z of the generator G in the label map area corresponding to class $c - M_c$.

The evaluation of GAN control discovery methods is commonly left to subjective visual inspection. To address this, we introduce new metrics to quantitatively assess diversity, spatial disentanglement, and consistency properties of learned latent directions (see Sec. 5.3.2). We compare Ctrl-SIS with other GAN control methods for different SIS GANs (Schönfeld et al., 2021; Park et al., 2019b; Wang et al., 2021c) on two datasets (Zhou et al., 2017; Caesar et al., 2018). Our experiments show that latent directions found by prior methods adapted to SIS (Härkönen et al., 2020; Shen and Zhou, 2021) lead to weaker class edits, comparable to random directions (see Sec. 5.3). In contrast, Ctrl-SIS finds directions that enable diverse and semantically meaningful class edits while maintaining high image quality.

In summary, our contributions are as follows: 1) We propose Ctrl-SIS – a method to discover interpretable latent controls for individual semantic classes in pretrained SIS GANs. To the best of our knowledge, the discovery of class-specific latent direction has not yet been addressed in the SIS literature. 2) We define diversity, consistency, and spatial disentanglement as desirable properties of class-specific latent controls and propose new metrics to quantify them.

5.2 Ctrl-SIS method

The goal of this work is to discover steerable latent directions for GAN-based SIS models. Enabled by the given semantic label maps, we aim to find GAN controls specific to semantic classes, e.g. a set of latent directions for controlling the appearance of the street and another set of directions for the appearance of house facades, see Fig. 5.1. However, this task presents two major challenges.

The first challenge is that SIS GANs commonly do not provide the same image diversity as unconditional models (Brock et al., 2018; Karras et al., 2020b), nor have

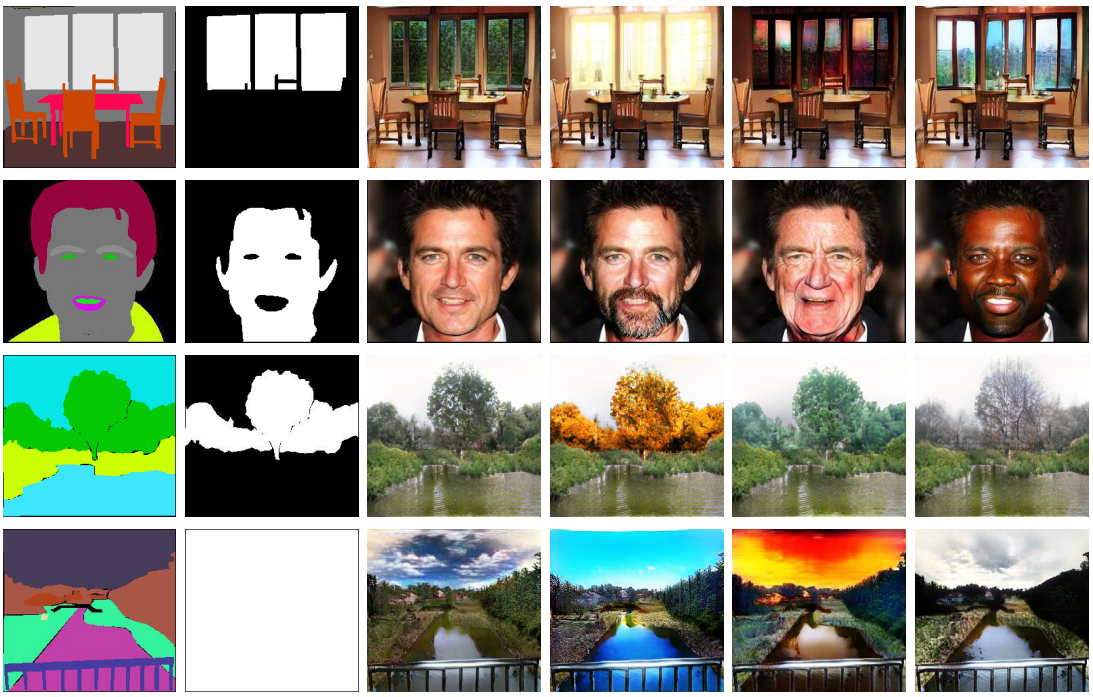

Figure 5.3: Examples of directions discovered by Ctrl-SIS for various classes, such as different views of a window, face appearances, or tree leafage for different seasons of the year. The directions give insight into the concepts that the pretrained SIS model is able to represent. The last row shows global edits with different directions per class.

region-specific latent codes. We alleviate both problems by applying a 3D latent code injection (Schönfeld et al., 2021), which we describe in Sec. 5.2.1. The second challenge is that prior GAN control discovery methods are not designed to consider label maps, nor to find *class-specific* directions, as they are devised for unconditional GANs. We address both aspects in Sec. 5.2.2 with a simple, efficient optimization method which we call Ctrl-SIS.

5.2.1 GAN controls for SIS models

Current SIS models employ different ways to inject latent code into the generator, which affects its ability to perform class-specific edits. The default approach is to feed a one-dimensional latent vector as input to the generator (Park et al., 2019b; Wang et al., 2021c; Liu et al., 2019), resulting in no direct opportunity to perform local region-based edits of the image. Thus, in order to enable local editing in SIS, we employ the 3D latent code injection scheme from Schönfeld et al. (2021) (see Chapter 4), adopting it to all SIS models considered in this work. The 3D latent codes $z \in \mathbb{R}^{H \times W \times D}$ are created by replicating the original noise vector along the height H and width W of the label map. The 3D latent code allows to apply different

latent vectors to different image regions (see Fig. 5.2). In practice, altering the 3D latent code only for a specific image region can still affect other image areas, due to spatial correlations learned by the generator during training. Nevertheless, the 3D latent space provides better spatial disentanglement, and thus improves image manipulation control for local edits compared to 1D latent codes. In the remainder of this paper, we assume a 3D latent space for the discovery of SIS GAN controls.

Let G be a well-trained GAN generator of an SIS model. The generator $G(z, y)$ synthesizes an image given a 3D latent code z and label map y, i.e., $x = G(z, y) = F(h(z, y))$, where $h = \{G_l(z, y)\}_{l \in L}$ is a chosen subset of features from intermediate layers $l \in L$ in the network G, and C is the total number of semantic classes. The latent code z controls the appearance of the synthetic image, while the label map y specifies the scene layout. Then an image x can be globally edited by moving z along a specific direction v_k:

$$x(v_k) = F(h(z, v_k, y)) = G(z + \alpha v_k, y), \tag{5.1}$$

where α controls the intensity of the change, and the latent direction v_k determines the semantics of the image transformation. Local editing of class c in x is carried out by moving z along a class-specific direction v_k^c only in the area of class c in the label map y:

$$x(v_k^c) = F(h(z, v_k^c, y)) = G(z + \alpha M_c \odot v_k^c, y), \tag{5.2}$$

where $M_c = \mathbb{1}_{[y=c]}$ is a binary mask indicating pixels in the image belonging to c (see Fig. 5.2). We next define the task of class-specific GAN control discovery and introduce an optimization objective to find v_k^c directions for any pretrained SIS model with a spatially-aware generation process induced by 3D latent codes.

5.2.2 Discovery of class-specific GAN controls

For the class of interest $c \in C$ we aim to find a diverse set of class-specific directions $V^c = \{v_0^c, v_1^c, ..., v_K^c\}$, $K > 1$, that can meaningfully edit the appearance of class c in the synthetic image x, such that image $x(v_k^c)$ has a visually distinct appearance of class c compared to x, but all other classes have the same appearance as in x. Based on this logic, we form an optimization objective, which consists of diversity, disentanglement, and consistency loss terms:

$$\min_{V^c} \mathcal{L}_{div} + \mathcal{L}_{dis} + \mathcal{L}_{const} \, . \tag{5.3}$$

The diversity loss $\mathcal{L}_{div}$ encourages a set of class-specific GAN controls V^c to be diverse and introduce different semantic changes to class c, the disentanglement loss $\mathcal{L}_{dis}$ prevents changes outside the class area, and the consistency loss $\mathcal{L}_{const}$ ensures that the semantics of an edit are consistent between different initial latent codes z. We next provide the mathematical formulation of these loss terms.

Diversity loss. Given a label map y and a class of interest c, the diversity loss aims to ensure that the set of found latent directions V^c applied to identical input latent code z yields maximally different semantic visual effects, i.e., change the appearance of class c in a different way. It is formulated as

$$\mathcal{L}_{div} = -\mathbb{E}_{(z,y)}\Big[\sum_{\substack{k_{1,2}=1 \\ k_1 \neq k_2}}^{K} M_c \cdot ||h(z, v^c_{k_1}, y) - h(z, v^c_{k_2}, y)||_2\Big], \tag{5.4}$$

where $\|\cdot\|$ is the L_2 norm, and for the class-specific area M_c the distance between the two resulting images $x(v^c_{k_1})$ and $x(v^c_{k_2})$ is maximized in the generator feature space h, ensuring semantically different directions for class c. Depending on the selected feature space in G, i.e., the subset of intermediate layers L in $h = \{G_l(z, y)\}_{l \in L}$, we can find various GAN control directions which correspond to different semantics encoded in the selected feature space of G.

Disentanglement loss. The discovered latent direction v^c_k for class c should only affect the image area belonging to c in the label map y and leave the rest of the image unaffected. Thus, we also minimize the change for images $x(v^c_{k_1})$ and $x(v^c_{k_2})$ in the feature space h in the area outside of M_c:

$$\mathcal{L}_{dis} = \mathbb{E}_{(z,y)}\Big[\sum_{\substack{k_{1,2}=1 \\ k_1 \neq k_2}}^{K} (1 - M_c) \cdot ||h(z, v^c_{k_1}, y) - h(z, v^c_{k_2}, y)||_2\Big]. \tag{5.5}$$

Consistency loss. Identical GAN control directions should cause consistent semantic edits of class c for different input latent codes and the same label map y given to the generator. Therefore, for every found direction v^c_k we minimize the feature space distance between two images generated with z_1 and z_2 in the class-specific area M_c :

$$\mathcal{L}_{const} = \mathbb{E}_{(z,y)}\Big[\sum_{k=1}^{K} M_c \cdot ||h(z_1, v^c_k, y) - h(z_2, v^c_k, y)||_2\Big]. \tag{5.6}$$

Note that the directions in V^c are the only parameters to be optimized; the weights of the pretrained image generator $G(z, y)$ are kept frozen. The parameters are optimized by iterating over batches of label maps in the training set and minimizing the objective for selected classes at every step. During optimization, the directions v^c_k are normalized along the channel dimension to unit length 1 and subsequently scaled by α, sampled from the interval $[-n; n]$, where $n = \mathbb{E}[||z||_2]$ is the average norm of the latent code along the channel dimension. This ensures that the latent edits are neither too small nor too extreme.

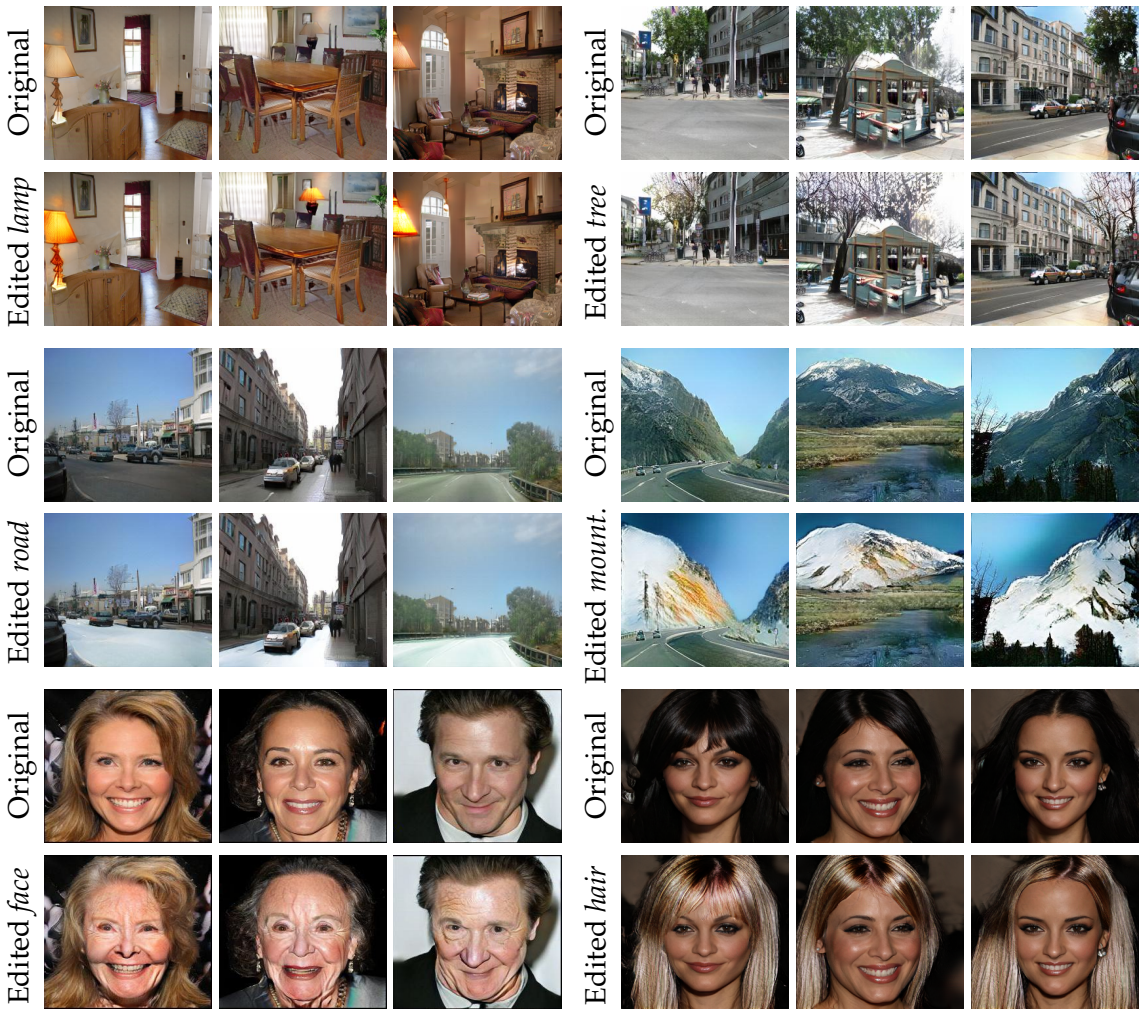

Figure 5.4: Interpretable latent directions learnt by Ctrl-SIS for various classes. Each triplet is edited with an identical direction. Class-specific edits, such as aging, snowy streets or bald trees, are highly consistent across different label maps and initial latent codes.

5.3 Experiments

5.3.1 Experimental setup

Datasets. We use three challenging datasets: ADE20K (Zhou et al., 2017), COCO-Stuff (Caesar et al., 2018), and CelebAMask-HQ (Lee et al., 2020). CelebAMask-HQ consists of 30k face images. ADE20K and COCO-Stuff contain 20k and 164k images of indoor and outdoor scenes, and are used for the main experiments and ablations.

SIS models. We consider three pretrained GANs for SIS: SC-GAN (Wang et al., 2021c), SPADE (Park et al., 2019b), and OASIS (Schönfeld et al., 2021), using the code

provided by the authors [1]. We additionally implement 3D latent codes for SPADE and SC-GAN, which do not originally support it, enabling local image editing for them.

GAN control methods. Ctrl-SIS is compared against two related latent discovery methods, GANSpace (Härkönen et al., 2020) and SeFa (Shen and Zhou, 2021), using the authors' code [2]. Following GANSpace-StyleGAN2 (Härkönen et al., 2020) and SeFA-StyleGAN2 (Shen and Zhou, 2021), we train all latent direction methods on features extracted from the normalization layers of each ResNet block in the generator.

Training details. Ctrl-SIS is trained with a batch size of 16 on a single NVIDIAv100 GPU, using the AdamW optimizer (Loshchilov and Hutter, 2017) and a learning rate of 1e-3. We train for 20 epochs on ADE20K, and 5 epochs on COCO-Stuff and CelebAMask-HQ, using $K = 5$. Finding class-specific directions with Ctrl-SIS takes ~1h. For evaluation we scale the directions with α sampled in $[-n; n]$ (see Sec. 5.2.2), to ensure that the direction magnitude is in the same range as the average latent code. By scaling the magnitudes of latent directions from all methods in the same way, we ensure that the effect of the edit only depends on the learned direction. For GANSpace and SeFa, we pick the directions corresponding to the first K components, as they cause the largest variations.

Image quality metrics. Following Isola et al. (2017), Park et al. (2019b), and Schönfeld et al. (2021), we monitor the visual quality of images generated with class-specific edits using FID (Heusel et al., 2017a) and mIoU metrics. FID is known to be well aligned with human judgement of image quality. mIoU assesses the alignment of images with ground truth label maps, calculated via a pretrained semantic segmentation network. We use UperNet101 (Xiao et al., 2018) for ADE20K and DeepLabV2 (Chen et al., 2015) for COCO-Stuff. In addition, we employ the precision and recall metrics of Kynkäänniemi et al. (2019), which correlate with image quality and diversity, respectively.

5.3.2 Evaluation of class-specific GAN controls

Prior GAN control methods were mostly evaluated by subjective visual inspection (Härkönen et al., 2020; Shen and Zhou, 2021; Yüksel et al., 2021). Consequently, it was challenging to assess the important properties of GAN control methods. In particular, a method for discovering semantically meaningful class-specific directions in the latent space of SIS GANs should exhibit the following three traits: First, the found directions should be as unique and different as possible. We assess this property via the *mean control diversity* - mCD. Second, a latent direction should invoke the same semantic edit independent of the initial latent code, which we

[1] Code for SIS models: SPADE, SC-GAN, OASIS
[2] Code for GAN control methods: GANSpace, SeFa

assess via the *mean control consistency* - mCC. Third, class-specific edits should not affect image areas outside of the target class area. We verify this requirement via the *mean outside class diversity* - mOD. The scores are based on computing the LPIPS distance between pairs of images with different edits and the same initial latent code (mCD and mOD), or the same edits but different initial latent codes (mCC). For the global mCD and mCC scores the edits are applied to all classes simultaneously with latent directions that are randomly picked from the set of discovered class-specific directions. On the other hand, the local scores mCD_l, mCC_l, and mOD rely on pairwise distances between images where only one class is edited at a time. To compute the pairwise distance between images where only one class is edited, we use the *masked* LPIPS distance. In the following, we explain the masked LPIPS distance and provide the formulations of the local scores mCD_l, mCC_l, and mOD, as well as the global scores mCD and mCC.

The masked LPIPS distance. The default LPIPS distance between two images is based on extracting deep features from both images using a VGG network pre-trained on ImageNet classification (Zhang et al., 2018c). The features of all layers are normalized and re-scaled along the channel dimension. The final LPIPS distance is the L2 distance between these features. To compute the *masked* LPIPS distance, we multiply the deep features with a binary mask before computing the L2 distance. We distinguish between LPIPS^{M_c} and LPIPS^{1-M_c}. The former uses the binary mask M_c, which is 1 where the label map contains class c and 0 everywhere else. The latter applies the inverted mask $1 - M_c$.

Mean control diversity. The mean control diversity is computed for global edits (mCD) and local edits (mCD_l). The mCD_l is computed via:

$$\text{mCD}_l = \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}_c \left[\mathcal{P}_{CD} \right], \tag{5.7}$$

where C is the total number of classes and $\mathcal{P}_{CD}$ denotes the control diversity measured for a label map containing class c. To compute $\mathcal{P}_{CD}$, a fixed initial latent code is sampled for each label map containing class c. Given a label map and its initial latent code, one locally edited image is created for each of the K latent directions specific to class c. Next, the average locally masked LPIPS distance is computed between all pairs of the K edited images. This score is averaged over Z initial latent codes, which can be formulated as follows:

$$\mathcal{P}_{CD} = \frac{1}{ZK} \sum_{z}^{Z} \sum_{\substack{k_{1,2}=1 \\ k_1 \neq k_2}}^{K} \text{LPIPS}^{M_c}_{z,k_1,k_2}. \tag{5.8}$$

Here, $\text{LPIPS}^{M_c}_{z,k_1,k_2}$ denotes the LPIPS distance masked with M_c between two images created with the same initial latent code z, where class c is edited with latent direction k_1 and k_2, respectively.

The mCD for global edits is computed as the average distance between globally edited images on the same label map. For each label map, we create pairs of images with different global edits, changing all classes at once. The class-specific latent directions are randomly chosen for each class. We compute the mean of the default LPIPS distance over all pairs and different initial latent codes. The score is averaged over all label maps in the test set. Higher mCD and mCD_l scores indicate better diversity.

Mean outside class diversity. The spatial disentanglement metric mOD is computed for local edits via

$$\text{mOD} = \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}_c \left[\mathcal{P}_{OD} \right], \tag{5.9}$$

where $\mathcal{P}_{OD}$ is the outside class diversity measured for a label map containing class c. In contrast to mCD_l, the masked LPIPS is computed for the area outside the target class:

$$\mathcal{P}_{OD} = \frac{1}{ZK} \sum_{z}^{Z} \sum_{\substack{k_{1,2}=1 \\ k_1 \neq k_2}}^{K} \text{LPIPS}_{z,k_1,k_2}^{1-M_c}. \tag{5.10}$$

$\text{LPIPS}_{z,k_1,k_2}^{1-M_c}$ denotes the LPIPS distance masked with $1 - M_c$ between two images created with the same initial latent code z, where class c is edited locally with the latent direction k_1 and k_2, respectively. A lower mOD indicates better spatial disentanglement.

Mean control consistency. Lastly, to measure the consistency of an edit under different initial latent codes, we compute the mean control consistency for global edits (mCC) and local edits (mCC_l). The mCC_l is

$$\text{mCC}_l = \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}_c \left[\mathcal{P}_{CC} \right], \tag{5.11}$$

where $\mathcal{P}_{CC}$ is the control consistency of a label map containing class c. We compute the pairwise distances between images with different initial latent codes and the same local edit:

$$\mathcal{P}_{CC} = \frac{1}{ZK} \sum_{k}^{K} \sum_{\substack{z_{1,2}=1 \\ z_1 \neq z_2}}^{Z} \text{LPIPS}_{k,z_1,z_2}^{M_c}. \tag{5.12}$$

Here, $\text{LPIPS}_{k,z_1,z_2}^{M_c}$ denotes the LPIPS distance masked with M_c between two images created with different initial latent codes z_1 and z_2, where class c is edited locally with latent direction k for both images.

The global mCC score is computed as the average distance between images with the same global edit but different initial latent codes. For each label map, we create pairs of images with different initial latent codes, but a shared global edit. We

compute the mean of the default LPIPS distance over all pairs and across different shared global edits. The score is averaged over all label maps in the test set. Ideally, the mCC and mCC_l are low, indicating high consistency under different initial latent codes.

Relation to prior diversity and disentanglement scores. The mCD_l and mCC_l are related to the *mean class diversity* (mCSD) and *mean other class* (mOCD) proposed by Zhu et al. (2020b). These two metrics evaluate diversity and spatial disentanglement for SIS models that allow class-specific manipulations (Zhu et al., 2020b; Schönfeld et al., 2021). Note that mCSD and mOCD measure the class-specific diversity and disentanglement of a *SIS model*, while our metrics evaluate the class-specific diversity and disentanglement of *a set of discovered latent directions*, allowing us to compare different control discovery methods on the same SIS model. The mCSD measures intra-class diversity as a property of the SIS model itself. In contrast, mCD_l measures the diversity of a set of latent directions, which is a property of the GAN control discovery method. The same relationship holds between mOCD and mOD. We next present an extended evaluation using our proposed local metrics mCD_l, mCC_l, and mOD.

Human evaluation. We also conduct a human evaluation of the learned latent directions. To this end, we employ the SHE score from Zhu et al. (2020b) and introduce a Human Diversity Rank (HDR) metric. For SHE, participants are shown two images edited only in the corresponding class area by applying the learned class-specific latent direction. The final SHE score is the percentage of image pairs that the participants judge to be semantically different in the area of only one class. For HDR, participants are shown rows of locally edited images from four different methods (Random, SeFa, GANSpace, Ctrl-SIS), as in Fig. 5.5 but in a randomized order. The task is to rank the methods by their diversity. The final HDR score is an average rank (range 1 to 4) assigned to a GAN control discovery method. Each participant is provided with 50 questions and unlimited answering time for both scores.

5.3.3 Main results

We compare Ctrl-SIS, GANSpace, and SeFa on global and local image editing. While local edits target a single class per image, global edits combine all class-specific edits within an image. Examples of local and global edits with Ctrl-SIS are shown in Fig. 5.3. The local edits show that the found directions encode semantic meaning, such aging faces, covering mountains in snow or turning on lamps (see Fig. 5.3 and 5.4). Global edits change the whole image globally and are the result of combining all class edits in one image (see last row of Fig. 5.3). In addition, we compare all methods to the performance of randomly sampled directions ("Random"), as well as to the performance on unedited images ("Baseline").

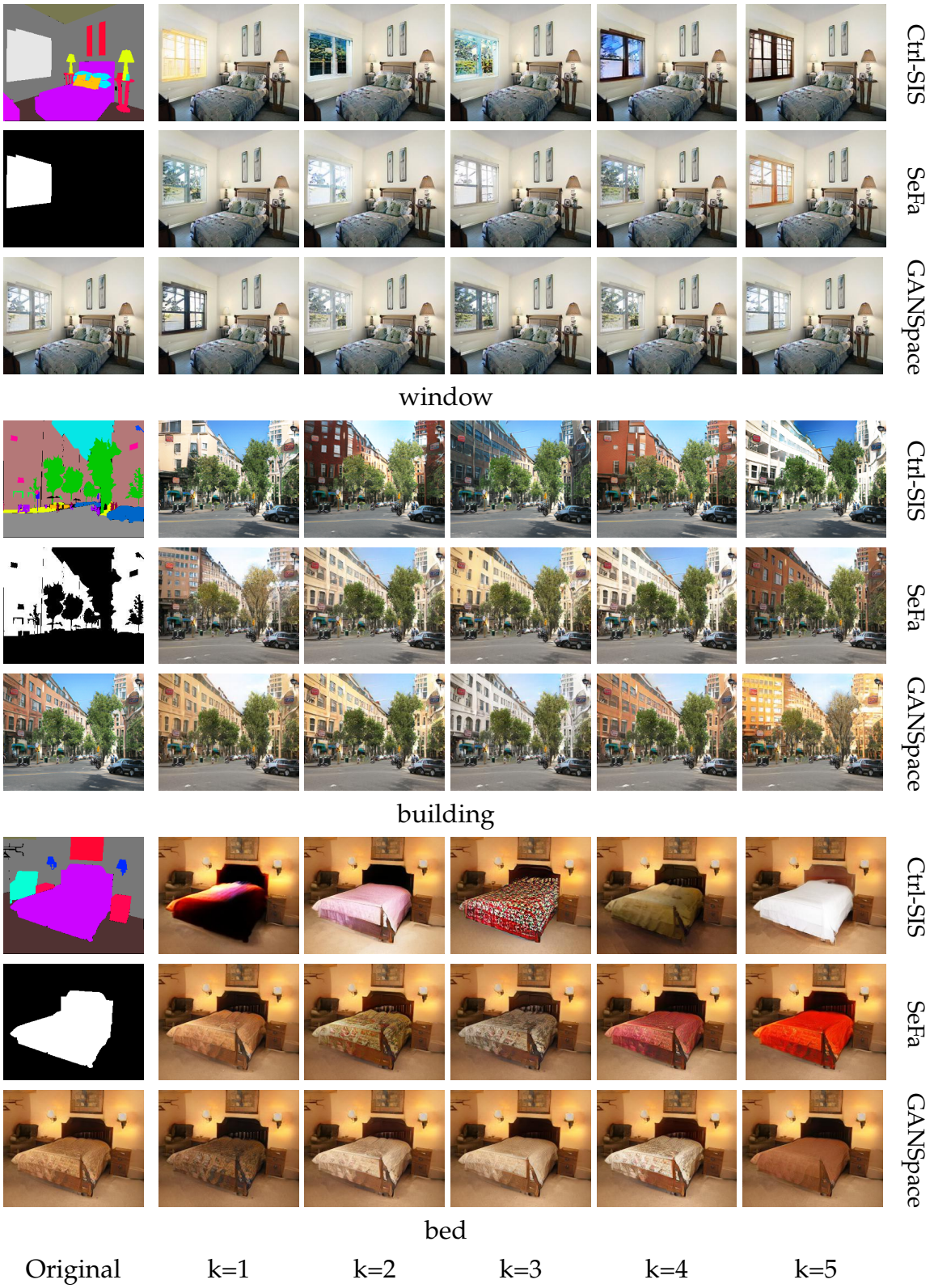

Figure 5.5: Qualitative comparison of Ctrl-SIS against SeFa and GANSpace. The learned directions $k = 1, ..., 5$ are applied for different classes. Ctrl-SIS *class-specific* latent directions result in more diverse edits for a selected class.

Method	ADE20K					COCO-Stuff				
	mCD ↑	mCC ↓	mOD ↓	FID↓	mIoU↑	mCD ↑	mCC ↓	mOD ↓	FID↓	mIoU↑
Baseline	-	-	-	28.6	52.2	-	-	-	**17.1**	42.4
Random	0.11	0.30	**0.01**	31.3	49.4	0.16	0.07	**0.00**	17.6	42.3
GANSpace	0.09	0.29	**0.01**	**28.1**	**53.3**	0.15	**0.06**	**0.00**	17.2	42.1
SeFa	0.12	**0.28**	**0.01**	**28.1**	53.2	0.15	**0.06**	**0.00**	**17.1**	**43.8**
Ctrl-SIS	**0.26**	**0.28**	**0.01**	30.9	48.9	**0.30**	0.07	0.01	21.1	43.6

Table 5.1: Evaluation of OASIS GAN controls on ADE20K and COCO-Stuff.

	mCD_l ↑	Precision↑	Recall↑	Human eval. SHE↑	HDR↓
Baseline	-	0.84	0.63	-	-
Random	0.04	0.82	0.62	32.9	2.62
GANSpace	0.03	**0.87**	0.61	29.1	3.43
SeFa	0.05	**0.87**	0.62	30.3	2.88
Ctrl-SIS	**0.12**	0.85	**0.64**	**60.7**	**1.07**

Table 5.2: Evaluation of local class-specific image edits on ADE20K with OASIS.

As seen in Table 5.1, Ctrl-SIS achieves improved diversity by at least a factor of two, e.g., mCD of 0.26 vs. 0.12 of SeFa on ADE20K. Interestingly, the diversity of GANSpace and SeFa is lower (underlined numbers) or close to random directions. Neither of these methods are designed to find class-specific directions. Yet, they still capture class-agnostic variations in the data, leading to directions that are closer to the mean of the image distribution and thus slightly better FID and mIoU. All methods exhibit similar consistency (mCC), with Ctrl-SIS and SeFa performing best on ADE20K, and SeFa and GANSpace on COCO-Stuff. The consistency of Ctrl-SIS is demonstrated in Fig. 5.4, e.g., where the learned latent direction consistently defoliates trees or covers streets in snow. Similar to the consistency, the disentanglement (mOD) is strong for all methods, due to the spatially disentangled 3D latent space of the OASIS model.

Due to the higher diversity of edited images, FID increases slightly for Ctrl-SIS compared to the baseline of unedited images (see Table 5.1). Since FID measures the overlap between the real and synthetic image distributions, images with weaker edits are closer to the original data. This is illustrated in Fig. 5.5, where edits are shown side-by-side for all methods. Since SeFa and GANSpace only minimally change the class, their FID is close to FID of unedited images (see Baseline in Table 5.1). Likewise, mIoU of images edited with Ctrl-SIS decreases, as the edited images move away from the mean mode of the synthetic image distribution. In contrast, for SeFa and GANSpace FID and mIoU are slightly better with respect to the baseline, while their diversity (mCC) is comparable to random directions. This observation suggests that SeFa and GANSpace images are closer to typical samples of the test set, while Ctrl-SIS learns more distinct directions.

Model	Method	Global edits			Local edits		
		mCD ↑	FID ↓	mIoU ↑	mCD_l ↑	FID ↓	mIoU ↑
OASIS	Random	0.11	31.3	49.4	0.04	30.6	50.1
	GANSpace	0.09	**28.1**	**53.3**	0.03	**28.3**	**53.9**
	SeFa	0.12	**28.1**	53.2	0.05	**28.3**	53.7
	Ctrl-SIS	**0.26**	30.9	48.9	**0.12**	28.8	51.6
SC-GAN	Random	0.08	34.3	38.1	0.05	**34.2**	38.6
	GANSpace	0.11	**34.2**	**38.3**	0.06	34.3	38.8
	SeFa	0.10	34.4	37.8	0.06	34.4	**38.9**
	Ctrl-SIS	**0.25**	36.4	34.7	**0.18**	**34.2**	38.4
SPADE	Random	0.08	**34.6**	**39.4**	0.05	34.6	39.6
	GANSpace	0.12	35.1	39.3	0.08	**34.6**	**39.7**
	SeFa	0.09	34.7	**39.4**	0.06	34.8	**39.7**
	Ctrl-SIS	**0.14**	35.4	38.6	**0.09**	**34.6**	39.4

Table 5.3: Comparison of GAN control methods across SIS models on ADE20K.

Method	mCD ↑	mCC ↓	mOD ↓	FID ↓	mIoU ↑
Ctrl-SIS	0.26	**0.28**	**0.01**	30.9	48.9
No $\mathcal{L}_{div}$	0.24	**0.28**	**0.01**	**30.5**	**49.4**
No $\mathcal{L}_{const}$	0.26	0.29	**0.01**	30.9	48.7
No $\mathcal{L}_{dis}$	**0.27**	**0.28**	0.02	31.6	48.3

Table 5.4: Loss ablation of Ctrl-SIS on ADE20K.

We perform alternative evaluations of local class-specific edits in Table 5.2. Ctrl-SIS shows the highest recall and diversity (mCD_l), and is the only method to improve both precision and recall over the OASIS baseline. Due to the precision-recall trade-off, SeFa and GANSpace have higher precision at the loss of recall, which is also reflected in their low mCD_l score and better FID over Baseline in Table 5.1. Moreover, both human diversity evaluation scores (SHE and HDR) are well aligned with the diversity metric mCD_l and recall, confirming the highest diversity of Ctrl-SIS.

Next, we compare Ctrl-SIS on different SIS models. Table 5.3 shows that Ctrl-SIS strongly improves diversity for local and global edits across all tested SIS models. The trade-off between diversity versus FID and mIoU is less pronounced for SPADE, which naturally suffers from lower sensitivity to input latent code due to the strong regularization effect of its perceptual loss, as shown in (Schönfeld et al., 2021) (see Chapter 4). While OASIS is trained without a perceptual loss, and SC-GAN uses a more powerful layer-wise conditioning strategy, leading to more diversity. Similarly to Table 5.1, the diversity of GANSpace and SeFa is comparable to random directions. In other words, the directions that SeFa and GANSpace find differ just as much from each other, as a set of randomly chosen directions. In contrast, the directions of Ctrl-SIS embody distinct appearances that are unlikely to appear in a

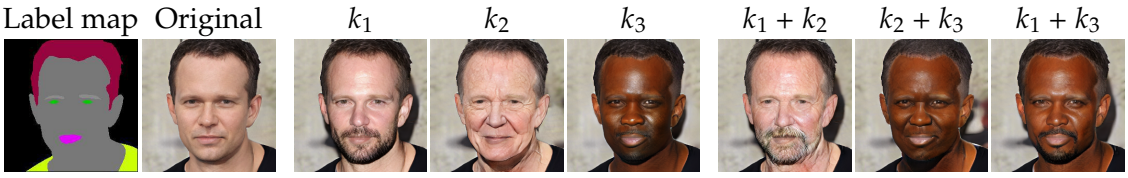

Figure 5.6: Combinations of directions k_1, k_2 and k_3 found for the *face* class (skin, neck, nose, ears) in the CelebAMask-HQ dataset. Different semantics can be combined, e.g., *k1 (beard)* and *k2 (age)* yield an old bearded person.

Figure 5.7: Optimizing Ctrl-SIS on the output features of ResNet blocks leads to an emphasis on structure (Res-Blk 1&2), while late normalization layers focus more on color (Norm-Blk 3&4). Directions from different layers can be combined: the last column combines Norm-Blk 4 with Res-Blk 1 (top) and 2 (bottom).

set of random directions.

Compositionality. Individual class-specific latent directions can be combined. For example, Fig. 5.6 shows that directions corresponding to "age" and "beard" can be combined into "old and bearded". Further, the latent directions found by Ctrl-SIS depend on the subset of feature layers $G_l(z, y)$ of the SIS generator G chosen for optimization (see Sec. 5.2). Fig. 5.7 highlights latent directions that were discovered by optimizing over layers from different ResNet blocks of the generator. For example, the set Norm-Block 4 in Fig. 5.7 minimizes the loss over the first convolution in all conditional normalization layers (Park et al., 2019b) within the fourth ResNet block. While for the set Res-Block 1 we minimized the loss for the final output features of the first ResNet block. We observe that the directions for Res-Block 1 and 2 differ strongly in the internal structure of semantic classes, while Norm-Block 3 and 4 encode changes in color. Interestingly, latent directions can be combined when synthesizing images, by injecting different directions in different layers. In the last column of Fig. 5.7, the directions of early ResNet blocks are injected into the first four layers of the SIS model, while directions from the conditional normalization layers of late ResNet blocks are injected from layer five onward. As the former directions encode structure, and the latter encode colors, the resulting image combines both aspects.

Ablation. Table 5.4 presents an ablation on the proposed objective, using OASIS on the ADE20K dataset. Without the diversity term of our loss function, the diversity decreases. Likewise, without the consistency or disentanglement term, consistency and disentanglement worsen. Further, the disentanglement term helps to improve synthesis and segmentation quality (FID and mIoU), by helping to restrict the area affected by the edit only to the selected class area.

5.4 Conclusion

We propose Ctrl-SIS, which to our knowledge, is the first method for discovering class-specific interpretable GAN controls of SIS models. This is achieved by optimizing a set of class-specific latent directions via proposed diversity, consistency, and disentanglement loss terms, making use of semantic label maps provided as part of the SIS task. The learned latent directions can locally change the appearance of targeted semantic classes without affecting other classes in the image, and can be combined to sequentially change the image. Quantitative and qualitative analysis shows that Ctrl-SIS results in image edits of high quality, that are significantly more diverse than prior methods adapted to SIS.

6 Conclusion and Future Perspectives

Contents

6.1 Discussion of contributions 103

6.1.1 Unconditional and class-conditional image synthesis with GANs 103

6.1.2 Semantic image synthesis with GANs 104

6.1.3 Discovering GAN controls for semantic image synthesis . 106

6.2 Future perspectives 107

6.2.1 Unconditional and class-conditional image synthesis with GANs 107

6.2.2 Semantic image synthesis with GANs 110

6.2.3 Discovering GAN controls for semantic image synthesis . 112

6.2.4 Broader outlook 113

At the current pace of progress in deep learning, impressive improvements in deep learning-based image synthesis can be seen on a year-to-year basis. The improvements in synthesis quality include higher resolution, a higher degree of detail, more diversity, better fidelity and correctness (e.g., dogs with no more or less than four legs), and the absence of artifacts introduced by the generation process. Improvements in synthesis controllability come from increased fidelity of conditional synthesis models, as well as better methods for image inversion (Richardson et al., 2021) and latent space control discovery (Härkönen et al., 2020; Shen and Zhou, 2021). In the area of high-fidelity conditional generation, significant strides have recently been made through large-scale training on ubiquitous text-image pairs from the web (Ramesh et al., 2021, 2022; Saharia et al., 2022; Yu et al., 2022; Rombach et al., 2022), which we also take into consideration in this chapter. These advances were also made possible through the use of transformers and diffusion models, which achieve the same or even better image synthesis quality as

GANs in some situations (Dhariwal and Nichol, 2021). In this thesis, we improved the image quality and controllability of GANs by working on three topics that we summarize in the following.

First, we focused on *unconditional and class-conditional image synthesis with GANs*. In Chapter 3, we proposed to rethink a GAN discriminator as a real-fake segmenter, based on a U-Net architecture. The new architecture more readily integrates features from all scales and image positions for the real-fake classification. The U-Net discriminator allows treating the discriminator prediction like an "image", which enables regularization that would not be possible otherwise, such as our proposed CutMix-based consistency regularization. Our discriminator improves synthesis quality by counteracting loss saturation (see Fig. 3.10) and self-supervision through our consistency regularization. The resulting U-Net GAN improved over the previous state-of-the-art BigGAN model and established a new state of the art among all models on the CelebA dataset.

Second, we focused on *semantic image synthesis with GANs*. In Chapter 4, we solved two core problems of previous semantic image synthesis (SIS) models: the necessity of a perceptual loss and the insensitivity to noise. We address both problems by introducing a segmentation-based discriminator. In addition, we strongly increase the noise sensitivity by proposing 3D noise injection. Our changes lead to a significant increase in synthesis quality and diversity and new possibilities in image manipulation due to the 3D structure of our noise. In addition to the standard evaluation, we propose an evaluation based on synthetic data augmentation, as well as a large-scale evaluation on the LVIS dataset. As shown in our LVIS experiments, we also successfully address dealing with imbalanced classes and sparse label maps, where previous work fails. The proposed OASIS model outperforms all previous state-of-the-art works.

Third, we focused on *discovering GAN controls for semantic image synthesis* - the automatic identification of meaningful directions in the latent space of SIS GANs. Previously, GAN control discovery methods existed only for unconditional and class-conditional GANs. However, the existing techniques are unsuitable for application in semantic image synthesis. Thus, in Chapter 5 we proposed such a method for semantic image synthesis, which learns a set of different controls for each class. These controls allow changing only one class in the image without affecting neighboring pixels. Since previous methods are primarily evaluated by subjective visual inspection, we also introduce evaluation metrics for diversity, consistency, and spatial disentanglement. The resulting method is termed Ctrl-SIS and can serve as a tool for manipulating synthetic images and inspecting the latent space of SIS GANs.

In this chapter, we discuss our contributions in more detail in Section 6.1. Lastly, we elaborate on open problems and future perspectives in Section 6.2.

6.1 Discussion of contributions

This thesis aimed to improve the image quality and controllability of GANs. To this end, we covered the three topics *unconditional and class-conditional image synthesis with GANs*, *semantic image synthesis with GANs*, and *discovering GAN controls for semantic image synthesis* in separate chapters. In the following, we summarize the contributions of each chapter.

6.1.1 Unconditional and class-conditional image synthesis with GANs

In Chapter 3, we focused on unconditional and class-conditional image synthesis with GANs. The goal was to improve image quality by redesigning the GAN discriminator. The key contribution is the introduction of a segmentation-based discriminator and a CutMix-based regularization enabled by the new discriminator design. Previously, classification-based discriminators were the norm for GAN architectures. On the other hand, we base our discriminator on the popular U-Net segmentation network. In contrast to a standard U-Net segmentation network, we also compute the loss at the bottleneck layer between encoder and decoder. The advantages of a U-Net discriminator are two-fold:

First, it performs local real-fake classifications of image patches, taking into account the larger image context. In contrast, a classification-based discriminator can easily classify an image as real or fake by only considering one distinctive trait. Consequently, the U-Net discriminator loss saturates much slower during training, since a small loss would require uniform predictions over all image pixels. Likewise, the loss predictions for the encoder and decoder also differ, contributing to the slower loss saturation. In the BigGAN baseline, on the other hand, the generator stops improving earlier due to faster loss saturation.

Second, the fact that the discriminator output is two-dimensional with the same height and width as the input image allows for new forms of regularization. We show that the new design enables introducing the nonleaking CutMix data augmentation, as well as a consistency regularization promoting equivariance under the CutMix transformation. The CutMix equivariance is an inductive bias that prevents overfitting to changes in the image that do not affect the realness of patches. For example, the CutMix regularization discourages local real-fake classifications from taking shortcuts by copying the real-fake classification of neighboring patches. Note that the CutMix augmentation is nonleaking, i.e., it avoids the known problem of augmentation leakage in GANs (Zhao et al., 2020a; Karras et al., 2020a), due to the unique property of computing the GAN loss in 2D.

Importantly, almost any GAN discriminator can be converted into a U-Net discriminator, without changing the base architecture or the loss functions. When

it comes to the architecture, the only necessary modification is to add a decoder to the already existing encoder-shaped discriminator net. With respect to the GAN loss, we showed that different GAN losses are trivial to extend to a U-Net decoder loss, such as the hinge loss, binary cross-entropy, and the class-conditional projection loss. Although our experiments were conducted on a BigGAN backbone, the U-Net discriminator has also been successfully implemented for StyleGAN2, where it works well in combination with the standard StyleGAN2 perceptual path length regularization and R1 regularization[1]. Interestingly, we observed that the self-attention layers of BigGAN do not contribute to performance in the U-Net GAN setting, likely because the U-Net architecture already achieves a similar effect of integrating information from distant image regions.

With our proposed changes to the discriminator, we achieved an average improvement of 2.7 FID compared to the previous state-of-the-art baseline BigGAN, measured on three different datasets. Thereby, U-Net GAN improves performance in both the unconditional and class-conditional setting. To the best of our knowledge, our model also achieved the best FID score on the CelebA dataset (2.95) at the time of publication. Our publication (Schönfeld et al., 2020) is the first to propose a segmentation-based discriminator, opening up further opportunities for research. Since our publication, the U-Net discriminator has also been popular in GAN-based super-resolution (Zhang et al., 2020c; Wang et al., 2021b; Jo et al., 2020; Wei et al., 2021; Wang, 2021; Li et al., 2021a). A widely-used super-resolution model that employs the U-Net discriminator is Real-ESR GAN (Wang et al., 2021b), which at the time of writing counts more than 16K stars on github[2].

6.1.2 Semantic image synthesis with GANs

In Chapter 4, we focused on semantic image synthesis (SIS), the task of converting label maps to images. Prior to our work, semantic image synthesis suffered from two entangled problems: First, SIS generators had low sensitivity to input noise and, therefore, severely limited diversity compared to unconditional GANs. Second, training a pure GAN was insufficient in previous architectures to achieve good image quality. Additional losses or discriminators were used by default. In particular, all previous works strongly depended on the perceptual loss, which unnecessarily constrains the generator and is partly responsible for the first problem. We addressed these two problems by introducing the OASIS model, and summarize the associated contributions in the following.

We showed in our experiments that the perceptual loss significantly limits the diversity of SIS models. The reason is that the perceptual loss acts like a reconstruction loss, but in VGG feature space. Consequently, a one-to-one matching between the synthetic and real image is encouraged, independent of the input

[1] https://github.com/lucidrains/unet-stylegan2/
[2] https://github.com/xinntao/Real-ESRGAN

noise, despite the fact that SIS requires a one-to-many mapping. Yet, without the perceptual loss, the image quality of the previous state-of-the-art baseline degrades strongly, e.g., from 21.7 to 99.1 FID for COCO-Stuff. Our ablations showed that the discriminator is the root of the problem. Notably, without the perceptual loss, training of the SPADE baseline collapses unless multiple discriminators are used.

We solved the problems mentioned above by using a segmentation-based discriminator. The explanatory insight is that it is more appropriate to use the label map as a loss target than to use it as discriminator input: As input, the label map can be ignored. As loss targets, labels cannot be ignored and provide strong supervision that was previously lacking. In effect, we showed that the perceptual loss becomes superfluous and that image quality and diversity are improved out-of-the-box, compared to previous models using the perceptual loss. Compared to U-Net GAN, the OASIS discriminator employs an N+1 cross-entropy loss as GAN objective and uses only the decoder loss, making OASIS the first GAN to employ a "pure" segmentation loss as GAN objective. Importantly, the OASIS discriminator unlocks new capabilities: Compared to all other discriminators, the OASIS discriminator can balance the loss among individual classes. The pixel-wise loss of underrepresented classes, i.e., rare or small objects, can be given a higher weight, which benefits the overall image quality by boosting the synthesis quality of underrepresented classes. Further, the OASIS discriminator enables our proposed LabelMix regularization, which improves synthesis quality. Interestingly, we demonstrate that the discriminator can be used as a regular segmenter after training, which allows the generator to resynthesize unannotated images with different textures.

Next to the discriminator design, we also overhauled the generator by proposing 3D noise injection. While previous works used the noise vector as input to the first generator layer, we inject a 3D noise tensor together with the label map into a spatially sensitive conditional batch normalization layer at every generator block. This setup makes the noise hard to ignore and strongly improves diversity, in addition to the diversity boost resulting from abandoning the perceptual loss. Our generator also improves the controllability of the synthesis process, since the 3D noise allows resampling images globally or locally at different spatial positions.

Taken together, the new discriminator and generator design constitute the OASIS model. We showed that OASIS outperforms all previous works on the standard benchmark datasets. We achieved improvements in terms of image quality, alignment with the label maps, and synthesis diversity. Further, we proposed to evaluate the SIS model via synthetic data augmentation, using synthesized images and their label maps as additional data for semantic segmentation. We observed a stonger increase in segmentation performance compared to the SPADE baseline, suggesting a potential use of SIS models for data augmentation in semantic segmentation.

In addition to the standard benchmark datasets, we were the first to compare SIS models on the LVIS dataset and specifically propose to use it to evaluate performance in a large-scale setting (more than 1000 classes), strong class imbalance, and

sparse annotations. An evaluation in a data regime that is closer to a real-world use case has been previously missing. We showed that OASIS significantly outperforms previous work on the LVIS dataset, lowering FID by 43%. Importantly, we demonstrated that OASIS works well even on very sparsely annotated label maps, where the previous state-of-the-art baseline collapses. The fact that OASIS works on sparse label maps can be attributed to the 3D noise. After all, the SPADE baseline has no significant source of stochasticity and therefore regards two different sparse label maps as almost identical, leading to collapsed predictions. Further, OASIS better synthesizes underrepresented classes under the heavy class-inbalance of LVIS, thanks to the class-balancing enabled by the OASIS discriminator design.

Lastly, it is important to note the similarity between OASIS and common class-conditional GANs, like BigGAN. First, like BigGAN, OASIS concatenates noise and class labels and uses them for conditional batch normalization at every layer. The difference is that in OASIS, the noise, the labels, and the conditional batch normalization have the spatial dimensions of height and width. Interestingly, previous SIS works did not adhere to a layerwise conditional normalization scheme, which is common in unconditional and class-conditional GANs. Part of the reason may be the damping effect of the perceptual loss on noise sensitivity. Moreover, it is common in class-conditional GANs to use the class label as a loss target via an additional classification head, or to inject it in the last layer to compute the loss input. Only the very first conditional GANs used the class label directly as input to the discriminator (Mirza and Osindero, 2014). Yet, all previous SIS models used the label map directly as discriminator input, since there is no straightforward way of incorporating the label map into the loss of a classification-based discriminator. Therefore, through our proposed changes, SIS GANs have become more similar to regular class-conditional GANs, which work perfectly well without multiple discriminators or perceptual losses, and enjoy high image diversity.

6.1.3 Discovering GAN controls for semantic image synthesis

In Chapter 5, we proposed a method for discovering class-specific semantically meaningful directions in the latent space of SIS GANs. Our technique allows to analyze the latent space of SIS GANs and provides a user with controls for class-specific edits in synthetic images. Previous works on GAN control discovery focused on unconditional GANs, and were therefore not designed to deal with image generation in the presence of label maps. In particular, the baselines evaluated in Chapter 5 yield latent directions that are comparable to completely random directions. Another line of work focuses on controlling the appearance of SIS images via style transfer from a reference image, rather than identifying latent directions. One reason why latent direction discovery for SIS GANs remained unexplored is the very limited diversity in the latent space of previous SIS GANs. Hence, the strongly improved diversity of OASIS enabled our work in Chapter 5.

In particular, we propose a method named Ctrl-SIS, which is the first method to apply GAN control discovery to SIS. In addition, Ctrl-SIS is the only method that discovers *class-specific* directions. We identify three properties of a good SIS GAN control discovery method: The class-specific discovered directions should be all uniquely different (diversity), evoke the same semantic changes regardless of the initial conditions (consistency), and should only affect the selected class (disentanglement). In contrast to previous works, Ctrl-SIS specifically optimizes for these three properties using dedicated losses.

Since previous works also primarily evaluate their method via subjective visual inspection, we propose evaluation metrics to assess the three properties mentioned above. We thus show quantitatively, but also qualitatively, that Ctrl-SIS identifies directions that are consistent, spatially disentangled, and significantly more diverse than the tested baselines. In addition, we demonstrate that the learned directions can be combined on the same or different objects. When two directions are applied on the same object, the individual semantics of both directions are preserved. For example, "old" and "beard" combine to "old and bearded" for the face class.

In conclusion, the controls identified by Ctrl-SIS allow a user to edit synthesized images in a predefined manner for specific classes. Additionally, Ctrl-SIS can provide a user with insight into the representations that the SIS model learns: First, Ctrl-SIS allows a user to visually assess the diversity that the SIS GAN model managed to capture for every class, which can help identify which classes and class-appearances the SIS model struggles to learn. Second, it enables a user to see which visual aspects each generator layer focuses on, as well as how sensitive each generator layer is to changes in the latent space. In doing so, Ctrl-SIS may help with the design of better SIS models.

6.2 Future perspectives

This thesis presented works to improve the synthesis quality and controllability of GANs. Achieving highly detailed image synthesis that accurately follows a user's input still requires much work. This section discusses future steps in the research topics covered in this thesis. Finally, we give a broader outlook for the general field of controllable image synthesis. In doing so, we do not limit ourselves to GANs, but also discuss current transformer and diffusion models.

6.2.1 Unconditional and class-conditional image synthesis with GANs

In the following, we discuss possible further steps for unconditional and class-conditional GANs, including U-Net GAN (see Chapter 3).

Pretrained feature extractors for the discriminator. GAN training benefits from an increased amount of data, but large-scale training is time-consuming. It is therefore particularly useful to incorporate networks pretrained on massive amounts of data. The data used for pretraining may be unlabeled, while the downstream GAN is trained on a smaller set of labeled data. Pretrained feature extractors have been widely used in GANs for image-to-image translation and semantic image synthesis (Wang et al., 2018a; Park et al., 2019b), but not in unconditional or class-conditional GANs. Only recently, pretrained networks were incorporated into the GAN discriminator of unconditional and class-conditional GANs: Kumari et al. (2022) form an ensemble between the original discriminator and several different pretrained feature extractors to classify images into real and fake. On the other hand, ProjectedGAN (Sauer et al., 2021) and StyleGAN-XL (Sauer et al., 2022) feed the real and fake images into a fixed feature extractor. The feature channels are mixed via random projections before being processed as a U-Net. Multiple discriminators classify the features of each individual layer. It would be straightforward to adapt this principle to U-Net GAN, by exchanging the encoder with a pretrained network and adding random projections. After all, ProjectedGAN has been shown to train faster by an order of magnitude while also achieving better FIDs (Sauer et al., 2021). Yet, in this particular case it is not clear if the better FID really corresponds to better images, since the visual results do not clearly confirm the improvement in FID. More research would be helpful to better understand the effect of using pretrained feature extractors in GAN discriminators.

Transfer learning for the generator. Next to the discriminator, the generator also benefits from pretrained networks. The most straightforward approach is to fine-tune a set of layers of a pretrained GAN on a new dataset (Wang et al., 2018b; Mo et al., 2020). Unbalanced GAN (Ham et al., 2020) reuses the trained decoder of a VAE as a GAN generator. Orthogonally, Baek and Shim (2022) proposes to pretrain a generator on potentially unlimited simulated data of visual primitives, consisting of shapes, colors, and textures. Interestingly, this approach outperforms models pretrained on real data on the task of few-shot learning from a new dataset of only 100 images. However, while discriminators can use any off-the-shelf pretrained feature extractor, transfer learning for generators is currently limited to reusing pretrained generators of the exact same architecture. Future research is needed to devise more flexible and general methods for transfer learning in GAN generators. Amongst others, there may be room for improvement for pretraining with simulated (and perhaps more realistic) data beyond the visual primitives chosen by Baek and Shim (2022).

Continual learning. Continual learning addresses the problem of catastrophic forgetting (French, 1999). Since training is expensive and time-consuming, a lot of time and money could be saved if GANs could be trained on new incoming data without having to retrain from scratch. Lifelong GAN (Zhai et al., 2019) uses

knowledge distillation (Hinton et al., 2015) to enable producing images learned in previous tasks. More recently, Cong et al. (2020) show that a pretrained GAN can learn from different datasets by freezing all weights and only retraining the style-modulation layers for each dataset. A similar approach is taken by Jain et al. (2022), to equip OASIS (see Chapter 4) with continual learning capabilities. However, it remains to be seen whether popular continual learning techniques like *Elastic Weight Consolidation* (Kirkpatrick et al., 2017) can be applied to GANs and whether learning from a continuous stream of data will be feasible at some point in the future.

Transformer-based architectures. Almost all GANs are built on convolutions. However, recent GANs combine convolutions with transformers (Hudson and Zitnick, 2021; Arad Hudson and Zitnick, 2021) or are entirely transformer-based (Lee et al., 2021; Jiang et al., 2021; Zhao et al., 2021). ViT-GAN (Lee et al., 2021), TransGAN (Jiang et al., 2021), and HiT (Zhao et al., 2021) use a vision transformer (Dosovitskiy et al., 2020) for both generator and discriminator, which models a sequence of flattened image patches. Transformers may be especially well suited for learning the interdependence between spatially separated image patches. On the other hand, GANformer 1 and 2 (Hudson and Zitnick, 2021; Arad Hudson and Zitnick, 2021) implement attention between the latent codes and the generator features, as well as between learned embeddings and discriminator features. These transformer-based GANs each improved synthesis quality over their respective baselines and developing them further may lead to even better-performing GANs.

Positional embeddings and implicit neural representations. A recent development is the use of positional embeddings as input to the generator, allowing the generation of arbitrarily sized or shaped images (Ntavelis et al., 2022; Skorokhodov et al., 2021b,a; Lin et al., 2021), out-of-the-box superresolution (Skorokhodov et al., 2021a; Ntavelis et al., 2022), out-of-the-box outpainting (Skorokhodov et al., 2021a), filling the gap between two images (Lin et al., 2021), image translation (Karras et al., 2021b), and arbitrary geometric transformation (Ntavelis et al., 2022). Some of these works go a step further and base the generator entirely on implicit neural representations (INRs), which are MLPs that produce an RGB value for a given positional coordinate (Anokhin et al., 2021; Skorokhodov et al., 2021a; Lin et al., 2021; Karras et al., 2021b). Depending on the implementation, INR-based generators have the additional advantage that image generation can be parallelized, since image patches can be produced independently. In doing so, InfinityGAN (Lin et al., 2021) reports a 7.2× inference speed-up. Besides, StyleGAN3 demonstrates how using MLP generator layers implemented via 1×1 convolutions allows achieving rotation equivariance. Despite the listed advantages, further research is needed to better synchronize local and global structure in image generation of arbitrary scale and size. In particular, spatial embeddings need to effectively encode positions along wide distances while preventing repeated motives, e.g., for panorama im-

ages. Next to better spatial embeddings, regularization schemes can be improved for better consistency between global and local structure. These developments will help to generate large high-resolution images with a lot of detail. INR-based and transformer-based generators appear to perform best when paired with a convolutional discriminator (Lee et al., 2021; Anokhin et al., 2021), suggesting there is room for improvement when it comes to incorporating new components into discriminator architectures.

Data augmentation. Data augmentation is widely used in most areas of deep learning. However, common augmentations like color jitter, adding noise, or cutting out parts of the image leak into the synthetic images (Karras et al., 2020a). For example, adding noise during training would lead to noisy synthetic images. The solution is to use differentiable data augmentation, where augmentations are not just applied to real data, but to both synthetic and real images in the D-step, as well as in the G-step (Karras et al., 2020a; Zhao et al., 2020a). However, as shown in Chapter 3, the CutMix augmentation is not leaking in U-Net GAN, despite not being used as a differentiable augmentation. U-Net GAN achieves this by setting the loss target for augmented images to fake in the encoder loss, while the decoder loss evaluates individual pixels and does therefore not leak. In future work, it may be worth exploring other augmentations based on cutting and mixing images, as well as geometric transformations, which should theoretically not leak through the U-Net decoder loss. These augmentations should additionally be tested for consistency regularization and also under the differentiable augmentation framework.

6.2.2 Semantic image synthesis with GANs

Semantic image synthesis is the focus of Chapter 4 and 5. In this section, we discuss future steps to increase the performance of semantic image synthesis models.

Applying methods from the unconditional GAN literature. Note that all future steps discussed in the previous subsection on unconditional and class-conditional (Sec. 6.2.1) also apply to SIS GANs. This includes leveraging pretrained networks, e.g., general feature extractors, continual learning, new architectural components like transformers or implicit neural representations, and data augmentation. Importantly, many of these things have not yet been applied to semantic image synthesis. Pretrained networks have been used in the form of the VGG perceptual loss, which limits synthesis diversity (see Chapter4). However, incorporating a pretrained network as part of the discriminator weights in the spirit of ProjectedGAN (Sauer et al., 2021) would not reduce diversity, since no matching between the real and fake VGG image features takes place. After all, the enforced one-to-one mapping independent of the noise is the reason for the diversity reduction seen in previous works. Further, OASIS has been extended to support continual learning (Jain et al., 2022), but much more work remains to be done. Moreover, vision

transformers and MLP-based generators still remain to be tried on semantic image synthesis GANS. In addition, self-supervised losses like the rotation loss (Chen et al., 2019) or the consistency loss of Zhao et al. (2020b) have not been applied to SIS GANs, despite the success in unconditional and class-conditional GANs. Lastly, the widely used differentiable data augmentation (Zhao et al., 2020a; Karras et al., 2020a) has yet to be applied to SIS GANs. Differentiable data augmentation may strongly impact performance, since the size of segmentation datasets is typically small due to the high annotation cost.

Semi-supervised learning. Segmentation-based discriminators in SIS can make use of the rich literature in the field of semi-supervised segmentation. Consequently, SIS performance could increase by making use of a large number of unlabeled images. In this case, the discriminator would learn to better segment images and could therefore give better feedback to the generator. Another self-supervised option would be to add an auxiliary binary real-fake classification head for unlabeled images. In doing so, OASIS could directly follow the U-Net GAN design, where the encoder part of the discriminator serves as a binary real-fake classifier.

Alternative segmentation losses. Next to the standard cross-entropy loss for semantic segmentation, alternative losses have been formulated such as the focal loss (Lin et al., 2017), lovász-softmax loss (Berman et al., 2018), region mutual information loss (Zhao et al., 2019), or poly loss (Leng et al., 2021). Future steps for SIS with segmentation-based discriminators may resort to one of these losses, or a combination thereof. Note that in our ablations in Chapter 4 we also demonstrated that a segmentation loss based on a pixel-wise projection loss (Miyato and Koyama, 2018) yields acceptable results, while this loss is not used in the segmentation literature. Thus, it is possible for follow-up work to use another known segmentation loss or propose a new loss function for the segmentation-based discriminator. Lastly, a segmentation loss that is beneficial for OASIS or follow-up work in semantic image synthesis may also be beneficial for U-Net based discriminators in unconditional and class-conditional GANs.

GAN inversion for semantic image synthesis. GAN inversion is a powerful tool that allows editing a real image, by encoding it into the latent space of a generator and re-synthesizing it with edits (Richardson et al., 2021). This principle could be applied to segmentation-based discriminators as well. As we show in Chapter 4, OASIS can already predict the label map of an *unlabeled* image and recreate it in a different style. Ideas from the GAN inversion literature can be used to also encode the style of an input image, allowing it to encode and decode an image accurately. Before decoding, the image can be edited in the predicted label map and the encoded style.

Unifying semantic image synthesis and image-to-image translation. Semantic image synthesis is a specific case of image-to-image translation, which also includes

sketch-to-image, image inpainting, image denoising, simulation-to-real, image colorization, and other tasks. While OASIS is tailored to use a semantic segmentation loss, it is not straightforward to use other forms of spatial conditional information directly in the GAN loss, e.g., hand drawn sketches. That is why general image-to-image models used the label map and other forms of spatial conditioning as direct input to the discriminator (Isola et al., 2017). Yet, the fact that segmentation-based discriminators output a spatial prediction may help to incorporate spatial conditions in future research. Interestingly, GauGAN2 (Huang et al., 2022) performs image-to-image translation from sketches and label maps, and incorporates the 2D input condition into the GAN loss via projection (Miyato and Koyama, 2018). Since a projection-based segmentation loss works for OASIS, it may be possible to use the findings of Huang et al. (2022) to enable general image-to-image translation with OASIS.

6.2.3 Discovering GAN controls for semantic image synthesis

In Chapter 5, we present a method for discovering semantically meaningful class-specific latent directions in the latent space of semantic image synthesis models. In the following, we discuss future steps for this task.

Contrastive learning. Contrastive learning (Chen et al., 2020b) is a highly effective unsupervised method for learning representations, based on the similarity and differences of features. This principle can be exploited to discover unique latent directions, by letting the directions take the role of the features in a contrastive objective, as demonstrated in LatentCLR (Yüksel et al., 2021). Adapting this approach to GAN control discovery to semantic image synthesis may be an interesting next step.

Closed-form solutions. Methods like SeFa (Shen and Zhou, 2021) and ReSeFa (Zhu et al., 2022) have the advantage of finding meaningful directions in seconds. However, it is not clear how to extend these methods to find class-specific directions for semantic image synthesis. In particular, it is not clear how to factorize out the image changes caused by the noise from those caused by the label. In addition, a method ideally finds general directions, rather than requiring test-time optimization on single images like Zhu et al. (2022). However, finding a practical solution to discovering general class-specific latent directions for SIS based on closed-form methods would be of great value.

More diverse semantic image synthesis models. The more diverse the semantic image synthesis model is, the more useful methods such as Ctrl-SIS will be. It was the improved diversity through our proposed 3D noise (see Chapter 4), which enabled our investigation of the SIS GAN latent space (see Chapter 5). Follow-up work will likely find ways to improve diversity further. Additionally, revisiting

older works on improving diversity (Zhu et al., 2017) may already provide solutions.

6.2.4 Broader outlook

In this section, we comment on the longer-term perspective of the field of controllable image synthesis. For this, we want to answer the question of where the field is headed and how to get there. If we had to define an end goal for the field of controllable image synthesis, it might look as follows:

Goal of the field of controllable image synthesis: *Given any description of the image content, be it in the form of text, sketches, label maps, or others, we can synthesize an image with exactly the specified content. Different forms of image descriptions can be combined. Noise can generate seemingly endless image variations, respecting the descriptions. Noise can also be applied alone to yield completely random images. The image looks perfectly realistic, and we can specify any output shape and resolution. If the model does not synthesize the image we have in mind on the first attempt, we can iteratively refine the image until we see exactly what we want. The refinement includes changing images only in certain areas without affecting the rest of the image. We can also show the synthesis model new examples of unseen objects, and it will be able to remember and synthesize them.*

Thus, the two key axes along which models must improve are *quality* and *controllability*, which were at the center of this thesis. Recent developments in image synthesis indicate that the goal outlined above may be achievable. Particularly, text-to-image models such as DALL-E (Ramesh et al., 2021), DALL-E 2 (Ramesh et al., 2022), Imagen (Saharia et al., 2022), Parti (Yu et al., 2022), and latent diffusion (Rombach et al., 2022) demonstrate high image quality, controllable through text. These models demonstrate compositionality and zero-shot capabilities. For example, they allow generating convincing "avocado chairs" or people, objects, and other concepts in unusual combinations (see Fig. 6.1). The stunning results suggest that scaling up the size of the dataset and the number of parameters plays a crucial role in achieving the goals of controllable image synthesis. If we believe in Richard Sutton's "bitter lesson"[3], then models that can best leverage computation will outperform solutions built on specialized domain knowledge by a large margin. Following this reasoning, big strides towards our goal will come from inventing more efficient and scalable architectures, leveraging massive datasets, and strategies that increase a model's exposure to more data, such as semi-supervised learning and transfer learning. On the other hand, some problems are not solved by scale alone, but by how models deal with problems inherent to the data. For example, the image content in large datasets follows a long-tailed distribution. Being able to generate rare content with similar quality as well-represented content requires algorithmic innovation. Hereafter, we discuss future developments in the light of the aforementioned aspects.

[3] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Consolidation of methods across modalities. Recent text-to-image models have achieved impressive results thanks to scale. The fact that text-image pairs are readily available on the web allowed for the creation of datasets with 250 million (Ramesh et al., 2021), 400 million (Schuhmann et al., 2021), and very recently 5 billion (Schuhmann et al.) samples. More data also means that models must scale in capacity through increased parameter counts. Since training such large models is expensive, any source of dysfunction must be eliminated. Complex models with many moving parts may contain a component that malfunctions at larger scales, and therefore compromises the ability of the model to scale well. Thus, Yu et al. (2022) argue with regard to their Parti model (see Fig. 6.1) that it is beneficial for scaling if a model is conceptually simple. Accordingly, they propose a text-to-image pipeline built entirely from standard transformers, making scaling to 20B parameters straightforward. Transformers are routinely used in natural language processing and are increasingly applied in computer vision. The consolidation of architectures and training methods for vision and language is likely to continue in the near future, and improvements in the field of language models will translate into improvements in image synthesis, and vice versa.

Multi-modal conditioning. Another form of consolidation is models that can work simultaneously with different types of conditioning information. A pioneering work is Product-of-Expert GAN (PoE-GAN), also known as GauGAN2 (Huang et al., 2022), which can generate images from text, sketches, and label maps, or a combination thereof. The model acts as an unconditional GAN when no conditioning input is given. Training with multi-modal conditioning faces several challenges. Some modalities have to be ignored when the modalities contain contradicting information. Huang et al. (2022) show that their model performs better when trained with unimodal data, indicating much room for improvement when it comes to fusing multi-modal data. Importantly, acquiring multi-modal annotations is especially expensive: More annotations are required, and the content of different modalities has to be well aligned, too. One solution may be to train with unaligned data and learn the alignment implicitly. Another problem to solve is dealing with unbalanced annotations between modalities: There will be more text and image pairs than sketch and image pairs. Learning under such annotation-imbalance amounts to a particular form of semi-supervised learning, where some images miss some form of annotation, while other images miss another form of annotation. Successfully addressing these challenges will lead to powerful content creation tools.

Parameter-efficient models. The results of the aforementioned Parti model are among the best the field currently has to offer. Especially the correct rendering and placement of written text in images is unprecedented, as shown in Figure 6.1. Figure 6.1 also demonstrates the necessity of a large number of parameters to generalize well. Parti is built entirely from transformers. While several efficient

transformers have been proposed, many of them do not scale well (Tay et al., 2022). More research is necessary to achieve such impressive results as in recent text-to-image models using a smaller parameter count.

A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says "Welcome Friends!".

A green sign that says "Very Deep Learning" and is at the edge of the Grand Canyon. Puffy white clouds are in the sky.

Figure 6.1: Scaling behavior of the Parti model (Yu et al., 2022) from 350M to 20B parameters. Higher parameter counts result in better image quality and higher alignment with the input prompt. High fidelity placement of embedded text only occurs at 20B parameters.

Scaling up GANs. Recent breakthroughs in text-to-image generation were built on transformers and diffusion models. The GAN loss is mainly used in combination with a pixel-wise reconstruction loss to train the decoder that translates the compressed image representation into pixel space (Yu et al., 2022; Rombach et al., 2022; Ding et al., 2022; Wu et al., 2022). Transformer-based models that do not use a GAN loss for their decoders produce blurry images (Ramesh et al., 2021). The likely reason for the widespread use of transformers and diffusion models is the better scaling properties. GANs still suffer from more training instability than other models. The performance and training stability of GANs on large-scale datasets of hundreds of millions of images is still underexplored and an interesting topic for future research.

Transfer learning via pretrained models. The success seen in text-to-image synthesis is unlikely to directly translate to semantic image synthesis, since the training data is more limited. It is not possible to scrape millions or even billions of pairs of label maps and images from the web. The same holds true for any type of conditioning information, apart from text. For conditional image synthesis models to

profit from the effect of scale, they could leverage pretrained networks. A purely GAN-based model may make use of a pretrained feature extractor in the discriminator, similar to Sauer et al. (2021, 2022), who apply this principle to unconditional and class-conditional GANs. On the other hand, diffusion-based conditional image synthesis models can benefit from finetuning a diffusion model that was trained on a large unlabeled dataset, as shown by Wang et al. (2022a) for the task of semantic image synthesis. This idea may also be tried for GANs, by finetuning a large pretrained unconditional GAN on the conditional image synthesis task.

Semi-supervised learning. Another strategy for conditional image synthesis tasks, besides the data-rich text-to-image task, is the leverage of semi-supervised learning. Recently, OSSGAN (Katsumata et al., 2022) proposed a model for class-conditional image synthesis in an "open set" setting. In the open set task, the model is trained jointly on a small set of labeled images and a large set of unlabeled images with classes that are not contained in the labeled image set. Straightforward strategies for open set learning are either using pseudo-labels, or using different losses for labeled and unlabeled images. It would be useful to try and further develop semi-supervised strategies for semantic image synthesis and other conditional image generation tasks.

Strategies for long-tailed data. Almost all datasets follow a long-tailed distribution, with the exception of artificially balanced datasets. However, artificially balancing large datasets by their content is not possible. For example, it is impossible to balance a 400M dataset of text and image pairs to contain every concept or combination of concepts equally often. Likewise, a dataset of pairs of label maps and images will always contain classes that are far more rare than others, or have vastly different object sizes. In effect, text-to-image models can generate pictures of Obama in many different contexts but struggle to render celebrities that are only known in some countries or regions. Semantic image synthesis models perform well on buildings but poorly on the grandstand class. This motivates the per-class balancing in OASIS (see Chapter 4). Much future research is needed to better model underrepresented classes or concepts in image synthesis models, in order to obtain models that are more faithful to the input conditioning and therefore more controlable.

Incremental learning and few-shot learning. It would be useful to have a trained image generation model that can be shown a single picture and label of a concept, and subsequently synthesize this concept in all kinds of other contexts. Ideally, the model can learn many new concepts without retraining and without catastrophic forgetting (French, 1999). The first such model was recently presented by Gal et al. (2022) for the text-to-image task, and we are likely to see more follow-up works in this direction. More research is needed to also translate this capability to other conditional image synthesis tasks. The work of Gal et al. (2022) highlights the importance of data and model scale for incremental and few-shot learning. If a

model is so big that it can represent almost any visual concept, we only need to find new "words" for the incrementally presented concepts, without having to update the weights of the model.

6 List of Figures

1.1 Overview of common GAN tasks, using unconditional or conditional generators (green) to generate images from noise. 2

2.1 4.5 years of GAN progress on face generation, as tweeted by GAN creator Ian Goodfellow (Goodfellow, 2019). 12

2.2 Illustration of a generative adversarial network (GAN) 12

2.3 An example of heavy checkerboard artifacts[4]. 15

3.1 Images produced throughout the training by our U-Net GAN model (top row) and their corresponding per-pixel feedback of the *U-Net discriminator* (bottom row). The synthetic image samples are obtained from a fixed noise vector at different training iterations. Brighter colors correspond to the discriminator confidence of pixel being real (and darker of being fake). Note that the U-Net discriminator provides very detailed and spatially coherent response to the generator, enabling it to further improve the image quality, e.g., the unnaturally large man's forehead is recognized as fake by the discriminator and is corrected by the generator throughout the training. 31

3.2 U-Net GAN. The proposed U-Net discriminator classifies the input images on a global and local *per-pixel* level. Due to the skip-connections between the encoder and the decoder (dashed line), the channels in the output layer contain both high- and low-level information. Brighter colors in the decoder output correspond to the discriminator confidence of pixels being real (and darker of being fake). 34

3.3 Visualization of the CutMix augmentation and the predictions of the U-Net discriminator on CutMix images. 1st row: real and fake samples. 2nd&3rd rows: sampled real/fake CutMix ratio r and corresponding binary masks M (color code: white for real, black for fake). 4th row: generated CutMix images from real and fake samples. 5th&6th row: the corresponding real/fake segmentation maps of D^U with its predicted classification scores. 37

3.4 FID curves over iterations of the BigGAN model (blue) and the proposed U-Net GAN (red). Depicted are the FID mean and standard deviation across 5 runs per setting. 43

3.5 Images generated with U-Net GAN trained on FFHQ with resolution 256×256 when interpolating in the latent space between two synthetic samples (left to right). Note the high quality of synthetic samples and very smooth interpolations, maintaining *global* and *local* realism. 45

3.6 Images generated with U-Net GAN trained on COCO-Animals with resolution 128×128. 46

3.7 Qualitative comparison of uncurated images generated with the unconditional BigGAN model (top) and our U-Net GAN (bottom) on FFHQ with resolution 256×256. Note that the images generated by U-Net GAN exhibit finer details and maintain better local realism. . . 48

3.8 Generated samples and the corresponding U-Net decoder predictions for COCO-Animals (row 1 & 2) and FFHQ (row 3 & 4). Brighter areas correspond to the discriminator confidence of pixels being real (and darker of being fake). 49

3.9 Visualization of the predictions of the encoder D^U_{enc} and decoder D^U_{dec} modules during training, within a batch of 50 generated samples. For visualization purposes, the D^U_{dec} score is averaged over all pixels in the output. Note that quite often decisions of D^U_{enc} and D^U_{dec} are not coherent with each other. As judged by the U-Net discriminator, samples in the upper left consist of locally plausible patterns, while not being globally coherent (example in orange), whereas samples in the lower right look globally coherent but have local inconsistencies (example in purple: giraffe with too many legs and vague background). 50

3.10 Comparison of the generator and discriminator loss behavior over training for U-Net GAN and BigGAN. The generator and discriminator loss of U-Net GAN is additionally split up into its encoder- and decoder components. 51

4.1 Existing semantic image synthesis models heavily rely on the VGG-based perceptual loss to improve the quality of generated images. In contrast, our model (OASIS) can synthesize diverse and high-quality images while only using an adversarial loss, without any external supervision. 55

4.2 OASIS multi-modal synthesis results. The 3D noise can be sampled globally (first 2 rows), changing the whole scene, or locally (last 2 rows), partially changing the image. For the latter, we sample different noise per region, like the bed segment (in red) or arbitrary areas defined by shapes. 56

4.3 SPADE (left) vs. OASIS (right). OASIS outperforms SPADE, while being simpler and lighter: it uses only an adversarial loss as supervision and a single segmentation-based discriminator, without relying on heavy external networks. Furthermore, OASIS learns to synthesize multi-modal outputs by directly resampling the 3D noise tensor, instead of using an image encoder as in SPADE. 60

4.4 LabelMix regularization. Real x and fake $\hat{x}$ images are mixed using a binary mask M, sampled based on the label map, resulting in $\text{LabelMix}_{(x,\hat{x})}$. The consistency regularization minimizes the L2 distance between the logits of $D_{\text{LabelMix}_{(x,\hat{x})}}$ and $\text{LabelMix}_{(D_x,D_{\hat{x}})}$. In this visualization, **black** corresponds to the fake class in the N+1 segmentation output. 61

4.5 VGG and adversarial generator loss functions for SPADE and OASIS trained with VGG loss on ADE20k dataset. The adversarial loss scales are different due to different objectives (binary or (N+1)-class cross entropy loss). 64

4.6 Comparison of class distributions of the COCO and LVIS datasets. LVIS has a much larger vocabulary of 1203 classes with a long tail of underrepresented classes. 67

4.7 Histogram distances to real data on the ADE20K validation set. While SPADE+ relies on the VGG loss to learn colors and textures, OASIS achieves low scores without it. 69

4.8 Qualitative comparison of OASIS with other methods on ADE20K and Cityscapes. Trained with only adversarial supervision, our model generates images with better perceptual quality and structure. 70

4.9 Failure mode of OASIS. Without the VGG loss, OASIS has less constraints on the diversity in colors and textures. This helps to achieve higher diversity among the generated samples, but sometimes leads to synthesis of objects with outlier colors and textures which may look less realistic compared to Park et al. (2019b) and Liu et al. (2019). 71

4.10 Qualitative comparison between OASIS and SPADE+ on the long-tailed LVIS dataset with 1203 classes. OASIS generates higher-quality images with more natural colors and textures. For label maps covered mostly by the background class (four right columns), OASIS hallucinates plausible and diverse images, while SPADE+ suffers from mode collapse. 73
4.11 Images generated by OASIS on ADE20K with 256×256 resolution using different 3D noise inputs. For both input label maps, the noise is resampled globally (first row) or locally in the areas marked in red (second row). 74
4.12 Latent space interpolations between images generated by OASIS for the ADE20K dataset at resolution 256×256. The first two rows display *global* interpolations. The second two rows show *local* interpolations of the floor or water only. 75
4.13 After training, the OASIS discriminator can be used to segment images. The first two columns show the real image and the segmentation of the discriminator. Using the predicted label map, the generator can produce multiple versions of the original image by resampling noise (Recreations 1-3). Note that no ground truth maps are required. 76

5.1 Ctrl-SIS learns class-specific directions in the latent space of a SIS model, which can be applied jointly for different classes for local editing of the image. 84
5.2 Ctrl-SIS provides a set of K class-specific latent directions which control the appearance of C semantic classes. To alter the appearance of class c, a class-specific latent direction is added to the input 3D latent code z of the generator G in the label map area corresponding to class $c - M_c$. 86
5.3 Examples of directions discovered by Ctrl-SIS for various classes, such as different views of a window, face appearances, or tree leafage for different seasons of the year. The directions give insight into the concepts that the pretrained SIS model is able to represent. The last row shows global edits with different directions per class. 87
5.4 Interpretable latent directions learnt by Ctrl-SIS for various classes. Each triplet is edited with an identical direction. Class-specific edits, such as aging, snowy streets or bald trees, are highly consistent across different label maps and initial latent codes. 90
5.5 Qualitative comparison of Ctrl-SIS against SeFa and GANSpace. The learned directions $k = 1, ..., 5$ are applied for different classes. Ctrl-SIS *class-specific* latent directions result in more diverse edits for a selected class. 95

5.6 Combinations of directions k_1, k_2 and k_3 found for the *face* class (skin, neck, nose, ears) in the CelebAMask-HQ dataset. Different semantics can be combined, e.g., *k1 (beard)* and *k2 (age)* yield an old bearded person. 98
5.7 Optimizing Ctrl-SIS on the output features of ResNet blocks leads to an emphasis on structure (Res-Blk 1&2), while late normalization layers focus more on color (Norm-Blk 3&4). Directions from different layers can be combined: the last column combines Norm-Blk 4 with Res-Blk 1 (top) and 2 (bottom). 98

6.1 Scaling behavior of the Parti model (Yu et al., 2022) from 350M to 20B parameters. Higher parameter counts result in better image quality and higher alignment with the input prompt. High fidelity placement of embedded text only occurs at 20B parameters. 115

6 List of Tables

3.1 The BigGAN (Brock et al., 2019) generator and discriminator architectures for *class-conditional* image generation. 39
3.2 The BigGAN (Brock et al., 2019) generator and discriminator architectures, modified for *unconditional* image generation. 40
3.3 The U-Net GAN discriminator architectures for class-conditional (a) and unconditional (b) tasks of generating images at resolution 128×128 and 256×256, respectively. 40
3.4 Hyperparameters of U-Net GAN for resolution 256^2 and 128^2. 42
3.5 Evaluation results on FFHQ and COCO-Animals. We report the best and median FID score across 5 runs and its corresponding IS, see Section 3.3.2 for discussion. 43
3.6 Best, median, mean, and std of FID values across 5 runs. 44
3.7 Ablation study of the U-Net GAN model on FFHQ and COCO-Animals. Shown are the median FID scores. The proposed components lead to better performance, on average improving the median FID by 3.7 points over BigGAN. 47
3.8 Comparison with the state-of-the-art models on CelebA (128×128). . 47
3.9 Evaluation results on FFHQ, COCO-Animals, and CelebA with PyTorch and TensorFlow FID/IS scores. The difference lies in the choice of framework in which the inception network is implemented, which is used to extract the inception metrics. 49

4.1 Comparison with other methods across datasets. Bold denotes the best performance. 68
4.2 Multi-modal synthesis evaluation on ADE20K. Bold and red denote the best and the worst performance. 69
4.3 Per-class IoU scores on ADE20k, grouped by pixel-wise frequency (the fraction of all pixels in the datasets belonging to one class). Bold denotes the best performance. Training with per-class loss balancing is denoted by α_c. 72

4.4 Comparison of SPADE+ and OASIS on the LVIS dataset with 1203 classes and a long tail of underrepresented classes. Bold denotes the best performance. Last row shows the scores for the LVIS validation set. 72
4.5 Semantic segmentation performance of ResNeSt-50 with and without synthetic data augmentation (DA). Bold denotes the best performance. 77
4.6 Per-class IoU scores on Cityscapes, obtained without (None) and with synthetic data augmentation using SPADE or OASIS. The classes are sorted and grouped by class pixel-wise frequency, as measured by the total fraction of pixels in the dataset belonging to one class. Bold denotes the best performance. The absolute (abs) and relative (rel) mIoU gain via data augmentation is shown. 78
4.7 Main ablation on ADE20K. The OASIS generator is a lighter version of the SPADE+ generator (72M vs 96M parameters). Bold denotes the best performance. 78
4.8 Ablation on the *D* architecture. Bold denotes the best performance, red highlights collapsed runs. 79
4.9 Ablation on the label map encoding. Bold denotes the best performance, red shows collapsed runs. 79
4.10 Different 3D noise sampling strategies during training. Bold denotes the best performance. 80
4.11 Ablation study on the impact of LabelMix and CutMix for consistency regularization (CR) in OASIS on Cityscapes. Bold denotes the best performance. 81
4.12 The effect of the discriminator feature matching loss (FM) in the absence or presence of the perceptual loss (VGG). Bold denotes the best performance. 81

5.1 Evaluation of OASIS GAN controls on ADE20K and COCO-Stuff. . . 96
5.2 Evaluation of local class-specific image edits on ADE20K with OASIS. 96
5.3 Comparison of GAN control methods across SIS models on ADE20K. 97
5.4 Loss ablation of Ctrl-SIS on ADE20K. 97

6 Bibliography

Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 23

Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 109, 110

Dor Arad Hudson and Larry Zitnick. Compositional transformers for scene generation. *Advances in Neural Information Processing Systems*, 34:9506–9520, 2021. 19, 109

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2017. 4, 71

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 19, 20, 32, 38

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*, pages 224–232. PMLR, 2017. 4

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 2016. 61

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 2017. 34

Kyungjune Baek and Hyunjung Shim. Commonality in natural images rescues gans: Pretraining gans with generic and privacy-free synthetic data. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7854–7864, 2022. 108

Shane T. Barratt and Rishi Sharma. A note on the inception score. *arXiv:1801.01973*, 2018. 42

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2019. 25

Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 111

David Berthelot, Nicholas Carlini, Ian G Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 33

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 17, 20, 85, 86

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019. 4, 30, 32, 33, 38, 39, 40, 42, 43, 47, 49, 51, 54, 55, 125

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 18

Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. In *International Conference on Learning Representations (ICLR)*, 2016. 24

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 57, 66, 86, 90

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 18

Arantxa Casanova, Marlène Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 54

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *International Conference on Learning Representations (ICLR)*, 2015. 67, 91

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018a. 61

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020a. 27

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *International Conference on Computer Vision (ICCV)*, 2017. 6, 24

Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018b. 38, 39

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 21, 30, 111

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020b. 112

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 18

A. V. Cherepkov, Andrey Voynov, and Artem Babenko. Navigating the gan parameter space for semantic image editing. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 85

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020. 27

Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of gans. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 85

Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. *Advances in Neural Information Processing Systems*, 33: 16481–16494, 2020. 109

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 57, 66

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 77

Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C. Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 38

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 24, 55

Emily L Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 17

Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017. 33

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 27, 102

Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022. 115

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 18, 109

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *International Conference on Learning Representations (ICLR)*, 2017. 38

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 27, 28

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. In *International Conference on Learning Representations*, 2018. 20

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 108, 116

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 116

L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 24

Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPs)*. 2015. 24

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *International Conference on Computer Vision (ICCV)*, 2019. 25, 84

Ian Goodfellow. 4.5 years of gan progress on face generation., 2019. URL `https://twitter.com/goodfellow_ian/status/1084973596236144640?lang=en`. 12, 119

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 13, 33, 38, 39, 63, 84

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017a. 20

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017b. 19, 32

Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7, 58, 59, 65, 66

Hyungrok Ham, Tae Joon Jun, and Daeyoung Kim. Unbalanced gans: Pre-training the generator of generative adversarial network using variational autoencoder. *arXiv preprint arXiv:2002.02112*, 2020. 108

Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14072–14082, 2021. 24

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 25, 84, 86, 91, 101

Louay Hazami, Rayhane Mama, and Ragavan Thurairatnam. Efficient-vdvae: Less is more. *arXiv preprint arXiv:2203.13751*, 2022. 27

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017a. 25, 91

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017b. 3, 13, 42, 67

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`. 109

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 25, 27

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 2018. 54

Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with product-of-experts GANs. In *ECCV*, 2022. 112, 114

Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR, 2021. 19, 109

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 22, 23, 24, 54, 56, 57, 59, 91, 112

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020. 25, 85

Himalaya Jain, Tuan-Hung Vu, Patrick Pérez, and Matthieu Cord. Csg0: Continual urban scene generation with zero forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3679–3687, 2022. 24, 109, 110

Jaebong Jeong, Janghun Jo, Jingdong Wang, Sunghyun Cho, and Jaesik Park. Realistic image synthesis with configurable 3d scene layouts. *arXiv preprint arXiv:2108.10031*, 2021. 24

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34:14745–14758, 2021. 19, 109

Younghyun Jo, Sejong Yang, and Seon Joo Kim. Investigating loss functions for extreme super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 424–425, 2020. 104

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 24

Alexia Jolicoeur-Martineau, Rémi Piché-Taillefer, Ioannis Mitliagkas, and Remi Tachet des Combes. Adversarial score matching and improved sampling for image generation. In *International Conference on Learning Representations*, 2020. 25

Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7799–7808, 2020. 15

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 15, 30, 42, 47, 49

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019a. 30, 32, 41, 55

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2019b. 17, 20

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a. 17, 55, 103, 110, 111

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Computer Vision and Pattern Recognition (CVPR)*, 2020b. 15, 19, 20, 85, 86

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020c. 17, 47, 55

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a. 85

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b. 16, 20, 55, 109

Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. Ossgan: Open-set semi-supervised image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11185–11193, 2022. 116

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 42, 66

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 27

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 109

Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for gan training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10651–10662, 2022. 108

Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. The GAN landscape: Losses, architectures, regularization, and normalization. *arXiv: 1807.04720*, 2018. 33

Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *International conference on machine learning*, pages 3581–3590. PMLR, 2019. 20

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 32, 41

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019. 3, 4, 91

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 84, 90

Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021. 19, 109, 110

Zhaoqi Leng, Mingxing Tan, Chenxi Liu, Ekin Dogus Cubuk, Jay Shi, Shuyang Cheng, and Dragomir Anguelov. Polyloss: A polynomial expansion perspective of classification loss functions. In *International Conference on Learning Representations*, 2021. 111

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 19, 32

Yuanzhuo Li, Junjie Liu, and Yu Chen. Learning structural coherence via generative adversarial network for single image super-resolution. In *2021 International Conference on Computer Engineering and Application (ICCEA)*, pages 184–188. IEEE, 2021a. 104

Yuheng Li, Yijun Li, Jingwan Lu, Eli Shechtman, Yong Jae Lee, and Krishna Kumar Singh. Collaging class-specific gans for semantic image synthesis. In *International Conference on Computer Vision (ICCV)*, 2021b. 22, 23, 24, 84

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv:1705.02894*, 2017. 38

Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Coco-gan: Generation by parts via conditional coordinating. In *International Conference on Computer Vision (ICCV)*, 2019. 30, 31, 32, 47, 49

Chieh Hubert Lin, Hsin-Ying Lee, Yen-Chi Cheng, Sergey Tulyakov, and Ming-Hsuan Yang. Infinitygan: Towards infinite-pixel image synthesis. In *International Conference on Learning Representations*, 2021. 109

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 32, 41

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *International Conference on Computer Vision (ICCV)*, 2017. 111

Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 26, 85

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. 21, 55

Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 22, 23, 24, 54, 55, 56, 67, 71, 87, 121

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, 2015. 32, 41

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 91

Mario Lučić, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 20, 33

Zhengyao Lv, Xiaoming Li, Zhenxing Niu, Bing Cao, and Wangmeng Zuo. Semantic-shape adaptive feature modulation for semantic image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11214–11223, 2022. 24

Anton Mallasto, Guido Montúfar, and Augusto Gerolin. How well do wgans estimate the wasserstein metric? *arXiv preprint arXiv:1910.03875*, 2019. 20

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 20

Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *International Conference on Learning Representations (ICLR)*, 2019. 30

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *International Conference on Machine learning (ICML)*, 2018. 4, 20

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014. 18, 54, 106

Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *International Conference on Learning Representations (ICLR)*, 2018. 17, 18, 23, 38, 39, 63, 79, 111, 112

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 4, 17, 20, 30, 32, 55

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964*, 2020. 108

Youssef Mroueh and Tom Sercu. Fisher gan. *Advances in Neural Information Processing Systems*, 30, 2017. 19

Youssef Mroueh, Tom Sercu, and Vaibhava Goel. Mcgan: Mean and covariance feature matching gan. In *International conference on machine learning*, pages 2527–2535. PMLR, 2017. 19

Youssef Mroueh, Chun-Liang Li, Tom Sercu, Anant Raj, and Yu Cheng. Sobolev gan. In *International Conference on Learning Representations*, 2018. 19

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997. 19

Valentina Musat, Daniele De Martini, Matthew Gadd, and Paul Newman. Depth-sims: Semi-parametric image and depth synthesis. *arXiv preprint arXiv:2203.03405*, 2022. 24

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016a. 32, 38

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016b. 20

Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: semantic editing of scenes by adding, manipulating or erasing objects. In *European Conference on Computer Vision (ECCV)*, 2020. 22, 23, 55, 56, 68

Evangelos Ntavelis, Mohamad Shahbazi, Iason Kastanis, Radu Timofte, Martin Danelljan, and Luc Van Gool. Arbitrary-scale image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11533–11542, 2022. 109

Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016. 15

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *International Conference on Learning Representations (ICLR)*, 2017. 18

Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 1996. 67

Ehsan Pajouheshgar, Tong Zhang, and Sabine Süsstrunk. Optimizing latent space directions for gan-based local image editing. *arXiv preprint arXiv:2111.12583*, 2021. 26, 85

Jeeseung Park and Younggeun Kim. Styleformer: Transformer based generative adversarial networks with style vector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8983–8992, 2022. 19

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH*. 2019a. 55

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Computer Vision and Pattern Recognition (CVPR)*, 2019b. 22, 23, 24, 26, 54, 55, 56, 57, 58, 59, 66, 67, 71, 84, 86, 87, 90, 91, 98, 108, 121

Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 2020. 54

William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020. 3

Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations (ICLR)*, 2019. 25, 84, 85

Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 25, 66

Yipeng Qin, Niloy Mitra, and Peter Wonka. How does lipschitz regularization influence gan training? In *European Conference on Computer Vision*, pages 310–326. Springer, 2020. 20

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1, 101, 113, 114, 115

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 101, 113

Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine learning (ICML)*, 2016. 18, 54

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 22, 24, 101, 111

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 27, 101, 113, 115

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 5, 22, 31, 32, 33, 34, 35, 57, 61

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 32

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision (IJCV)*, 2000. 67

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1, 101, 113

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, 2018. 3, 4

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016a. 3

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016b. 42

Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 55, 108, 110, 116

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*, pages 1–10, 2022. 15, 17, 18, 20, 108, 116

Edgar Schönfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4, 9, 30, 63, 85, 104

Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. 5, 9, 26, 54, 84, 85, 86, 87, 90, 91, 94, 97

Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. 114

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 114

Sarah Schwettmann, Evan Hernandez, David Bau, Samuel Klein, Jacob Andreas, and Antonio Torralba. Toward a visual concept vocabulary for gan latent space. In *International Conference on Computer Vision (ICCV)*, 2021. 25

Matt Shannon, Ben Poole, Soroosh Mariooryad, Tom Bagby, Eric Battenberg, David Kao, Daisy Stanton, and RJ Skerry-Ryan. Non-saturating gan training as divergence minimization. *arXiv preprint arXiv:2010.08029*, 2020. 20

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 25, 26, 84, 86, 91, 101, 112

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 25, 85

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 55

Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021a. 109

Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. Aligning latent and image spaces to connect the unconnectable. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14144–14153, 2021b. 109

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 25

Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *International Conference on Computer Vision (ICCV)*, 2017. 24

Nurit Spingarn, Ron Banner, and Tomer Michaeli. Gan "steerability" without optimization. In *International Conference on Learning Representations (ICLR)*, 2020. 25

Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*. 2017. 57, 71

Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. Oasis: Only adversarial supervision for semantic image synthesis. *International Journal of Computer Vision*, pages 1–21, 2022. 5, 9, 54

Ryohei Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially controllable image synthesis with internal representation collaging. *arXiv preprint arXiv: 1811.10153*, 2018. 26

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 17

Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, and Nenghai Yu. Rethinking spatially-adaptive normalization. *arXiv:2004.02867*, 2020. 24

Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Bin Liu, Gang Hua, and Nenghai Yu. Diverse semantic image synthesis via probability distribution modeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 84

Hao Tang, Song Bai, and Nicu Sebe. Dual attention gans for semantic image synthesis. In *ACM International Conference on Multimedia*, 2020a. 24

Hao Tang, Xiaojuan Qi, Dan Xu, Philip HS Torr, and Nicu Sebe. Edge guided gans with semantic preserving for semantic image synthesis. *arXiv:2003.13898*, 2020b. 22

Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020c. 22, 24

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? *arXiv preprint arXiv:2207.10551*, 2022. 115

Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. Warpedganspace: Finding non-linear rbf paths in gan latent space. In *International Conference on Computer Vision (ICCV)*, 2021. 25, 26, 84

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 27

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 14, 18

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, 2019a. 33

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *IJCAI*, 2019b. 33

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning (ICML)*, 2020. 25, 85

Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a. 67

Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022a. 25, 116

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018a. 22, 23, 55, 56, 57, 108

Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models. *arXiv preprint arXiv:2207.00050*, 2022b. 25

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914, 2021b. 104

Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 218–234, 2018b. 108

Yi Wang, Lu Qi, Ying-Cong Chen, Xiangyu Zhang, and Jiaya Jia. Image synthesis via semantic composition. In *International Conference on Computer Vision (ICCV)*, 2021c. 22, 23, 24, 26, 54, 55, 56, 84, 86, 87, 90

Yin Wang. Single image super-resolution with u-net generative adversarial networks. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 4, pages 1835–1840. IEEE, 2021. 104

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems & Computers*, 2003. 67, 69

Zihao Wei, Yidong Huang, Yuang Chen, Chenhao Zheng, and Jinnan Gao. A-esrgan: Training real-world blind super-resolution with attention u-net discriminators. *arXiv preprint arXiv:2112.10046*, 2021. 104

Chenfei Wu, Jian Liang, Xiaowei Hu, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zicheng Liu, Yuejian Fang, and Nan Duan. Nuwa-infinity: Autoregressive over autoregressive generation for infinite visual synthesis. *arXiv preprint arXiv:2207.09814*, 2022. 115

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 25, 26, 85

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learninga comprehensive evaluation of the good, the bad and the ugly. *Transactions on Pattern Analysis and Machine Intelligence*, 2018. 42

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision (ECCV)*, 2018. 67, 91

Yasin Yaz, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, Vijay Chandrasekhar, et al. The unusual effectiveness of averaging in gan training. In *International Conference on Learning Representations (ICLR)*, 2018. 68

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 67

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 28, 101, 113, 114, 115, 123

Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *International Conference on Computer Vision (ICCV)*, 2021. 25, 26, 84, 91, 112

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 32, 33, 36, 63, 80

Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2759–2768, 2019. 108

Dan Zhang and Anna Khoreva. PA-GAN: Improving GAN training by progressive augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 55

H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stack-GAN++: Realistic image synthesis with stacked generative adversarial networks. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018a. 54

Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine learning (ICML)*, 2019. 4, 20, 30, 31, 39

Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020a. 4, 21, 30, 33, 34, 47

Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 54

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv:2004.08955*, 2020b. 68, 75

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018b. 33, 36

Kai Zhang, Shuhang Gu, and Radu Timofte. Ntire 2020 challenge on perceptual extreme super-resolution: Methods and results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 492–493, 2020c. 104

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Vision and Pattern Recognition (CVPR)*, 2018c. 92

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018d. 67, 69

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017. 20

Long Zhao, Zizhao Zhang, Ting Chen, Dimitris Metaxas, and Han Zhang. Improved transformer for high-resolution gans. *Advances in Neural Information Processing Systems*, 34:18367–18380, 2021. 19, 109

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33:7559–7570, 2020a. 103, 110, 111

Shuai Zhao, Yang Wang, Zheng Yang, and Deng Cai. Region mutual information loss for semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 111

Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. *arXiv:2002.04724*, 2020b. 34, 47, 111

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 57, 66, 86, 90

Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zhengjun Zha, Jingren Zhou, and Qifeng Chen. Low-rank subspaces in gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 26, 85

Jiapeng Zhu, Yujun Shen, Yinghao Xu, Deli Zhao, and Qifeng Chen. Region-based semantic factorization in gans. *arXiv preprint arXiv:2202.09649*, 2022. 26, 85, 112

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 113

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Computer Vision and Pattern Recognition (CVPR)*, 2020a. 84

Zhen Zhu, Zhiliang Xu, Ansheng You, and Xiang Bai. Semantically multi-modal image synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b. 23, 84, 94