StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators

Rinon Gal[1,2*] Or Patashnik[1] Haggai Maron[2] Amit Bermano[1]
Gal Chechik[2] Daniel Cohen-Or[1]

[1]Tel-Aviv University [2]NVIDIA

Abstract

Can a generative model be trained to produce images from a specific domain, guided only by a text prompt, without seeing any image? In other words: can an image generator be trained "blindly"? Leveraging the semantic power of large scale Contrastive-Language-Image-Pre-training (CLIP) models, we present a text-driven method that allows shifting a generative model to new domains, without having to collect even a single image. We show that through natural language prompts and a few minutes of training, our method can adapt a generator across a multitude of domains characterized by diverse styles and shapes. Notably, many of these modifications would be difficult or outright impossible to reach with existing methods. We conduct an extensive set of experiments across a wide range of domains. These demonstrate the effectiveness of our approach, and show that our models preserve the latent-space structure that makes generative models appealing for downstream tasks.

1. Introduction

The unprecedented ability of Generative Adversarial Networks (GANs) [17] to capture and model image distributions through a semantically-rich latent space has revolutionized countless fields. These range from image enhancement [24, 61], editing [19, 45] and recently even discriminative tasks such as classification and regression [33, 59].

Typically, the scope of these models is restricted to domains where one can collect large sets of images. This requirement severely constrains their applicability. Indeed, in many cases (paintings by specific artists, rare medical conditions, imaginary scenes), there may not exist sufficient data to train a GAN, or even any data at all.

Recently, it has been shown that Vision-Language models (CLIP, [39]) encapsulate generic information that can bypass the need for collecting data. Moreover, these models can be paired with generative models to provide a simple and intuitive text-driven interface for image generation and manipulation [37]. However, such works are built on pre-trained generative models with fixed domains, limiting the user to *in-domain* generation and manipulation.

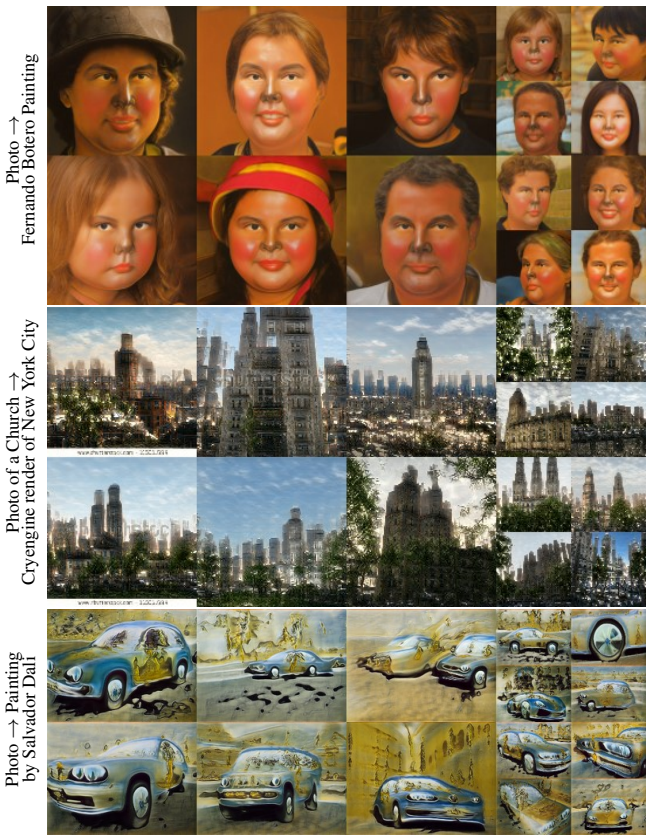

Figure 1. Examples of text-driven, out-of-domain generator adaptations induced by our method. The textual directions driving the change appear next to each set of generated images.

In this work, we present a text-driven method that enables *out-of-domain* generation. We introduce a training scheme that shifts the domain of a pre-trained model towards a new domain, using nothing more than a textual

* Work was done during an internship at NVIDIA
Code and videos available at: stylegan-nada.github.io/

prompt. Fig. 1 demonstrates out-of-domain generation for three examples. All three models were trained *blindly*, without seeing any image of the target domain.

Leveraging CLIP for text-guided training is challenging. The naïve approach – requiring generated images to maximize some CLIP-based classification score, often leads to adversarial solutions [18] (see Sec. 5). Instead, we encode the difference between domains as a textually-prescribed direction in CLIP's embedding space. We propose a novel loss and a two-generator training architecture. One generator is kept frozen, providing samples from the source domain. The other is optimized to produce images that differ from this source only along a textually-described, cross-domain direction in CLIP-space.

To increase training stability for more drastic domain changes, we introduce an adaptive training approach that utilizes CLIP to identify and restrict training only to the most relevant network layers at each training iteration.

We apply our approach to StyleGAN2 [23], and demonstrate its effectiveness for a wide range of source and target domains. These include artistic styles, cross-species identity transfer and significant shape changes, like converting dogs to bears. We compare our method to existing editing techniques and alternative few-shot approaches and show that it enables modifications which are beyond their scope – and it does so without direct access to *any* training data.

Finally, we verify that our method maintains the appealing structure of the latent space. Our shifted generator not only inherits the editing capabilities of the original, but it can even re-use any off-the-shelf editing directions and models trained for the original domain.

2. Related work

Text-guided synthesis. Vision-language tasks include language-based image retrieval, image captioning, visual question answering, and text-guided synthesis among others. Typically, to solve these tasks, a cross-modal vision and language representation is learned [10, 13, 25–27, 30, 42, 47, 48], often by training a transformer [52].

The recently introduced CLIP [39] is a powerful model for joint vision-language representation. It is trained on 400 million text-image pairs. Using a contrastive learning goal, both image and text are mapped into a joint, multimodal embedding space. The representations learned by CLIP have been used in guiding a variety of tasks, including image synthesis [12, 32] and manipulation [7, 37]. These methods utilize CLIP to guide the optimization of a latent code, generating or manipulating a specific image. In contrast, we present a novel approach in which a text prompt guides the *training* of the image generator itself.

Training generators with limited data. The goal of few-shot generative models is to mimic a rich and diverse data distribution using only a few images. Methods used to tackle such a task can be divided into two broad categories: training from-scratch, and fine-tuning — which leverages the diversity of a pre-trained generator.

Among those that train a new generator, 'few' often denotes several thousand images (rather than tens of thousands [23] or millions [8]). Such works typically employ data augmentations [20, 50, 64, 65] or empower the discriminator to better learn from existing data using auxiliary tasks [29, 60].

In the transfer-learning scenario, 'few' typically refers to numbers ranging from several hundred to as few as five [35]. When training with extremely limited data, a primary concern is staving off mode-collapse or overfitting, to successfully transfer the diversity of the source generator to the target domain. Multiple methods have been proposed to tackle these challenges. Some place restrictions on the space of modified weights [31, 38, 41]. Others introduce new parameters to control channel-wise statistics [34], steer sampling towards suitable regions of the latent space [55], add regularization terms [28, 51] or enforce cross-domain consistency while adapting to a target style [35].

While prior methods adapt generators with *limited* data, we do so *without* direct access to *any* training images. Additionally, prior methods constrained the space of trainable weights to fixed, hand-picked subsets. Our method introduces adaptive layer selection - accounting for both the state of the network at each training step, and for the target class.

3. Preliminaries

At the core of our approach are two components - StyleGAN2 [23] and CLIP [39]. In the following, we discuss relevant features in StyleGAN's architecture, and how CLIP has been employed in this context in the past.

3.1. StyleGAN

In recent years, StyleGAN and its variants [20–23] have established themselves as the state-of-the-art unconditional image generators. The StyleGAN generator consists of two main components. A mapping network converts a latent code z, sampled from a Gaussian prior, to a vector w in a learned latent space $\mathcal{W}$. These latent vectors are then fed into the synthesis network, to control feature (or convolutional kernel) statistics. By traversing $\mathcal{W}$, or by mixing different w codes at different network layers, prior work demonstrated fine-grained control over semantic properties in generated images [3, 19, 37, 45]. These latent-space editing operations, however, are typically limited to modifications that align with the domain of the initial training set.

3.2. StyleCLIP

In a recent work, Patashnik *et al*. [37] combine the generative power of StyleGAN with the semantic knowledge of

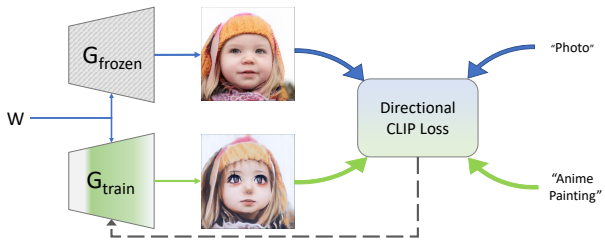

Figure 2. Overview of our training setup. We initialize two generators - G_{frozen} and G_{train} using the weights of a generator pre-trained on images from a source domain (*e.g.* FFHQ [22]). G_{frozen} remains fixed throughout the process. G_{train} is modified through optimization and an iterative layer-freezing scheme. The process shifts the domain of G_{train} according to a user-provided textual direction while maintaining a shared latent space.

CLIP to discover novel editing directions, using only a textual description of the desired change. They outline three approaches for leveraging the semantic power of CLIP. The first two aim to minimize the CLIP-space distance between a generated image and some target text. They use direct latent code optimization, or train an encoder (or *mapper*) to modify an input latent code. The third approach, which we adapt, uses CLIP to discover global directions of disentangled change in the latent space. They modify individual latent-code entries, and determine which ones induce an image-space change that is co-linear with the direction between two textual descriptors (denoting the source and desired target) in CLIP-space.

However, these approaches all share the limitation common to latent space editing methods - the modifications that they can apply are largely constrained to the domain of the pre-trained generator. As such, they can allow for changes in hairstyle, expressions, or even convert a wolf to a lion *if the generator has seen both* - but they cannot convert a photo to a painting, or produce cats when trained on dogs.

4. Method

Our goal is to shift a pre-trained generator from a given source domain to a new target domain, described only through textual prompts, with no images. As a source of supervision, we use only a pre-trained CLIP model.

We approach the task through two key questions: (1) How can we best distill the semantic information encapsulated in CLIP? and (2) How should we regularize the optimization process to avoid adversarial solutions and mode collapse? In the following section we outline a training scheme and a loss that seek to answer both questions.

4.1. CLIP-based guidance

Global loss. We rely on a pre-trained CLIP model to serve as the sole source of supervision for our target domain. Naïvely, one could use StyleCLIP's direct loss:

$$\mathcal{L}_{global} = D_{CLIP}\left(G\left(w\right), t_{target}\right), \tag{1}$$

where $G\left(w\right)$ is the image generated by the latent code w fed to the generator G, t_{target} is the textual description of the target class, and D_{CLIP} is the CLIP-space cosine distance. We name this loss 'global', as it does not depend on the initial image or domain.

We observe that in practice, this loss leads to adversarial solutions. In the absence of a fixed generator that favors solutions on the real-image manifold, optimization overcomes the classifier (CLIP) by adding pixel-level perturbations to the image. Moreover, this loss sees no benefit from maintaining diversity. Indeed, a mode-collapsed generator producing only one image may be the best minimizer for the distance to a given textual prompt. See Appendix B for an analysis of CLIP's embedding space, demonstrating this issue. These shortcomings make this loss unsuitable for training the generator. However, we do leverage it for adaptive layer selection (see Sec. 4.2).

Directional CLIP loss. To address these issues, we draw inspiration from StyleCLIP's global direction approach. Ideally, we want to identify the CLIP-space direction between our source and target domains. Then, we'll fine-tune the generator so that the images it produces differ from the source *only* along this direction.

To do so, we first identify a cross-domain direction in CLIP-space by embedding a pair of textual prompts describing the source and target domains (*e.g.* "Dog" and "Cat"). Then, we must determine the CLIP-space direction between images produced by the generator before and after fine-tuning. We do so using a two-generator setup. We begin with a generator pre-trained on a single source domain (*e.g.* faces, dogs, churches or cars), and clone it. One copy is kept frozen throughout the training process. Its role is to provide an image in the source domain for every latent code. The second copy is trained. It is fine-tuned to produce images that, for any sampled code, differ from the source generator's only along the textually described direction. We name these generators G_{frozen} and G_{train} respectively.

In order to maintain latent space alignment, the two generators share a single mapping network which is kept frozen throughout the process. The full training setup is shown in Fig. 2. An illustration of the CLIP-space directions is provided in Fig. 3. The direction loss is given by:

$$\begin{aligned} \Delta T &= E_T\left(t_{target}\right) - E_T\left(t_{source}\right), \\ \Delta I &= E_I\left(G_{train}\left(w\right)\right) - E_I\left(G_{frozen}\left(w\right)\right), \\ \mathcal{L}_{direction} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|}. \end{aligned} \tag{2}$$

Here, E_I and E_T are CLIP's image and text encoders, and t_{target}, t_{source} are the source and target class texts.

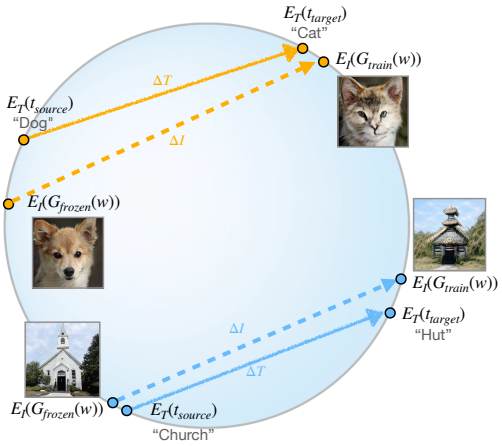

Figure 3. Illustration of CLIP-space directions in our directional loss (Eq. (2)). We embed images generated by both generators into CLIP-space and demand that the vector connecting them, ΔI, is parallel to the vector connecting a source and target text, ΔT.

Such a loss overcomes the global loss' shortcomings: First, it is adversely affected by mode-collapsed examples. If the target generator only creates a single image, the CLIP-space directions from all sources to this target image will be different. As such, they can't all align with the textual direction. Second, it is harder for the network to converge to adversarial solutions, because it has to engineer perturbations that fool CLIP across an infinite set of different instances.

4.2. Layer-Freezing

For domain shifts which are predominantly texture-based, such as converting a photo to a sketch, the training scheme described above quickly converges before mode collapse or overfitting occurs. However, more extensive shape modifications require longer training, which in turn destabilizes the network and leads to poor results.

Prior works on few-shot domain adaptation observed that the quality of synthesized results can be significantly improved by restricting training to a subset of network weights [31, 41]. The intuition is that some layers of the source generator are useful for generating aspects of the target domain, so we want to preserve them. Furthermore, optimizing fewer parameters reduces the model complexity and the risk of overfitting. Following these approaches, we regularize the training process by limiting the number of weights that can be modified at each training iteration.

Ideally, we would like to restrict training to those model weights that are most relevant to a given change. To identify these weights, we turn back to latent-space editing techniques, and specifically to StyleCLIP.

In the case of StyleGAN, it has been shown that codes provided to different network layers, influence different semantic attributes. Thus, by considering editing directions

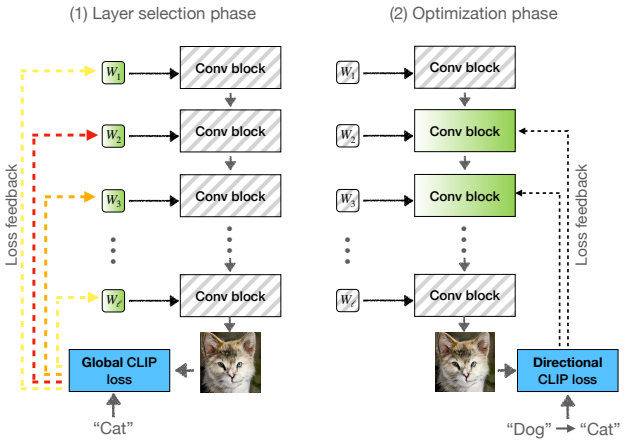

Figure 4. The adaptive layer-freezing mechanism has two phases. In the first phase (left), we optimize a set of latent codes in $\mathcal{W}+$ (turquoise), leaving all network weights fixed. This optimization is conducted using the Global CLIP loss (Eq. (1)) . We select the layers whose corresponding w entry changed most significantly (darker colors, left). In the second phase (right), we unfreeze the weights of the selected layers. We then optimize these layers using the directional CLIP loss (Eq. (2)).

in the $\mathcal{W}+$ space [2] – the latent space where each layer of StyleGAN can be provided with a different code $w_i \in \mathcal{W}$ – we can identify which layers are most strongly tied to a given change. Building on this intuition, we suggest a training scheme, where at each iteration we (i) select the k-most relevant layers, and (ii) perform a single training step where we optimize only these layers, while freezing all others.

To select the k layers, we randomly sample N_w latent codes $\in \mathcal{W}$ and convert them to $\mathcal{W}+$ by replicating the same code for each layer. We then perform N_i iterations of the StyleCLIP *latent-code* optimization method, using the global loss (Eq. (1)). We select the k layers for which the latent-code changed most significantly. The two-step process is illustrated in Fig. 4. In all cases we additionally freeze StyleGAN's mapping network, affine code transformations, and all toRGB layers.

Note that this process is inherently different from selecting layers according to gradient magnitudes during direct training. Latent-code optimization using a *frozen* generator tends to favor solutions which remain on the real-image manifold. By using it to select layers, we bias training towards similarly *realistic* changes. In contrast, direct training allows the model more easily drift towards unrealistic or adversarial solutions.

4.3. Latent-Mapper 'mining'

For some shape changes, we found that the generator does not complete a full transformation. For example, when transforming dogs to cats, the fine-tuning process results in a new generator that outputs both cats, dogs, and an assort-

Figure 5. Image synthesis using models adapted from StyleGAN2-FFHQ [23] to a set of textually-prescriped target domains. All images were sampled randomly, using truncation with $\psi = 0.7$. The driving texts appear below each generated set.

ment of images that lie in between. To alleviate this, we note that the shifted generator now includes *both* cats and dogs within its domain. We thus turn to in-domain latent-editing techniques, specifically StyleCLIP's latent mapper, to map all latent codes into the cat-like region of the latent space. Training details for the mapper are provided Appendix I.

5. Experiments

5.1. Results

We begin by showcasing the wide range of out-of-domain adaptations enabled by our method. These range from style and texture changes to shape modifications, and from realistic to fantastical, including zero-shot prompts for which real data does not even exist (*e.g.* Nicolas Cage dogs). All these are achieved through a simple text-based interface, and are typically trained within a few minutes.

In Figs. 5 and 6, we show a series of randomly sampled images synthesized by generators converted from faces, churches, dogs and cars to various target domains. Additional large scale galleries portraying a wide set of target domains are shown in Appendix G.

Fig. 7 shows domain adaptation from dogs to a wide range of animals. While Figs. 5, 6 focused on style and minor shape adjustments, here the model performs significant shape modifications. For example, many animals sport upright ears, while most AFHQ-Dog [11] breeds do not. Training details for all scenarios are provided in Appendix I.

5.2. Latent space exploration

Modern image generators (and StyleGAN in particular), are known to have a well-behaved latent space. Such a latent space is conductive for tasks such as image editing and image-to-image translation [4, 19, 37, 40, 45]. The ability to manipulate real images is of particular interest, leading to an ever-increasing list of methods for GAN inversion [2, 5, 40, 49, 58]. We show that our transformed generators can still support such manipulations, using the same techniques and inversion methods. Indeed, as outlined below, our model can even reuse off-the-shelf models pre-trained on the source generator's domain, with no need for additional fine-tuning.

GAN Inversion. We begin by pairing existing inversion methods with our transformed generator. Given a real image, we first invert it using a ReStyle encoder [5], pre-trained on the human face domain. We then insert the inverted latent-code $w \in \mathcal{W}+$ into our transformed generators. Fig. 8 shows results obtained in such a manner, using generators adapted across multiple domains. Our generators successfully preserve the identity tied to the latent code, even for codes obtained from the inversion of real images.

Latent traversal editing. The inversion results suggest that the latent space of the adapted generator is aligned with that of the source generator. This is not entirely surprising. First, due to our *intertwined* generator architecture and the nature of the directional loss. Second, because prior and concurrent methods successfully employed the natural alignment of fine-tuned generators for downstream applications [38, 46, 53, 57]. However, our fine-tuning approach is non-adversarial and thus differs from these prior methods. Consequently, verifying that latent-space alignment remains unbroken is of great interest. We use existing editing techniques and show that latent-space directions do indeed

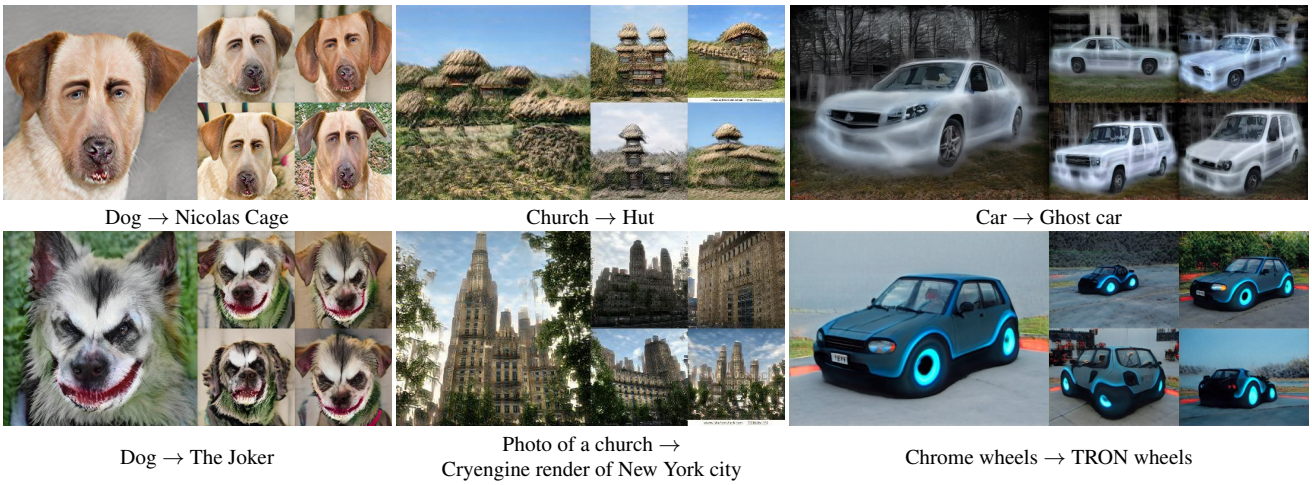

Figure 6. Image synthesis using models adapted from StyleGAN2's [23] LSUN Church, LSUN Car [62] models and StyleGAN-ADA [20] AFHQ-Dog [11]. All images were sampled randomly, using truncation with $\psi = 0.7$. The driving texts appear below each generated set.

Figure 7. Generator translation to multiple animal domains. In all cases we begin with a StyleGAN-ADA [20] AFHQ-Dog [11] model. All generators are adapted using our method and a StyleCLIP [37] latent mapper. For all experiments, the source domain text was 'Dog'. The target domain text is shown below each image.

maintain their semantic meaning. As a result, rather than finding new paths in the latent space of the modified generator, we can simply reuse the same paths and editing models found for the original, source generator.

In Figure 9, we edit a real image mapped into novel domains, using several off-the-shelf methods. We use StyleCLIP [37] to edit expression and hairstyle, StyleFlow [3] to edit pose, and InterFaceGAN [45] to edit age. We use the original implementations, pre-trained on the *source* domain.

Image-to-image translation. Generative objectives extend beyond image-editing. Richardson *et al*. [40] demonstrate a wide range of image-to-image translation applications approached by training encoders. These encoders learn a mapping from images in arbitrary domains to the latent space of a pre-trained generator. The generator is then used to re-synthesize an image in its own domain. They demonstrate this approach for conditional synthesis tasks, image restoration and super resolution. However, a significant limitation of their method is that the target domain of the generated images is restricted to domains for which a StyleGAN generator can be trained. We show that these pre-trained encoders can also be paired with our adapted generators, enabling more generic image-to-image translation. Specifically, Fig. 10 shows conditional image synthesis in multiple domains, using segmentation masks and sketch-based guidance, without re-training the encoder.

5.3. Comparison to other methods

We compare two aspects of our method to alternative approaches. First, we show that the text-driven out-of-domain capabilities of our method cannot be replicated by current latent-editing based techniques. Then, we demonstrate that StyleGAN-NADA can affect large shape changes better than current few-shot training approaches.

Figure 8. Out-of-domain editing through latent-code equivalence between generators. We invert an image into the latent space of a StyleGAN2 FFHQ model [23], using a pre-trained ReStyle encoder [5]. We then feed the same latent code into the transformed generators in order to map the same identity to a novel domain.

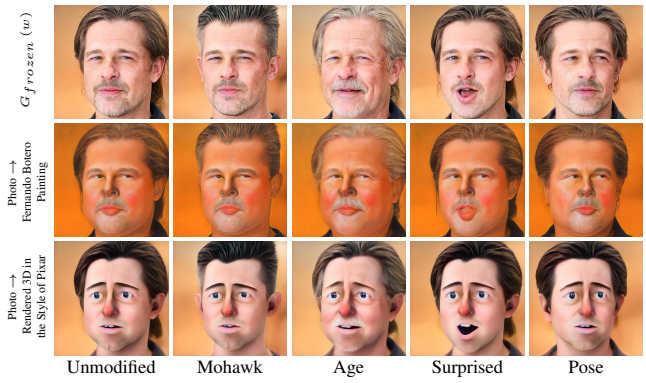

Figure 9. Multi-domain editing of a real, inverted image using StyleCLIP [37] (mohawk and surprised), InterFaceGAN [45] (age) and StyleFlow [3] (pose). The top row portrays editing in the source domain. All rows below show the same editing operations in our translated domains.

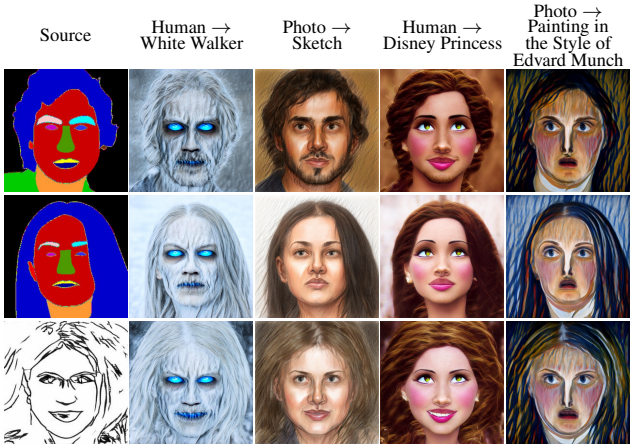

Figure 10. Conditional synthesis in multiple domains. We use the official pSp [40] FFHQ encoder to invert segmentation masks and simple sketches (left) into the latent space of the GAN. The inverted-codes work seamlessly with our adapted models.

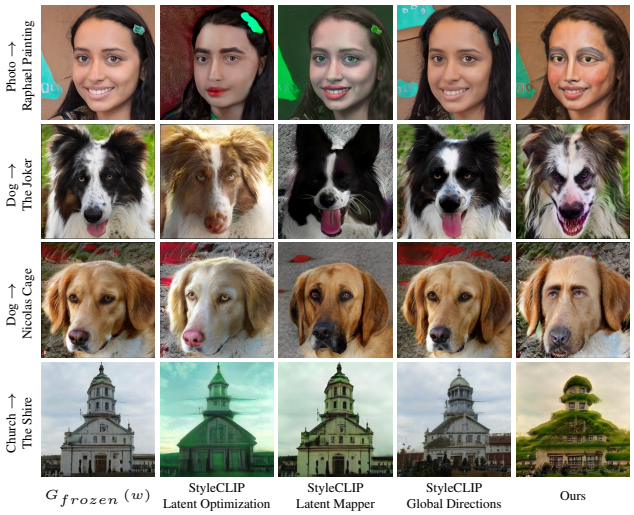

Figure 11. Out-of-domain manipulation comparisons to StyleCLIP [37]. In each row we show: an image synthesized with a randomly sampled code (left), the results of editing the same code towards an out-of-domain textual direction using all three StyleCLIP [37] methods, and the image produced for the same code with a generator converted using our method (right).

Text-guided editing. We show that existing editing techniques that operate *within* the domain of a pre-trained generator, fail to shift images beyond said domain. In particular, we are interested in text-driven methods that operate without additional data. A natural comparison is then StyleCLIP and its three CLIP-guided editing approaches. Fig. 11 shows the results of such a comparison. As can be seen, none of StyleCLIP's approaches succeed in performing out-of-domain manipulations, even when only minor changes are needed (such as inducing celebrity identities on dogs).

Few-shot generators. We compare StyleGAN-NADA with several few-shot alternatives: Ojha *et al*. [35], MineGAN [55], TGAN [56] and TGAN + ADA [20]. In all cases, we convert the official StyleGAN-ADA AFHQ-Dog model [11, 20] to a cat model. Our method operates in a zero-shot manner. Other methods were trained using samples of 5, 10 and 100 images from AFHQ-Cat [11]. We evaluate two aspects of these models — quality and diversity. Quality is measured using a user study with a two-alternative forced choice setup. Users were presented with one of our generated images, and one from a competing method. They were asked to pick the image portraying a higher-quality cat. We gathered 1200 responses from 218 unique users. We report the percentage of users which preferred each method to our own. For diversity, we follow Ojah *et al*. [35] by clustering the data and computing an average LPIPS [63] distance within the clusters. However, their method clusters according to LPIPS distances from the

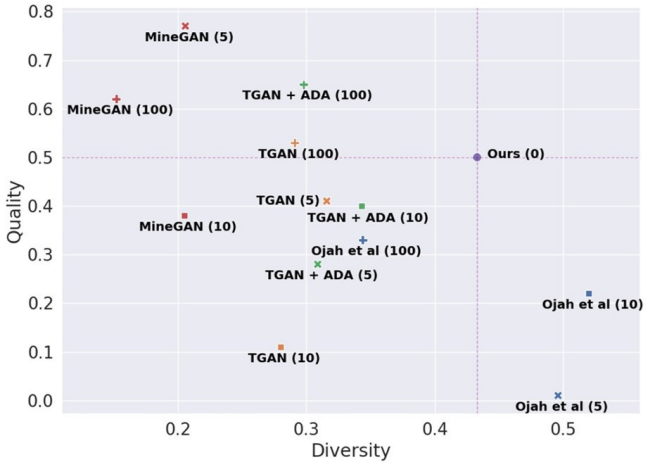

Figure 12. Quality (↑) and diversity (↑) comparison for our method and selected few-shot approaches. Numbers in parenthesis denote the number of training images. Our model pushes the Pareto front, and it does so without using a single image.

training set. As our own method does not use a training set, we cluster around generated images using K-Medoids [1].

Figure 13. Image synthesis using models adapted from StyleGAN-ADA [20] AFHQ-Dog [11] to the cat domain. We compare to two few shot models - Ojha *et al*. [35] and MineGAN [55]. Next to each method we list the number of training images used during training. With 5 examples, MineGAN memorizes the training set.

Results are shown in Fig. 12. Our model consistently outperforms most 5 and 10-shot methods in terms of quality, and displays improved diversity even when compared to methods trained with a hundred images. With 5 images, MineGAN [55] memorizes the data. Fig. 13 shows qualitative results for selected methods. See Appendix H for more.

In addition to these comparisons, we find that in many cases, using our method as a pre-training step before employing a few-shot method improves synthesis performance. See Appendix C.

5.4. Ablation study

We evaluate our suggested modifications through an ablation study. See results in Fig. 14. The global loss approach consistently fails to produce meaningful results, across all domains and modifications. Meanwhile, our directional-loss model with adaptive layer-freezing achieves the best visual results. In some cases, quality can be improved further by training a latent mapper.

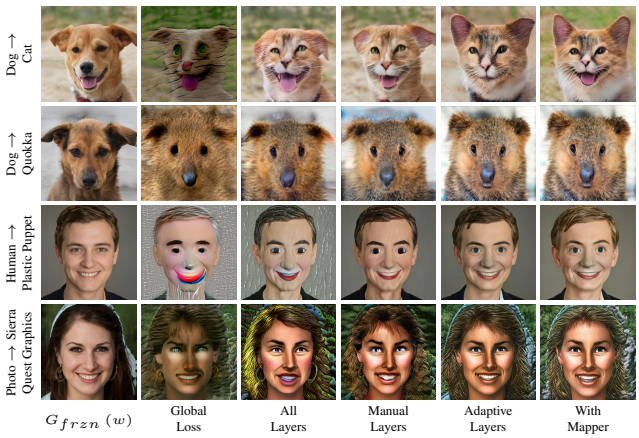

Figure 14. Images synthesized by a transformed generator when changing individual network components. The first column shows images created by the source generator. We then show images produced by generators trained using: the global CLIP loss (Eq. (1)), training all layers, manually selecting layers, using the adaptive layer-freezing method, and when adding a StyleCLIP [37] mapper. The mapper is only needed for more extensive shape changes.

6. Conclusions

We presented StyleGAN-NADA, a CLIP-guided zero-shot method for Non-Adversarial Domain Adaptation of image generators. By using CLIP to guide the training of a generator, rather than an exploration of its latent space, we are able to affect large changes in both style and shape, far beyond the generator's original domain.

The ability to train generators without data leads to exciting new possibilities - from editing images in ways that are constrained almost only by the user's creativity, to synthesizing paired cross-domain data for downstream applications such as image-to-image translation.

Our method, however, is not without limitations. By relying on CLIP, we are limited to concepts that CLIP has observed. Textual guidance is also inherently limited by the ambiguity of natural language prompts. When one describes a 'Raphael Painting', for example, do they refer to the artistic style of the Renaissance painter, a portrait of his likeness, or an animated turtle bearing that name?

Our method works particularly well for style and fine details, but it may struggle with large scale attributes or geom-

etry changes. Such restrictions are common also in few-shot approaches. We find that a good translation often requires a fairly similar pre-trained generator as a starting point.

We focused on transforming existing generators. An intriguing question is whether one can do away with this requirement and train a generator from scratch, using only CLIP's guidance. While such an idea may seem outlandish, recent progress in inverting classifiers [14] and in generative art [12, 32] gives us hope that it is not beyond reach.

We hope our work can inspire others to continue exploring the world of textually-guided generation, and particularly the astounding ability of CLIP to guide visual transformations. Perhaps, not too long in the future, our day-to-day efforts would no longer be constrained by data requirements - but only by our creativity.

Acknowledgments We thank Yuval Alaluf, Ron Mokady and Ethan Fetaya for reviewing early drafts and helpful suggestions. Assaf Hallak for discussions, and Zonzge Wu for assistance with StyleCLIP comparisons.

References

[1] *Partitioning Around Medoids (Program PAM)*, chapter 2, pages 68–125. John Wiley and Sons, Ltd, 1990.

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

[3] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021.

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model, 2021.

[5] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. *arXiv preprint arXiv:2104.02699*, 2021.

[6] Christos Aridas, Jan-Oliver Joswig, Timothée Mathieu, and Roman Yurchak. scikit-learn-extra module, 2019. https://scikit-learn-extra.readthedocs.io/en/stable/index.html.

[7] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word, 2021.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[9] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, A. E. Kholy, Faisal Ahmed, Zhe Gan, Y. Cheng, and Jing jing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[12] Katherine Crowson. Vqgan + clip, 2021. https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN.

[13] Karan Desai and J. Johnson. VirTex: Learning visual representations from textual annotations. *ArXiv*, abs/2006.06666, 2020.

[14] Xin Dong, Hongxu Yin, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. Deep neural networks are surprisingly reversible: A baseline for zero-shot inversion, 2021.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[16] Karl Pearson F.R.S. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[19] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

[20] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[21] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks, 2021.

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[25] Gen Li, N. Duan, Yuejian Fang, Daxin Jiang, and M. Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *Proc. AAAI*, 2020.

[26] Liunian Harold Li, Mark Yatskar, Da Yin, C. Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019.

[27] Xiujun Li, Xi Yin, C. Li, X. Hu, Pengchuan Zhang, Lei Zhang, Longguang Wang, H. Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[28] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*, 2020.

[29] 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*, 2020.

[30] Jiasen Lu, Dhruv Batra, D. Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[31] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. In *CVPR AI for Content Creation Workshop*, 2020.

[32] Ryan Murdock. The big sleep, 2021. https://twitter.com/advadnoun/status/1351038053033406468.

[33] Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or. Large: Latent-based regression through gan semantics, 2021.

[34] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2750–2758, 2019.

[35] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021.

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[37] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021.

[38] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[40] 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. *arXiv preprint arXiv:2008.00951*, 2020.

[41] Esther Robb, Wen-Sheng Chu, Abhishek Kumar, and Jia-Bin Huang. Few-shot adaptation of generative adversarial networks. *ArXiv*, abs/2010.11943, 2020.

[42] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. *arXiv preprint arXiv:2008.01392*, 2020.

[43] 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*, 2021.

[44] Kim Seonghyeon. Stylegan2-pytorch. https://github.com/rosinality/stylegan2-pytorch, 2020.

[45] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020.

[46] Guoxian Song, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. Agilegan: Stylizing portraits by inversion-consistent transfer learning. *ACM Trans. Graph.*, 40(4), July 2021.

[47] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *Proc. ICLR*, 2020.

[48] Hao Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP/IJCNLP*, 2019.

[49] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.

[50] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021.

[51] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. *ArXiv*, abs/2104.03310, 2021.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[53] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Cross-domain and disentangled face manipulation with 3d guidance, 2021.

[54] Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition, 2018.

[55] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: Effective knowledge transfer from gans to target domains with few images. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[56] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and B. Raducanu. Transferring gans: generating images from limited data. In *ECCV*, 2018.

[57] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models, 2021.

[58] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey, 2021.

[59] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021.

[60] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. *arXiv preprint arXiv:2106.04566*, 2021.

[61] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 672–681, 2021.

[62] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep

learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[64] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.

[65] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

Supplementary Materials
StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators

Rinon Gal[1,2*] Or Patashnik[1] Haggai Maron[2] Amit Bermano[1]
Gal Chechik[2] Daniel Cohen-Or[1]

[1]Tel-Aviv University [2]NVIDIA

A. Broader impact

This project aims to provide an effective content creation tool so that artists and others can unleash their creativity, as well as to support machine learning efforts in areas with limited data. At the same time, our tool can also be used for nefarious purposes in the wrong hands.

As CLIP was trained on large collections of images and text from the internet, models using it for supervision are likely to propagate the biases inherent to such data – and our model is no exception. Attempting to guide a face-generator conversion with the text 'doctor', for example, causes the generator to produce mostly males, while using the text 'nurse' has the opposite effect. In Appendix C we propose a method for mitigating this limitation using a small set of images.

B. CLIP-space analysis

We analyze the differences between our directional CLIP loss and the traditional global distance minimization approach by visualizing their behavior in CLIP's embedding space. We first embed all images from the AFHQ cat and dog data sets [11] into CLIP's multi modal space. We then project them to 2D using PCA [16]. We similarly embed and project the texts "Cat" and "Dog", as well as the fake images synthesized by our generator after training with both the global loss and the directional loss. The results are shown in Fig. 15.

Our results align with the intuition presented in Sec. 4. In the case of the global CLIP loss (Fig. 15b), we are optimizing towards a single target. There is no benefit to maintaining a diverse distribution, and results visibly collapse to a single region of the embedding space. In contrast, the directional loss (Fig. 15c) discourages this collapse and successfully maintains a higher degree of diversity.

C. Few-Shot CLIP-Guidance

While our method focused on zero-shot domain adaptation, it is possible to leverage similar ideas for few-shot training. We investigate two alternative approaches for

* Work was done during an internship at NVIDIA

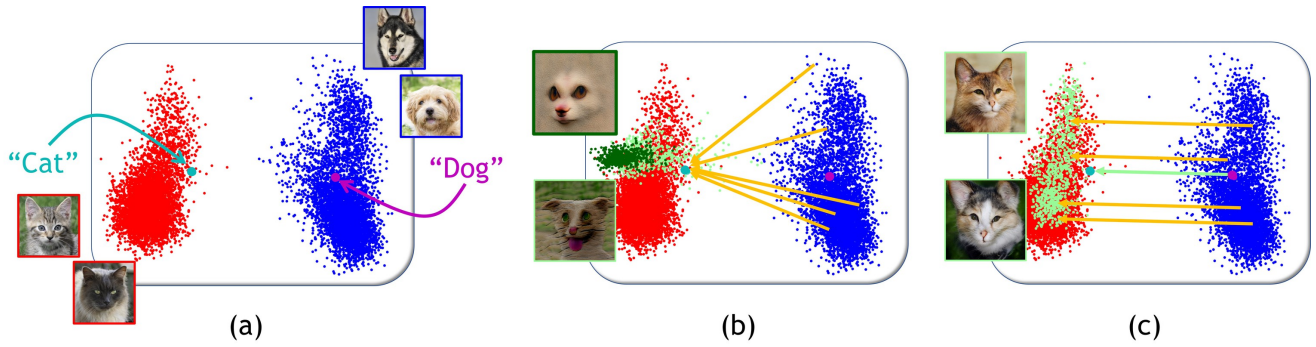

Figure 15. PCA-based visualization of the effects of the directional loss. (a) We embed real dog (blue) and cat (red) images [11] into CLIP's multi-modal space. We additionally embed the strings "Dog" (purple) and "Cat" (cyan). (b) A visualization of the direct, global loss. When the training objective is minimizing distances to one fixed point, we observe adversarial solutions (light green) and eventual collapse of all images to a single region (dark green). (c) A visualization of our directional loss. Generated dogs are shifted along a cross-domain direction, avoiding adversarial attacks and maintaining better diversity (light green).

leveraging the semantic knowledge of CLIP for few-shot domain adaptation.

Image-based directions In the first approach we consider the scenario where a small image set ($\sim 3-5$ images) is available. In such a scenario, rather than using a CLIP-space direction between two pairs of textual descriptions, it is possible to instead consider the CLIP-space direction between the images produced by the original generator and the domain represented by the small image set. Such an approach leverages CLIP in order to encode the *semantic* difference between images from both domains. The directional loss then takes the form:

$$\begin{aligned} \Delta I_{real} &= \frac{1}{N_r}\sum_{i=1}^{N_r} E_I\left(I_i\right) - \frac{1}{N_s}\sum_{i=1}^{N_s} E_I\left(G_{frozen}\left(w_i\right)\right) \ , \\ \Delta I_{gen} &= E_I\left(G_{train}\left(w\right)\right) - E_I\left(G_{frozen}\left(w\right)\right) \ , \\ \mathcal{L}_{direction} &= 1 - \frac{\Delta I_{gen}\cdot\Delta I_{real}}{|\Delta I_{gen}|\,|\Delta I_{real}|} \ . \end{aligned} \tag{3}$$

Here N_r is the size of the real-image set, I_i is the i-th image in the set, and N_s is the number of images sampled from the source domain generator. Our experiments use $N_s = 16$.

This approach holds several advantages over standard few-shot methods: it better maintains the structure of the latent space, shows a higher degree of identity preservation (Fig. 26), trains in a fraction of the time, and does not require the images describing the target domain to be aligned nor preprocessed in a manner fitting the source domain.

In comparison to our proposed zero-shot method, the few-shot approach can help alleviate some of the limitations of the model. In particular, it offers a way to avoid linguistic ambiguity by presenting an example of the specific domain we wish to mimic. Moreover, it can be used to better target specific styles which may be difficult to describe through text. However, as CLIP's embedding space is semantic in nature, this process is not guaranteed to converge to the exact realization of the style. Finally, the few-shot approach can be used to combat the biases learned by CLIP. For example, by providing images of medical professionals from both sexes, one may avoid CLIP's preference for male doctors or female nurses. See Fig. 16 for an example.

Zero-shot pre-training In our second approach, we consider the scenario where a few dozen or hundred images are available. In such a scenario, few-shot models can already achieve reasonable performance. However, their performance typically depends on the similarity between the source generator and their desired target domain [35]. We show that their performance can therefore be improved (and in some situations, considerably so) if, rather than fine-tuning a model from some semi-related source domain, we first use our zero-shot approach to decrease the gap between the two domains.

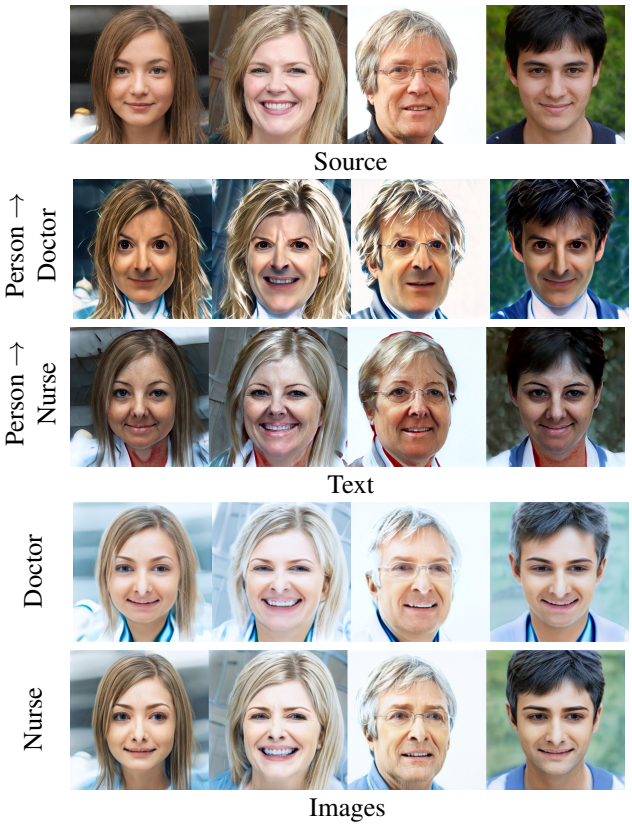

Figure 16. Textual bias and ambiguity in domain adaptation. We transform an FFHQ [22] model into doctors and nurses. When the conversion is done using text, CLIP's learned biases manifest in the new domain. In particular, the text 'doctor' converts the individuals to males, while the text 'nurse' converts them to females. Furthermore, the images generated when using the text 'doctor' resemble the actor David Tennant from the television show 'Doctor Who'. When using image targets, both issues dissapear. However, the model may instead pick up biases found in the few-shot set, such as facial expressions or hair color.

Let F denote a few-shot method that converts a generator between domains using an image set $\{I_{real}\}$, and let N denote our zero-shot method, then we propose a few-shot adaptation of the form:

$$\begin{aligned} G_{\text{NADA}} &= N\left(G_S, t_{source}, t_{target}\right) \\ G_T &= F\left(G_{\text{NADA}}, \{I_{real}\}\right) \ , \end{aligned} \tag{4}$$

where G_S and G_T are the source and target generators respectively, G_{NADA} is an intermediate generator, created by applying our zero-shot method to G_S using the textual prompts t_{source}, and t_{target}.

Typical few-shot adaptation methods require a well-trained discriminator for the source domain. Our method, however, modifies only the generator. We therefore investigate three possible solutions to this generator-discriminator divergence: In the first, we simply ignore the divergence

Table 1. FID (↓) of alternative methods for applying StyleGAN-NADA as a pre-training approach for improving few-shot models. We show results for conversion of cat models from an AFHQ-Dog [20] generator using 10 images. For each model we show the baseline results achieved without pre-training, and the results achieved when: pre-training only the generator ('Pretrained G'), fine-tuning both the generator and the discriminator on StyleGAN-NADA generated images ('Finetuned G + D'), and when allowing the first 50 steps of the few-shot training to modify only the discriminator ('catch-up').

Method	Ojha *et al*.	MineGAN	TGAN	TGAN + ADA
Baseline	45.13	79.31	87.11	52.70
Pre-trained G	45.61	**48.15**	100.61	51.48
Fine-tuned G + D	48.38	56.89	56.46	60.82
D 'catch-up'	**41.22**	51.81	**54.49**	**49.76**

Table 2. FID (↓) for selected few-shot models when fine-tuning a source generator with and without a pre-training step using our method. Cat models were converted from an AFHQ-Dog generator. The sketch model was converted from an official FFHQ 256x256 checkpoint. For every model and training set we compare the pre-trained and direct fine-tuning results and mark the winner in bold. For each set we further highlight the best performing model in blue.

Model	Pre-training?	Set		
		Cats (10)	Cats (100)	Sketches (10)
Ojha *et al*.	×	45.13	27.54	72.74
	✓	**41.22**	**16.17**	**69.92**
MineGAN	×	79.31	27.34	62.27
	✓	**51.81**	**21.33**	**60.55**
TGAN	×	87.11	**19.14**	69.44
	✓	**54.49**	19.41	**56.52**
TGAN + ADA	×	52.70	**19.09**	**56.76**
	✓	**49.76**	25.47	57.18

and employ the few-shot methods as usual, using G_{NADA} along with the source discriminator, D_S. In the second approach, we allow the discriminator to 'catch-up' by first performing a few training iterations where only the discriminator is updated, using G_{NADA} as the source of fake data, and the few-shot set for samples of the target domain. In the final approach, we use G_{NADA} and an accompanying StyleCLIP mapper to synthesize a large collection of images and then fine-tune G_S and D_S using this large collection before using them as the sources for the few-shot method.

We first compare all three methods using the dog-to-cat 10-image setup. In all cases we use the same number of training iterations as the baseline model (including any iterations where only the discriminator is trained). For the 'Fine-tuned G + D' configuration we generated 25 thousand images and fine-tuned the original dog model on this synthetic set for 5000 iterations. FID metrics were calculated following Ojah *et al*. [35], by sampling 5000 images from the fine-tuned model and comparing with the full (non few-shot) target set. The results are shown in Tab. 1. These results indicate that, of the three alternatives, starting with a brief discriminator-only training session yields the most consistent improvements. We hypothesize that the initial synchronization of the discriminator helps focus it on features that differentiate real cats from fake ones, rather than those that differentiate cats from dogs. In the fine-tuning case, it is possible that training on images generated with an additional StyleCLIP-mapper step leads to a reduction in diversity that harms the downstream adaptation methods. MineGAN portrays a particularly large improvement even when using just the pre-trained generator, likely because it manages to identify the latent-regions where the good cats reside and focus the network's attention on them.

Having determined that a discriminator 'catch-up' session produces the most consistent improvements, we turn to evaluating our pre-training method on additional domains and levels of supervision. The results are shown in Tab. 2. In all experiments we show the results of the 'catch-up' method, even where one of the alternatives achieves superior results. In almost all cases, pre-training using our zero-shot method leads to lower FIDs. In some cases, pre-training using StyleGAN-NADA leads to remarkable improvements of more than 40% in FID scores. These results indicate that our method can aid in reducing the domain gap before the application of the few-shot method, giving the later a much more convenient starting point.

D. Beyond StyleGAN

In addition to StyleGAN, we investigated our model's ability to convert existing classes in a more localized manner using OASIS [43], a SPADE-like [36] model that synthesizes images from segmentation masks. In this setup, we utilize a model pre-trained on the COCO-stuff dataset [9], and aim to change one of the model classes to a novel class which shares the shape of the source class, but differs in texture. To this end, we employ the same training architecture and training losses presented in the core paper, with two modifications: First, before passing a generated image to CLIP, we mask all regions which do not belong to our designated class. In this manner, CLIP-space directions are calculated using only the regions with the class we wish to change. Second, we seek to minimize change throughout all regions outside the mask, as well as all images where the designated class does not appear. As such, we employ both an L2 and an LPIPS [63] loss between all the masked regions in the source and target generator outputs. Qualitative results are shown in Fig. 17.

These results demonstrate that our framework can be readily applied to other generative models with minimal effort. In this sense, our method is not a StyleGAN-specific tool, but rather a general framework for training generative models without data.

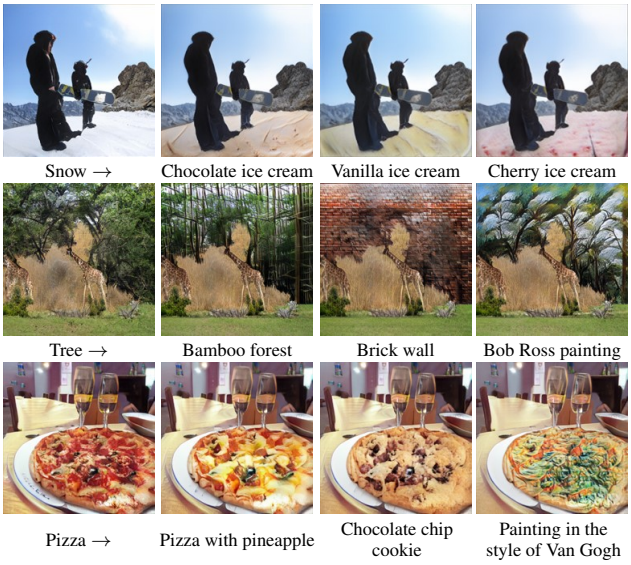

Figure 17. Replacing COCO-Stuff [9] classes with novel classes decribed through text. All images were generated using OASIS [43]. Images in the left column were created by the official pre-trained model. In each row we fine-tune the model so that one class changes its identity in the synthesized images. Target texts are provided below each modified image.

E. Identity Preservation

We provide additional examples that showcase our method's ability to preserve identity between different domains. Fig. 18 shows domain adaptation results using both synthetic and real images (inverted using e4e [49]). StyleGAN-NADA succesfully preserves the source identity in the new domain and even converts appropriate accessories such as hats and eyeglasses.

F. Cross-Model interpolation

In addition to supporting latent-space interpolations and editing in the new domains, we observe that our models allow for an additional form of interpolation - between the model parameters in two different domains. This serves as further demonstration for the strong coupling of the latent spaces across our transformed models. Furthermore, doing so enables additional applications such as generating videos of smooth transitions across a wide range of domains. Fig. 19 demonstrates such transitions. Our project page includes a video with further demonstrations. For more applications of network interpolations, see [54] or the concurrent work by Wu *et al*. [57].

G. Additional samples

We provide additional synthesized results from a large collection of source and target domains. In Figs. 20 and 21 we show results from models converted from the face domain. In Fig. 22 we show results from models converted from the church domain. In Fig. 23 we show additional results from the dog domain. Finally, in Fig. 24 we show additional animal transformations.

H. Qualitative few-shot comparisons

We provide qualitative comparisons to few shot methods. In Fig. 25 we transform the official StyleGAN-ADA [20] AFHQ-Dog [11] model to the cat domain using our zero-shot method, and using few-shot models trained on samples from the AFHQ-Cat [11] set.

In the extreme low-data regime, competing methods either fail to produce meaningful images when faced with such a domain gap, or they simply memorize the training set. Moreover, only our method and the 100-image variant of Ojha *et al*. [35] maintain consistent correspondence between images in the source and target domain.

In Fig. 26 we transform the official StyleGAN-ADA [20] FFHQ [22] 256x256 checkpoint to sketches, using the few-shot set of Ojha *et al*. [35]. Our zero-shot transformation used the texts "Photo" to "Black and white pencil sketch". Our method maintains significantly higher quality and preserves more detail from the source domain. However, designing a textual prompt which targets the exact style portrayed in those images is difficult. By using 3 images from the target set (Appendix C) rather than text, we are able to reduce the gap between the styles, but not completely eliminate it. Even when transferring across 'close' domains, most competing methods produce considerable artifacts, display significant mode collapse, or maintain very limited correspondence with the source domain.

I. Training details

Hyper parameter choices. For texture-based changes we find that a training session typically requires 300 iterations with a batch size of 2, or roughly 3 minutes on a single NVIDIA V100 GPU. In some cases ("Photo" to "Sketch") training can converge in 50 iterations and using less than a single minute, an improvement of two orders of magnitude compared to recent adversarial methods designed for training speed [46].

In practice, calibrating the number of training iterations is trivial. Since training converges quickly, one may simply set the number of iterations arbitrarily high and investigate intermediate model outputs to identify the point where the model produced the most pleasing results.

For animal changes, training typically lasts 2000 iterations. We then train a StyleCLIP mapper using the modified G_{train} as a base model. The entire process takes roughly 6 hours on a single NVIDIA V100 GPU.

For all experiments, we use an ADAM Optimizer with a learning rate of 0.002.

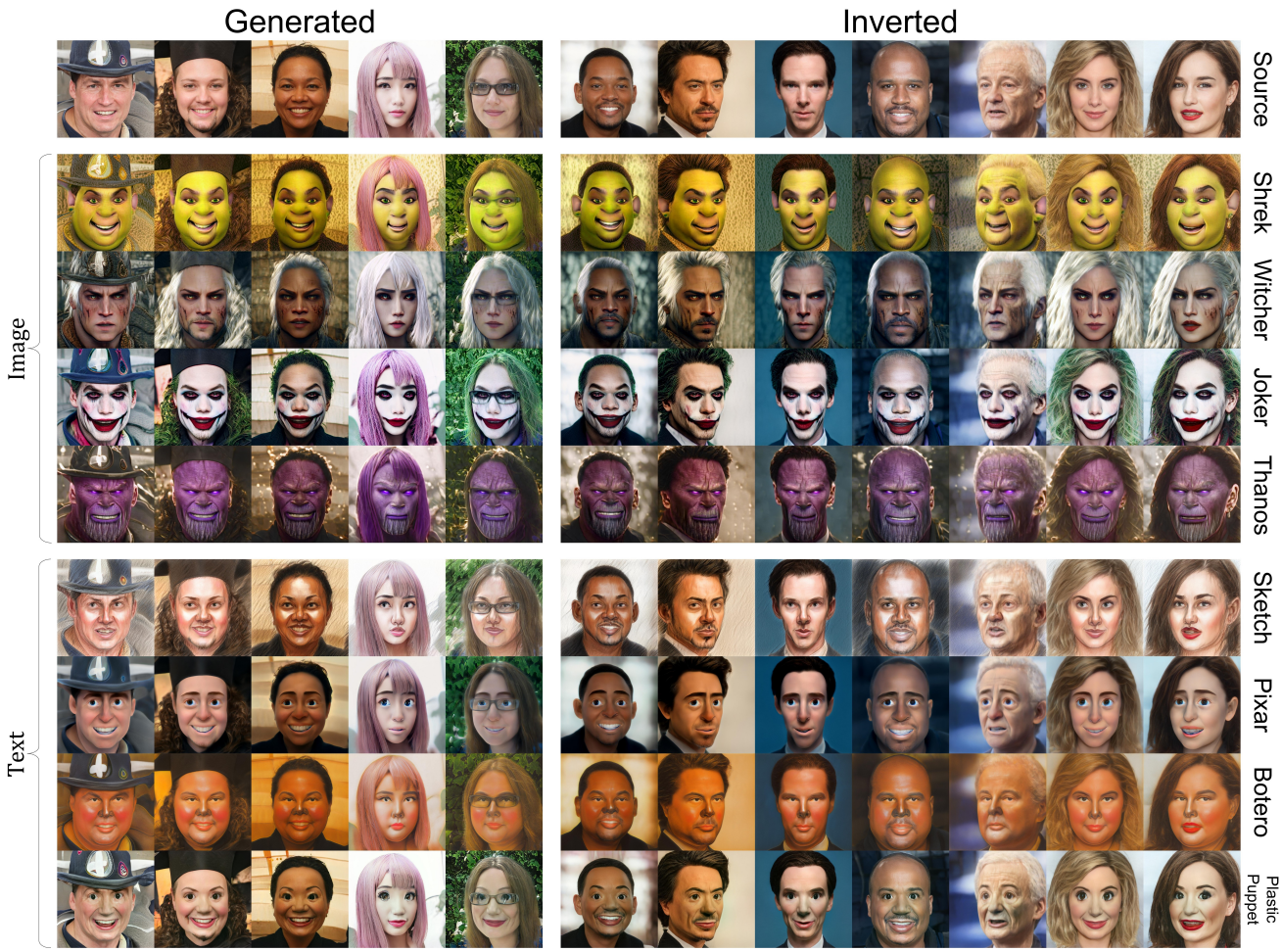

Figure 18. Cross-domain identity transfer for generated and real (inverted) images. Our method maintains a high degree of identity preservation, including accessories such as hats and eyeglasses. The method works equally well with both synthesized and real images. Furthermore, it can be employed using both textual and image targets.

When using our adaptive layer-freezing approach, we set the optimization batch size N_w to 8, and the number of optimization iterations N_i to 1. For texture-based changes we allow the network to modify all layers. For small shape changes (*e.g.* 'Human' to 'Werewolf') we set the number of trainable layers k to $\frac{2}{3}$ of the number of network layers (*i.e.* $k = 12$ for FFHQ). When modifying animals we set $k = 3$.

We use the Vision-Transformer [15] based CLIP models, 'ViT-32/B' and 'ViT-B/16'. We observe that the two models tend to focus on different levels of detail. For texture-based changes, the choice of a model leads to minor variations in artistic styles. As such, the 'optimal' choice is a matter of individual preference. For shape changes we obtain the best results by using both models in tandem, allowing the model to better focus on both global shape and more local texture.

In a similar manner, we observe that the choice of style mixing probability in StyleGAN2's training can affect the artistic style of the generated images. All animal change experiments were conducted without mixing. For other experiments, we report our hyperparameter choices in Tab. 3.

All remaining StyleGAN2 parameters were unmodified.

StyleCLIP mapper As discussed in Sec. 5, in some scenarios we use the latent mapper from StyleCLIP [37], to better identify latent-space regions which match the target domain. Unfortunately, the mapper occasionally induces undesirable semantic artifacts on the image, such as opening an animal's mouth and enlarging tongues. We observe that these artifacts correlate with an increase in the norm of the CLIP-space embeddings of the generated images. We thus discourage the mapper from introducing such artifacts by constraining these norms by introducing an additional loss during mapper training:

$$\mathcal{L}_{norm} = \left| E_I\left(G\left(w\right)\right) - E_I\left(G\left(M\left(w\right)\right)\right)\right|^2 , \quad (5)$$

Figure 19. Cross-domain image interpolations. Our models can be used to smoothly transform an image between domains by interpolating the model weights, rather than latent codes. The left-most image in each row is a reconstruction in the original StyleGAN2-FFHQ [23] model, obtained using e4e [49]. The other images demonstrate model-based interpolation through two different domains. Interpolation works not only from the source domain to new targets, but also between different target domains.

Table 3. Hyper parameter choices for select models shown in the paper. See Appendix I for more details on parameter choices.

Experiment	Iterations	ViT-B/32	ViT-B/16	Mixing	Adaptive k
White Walker	200	✓	✓	0.9	18
Werewolf	300	✓	✓	0.9	12
Elf	200	✓	✓	0.9	18
Edvard Munch	300	✓	✓	0.9	18
Sketch	300	✓	×	0.0	18
Pixar	130	✓	✓	0.9	18
Zombie	150	✓	×	0.9	18
Cubism	300	✓	×	0.0	18
Princess	200	✓	✓	0.9	18
Modigliani	400	✓	✓	0.0	18
Shire	300	✓	×	0.9	14
Nicolas Cage	300	✓	×	0.9	12
Cat	2000	✓	✓	0.0	3
Bear	2000	✓	✓	0.0	3

where E_I is the CLIP image encoder, G is the fine-tuned generator, w is a sampled latent code and M is the latent mapper. When training a latent mapper, we set $\lambda_{L2} = 0.5$ and $\lambda_{embedding-norm} = 0.2$.

J. Licenses and data privacy

Tables 4 and 5 outline the models used in our work, the datasets used to train them, and their respective licenses.

The FFHQ [22] dataset contains biometric data in the form of face images. Images in this set were crawled from Flickr, without reaching out to their owners. However, they were all uploaded under permissive licenses which allow free use, redistribution, and adaptation for non-commercial purposes. The FFHQ curators provide contact details for individuals that want their images removed from the set.

Table 4. Models used in our work, their sources and licenses.

Model	Source	License
StyleGAN2	[23]	Nvidia Source Code License-NC
scikit-learn-extra	[6]	BSD 3-Clause
pSp	[40]	MIT License
e4e	[49]	MIT License
ReStyle	[5]	MIT License
InterFaceGAN	[45]	MIT License
StyleCLIP	[37]	MIT License
StyleFlow	[3]	CC BY-NC-SA 4.0
CLIP	[39]	MIT License
StyleGAN2-pytorch	[44]	MIT License
StyleGAN-ADA	[20]	Nvidia Source Code License
MineGAN	[55]	MIT License
Ojha *et al.*	[35]	Adobe Research License
OASIS	[43]	GNU Affero GPL

Table 5. Datasets used in our work, their sources and licenses.

Dataset	Source	License
FFHQ	[22]	CC BY-NC-SA 4.0[I]
LSUN	[62]	No License
AFHQ	[11]	CC BY-NC 4.0
Sketches	[35]	Adobe Research License
COCO-Stuff	[9]	CC BY 4.0

Figure 20. Additional images synthesized using models adapted from StyleGAN2-FFHQ [23] to a set of textually-prescriped target domains. All images were sampled randomly, using truncation with $\psi = 0.7$. The driving texts appear to the left of each row.

Figure 21. Additional images synthesized using models adapted from StyleGAN2-FFHQ [23] to a set of textually-prescriped target domains. All images were sampled randomly, using truncation with $\psi = 0.7$. The driving texts appear to the left of each row.

Figure 22. Additional images synthesized using models adapted from StyleGAN2 [23] LSUN Church [62] to a set of textually-prescriped target domains. All images were sampled randomly, using truncation with $\psi = 0.7$. The driving texts appear to the left of each row.

Figure 23. Additional images synthesized using models adapted from StyleGAN-ADA [20] AFHQ-dog [11] to a set of textually-prescriped target domains. All images were sampled randomly, using truncation with $\psi = 0.7$. The driving texts appear to the left of each row.

Figure 24. Additional generator transformations to multiple animal domains. In all cases we begin with a StyleGAN-ADA [20] AFHQ-Dog [11] model. All generators are adapted using our method and a StyleCLIP [37] latent mapper. For all experiments, the source domain text was 'Dog'. The target domain text is shown below each image.

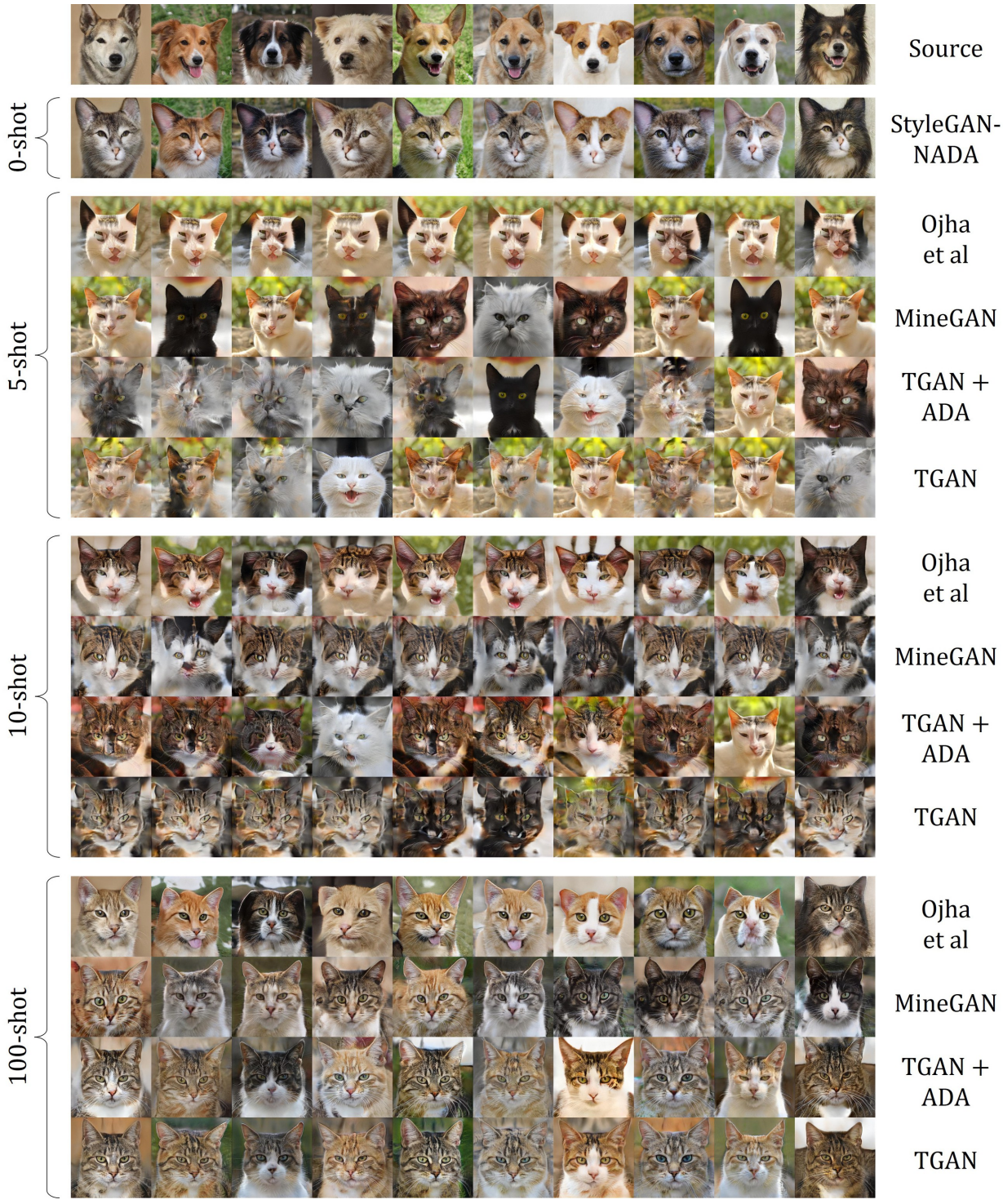

Figure 25. Comparison to few-shot methods on a challenging shape-modifying domain adaptation task. In all cases we begin with a StyleGAN-ADA [20] model pre-trained on AFHQ-Dog [11]. Our method transforms the domain of the generator in a 0-shot manner. For the remaining approaches, we train using 5, 10 and 100 image sets. In low data settings, competing methods produce considerable artifacts or simply memorize the training set. Even with a hundred images, most competing methods fail to maintain correspondence between the source and target domain beyond broad attributes such as pose.

Figure 26. Comparison to few-shot methods on an image-to-sketch translation task. Our method creates high quality sketches and maintains considerably more diversity and better correspondence with the source domain. Targeting the exact artistic style using natural language prompts, however, is difficult. By guiding domain adaptation using a few samples from the set (Appendix C), we can adapt more features from the specific style. However, a gap still remains.