Development of domain adaptation methods for generative models

Literature review

Kirill Korolev

Contents

1 StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery

In this paper [6] authors try to incorporate pretrained CLIP into StyleGAN within text-based image manipulation task. In particular, they propose 3 techniques:

1 Latent optimization by using CLIP loss between generated stylized image and embedding of text.

2 Training a latent mapper for a specific text prompt that learns a residual added to $w \in \mathcal{W}+$ corresponding to input image to be manipulated.

3 Mapping a prompt to a global direction in a style space $\mathcal{S}$.

1.1 Latent optimization

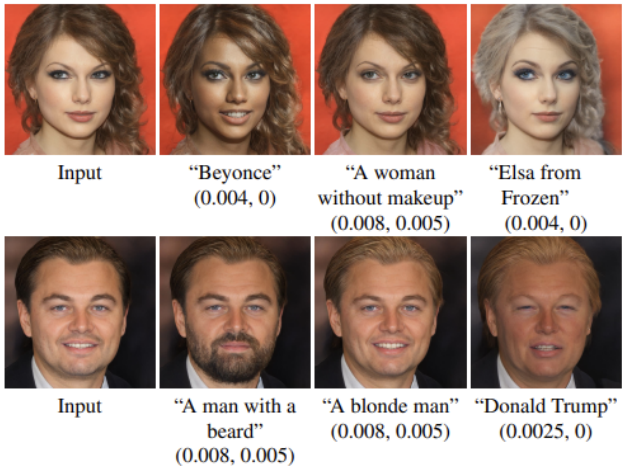


Figure 1.1: Edits obtained by a latent optimization with different values of λ_1 and λ_2.

More formally, given a latent code $w_s \in \mathcal{W}+$ after input image inversion and a text prompt t, the task is to solve the following optimization problem

$$w = \arg\min_{w \in \mathcal{W}+} D_{CLIP}(G(w), t) + \lambda_1 ||w - w_s||_2 + \lambda_2 \mathcal{L}_{ID}(w) \tag{1}$$

Here G is a pretrained StyleGAN generator, D_{CLIP} is a cosine similarity between corresponding CLIP embeddings and $\mathcal{L}_{ID}$ is an identity loss that controls similarity of a stylized image to an input image.

This approach is straightforward, but very limited, because the optimization must be done every time for each input image and text.

1.2 Latent mapper

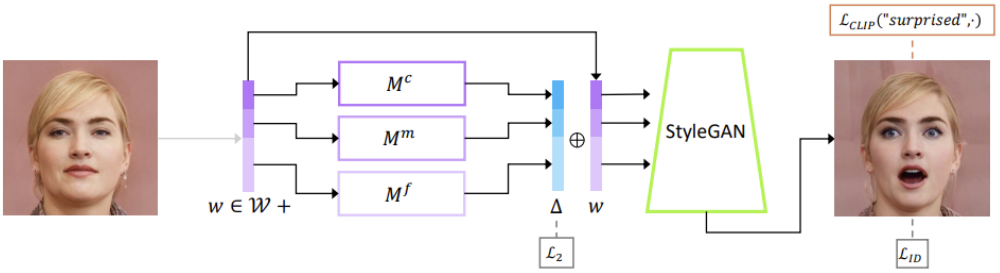


Figure 1.2: A scheme of training of a latent mapper. The source image is inverted into a latent w. Three mapping functions are trained to generate a residual that is added to w to yield a target code, from which the stylized image is generated.

To avoid optimization by $w \in \mathcal{W}+$, it is possible to train a latent mapper for a fixed text prompt that learns a residual to be added to a latent code of an input image. It is known that different layers of a synthesized network in StyleGAN responsible for various semantics of an image. Therefore, authors split a latent into coarse, medium and fine parts $w = (w_c, w_m, w_f)$ and the mapper is defined by $M_t(w) = (M_t^c(w_c), M_t^m(w_m), M_t^f(w_f))$. It is trained by optimizing the following loss rather similar to the previous approach

$$\mathcal{L}(w) = D_{CLIP}(G(w + M_t(w)), t) + \lambda_1 ||M_t(w)||_2 + \lambda_2 \mathcal{L}_{ID}(w) \tag{2}$$

1.3 Global directions

However, this mapper poorly deals with fine-grained details and also the directions in a latent space tend to be similar for a fixed text prompt. Therefore, authors propose to learn a global direction Δs in a style space. Given an original image $G(s)$ and a stylized image $G(s+\Delta s)$, denote their CLIP embeddings as i and $i + \Delta i$. Also, denote CLIP embeddings of their text descriptions as t and $t + \Delta t$. Generally, in CLIP manifolds Δt and Δi are colinear. So, the idea is to find components in a style vector s that influence the collinearity of these two vectors.

Figure 1.3: Edits made with global directions using StyleGAN2 pretrained on FFHQ.

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

In a StyleGAN-NADA paper [4] authors elaborate the idea of global directions for domain adaptation task. They clone a StyleGAN generator G_{frozen}, pretrained on a source domain and freezed in this process, and train G_{train} such that deltas of CLIP embeddings of texts and images are colinear. More formally, they optimize the following

$$\begin{aligned} \Delta T &= E_T(t_{target}) - E_T(t_{source}) \\ \Delta I &= E_I(G_{train}(w)) - E_I(G_{frozen}(w)) \\ \mathcal{L}_{direction} &= 1 - \frac{\langle \Delta I, \Delta T \rangle}{||\Delta I|| \cdot ||\Delta T||} \end{aligned} \tag{3}$$

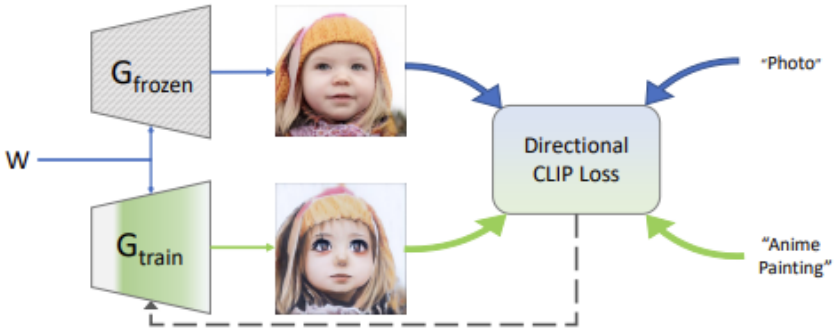


Figure 2.1: Overview of a training setup in StyleGAN-NADA.

They found out that more complex domains require longer training, which destabilizes the network, if the full fine-tuning was done. Therefore, they freeze some of the layers of G_{train} for training only on a subset of network weights.

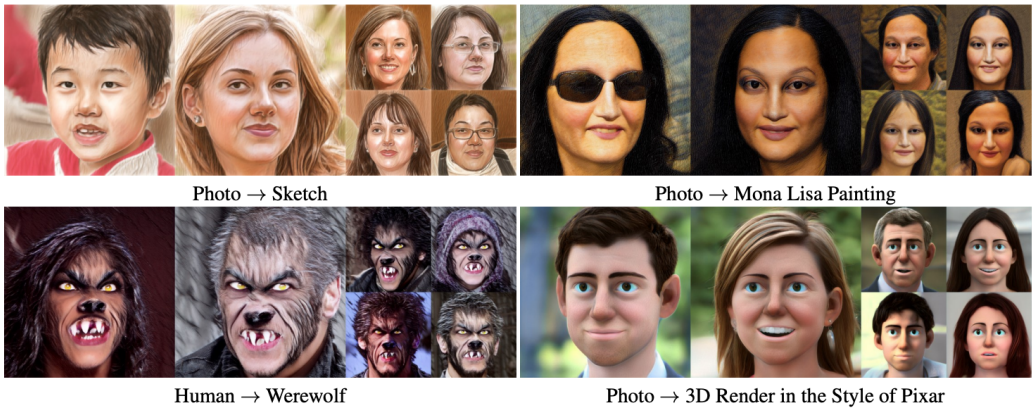
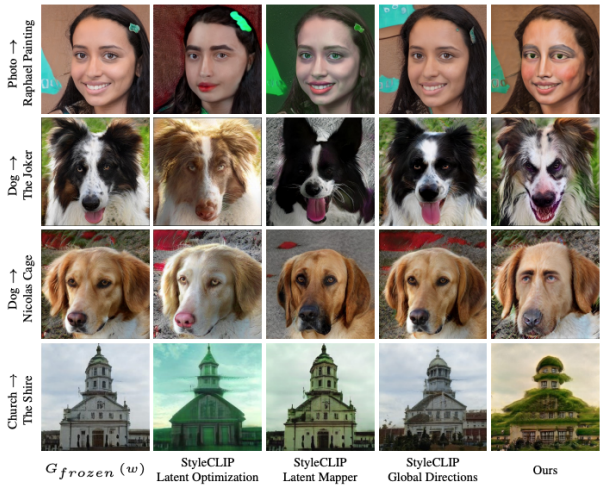
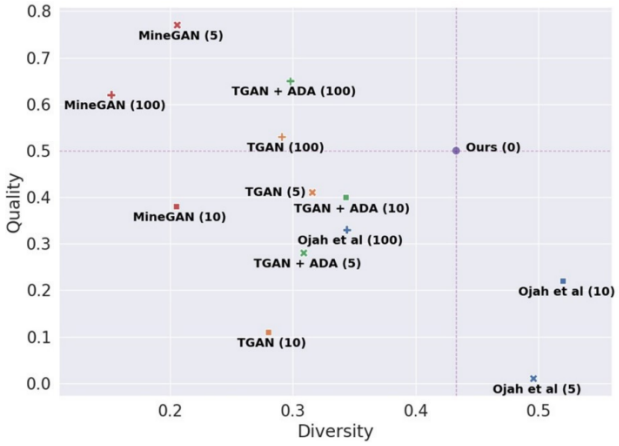


Figure 2.2: Image synthesis using models adapted from StyleGAN2-FFHQ to a set of textually-prescriped target domains.

Authors compare StyleGAN-NADA with several few-shot alternatives: Mine-GAN, TGAN and TGAN + ADA by converting a StyleGAN-ADA AFHQ-Dog model to a cat model. They evaluate quality, which is a percentage of people which preferred other methods instead of theirs, and diversity, which is an average LPIPS distance between clusters of generated images. Authors state that their zero-shot model outperform models, which were fine-tuned on a few or even dozens of images as can be seen in 2.3b.



(a) Comparison with StyleCLIP. None of the three approaches from StyleCLIP succeed in performing an out-of-domain manipulation.



(b) Quality(↑) and diversity(↑) comparison for StyleGAN-NADA and selected few-shot approaches.

Figure 2.3: Visual and quantitative comparison of StyleGAN-NADA to other methods.

3 BlendGAN: Implicitly GAN Blending for Arbitrary Stylized Face Generation

Here [5] authors goal is to train a generator

$$\hat{x}_f, \hat{x}_s = G(z_f, z_s, i) \tag{4}$$

that generates a pair of natural and stylized faces $(\hat{x}_f, \hat{x}_s)$ given their latent codes (z_f, z_s) and a blending factor i.

3.1 Style encoder

Additionaly, they independently train a style encoder E_{style}, which takes an image of a desired style and outputs a latent z_s. Basically, it extracts a Gram matrix of features from pretrained VGG network, which then flattens into a vector. Because of a huge size, which leads to a sparser distribution, an additional MLP is used to reduce its dimensionality and output z_s. Finally, the encoder is trained in a contrastive fashion using a NT-Xent loss, where an input image is augmented using an affine transformation and a similarity is maximized between z_i, z_j that correspond to the same image.

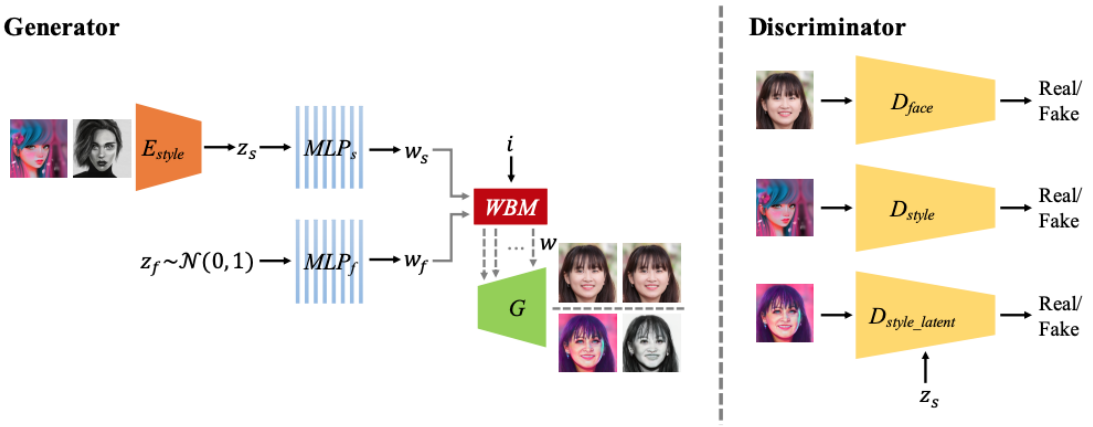


Figure 3.1: Overview of a pipeline used in BlendGAN. E_{style} encoder extracts a style latent z_s, which is propagated with a natural latent z_f separately through MLP to get w_s and w_f. Then they are combined using WBM and fed to a generator.

3.2 WBM

Given $z_f \in \mathcal{N}(0, I)$ and z_s from a style encoder, they are independently propagated through a mapping network of a StyleGAN2, which returns w_f and w_s from $\mathcal{W}$. As it was said different resolution layers of a StyleGAN are responsible for different features of the generated image. Hence, authors introduce the weighted blending module (WBM), controlled by the blending indicator i, which combines these latents in such a way that the blending factor is different for different layers:

$$
\begin{aligned}
w &= w_s \odot \hat{\alpha} + w_f \odot (1 - \hat{\alpha}) \\
\hat{\alpha} &= \alpha \odot m(i; \theta) \\
m(i;\theta) = (m_0, m_1, \ldots, m_{17}),\ m_j &= \begin{cases} 0, j < i \\ \theta, j = i \quad \theta \in (0,1) \\ 1, j > i \end{cases}
\end{aligned} \tag{5}
$$

where α is a learnable parameter and $m(i; \theta)$ controls which layers should be blended.

3.3 Dicriminators

They use three discriminators: D_{face}, which distinguishes between real and fake natural faces, D_{style}, which recognizes real and fake stylized images and D_{style_latent}, which receives a stylized image and a latent and predicts whether this image was generated using this latent. Optimization is done using standard adversarial losses.

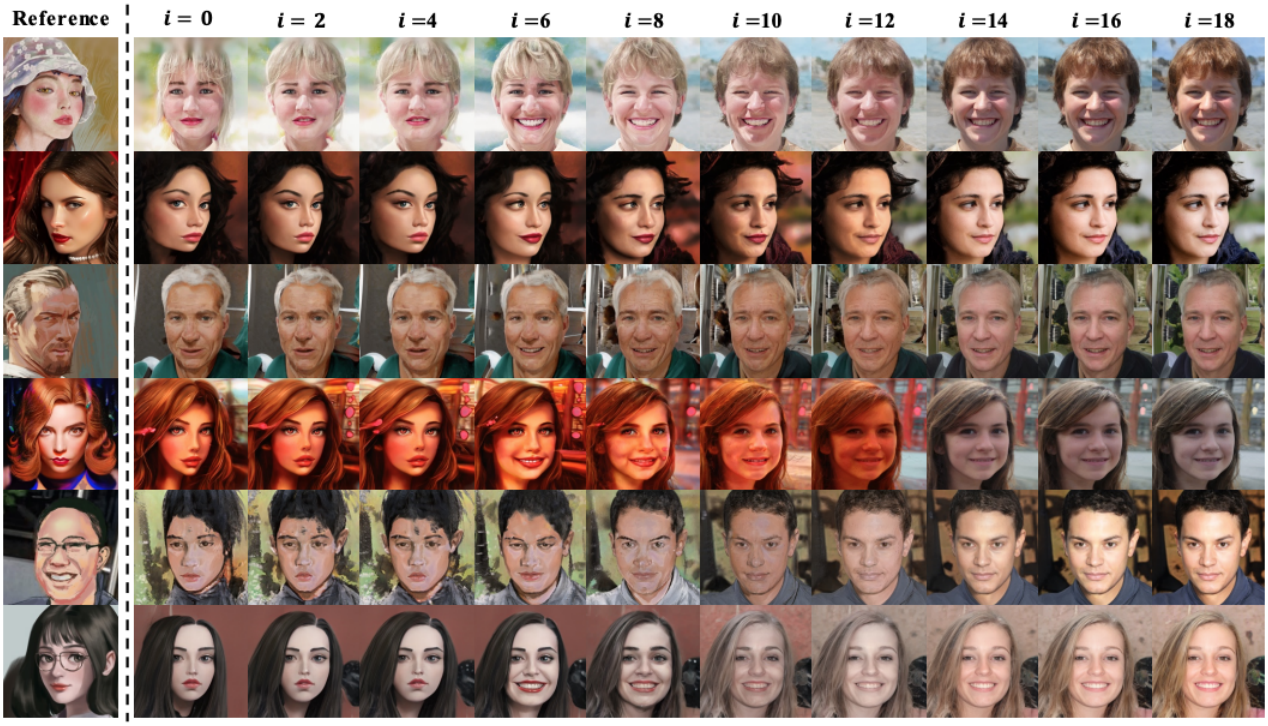


Figure 3.2: Reference-guided stylized-face images with different blending indicators.

3.4 Results

To evaluate the quality of results, authors use FID metric to measure the divergence between the generated images and AAHQ dataset. In addition, they adopt the LPIPS metric to measure the style diversity of generated images.

Indicator i		0	2	4	6	8	10	12	14	16	18
latent	FID ↓	**8.97**	12.45	12.75	23.17	42.45	57.34	68.31	76.91	78.25	76.73
	LPIPS ↑	**0.581**	0.571	0.568	0.515	0.459	0.367	0.304	0.207	0.145	0.159
reference	FID ↓	**3.79**	6.39	6.82	15.08	34.33	51.49	63.73	76.00	77.11	76.97
	LPIPS ↑	**0.661**	0.651	0.650	0.599	0.540	0.450	0.377	0.237	0.191	0.160

Figure 3.3: FID and LPIPS comparison with different blending indicators.

For latent-guided generation, the style latent code z_s is randomly sampled from $\mathcal{N}(0, I)$. Quantitative comparison with other methods is shown in 3.4a. For reference-guided generation, the style latent code z_s is embedded by the style encoder E_{style} from a reference artistic face image. Quantitative comparison is shown in 3.4b.

Method	FID ↓	LPIPS ↑
MUNIT	29.69	0.394
FUNIT	176.96	0.500
DRIT++	22.53	0.448
StarGANv2	50.20	0.312
BlendGAN ($i = 6$)	23.17	0.515
BlendGAN ($i = 0$)	**8.97**	**0.581**

(a) Quantitative comparison on latent-guided stylized-face generation.

Method	FID ↓	LPIPS ↑
AdaIN	37.23	0.345
MUNIT	103.61	0.192
FUNIT	87.71	0.327
DRIT++	31.71	0.241
StarGANv2	50.03	0.307
BlendGAN ($i = 6$)	15.08	0.599
BlendGAN ($i = 0$)	**3.79**	**0.661**

(b) Quantitative comparison on reference-guided stylized-face generation.

4 Image-based CLIP-Guided Essence Transfer

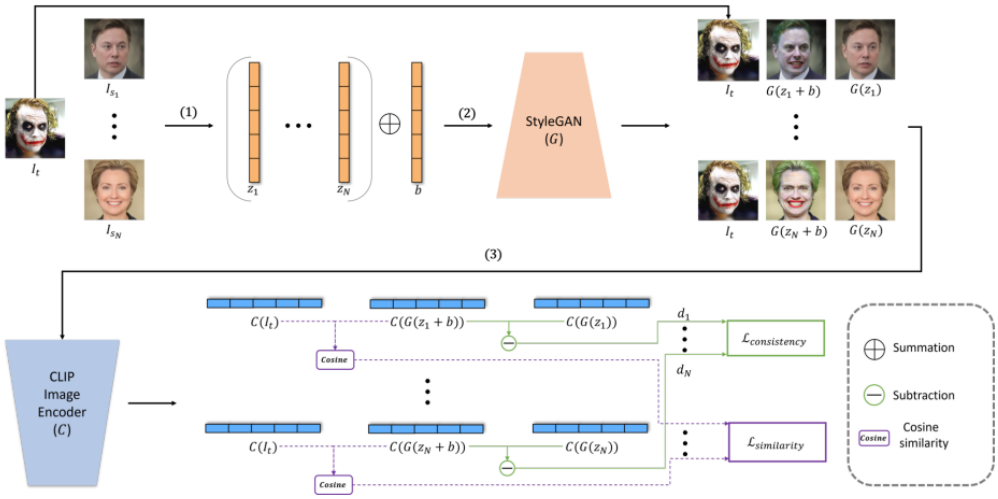


Figure 4.1: An illustration of a loss calculation flow.

The idea of this paper [3] is very similar, what is proposed in StyleCLIP. The problem is to transfer the essence or high level features of an image I_t to source images $I_s = \{G(z) \mid z \in \mathcal{N}(0, I)\}$. So, authors try to find a global direction b, which encodes an essence, such that

$$E_{CLIP}(G(z + b)) - E_{CLIP}(G(z)) = const \quad \forall z \tag{6}$$

This can be equivalently expressed as each difference in a CLIP space is close to each other:

$$\mathcal{L}_{consistency} = \frac{1}{\binom{N}{2}} \left(\sum_{i_{src_1}, i_{src_2} \in I_s} 1 - \frac{\Delta i_{src_1} \cdot \Delta i_{src_2}}{||\Delta i_{src_1}|| \cdot ||\Delta i_{src_2}||} \right) \tag{7}$$

where $\Delta i_{src_i} = E_{CLIP}(G(src_i + b)) - E_{CLIP}(G(src_i))$.

To add a constraint that ties shifts to I_t it is reasonable to optimize the following loss

$$\mathcal{L}_{similarity} = \frac{1}{N}\left(\sum_{z} 1 - \frac{E_{CLIP}(G(z+b)) \cdot E_{CLIP}(I_t)}{||E_{CLIP}(G(z+b))|| \cdot ||E_{CLIP}(I_t)||}\right) \tag{8}$$

Finally, the global direction is found by optimizing

$$b^* = \arg\min_{b} \mathcal{L}_{similarity} + \lambda_1 \mathcal{L}_{consistency} + \lambda_2 ||b||_2 \tag{9}$$

Figure 4.2: Examples of using optimization-based method.

In their second method, instead of optimization by b, they fine-tune an essence encoder, which is a pretrained e4e encoder, that produces b^*.

		Quality	Identity scores		Semantic scores	
		FID (↓)	Source (↑)	Target (↓)	BLIP (↑)	CLIP (↑)
Celebrities Test	StyleGAN-NADA [11]	215.7±26.1	23.0±4.7	33.0±7.1	**84.5±3.6**	**94.0±1.3**
	Mind The Gap [52]	180.4±19.3	27.2±5.6	39.4±8.1	75.8±5.6	75.4±7.0
	JoJoGAN [6]	186.1±12.7	36.0±6.1	50.7±6.9	72.6±7.3	71.8±6.2
	BlendGAN [30]	177.8±12.6	37.6±6.5	**5.2±7.7**	60.8±6.2	58.4±5.2
	StyleCLIP [36]	166.9±9.0	**70.7±26.0**	6.2±6.8	54.8±6.6	55.7±5.0
	Our encoder	188.7±23.2	39.0±6.5	31.9±5.7	69.0±6.0	72.6±5.5
	Our optimization	**163.6±16.7**	43.5±6.8	17.0±6.6	66.9±6.0	74.4±3.2
FFHQ Test	StyleGAN-NADA [11]	220.2±41.8	24.1±5.5	28.3±9.2	**81.1±4.2**	**91.0±3.2**
	JoJoGAN [6]	175.2±15.2	42.3±4.0	41.7±11.4	76.0±6.0	67.1±7.4
	BlendGAN [30]	175.1±14.5	37.6±5.3	**2.4±6.0**	64.4±6.7	54.7±7.8
	Our encoder	175.6±23.5	42.5±5.5	30.8±6.9	72.8±4.9	66.7±6.1
	Our optimization	**161.1±17.2**	**45.2±8.6**	17.0±7.2	74.1±4.9	74.8±5.8

Figure 4.3: Quantitative comparison with baselines. Results that indicate identity loss of the source are marked in orange; results that indicate that no semantic attributes were transferred are marked in red.

Authors introduce two types of metrics: an Identity score, which measures, how well the identity of a source image is maintained, and a Semantic score, which evaluates the degree of transformation.

$$\text{ID-Score}_{source}(I_{s,t}) = \langle R(I_s), R(I_{s,t}) \rangle,\ \text{ID-Score}_{target}(I_{s,t}) = \langle R(I_t), R(I_{s,t}) \rangle \tag{10}$$

where R is a pretrained ArcFace face recognition representation.

$$\text{Semantic-Score}(I_{s,t}) = \langle E_{CLIP}(I_t), E_{CLIP}(I_{s,t}) \rangle \tag{11}$$

5 HyperDomainNet: Universal Domain Adaptation for Generative Adversarial Networks

In a HyperDomainNet paper [2] authors propose several contributions for domain adaption of GANs.

5.1 Domain modulation technique

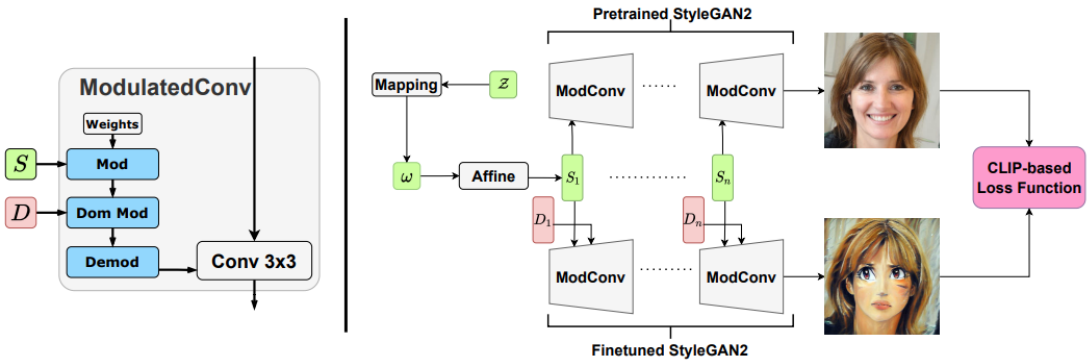
Because it was observed that the mostly changed part during fine-tuning of a generator is a synthesized network, authors decided to revisit modulation and demodulation techniques used in StyleGAN2. They introduce a compact parameterization — a domain vector d of dimension 6 thousand that is embedded in a convolution blocks like a style vector

$$w'_{ijk} = d_i \cdot w_{ijk} \tag{12}$$

The optimization is done in a similar fashion like in StyleGAN-NADA, but they optimize only a vector d.

5.2 Improving diversity

Authors empirically observe that StyleGAN-NADA and MindTheGap struggle with the mode collapsing problem. The main hypothesis for this behaviour is that ΔT and ΔI no longer lie on a CLIP sphere and then it is not reasonable to calculate cosine similarity between them. So, the idea is to preserve the CLIP distances between images before and after domain adaptation



(a) Domain-modulation technique (b) Fine-tuning StyleGAN2 by optimizing the domain vector D

with the following loss.

$$\mathcal{L}_{indomain-angle} = \sum_{i,j}(\langle E_{CLIP}(G_{frozen}(w_i)), E_{CLIP}(G_{frozen}(w_j))\rangle - \langle E_{CLIP}(G_{train}(w_i)), E_{CLIP}(G_{train}(w_j))\rangle)^2 \tag{13}$$

5.3 HyperDomainNet

A compact representation of a domain as a vector d allows to formulate a task of training an encoder $D_\phi(\cdot)$ that predicts a domain parameters given the input target domain, for instance, as a text prompt. Also, authors introduce $\mathcal{L}_{tt-direction}$ loss, which is very similar to a $\mathcal{L}_{direction}$ loss, except that the directions ΔI and ΔT are calculated not between source and target domains, but between different target domains. They use different regularization, instead of $\mathcal{L}_{indomain-angle}$, because it becomes inefficient, if there are small number of domain images. In particular, the norm of a domain parameterization is constrained with $\mathcal{L}_{domain-norm} = ||D_\phi(E_{CLIP}(t)) - 1||^2$, where t is a text description of a domain.

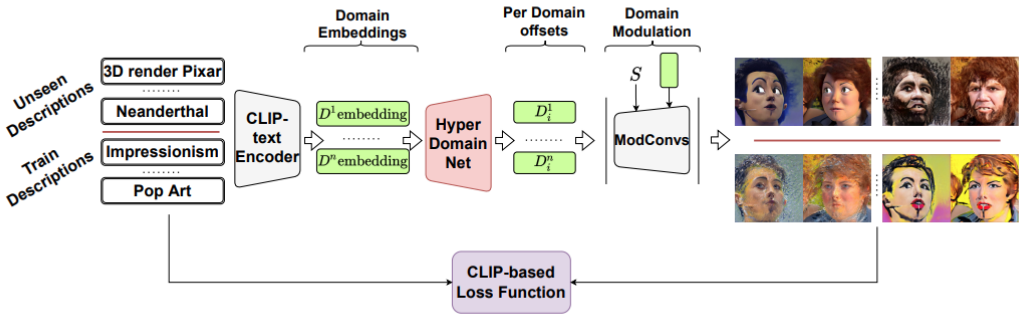


Figure 5.1: Detailed training process of the HyperDomainNet.

5.4 Results

Figure 5.2: Comparison with the original StyleGAN-NADA method and its version with their parameterization.

Besides FID, authors evaluate a precision and recall metrics. Precision is the probability that a random image from the first distribution falls within the support of the second one. Recall is the probability that a random image from the second distribution falls within the support of the first one.

	Model quality			**Model complexity**
Model	FID	Precision	Recall	# trainable parameters
TargetCLIP [4]	199.33	0.000	0.293	9K
Cross-correspondence [24]	158.86	0.001	0	30M
StyleGAN-NADA [6]	124.55	0.118	0	24M
MindTheGap [48]	78.35	0.326	0.017	24M
MindTheGap (our param.)	79.83	0.452	0.017	6k
MindTheGap+indomain	71.46	0.503	0.014	24M
MindTheGap+indomain (our param.)	72.71	0.472	0.028	6k

Figure 5.3: Evaluation of one-shot adaptation methods.

6 StyleDomain: Efficient and Lightweight Parameterizations of StyleGAN for One-shot and Few-shot Domain Adaptation

Authors [1] explore the importance of each part of the StyleGAN2, in particular, a mapping network f_M, affine layers $f_1^A, \ldots, f_N^A$ and a synthesis network are being considered. To analyze the impact of each component, they optimize with respect to only one component at a time. Two settings are being considered: one-shot domains, for example, when only the style of an image is changed and few-shot domains, when the domains are more distant from each other. For one-shot domains in a case of text-based domains the objective from StyleGAN-NADA is utilized. The analysis showed that the optimization by affine layers is sufficient, besides the optimization of the synthesis network. For few-shot domain adaptation the fine-tuning procedure from StyleGAN-ADA was used and in that case affine parameterization didn't show the same results as synthesis network did.

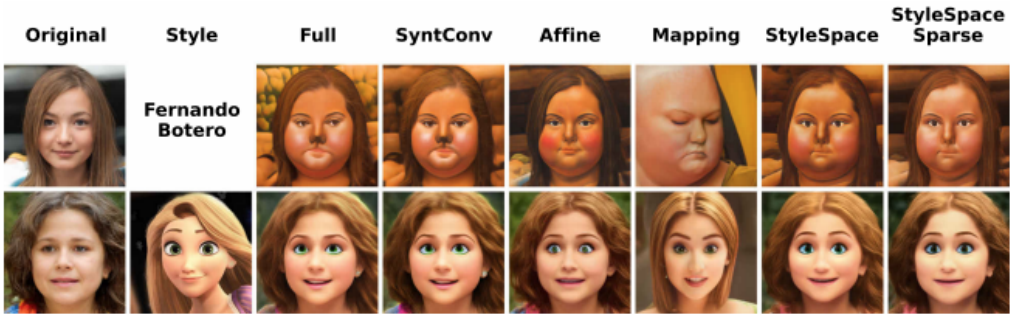


Figure 6.1: Text-based and image-based adaptation for different parameterizations for one-shot domains.

		Botero		Sketch		Disney (image)		Titan Erwin (image)	
Parameter Space	Size	Quality	Diversity	Quality	Diversity	Quality	Diversity	Quality	Diversity
Full	30.3M	0.312	0.228	0.208	0.296	0.713	0.247	0.760	0.194
SyntConv	23.6M	0.311	0.224	0.191	0.292	0.711	0.259	0.741	0.217
Affine	4.6M	0.298	0.221	0.194	0.296	0.565	0.359	0.650	0.314
Mapping	2.1M	0.226	0.115	0.182	0.143	0.717	0.080	0.645	0.102
StyleSpace	6.0K	0.309	0.23	0.193	0.306	0.627	0.308	0.672	0.296
StyleSpaceSparse	**1.2K**	0.322	0.213	0.201	0.269	0.617	0.304	0.659	0.303

Figure 6.2: Quality and Diversity metrics for text-based and one-shot image-based domain adaptations with different parameterizations.

Figure 6.3: Domain adaptation for dissimilar domains.

6.1 StyleSpace

Because of the promising results of an affine parameterization, authors check the hypothesis about optimization of a style vector

$$\mathcal{L}_{domain}\left(\{G(s(z_i)+\Delta s)\}_{i=1}^{K}\right) \to \min_{\Delta s} \tag{14}$$

Also, the StyleSpaceSparse parameterization is introduced as the authors explored that some components of Δs can be zeroed out by some threshold heuristics without a serious degradation. It turns out that StyleSpace achieves the same quality visually as a full parametrization for one-shot domains, but for few-shot domains there is also a decrease in quality as for affine layers.

6.2 Affine+ and AffineLight+

To improve the quality of an affine parameterization, additionally, the shifts $\Delta\theta_1, \Delta\theta_2 \in \mathbb{R}^{512\times512\times1\times1}$ for convolution layers weights are introduced. Also, to reduce the number of parameters even more, the low-rank decomposition is used in affine layers for a matrix, which authors call an AffineLight+ parametrization. Authors observe that Affine+ removes the performance gap with full fine-tuning and uses only 2% of parameters of the synthesis network. On the other hand, AffineLight+ still shows adequate perfomance and has 100 times less parameters than full parameterization.

		Domains	
Parameter Space	Size	Dog	Cat
Full	30.3M	20.3	7.1
SyntConv	23.6M	19.7	7.2
Affine	4.6M	70.1	27.6
Mapping	2.1M	208.2	226.1
Affine+	5.1M	**18.6**	**7.0**
AffineLight+	0.6M	20.6	8.9
StyleSpace	**6.0K**	75.8	22.0

Figure 6.4: FID scores for domain adaptation with different parameterizations.

References

[1] Aibek Alanov, Vadim Titov, Maksim Nakhodnov, and Dmitry Vetrov. "StyleDomain: Analysis of StyleSpace for Domain Adaptation of StyleGAN". In: *arXiv preprint arXiv:2212.10229* (2022).

[2] Aibek Alanov, Vadim Titov, and Dmitry P Vetrov. "Hyperdomainnet: Universal domain adaptation for generative adversarial networks". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 29414–29426.

[3] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. "Image-Based CLIP-Guided Essence Transfer". In: *arXiv preprint arXiv: 2110.12427* (2021).

[4] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. *StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators*. 2021. arXiv: 2108.00946 [cs.CV].

[5] Mingcong Liu, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, and Wen Zheng. "BlendGAN: Implicitly GAN Blending for Arbitrary Stylized Face Generation". In: *Advances in Neural Information Processing Systems*. 2021.

[6] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 2085–2094.