Development of domain adaptation methods for generative models

Kirill Korolev
Faculty of Computer Science, *Higher School of Economics*
Moscow, Russia

***Abstract*—Modern generative adversarial networks (GANs) enable the synthesis of high-quality images and provide tools for fine-grained image manipulation. However, out-of-domain generation requires an additional fine-tuning of a generator or non-flexible latent optimization, which requires training for each new image. A novel encoder, which allows a GAN generator to adapt to a new domain in a single pass, is proposed and implemented. A thorough comparison and a variety of experiments are made with prior domain adaptation methods.**

***Index Terms*—neural networks, generative adversarial networks, domain adaptation**

I. Introduction

In recent years, GANs [1] have shown an amazing ability for image generation and manipulation. In particular, StyleGAN2 [2] synthesizes high-quality images and, by its design and architecture, provides a flexible toolset for semantic and stylistic alteration of an image. Generative models for image synthesis are usually trained on large datasets, such as FFHQ or LSUN, which contain a vast variety of objects, but are limited to a particular domain. This limitation affects the user experience with these models and requires training from scratch for synthesizing out-of-domain images. Thus, domain adaptation is a technique for transferring knowledge from one domain to a pre-trained generative model in another domain. Prior works incorporate information about a new domain by either learning a domain direction in a latent space [3], [4] or by training an additional encoder [5], [6], [7] for each domain. In this work we also develop an encoder-based approach, when the generator adapts to a new domain in one encoder pass. But, in contrast to BlendGAN [5], primarily used for image stylization, and TargetCLIP [6], utilized for semantic transferring, our encoder is oriented towards general image-based domain adaptation, even for dissimilar domains. We aim to construct a compact parameterized model that competes with other encoder-based approaches, such as [5], [6], [7], in terms of semantic transfer and distribution diversity. In Section II, we discuss prior works in detail. Section III is devoted to the methodology of our approach, and finally, in Section IV, we list the expected results.

II. Literature Review

In the paper [8] authors try to incorporate pretrained CLIP [9] 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 $\mathcal{S}$.

In the first method, 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.

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

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.

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.

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

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.

In a StyleGAN-NADA paper [3] 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 loss.

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

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.

In the paper [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.

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.

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:

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

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

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.

The idea of the paper [6] 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

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

where $E(\cdot)$ is a CLIP visual-encoder. This can be equivalently expressed as each difference in a CLIP space is close to each other:

$$\mathcal{L}_{consistency} = \frac{1}{\binom{N}{2}} \left(\sum_{i_1, i_2 \in I_s} 1 - \frac{\Delta i_1 \cdot \Delta i_2}{||\Delta i_1|| \cdot ||\Delta i_2||} \right) \tag{7}$$

where $\Delta i_j = E(G(i_j + b)) - E(G(i_j))$.

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

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

Finally, the global direction is found by optimizing

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

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

In a HyperDomainNet paper [4] authors propose several contributions for domain adaption of GANs.

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

$$w'_{ijk} = d_i \cdot w_{ijk} \tag{10}$$

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

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 with the following loss.

$$\begin{aligned} \mathcal{L}_{indomain-angle} = \sum_{i,j} (\langle E(G_{frozen}(w_i)), E(G_{frozen}(w_j)) \rangle - \\ \langle E(G_{train}(w_i)), E(G_{train}(w_j)) \rangle)^2 \end{aligned} \tag{11}$$

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. Authors in [7] explore the importance of each part of the StyleGAN2, in particular, a mapping network f_M, affine layers $f_1^A, \dots, f_N^A$ 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.

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

$$\mathcal{L}_{domain}\left(\{G(s(z_i) + \Delta s)\}_{i=1}^{K}\right) \to \min_{\Delta s} \tag{12}$$

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.

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.

III. Methods

In this work, we studied six base papers [8], [3], [5], [6], [4], [7], and three of them [5], [6], [7] were chosen as a baseline for comparison.

For comparison with other methods it is essential to fix the testing protocol. For evaluation 100 face images from FFHQ dataset were sampled and 15 domains were selected (10 style images from AAHQ dataset, 'Ariel', 'Dumbledore', 'Trump' domains from TargetCLIP and 'Sketch', 'Anastasia' domains from StyleDomain). Inspired by [6] and [4], the semantic score is used to estimate how close the adapted images to the reference ones. For a given style or domain define

$$\text{Semantic-Score} = \frac{1}{|\mathcal{S}|}\sum_{I_s \in \mathcal{S}} \langle E(I_d), E(I_{s,d})\rangle \tag{13}$$

where $E(\cdot)$ is a CLIP visual-embedding and $I_{s,d}$ is an adapted image.

To evaluate the diversity of the generated images for a specific domain the following metric is calculated from [4]

$$Diversity = \frac{1}{\binom{|\mathcal{S}|}{2}}\sum_{s_1 < s_2}\left(1 - \langle E(I_{s_1,d}), E(I_{s_2,d})\rangle\right) \tag{14}$$

Eventually, both Semantic-Score and Diversity metrics are averaged across 10 styles.

IV. Results Anticipated

Method	Semantic-Score	Diversity
BlendGAN	0.663 ± 0.0685	0.187 ± 0.0222
TargetCLIP	0.702 ± 0.036	0.248 ± 0.008
StyleDomain	0.617 ± 0.04	0.267 ± 0.035

TABLE I: Evaluation results for several prior methods.

1) Analysis of foundational and key papers in the form of a literature review for the final text.
2) Report describing the selected articles for reproduction, with motivation for why they were chosen.
3) Report with a detailed description of the testing protocol and visualization methods.
4) Report describing the results of method execution, evaluating their quality based on a fixed testing protocol, visualizing these results, and comparing the methods among themselves.
5) Description of the structure and organization of the codebase for the project.
6) Report with the results of a method implemented with its own code and a comparison with the results of its original implementation.
7) Report with a more in-depth analysis of the method's performance, including various examples where the method works well/poorly.
8) Report with clear formulation of hypotheses and a list of experiments to be conducted for their verification.
9) Report with the results of conducted experiments and their analysis. Description of the final method proposed.
10) Report with the results of comparing the final method with existing approaches.

V. Conclusion

Contemporary generative adversarial networks (GANs) empower the creation of high-quality images and offer capabilities for intricate image manipulation. Domain adaptation for GANs requires an exhaustive fine-tuning of a generator or solving the latent optimization problem for each new image. To address this challenge, an encoder-based approach is proposed and implemented in order to effortlessly adapt to new domains in a one encoder pass. It significantly improves a user-experience of GANs for generating novel out-of-domain views and objects by reducing time and resources for adaptation.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

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

[3] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," 2021.

[4] A. Alanov, V. Titov, and D. P. Vetrov, "Hyperdomainnet: Universal domain adaptation for generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 414–29 426, 2022.

[5] M. Liu, Q. Li, Z. Qin, G. Zhang, P. Wan, and W. Zheng, "Blendgan: Implicitly gan blending for arbitrary stylized face generation," in *Advances in Neural Information Processing Systems*, 2021.

[6] H. Chefer, S. Benaim, R. Paiss, and L. Wolf, "Image-based clip-guided essence transfer," *arXiv preprint arXiv: 2110.12427*, 2021.

[7] A. Alanov, V. Titov, M. Nakhodnov, and D. Vetrov, "Styledomain: Analysis of stylespace for domain adaptation of stylegan," *arXiv preprint arXiv:2212.10229*, 2022.

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

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

Word count: 2090