CLIP-Guided Domain Adaptation of Image Generators with Scaled Layer Freezing

Daniel Trippa
Sapienza Università di Roma
Facoltà di ingegneria dell'informazione informatica e statistica
Roma, Italia
daniel.trippa@gmail.com

***Abstract*—Can a generative model be trained to produce images from a specific domain without seeing any images from it, guided only by text prompts? The original paper, StyleGan-Nada, leverages the semantic power of large scale Contrastive-Language-Image-Pretraining (CLIP) models and a text-driven method is used to shift the domain of a generator without having to collect even a single image. A set of experiments is conducted to show the power and the defects of this method and additional work is done in order to improve the original metodology.**

***Index Terms*—CLIP, domain adaption, layer freezing, prompt templates, image generation, StyleGan-Nada**

I. Introduction

StyleGan-Nada [1] leverages the semantic power of CLIP models [2] togheter with the generative power of the most recent Generative Adversarial Networks (GANs) [3] to create zero-shot generators [4]. Typically, a generator is restricted to thoose domains that have a large number of images avaiable. There are a lot of domains for wich little or no data is avaiable. In fact it may not be possible to create a generator for these domains following the classic GAN approach.

It Has been shown that Vision-Language models like CLIP are able to abstract generic informations and can be used for in-domain image manipulation, such as image editing. However this method allows also for out-domain changes that were not previously possible.

In this work we will experiment and conduct some testings on the original StyleGan-Nada paper, as well as proposing and testing some improvents that allow for better training speed and less artifacts under some circumstances.

Originally the paper proposes an adaptive layer freezing apprach to select the trainable layers, we will adapt this approach to make it scalable adding two new hyperparameters. Also old CLIP losses are reintroduced and other changes in respect to the original implementation are discussed.

II. Related Work

A. StyleGan

In recent years, StyleGan [5] and similars have been the state-of-the-art for unconditional image generation. Such models can generate ultra-realistic images that never appeared in the training set before. StyleGan generator is composed of two main components. A mapping network takes points from a latent space s and maps it to a latent space W. This new latent code is fed into a syntesis network that produces the desired image. In more recent works, different features of the output are mapped into specific parts of the latent code, allowing for editing and individual feature manipulation by only modifying parts of input code w. This control over the semantic proprieties of the output however are generally bounded to the training set and the specific domain of the generator itself.

Fig. 1. Some experiments with dog generator domain shifting. Ordered by rows we have the following shifts: (I) A photo of a dog. → A drawing of a dog. (II) A photo of a dog. → A photo of joker. (III) A photo of a dog. → A photo of a cute baby dog.

B. Contrastive Language-Image Pre-training

Contrastive Language-Image Pre-training (CLIP) is the first multimodal (vision and text) network, meaning it is able to semantically associate images and text. CLIP was trained on 400.000.000 image-text pairs, where a the text usually corresponds to a caption of the given image. The model was trained to connect text and images in latent space, meaning that text that describes correctly an image will be *near* the image itself. Once trained, it is able to prefict the best text that describes an image without directly training for that specific task. It can effectly produce any zero-shot image classifier.

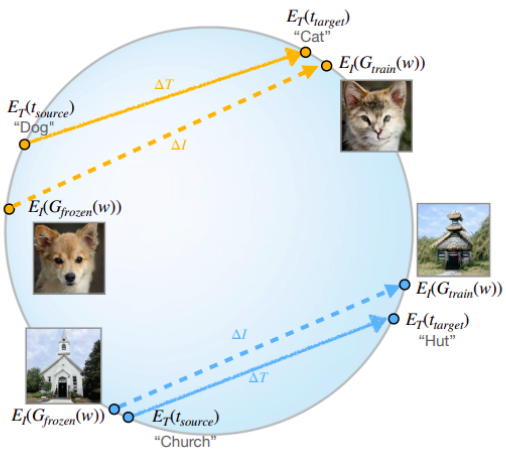

Fig. 2. Directional Loss rapresentation. Source Image,Source text and Target texts are mapped into latents space. The direction $\Delta T$of the shift is computed and is used to minimize the directional loss by aligning ΔI.

C. StyleCLIP

StyleCLIP [6] combines the generative power of StyleGan with the semantic knowledge of Contrastive Language-Image Pre-training to discover new editing directions by only using a text prompt as input. Once an editing direction is found it can be used for any image in that domain. The paper proposes three approaches for leveraging CLIP capabilities to do so. The one that is extended in StyleCLIP-Nada aims to find a global direction that allows for co-linear changes in the image and the two text prompts. This approaches are limited to the initial domain and as such can produce changes like hairstyle, expression or smaller editing.

III. StyleGan-NADA

Original work by *Patashnik et al.* has the goal of shifting from a given source domain to a given target domain using only text prompts. The only source of supervision is CLIP pretrained models. This faces two challenges: how to extract the semantic knowledge that are intrinsic in CLIP and how to prevent adversarial or collapsed solutions.

To use a pretrained clip model as supervisor, some kind of loss has to be computed. In this paper global loss and directional loss are used. StyleCLIP global loss can be computed as follows:

$$L_{Global} = D_{CLIP}(G(w), t_{target})$$

Where D_{CLIP} is the cosine distance, $G(w)$ is the generated image starting from latent code w and t_{target} is the text describing the target domain. Although this method is intuitive, it has been shown that it leads to adversarial solution for out-of-domain shifting. The above loss is still used during the layer freezing approach to compute the best layers to train.

To solve this problem, a new loss is introduced. The idea is to fix a direction in CLIP space given by the source and target text prompt. Then the new produced generator and the old one are aligned on the same direction. To do so, a pair of generators is used, one is fixed and is not trained and the other will be fine-tuned. At every iteration, the directional CLIP loss is computed using the image produced by the new generator and the old one. The goal is to align this shift over all possible latent codes. The two generators are called G_{train} and G_{frozen} respectively. Matematically, Directional loss for a given latent code w can be computed as follows:

$$\Delta T = E_T(t_{target}) - E_T(t_{source})$$

$$\Delta I = E_I(G_{train}(w)) - E_I(G_{frozen}(w))$$

$$L_{direction} = 1 - \frac{\Delta T \cdot \Delta I}{|\Delta T||\Delta I|}$$

where E_T and E_I are the CLIP encoders for Text and Images respectively. t_{source} and t_{target} are the source and target text prompt along we wish to shift the domain.

The above loss prevents collapsed solution, as the new generator needs to align for every latent code, and this can't be possible if it only produces one image. It also prevents adversarial solution, as to do so it would have to fool the network across an infinite set of instances.

IV. Layer Freezing

For simple domains shifts, such as texture based (e.g. $photo \rightarrow sketch$), the previously described method converges very quickly but for out of domain shifts that require a longer training time the network risks to collapse or overtrain. It has been shown in previous works that restricting the subset of trained weights can improve the quality. StyleGan-NADA proposes a method to choose the trainable weights called adaptive layer freezing. The inutuiton is to pick the layers that are most relevant to the change by testing for the w code that affects the most the output.

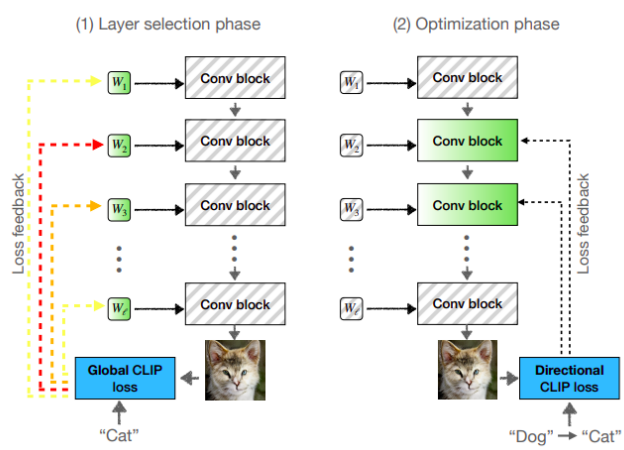

Fig. 3. Layer freezing rapresentation. First the most impactfull latent codes w are found by optimizing using Global Loss. Then only layers corrisponding to that latent codes are used for directional training

To select k layers, first we sample N_w latent codes $\in W$ and perform N_i steps of StyleCLIP $latentcode$ optimization method. The layers for wich the latent code changed the most are selected and thus all the others are $freezed$ and will not be trained in the next iteration. Note that this operation is conducted on the generator that is trained, not the freezed one.

V. Additional Work

A. Scaling the layer freezing approach

1) Less training: Our implementation scales the original layer freezing approach by adding two hyperparameters. During our experiments we noted that enabling layer freezing during training had a very bad impact on training speed performance (almost an order of magnitude of speed is lost). To solve this problem but still keep the advantages of the original approach, we propose to perform StyleCLIP latent code optimization only one every N_{int} iteration steps. The intuitive idea is that the layers that will be more impactful on a certain iteration will probably stay so for the next few iterations and reperforming the optimization step would be just a waste of resources. Based on the N_{int} used, training speed can be greatly improved (up to x5 speedup) without loosing image quality.

2) Better fine tuning: For better fine tuning, we also propose to decrease the number of trained layers k over time. This introduces a new parameter $N_{falloff}$, that controls the rate at wich k is decreasing. This approach is useful for very long training sessions that need a better fine tuning. Training less layers over time will ensure that the good traits of the shift are not lost over time and prevent an *overtraining* effect that sometimes occurred during the experiments where training too much produces unrealistic images and drifts away from the target domain.

Fig. 4. Above: Original, Below: Ours (Without StyleCLIP templates)

B. Reintroducing global loss during training

Originally, global loss is completely discarded because of the adversarial solution that it can cause. Despite of this, global loss can help to get closer to the target domain once the directional approach has done its work. We tested this hypothesis by reintroducing Global loss and computing the total loss as a weighted average of the two losses: global and directional. The loss weights are controlled by a parameter δ and can be computed as follows:

$$L_{total} = \delta * L_{Direction} + (1 - \delta) * L_{Global}$$

Generally the best results are obtained by keeping the $L_{Direction}$ higher by using a δ value around 0.8. See table I for values used during experiments and results.

C. Removing StyleCLIP templates

StyleCLIP uses a list of prompts generated starting of a template (e.g *a photo of a* {}, *a low resolution image of a* {}) and the average of the resulting encodings is used. This is motivated through the improved classification accuracy that they are getting. StyleGan-NADA doesn't describe this approach but the same templates are kept in the original implementation [7]. Without the need for a classification task, multiple prompts could actually mislead the training of the generator. We tested the model without the templates, using only the given source and target prompt and generally got the same results, although in some cases we noticed less artifacts.

TABLE I
Table shows parameters used for the experiments with the improved model

Experiment	Iterations	Adaptive k	N_{iter}	$N_{falloff}$
a photo of a cute baby dog	200	12	10	50
a photo of joker	200	10	5	50
a sketch of a dog	200	10	10	50
a church painted by van gogh	300	8	5	500
a cubism painting of a church	400	8	10	500
a sketch of a person	200	18	x	x

Fig. 5. Above: Original (100% Directional Loss), Below: Ours (80% Directional Loss, 20% Global loss). Notice More colorful images and less artifacts when global loss is reintroduced.

VI. Experiments and results

We conducted multiple experiments to compare the original results with the proposed improvements. All test were conducted on google colab with an allocated *Tesla K80* GPU. The code for our implementation is online and avaiable at github [8]. Also a notebook is available for experimeting with the new parameters [9].

Table I showcases some of the used parameters. Generally the starting numer of trained layers k corresponds to $\frac{2}{3}$ of the total layer for a given dataset. Different values for N_{iter} and $N_{falloff}$ are tested. An iteration that performs latent code optimization lasts around 7 times more that a normal iteration. Thus using a value N_{iter} of 10 almost removes this speed bottleneck as the optimization is performed only once every 10 steps. In our experiments training time was almost five times less then the original results.

For domain shifting with ffhq dataset, a parameter δ of 0.8 is used and layer freezing was disabled (all layers are trained). Results can be observed in 5. Generally less artifacts can be noticed. Also colors of the original generator are conserved.

In figures 4 we compare the original results of domain shifted dog generators and the one without StyleCLIP templates. Also in figure 6 we compare the domain shifting of churches, where a speedup of $x3$ and $x5$ where obtained respectively.

VII. Conclusions

We reimplemented StyleGan-NADA and conducted additional experiments. We also proposed some improvements that lead to a speedup in training time and give additional hyperparameters that can control the final output. A lot of creative works can benefit from the ability to generate arbitrary images starting only from text prompt. One such application can be image editing starting from a generator.

This method still has some limitations, for example it's limited to domain shifts that were seen by the CLIP model during training and by the ambiguity of natural language.

We hope that additional work can be done in the future to improve textually guided generation such as the one described here.

Fig. 6. Above: Original , Below: Ours. The domain shifts are respectively (I) a photo of a church. $\rightarrow$ a church painted by van gogh. (II) a photo of a church $\rightarrow$ a cubism painting of a church. A speedup of x3 and x5 where obtained respectively

References

[1] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[4] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly, 2017.

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2019.

[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)*, pages 2085–2094, October 2021.

[7] https://github.com/rinongal/stylegan-nada.

[8] https://github.com/dnl-trpp/stylegan-nada.

[9] https://colab.research.google.com/drive/1pexflahu89q9hm1_cf96jxcdslcn30w3.