LAMDA: LATENT MAPPING FOR DOMAIN ADAPTATION OF IMAGE GENERATORS

Anonymous authors
Paper under double-blind review

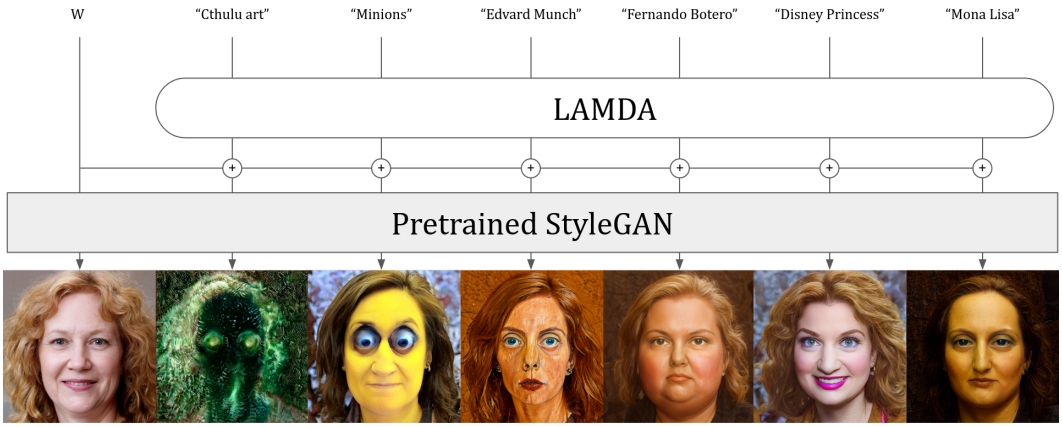

Figure 1: Our method trains a unified mapper on many keywords simultaneously to learn the relationship between CLIP text embeddings and latent embeddings of GANs.

ABSTRACT

Our paper tackles the problem of adapting image generators to new keyword-defined domains without training on any new images. We combine the power of CLIP models for image-text similarity with the disentangled representation of images found in the latent spaces of generative adversarial networks (GANs). We present the latent mapper (LAMDA) which maps directions in CLIP text space to directions in the GAN latent space. Using a latent mapper enables training on a large number of keywords simultaneously which was not previously possible, and allows benefiting from the interrelations between different keywords. It also leads to higher image quality while requiring only a fraction of the training time and parameters of state-of-the-art methods. As a result of learning those relationships, LAMDA produces excellent results when composing multiple words and generates semantically meaningful images. We demonstrate the generalizability of LAMDA by showcasing results on unseen words at test-time, as well as results on different kinds of style-based GANs.

1 INTRODUCTION

The success of unconditional GANs (Goodfellow et al., 2014; Radford et al., 2015; Brock et al., 2018; Karras et al., 2017; 2019) in generating realistic images inspired many researchers to investigate the latent space. Investigating how GANs map latent codes to images leads to methods that can control the generative process. This allowed generating images with a specific pose or facial expression (Patashnik et al., 2021; Gal et al., 2021; Wu et al., 2021; Singh et al., 2019; Alharbi & Wonka, 2020). On the other hand, investigating how to map a real image into the latent space allows finding the latent code that can generate an image that is as similar as possible to a real image. This is often referred to as GAN inversion in literature. GAN inversion allows leveraging the disentangled latent space for editing real images (Abdal et al., 2019; Richardson et al., 2021; Tov et al., 2021; Abdal et al., 2020).

The latent space offers many desirable properties in comparison with pixel space. For example, in the case of GANs trained on face images, editing the pose or facial expression is much easier in latent space than in pixel space. Although the latent space is still hard to explore because it requires manual examination, the emergence of language-image models alleviates this issue. The ability to compute similarity between a word and an image without human intervention is vital to domain adaption methods.

Domain adaptation of image generators enables performing edits on real images that are difficult and time-consuming to do manually, such as sketching a face or replicating a certain artistic style. However, the performance of previous methods is severely limited. For example, the most successful method, StyleGAN-NADA (Gal et al., 2021), shows results of high quality, but it needs to finetune a separate network per keyword. This is undesirable in multiple ways. First, it requires finetuning and storing a network per keyword, which is costly and not scalable. Second, it ignores the relationships between keywords and foregoes the opportunity to learn more information from these relationships. Third, it does not accommodate the composition of keywords.

We propose a novel approach for domain adaption of image generators tackling these opportunities for improvement. Our approach leverages both the inter-relationships between different keywords and the relationships between text embeddings and latent codes to train a latent mapper. Compared with the state-of-the-art, the benefits of our approach are:

- Scalable domain adaptation of GANs. We demonstrate results for training with 100 keywords simultaneously.
- Better quality of adapted images.
- Better composition of keywords.
- An order of magnitude fewer parameters and faster training time.
- Generalization to some unseen keywords, 3DGANs, and transformer style-based GANs.

2 Related work

2.1 Style transfer

In the case of style transfer, the goal is produce an image that reflects the content of one image but the style of another image. The input is usually a pair of images: one defining the style and the other defining the content. However, in our case, we aim to adapt a generator to exhibit certain keywords. It is not trivial how to extend style transfer to generators and keywords.

Pixel space is not suitable for this task as it does not disentangle between lower and higher level concepts. This sparks interest in disentangled deep representations of images. One of the most successful approaches (Gatys et al., 2015) proposes using deep features of pretrained networks as content and style descriptors. Specifically, the authors show that raw values of deeper features are good descriptors of content as they preserve high-level details while being less sensitive to low-level details. To describe style, the authors propose using feature map statistics in different layers of the network. This idea is refined further in Adaptive Instance Normalization (AdaIN) (Huang & Belongie, 2017), where the authors show that we could specify the style of an image by manipulating the mean and variance of deep features. This viewpoint influenced one of the most successful variant of GANs: Style-based GANs.

2.2 Unconditional GANs

GANs have shown considerable progress in generating realistic images (Goodfellow et al., 2014; Radford et al., 2015; Brock et al., 2018; Karras et al., 2017). In the unconditional case, the GAN is simply given a set of images with the goal of generating similar images without the ability to specify which class of images to generate.

The most successful variant of GANs for unconditional face image generation in terms of disentanglement and image quality is the style-based generator(Karras et al., 2019). It inspires many works that obtain the best results on the FFHQ dataset (Wang et al., 2022; Humayun et al., 2022; Sauer et al., 2022; Karnewar & Wang, 2020; Zhao et al., 2020). The success is due mainly to the

special attention dedicated to the latent space. Prior to StyleGAN, the common approach used a single linear layer to map a randomly-sampled code to a tensor. Then, the tensor is continuously upsampled and convolved to obtain the final image. StyleGAN, on the other hand, proposes using 8 fully-connected layers to map the random code to another space called W space. Rather than upsampling and convolving w codes to generate the images, they are used to modulate each feature map to generate the final image. The W space shows many desirable properties. First, it has better disentanglement scores than the randomly-sampled codes. Second, they can be used to encode real images into GANs with high fidelty (Abdal et al., 2019; Wu et al., 2021; Tov et al., 2021; Richardson et al., 2021).

The utilization of GANs to edit real images is driven by the success of two research areas: disentangled generation and GAN inversion. Research in disentangled generation provides GANs that allow control over most aspects of generation. For example, FineGAN (Singh et al., 2019) and structured noise injection (Alharbi & Wonka, 2020) design the GAN such that background and style information can be changed independently from local detail. Additionally, researchers found that it is possible to find disentangled editing directions within existing standard GANs(Abdal et al., 2019; Wu et al., 2021; Tov et al., 2021; Richardson et al., 2021). Research in GAN inversion enables mapping real images to latent spaces of GANs. In the case of style-based generators, recent papers show that we can encode real images with high fidelity into the W^+ and S spaces.

In summary, the W latent space of StyleGAN exhibits disentanglement between high-level and low-level details of the images. In addition, it seems capable of representing any real image in the same domain as the training data with high fidelity.

2.3 TEXT-TO-IMAGE GENERATION

Many recent advances in text-to-image generation are driven by the success of language-image training, specifically the CLIP model(Radford et al., 2021). CLIP learns a joint representation space between sentences and images allowing the ability to measure similarity between a sentence and an image. Dall-E 2 (Ramesh et al., 2022) and Stable Diffusion (Rombach et al., 2022) two of the most successful text-to-image models make use of the CLIP model as a core part of their methods.

Text-to-image generation often requires large scale training with billions of images and parameters. In contrast, text-driven domain adaptation does not require new training images for the adapted domains. Instead, it leverages the existing power of pretrained generators in addition to CLIP to generate images in new domains.

2.4 TEXT-DRIVEN DOMAIN ADAPTATION OF GANS

Our work falls in the category of text-driven domain adaptation of generated images. Given a trained generator that can generate realistic images, the goal is to be able to adapt it to generate image of other domains. For example, given a generator that can generate realistic face images, we aim to adapt it to generate sketches or paintings of those faces in specific artistic styles. Similarly to the text-to-image generation case, it is essential to be able to compare an image to a keyword. This is only possible due to the success of language-image models such as CLIP.

StyleCLIP (Patashnik et al., 2021) is one of the earliest works in literature that leverages text-image similarity for editing generated images. The authors explore the task from three different angles: optimizing a specific latent code to exhibit a certain keyword, training a mapper network to transform any latent code into a certain keyword, and optimizing global directions that exhibit certain keywords independently of input latent codes. The main drawback to StyleCLIP is the inability to handle out-of-domain keywords. While it can generate faces in a certain hairstyle, it cannot be used to generated sketches. StyleGAN-NADA (Gal et al., 2021) overcomes this issue by finetuning the GAN to allow generating out-of-domain concepts. In order to adapt StyleGAN to generate images exhibiting a certain keyword, StyleGAN-NADA finetunes it using a directional CLIP loss so that the generator would only generate images exhibiting that keyword. While the training procedure is fast, there are several main limitations to StyleGAN-NADA. First, StyleGAN-NADA requires a separate training procedure and a separate generator per keyword. This does not scale well if the generator needs to be adapted to many keywords. It is also oblivious to the natural interrelations between different keywords. Second, it produces lower quality images on certain keywords. Third, it lacks the ability

to compose different words. Since each keyword requires a separate network, producing an image that reflects multiple words is difficult.

A key difference between our method and previous methods is that we propose adapting generators to many keywords simultaneously. This enables simple composition of words and allows the network to benefit from the common information between different keywords. A single mapper network is trained to map directions in CLIP text space into directions in the latent space of GANs. As a result, we accommodate test-time translation of 100 keywords to latent space. To the best of our knowledge, all previous methods require a separate training procedure for each keyword as they use a single text embedding to guide the training. In contrast, the mapper in our networks sees many text embeddings which leads to meaningful composition of keywords at test time and even extends to some unseen keywords.

3 METHOD

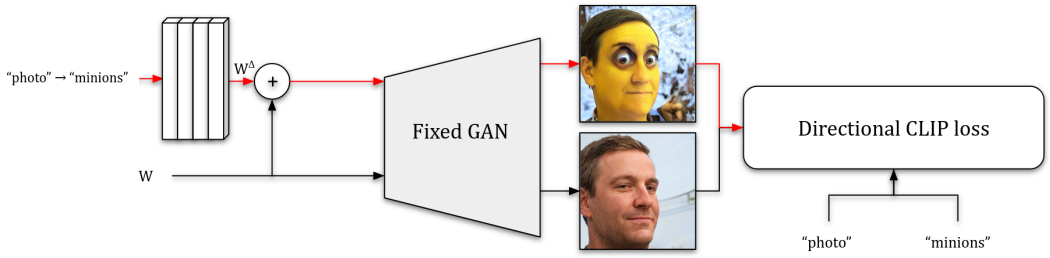

Figure 2: An illustration of our method. The directional CLIP loss is used to guide a network that maps CLIP text directions into GAN W directions.

3.1 INTUITION

There are two reasons that guide our design choice.

First, we hypothesize that we can adapt style-based image generators without finetuning by mapping into the latent space. The latent spaces of style-based GANs are heavily-explored in literature. They are understood to contain a high-level representation of the image. They are also disentangled to some degree, such that certain concepts can be mapped to specific layers (Wu et al., 2021). Furthermore, there is evidence that even when trained on realistic face images, the latent spaces are capable of encoding artistic style and even non-face images (Abdal et al., 2019).

Second, we hypothesize that modeling the relationship between CLIP text embeddings will improve training speed and image quality. Previous methods adapt generators to each keyword individually, which does not benefit from the similarity between many words. For example, when adapting the generator to produce certain facial expressions, it should be expected that many expressions will modify similar layers in W space.

Consequently, we propose LAMDA to simultaneously learn how to translate many keywords into direction in the latent space of a pretrained generator without finetuning.

3.2 LATENT MAPPING

The idea behind our work is to map directions in CLIP text space into directions in the GANs W^+ space. Given a source keyword and a target keyword, the mapper will translate the direction between them into a direction in W^+ such moving in that direction in W^+ space will introduce the target keyword. The learned directions are global and can be applied to any w^+ code.

The CLIP text embeddings are computed from the source and target words. They are used to obtain the normalized keyword direction.

$$CLIP_{text\_dir} = \frac{CLIP_{target} - CLIP_{source}}{|CLIP_{target} - CLIP_{source}|} \tag{1}$$

The latent mapper takes the word directions described above as inputs and translates them into W^+ directions.

$$LAMDA : CLIP_{text\_dir} \rightarrow W^+_{latent\_dir} \tag{2}$$

The new w^+ code that reflects the target word can be computed by adding the learned w^+ direction to any w^+ code.

$$w^+_{source \rightarrow target} = w^+ + LAMDA(CLIP_{text\_dir}) \tag{3}$$

Where:

$$\begin{gathered} CLIP_{text\_dir} \in R^{512 \times 1} \\ w^+_{latent\_dir} \in R^{512 \times 18} \end{gathered} \tag{4}$$

3.3 IDENTITY PRESERVATION

One issue with our current proposal is that there is no incentive for the network to maintain the identity of the original generated face. We propose two loss terms for identity preservation: the dictionary description loss, and the direction magnitude loss.

We compute a description of each generated face before and after translation. The description is the CLIP similarity loss between the generated image and each of the 100 training keywords (except for the two keywords used for this training step). At each training iteration, we incentivize the mapping network to maintain the identity by penalizing it based on the difference between the keyword similarities before and after translation.

$$\begin{aligned} \mathcal{L}_{ID} = \frac{1}{n} \sum_{\substack{i \\ i \neq s \\ i \neq t}}^{n} || & Global\_CLIP\_Loss(CLIP_{w^+}, CLIP_{keywords[i]}) - \\ & Global\_CLIP\_Loss(CLIP_{w^+_{s \rightarrow t}}, CLIP_{keywords[i]}) ||_2^2 \end{aligned} \tag{5}$$

Where:

$$w^+_{s \rightarrow t} = w^+ + LAMDA(CLIP_{keywords[s] \rightarrow keywords[t]}) \tag{6}$$

This can be achieved with minimal cost by leveraging the precomputed CLIP text embeddings of our 100 training keywords. The only additional cost is the computation of dot products which is negligible. The global CLIP loss is a dot product between the image and text embeddings.

Additionally, we penalize the learned directions based on their magnitudes. We find that the weight assigned to this loss term offers a trade-off between keyword resemblance and identity preservation.

$$\mathcal{L}_{magnitude} = ||LAMDA(CLIP_{text\_dir})||_2^2 \tag{7}$$

3.4 DIRECTIONAL MARGIN

One of the main benefits of learning latent directions is enabling composition of keywords. While previous domain adaptation methods can interpolate between keywords, they struggle to perform composition. Consider the task of adapting a generator to produce images that contain several keywords at once, for example ("sad", "bangs", "sketch", and "goatee"). It is not immediately clear how to obtain this with previous methods as simply training StyleGAN-NADA on longer sentences does not produce the required results.

In our method, the composition is as simple as adding more directions to the w code. The quality of composition can be used to assess entanglement. If composing two words introduces new features or leads to artifacts, then this is an indicator that the network might be overfitting to the training keywords.

	# of parameters	Training time
Ours	5M	1.5 min / word
StyleGAN-NADA	300M	5min / word

Table 1: A comparison of training time and number of parameters between our method and StyleGAN-NADA. Our method requires a fraction of the parameters and can be trained much faster.

	Quality ↑	Diversity ↑
Ours	**59%**	0.63
StyleGAN-NADA	41%	0.69

Table 2: A comparison of quality and diversity between our method and StyleGAN-NADA. Our method is consistenly preferred over StyleGAN-NADA while still being diverse.

We propose to reduce overfitting and improve composition quality by adding a margin loss. During training, each learned direction d is perturbed by ϵ.

$$w^{+}_{s \to t} = w^{+} + (LAMDA(keywords[s], keywords[t]) + \mathcal{N}(0, \epsilon)) \tag{8}$$

This serves two benefits: it increases disentanglement between learned directions, and it allows for a small amount of variation for each keyword.

3.5 Training loss

We use the directional CLIP loss from (Gal et al., 2021) to penalize the directions produced by our latent mapper. During each step of training, we randomly generate a w code and randomly pick two words out of the keyword list. We use the latent mapper to map the text direction between the two words into a latent direction in W space. The directional loss can be computed as:

$$\begin{aligned} CLIP_{text\_dir} &= \frac{CLIP_{target} - CLIP_{source}}{|CLIP_{target} - CLIP_{source}|}\,, \\ CLIP_{image\_dir} &= \frac{CLIP_{G(w+LAMDA(CLIP_{text\_dir}))} - CLIP_{G(w)}}{|CLIP_{G(w+LAMDA(CLIP_{text\_dir}))} - CLIP_{G(w)}|}\,, \\ \mathcal{L}_{direction} &= 1 - (CLIP_{text\_dir} \cdot CLIP_{image\_dir})\,, \end{aligned} \tag{9}$$

So our final loss function is:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{direction} + \beta \mathcal{L}_{ID} + \gamma \mathcal{L}_{magnitude} \tag{10}$$

In training, the hyper parameters α, β, γ, and ϵ control the relative importance of fidelity to keywords, fidelity to input identity, and composition quality.

4 Experiments

4.1 Quantitative analysis

We perform a user study to assess the quality of our model in comparison with the state-of-the-art model StyleGAN-NADA. We train 100 StyleGAN-NADA models, one for each keyword to compare against our latent mapper model. We present users with the keyword, an image generated by the StyleGAN-NADA model trained on that keyword, and an image generated by our model when given that keyword. Since both our model and StyleGAN-NADA preserve the original W space from StyleGAN, we input the same w code to both our model and the StyleGAN-NADA model.

To quantify the diversity of the adapted generators, we use LPIPS (Zhang et al., 2018) computed on 1000 pairs of images. The identity of the faces (w code) in each pair is kept the same between

StyleGAN-NADA and our method, such that the diversity in style would be the main source of variation.

Figure 3: Top row: StyleGAN-NADA. Bottom row: LAMDA. Even though StyleGAN-NADA finetunes a separate network per keyword, LAMDA produces higher quality results on difficult keywords.

4.2 Qualitative analysis

We show a visual comparison between our work and StyleGAN-NADA in Figure3 for several difficult words. We observe that for difficult keywords there is a clear advantage for our method. Even though StyleGAN-NADA finetunes the network parameters, it still fails to produce large appearance changes. On the other hand, our method finds global direction without finetuning.

Figure 4: The quality of our method when composing multiple words at test-time. This is done by adding the learned directions corresponding to each word together.

4.2.1 Composition quality

It is incredibly valuable to be able to generate images that reflect the composition of more than just word. This is highly desirable as it allows for a very large degree of freedom in the generated images after training. In addition, it also serves to asses the disentanglement of the global directions. If adding multiple learned global directions leads to artifacts or loss of quality, it would indicate entanglement and overfitting.

As shown in Figure 4, our method produces aesthetically-pleasing images that reflect the multiple keywords. This is a major advantage over StyleGAN-NADA as it does not naturally support composition.

4.3 Compatibility with different GANs

Our method is not restricted to StyleGAN and it can be used easily with other style-based GANs. We showcase our results when mapping text embeddings to latent spaces of EG3D(Chan et al., 2022) and StyleSwin(Zhang et al., 2022). Both architectures are complex and finetuning for each keyword is not scalable. For example, EG3D contains a rendering component and StyleSwin is a transformer-based GANs that has almost 10 times as many parameters as StyleGAN2.

By mapping into the latent spaces without finetuning we enable adaptation of those image generators to a 100 keywords. In Figure5 we show the results of using our method to find latent directions

Figure 5: Our method can be used to find meaningful directions in EG3D latent space. Top row: ”surprise”. Middle row: ”werewolf.” Bottom row: ”white walker.” First column: adapted images. Second column: depth images. Third column: 3D rendering.

Figure 6: Our method can be used to find meaningful directions in StyleSwin latent space. From left to right: original image, ”photo” →”angry”, ”photo” →”Raphael painting”, ”photo” →”zombie”

for EG3D. We find that in addition to the image quality, our method still preserves the desirable qualities of EG3D in terms of the quality of the rendering and the depth image. When mapping to the StyleSwin latent space we also obtain similar quality as show in Figure6. We also showcase our results on cat images in Figure 7.

4.4 Generalization to unseen words

After training on the 100 keywords we selected, we explore whether the joint training is beneficial at all in terms of knowing the relationship between words. One extreme is if the network overfits and learns to map each keyword in the training individually. The other extreme is if the network, giving only a small set of keywords, can generalize to all other keywords.

We find that the network does extend to some unseen keywords demonstrating that the network is learning at least some of the hidden relationships between words. In Figure8, we show that the latent mapper can still produce meaningful directions for unseen words of different kinds. We believe this proves that the network is not overfitting, and is instead learning relationships that can extend to words outside of the training dataset.

4.5 Discussion

Our experiments demonstrate the strengths of our method. We find that images adapted using out method have higher quality that the state-of-the-art while training faster with fewer parameters. This is because the latent mapper is able to assimilate information when learning from multiple keywords during training. Most keywords share some similarity with other keywords. Previous methods train from scratch for each keyword, while our method can use what it learns for one keyword to adapt to other keywords.

In addition to the benefits mentioned above, our experiments show that the network is not simply memorizing training keywords. We show that the network can compose the learned directions at test-time to combine keyword effects. We also show that the mapper is capable of extending to new unseen words at test-time.

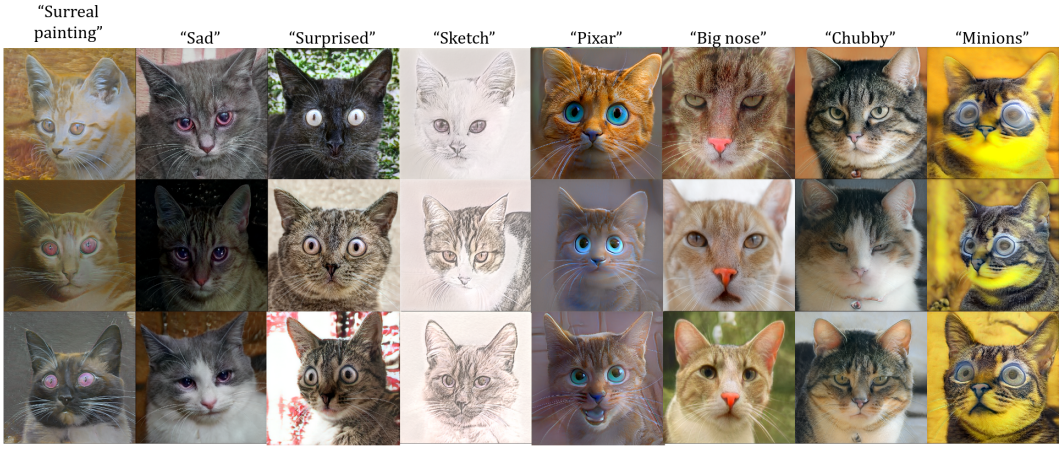

Figure 7: Our method can be used to find meaningful directions in StyleGAN-ADA latent space (trained to produce cat images). We use the same 100 keywords used for human faces.

Figure 8: After training, the latent mapper produces acceptable results for several unseen words. This shows that the network is not overfitting, and is learning the meaningful relationships between words.

Finally, our experiments show that our method is not restricted to StyleGAN and can be applied to other style-based generators. Specifically, we show that LAMDA can map into EG3D while still producing consistent depth and geometry. It can also be trained to map into transformer-based GANs such as StyleSwin.

5 Limitations and Future Work

The main limitation is the sensitivity of training to the hyperparameters. Very high quality images can be obtained but often at the cost of identity loss and worse composition performance.

Another limitation we share with previous method is the reliance on the source keyword. We observe that while our method is less sensitive, the results can still vary based on the choice of the input word.

In terms of future work, we observe that with just 100 words the mapper can extend to many other words. One interesting endeavor for future work is to investigate the possibility of training a mapper with a large enough number of keywords such that it can extend to any natural word.

Furthermore, it would be useful to analyze words and their relationships based on the mapping. For example, one can aim to produce an attention map per keyword to visualize its influence.

6 Reproducibility Statement

All codes and models used in this paper will be made publicly available. Additionally, results can be reproduced without too much difficulty by training the fully-connected network LAMDA. Many choices of parameters produce acceptable results. We report our specific training schedule and hyperparameter selection in the appendix.

REFERENCES

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*, pp. 4432–4441, 2019.

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8296–8305, 2020.

Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5134–5142, 2020.

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

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16123–16133, 2022.

Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Polarity sampling: Quality and diversity control of pre-trained generative networks via singular values. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10641–10650, 2022.

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

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

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

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*, pp. 2085–2094, 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

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. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

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

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2287–2296, 2021.

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

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022.

Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6490–6499, 2019.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12863–12872, 2021.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11304–11314, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, and Changyou Chen. Feature quantization improves gan training. *arXiv preprint arXiv:2004.02088*, 2020.

A Appendix

A.1 Training details and setup

We manually formed the 100 keywords list. They cover 4 broad categories: facial editing, facial expressions, artistic styles, and transformations. Facial editing keywords include keywords such as "bald" and "raising eyebrows." Facial expression keywords include "happy" and "disgusted." Artistic styles contains famous artists. Finally, transformations contains words that represent certain characters such as "minions" or "disney princess."

We use a fully-connected network to map input text directions into latent directions. For our network that was used to produce figures and results in this paper, we used the hyperparameters: $\alpha = 1$ $\beta = 0.1$ $\gamma = 1$ $\epsilon = 0.5$

We used $1e^{-3}$ as the learning rate and trained for a total of 30 thousands steps. The learning rate was multiplied by 0.8 every 2 thousand iterations.