MIND THE GAP: DOMAIN GAP CONTROL FOR SINGLE SHOT DOMAIN ADAPTATION FOR GENERATIVE ADVERSARIAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a new method for one shot domain adaptation. The input to our method is a trained GAN that can produce images in domain A and a single reference image I_B from domain B. The proposed algorithm can translate any output of the trained GAN from domain A to domain B. There are two main advantages of our method compared to the current state of the art: First, our solution achieves higher visual quality, e.g. by noticeably reducing overfitting. Second, our solution allows for more degrees of freedom to control the domain gap, i.e. what aspects of the image I_B are used to define the domain B. Technically, we realize the new method by building on a pre-trained StyleGAN generator as GAN and a pre-trained CLIP model for representing the domain gap. We propose several new regularizers for controlling the domain gap to optimize the weights of the pre-trained StyleGAN generator to output images in domain B instead of domain A. The regularizers prevent the optimization from taking on too many attributes of the single reference image. Our results show significant visual improvements over the state of the art as well as multiple applications that highlight improved control.

1 INTRODUCTION

We propose a new method for domain adaptation based on a single target image. Given a trained GAN for domain A, and a single image I_B from domain B, our approach learns to find a corresponding image in domain B for any image in domain A. We can achieve this by fine-tuning the GAN for domain A to obtain a second GAN that generates images in domain B. The two GANs share a latent space that a single latent code will generate two corresponding images, one in domain A and one in domain B. The main selling point of our method is that is achieves far superior quality than the state of the art in single shot domain adaption. In fact, the results are also better than most published few shot domain adaption methods. Our method is computationally lightweight and only takes a few minutes on a single GPU, so that it can be widely applied.

In order to do this, we leverage multiple existing components, including two excellent pre-trained networks: First, we use StyleGAN2 (Karras et al. (2020b)) as pre-trained GAN. A follow up version has been published on arXiv (Karras et al. (2021)), but the code is not yet available. Second, we use a pre-trained network for image embedding, CLIP (Radford et al. (2021)), to encode images as vectors. Third, we use the pioneering idea of and StyleGAN-NADA (Gal et al. (2021)) (building on StyleCLIP (Patashnik et al. (2021))) to use CLIP to encode a domain gap (or domain shift) as vector in CLIP embedding space.

Even though the visual quality of StyleGAN-NADA is already impressive when used as a single image domain adaption method, we identified multiple technical issues that can be improved to achieve another large jump in visual quality. First, and most importantly, StyleGAN-NADA does not have a good solution to model the domain gap. They model the domain gap as vector from the average image in domain A to the given image I_B in CLIP embedding space. However, this leads to overfitting in practice and the transfer results lose attributes of the input images, so that input images from domain A get mapped to images that are too similar to I_B in domain B. We identify a better solution to this problem. In fact, the domain gap should be modeled as a vector from the image I_B to its projection into domain A. We therefore need to solve an inverse domain transfer problem which

we propose to tackle using the state of the art GAN embedding method II2S (Zhu et al. (2020b)). A key insight is that we can use a heavily regularized version of II2S GAN inversion to do the reverse problem of transferring any related image (from a domain B) into the domain A, helping to characterize the semantic domain gap better than previous work. Further extensions enable us to fine tune the modeling of the domain gap to explicitly model which attributes of the input image should be kept. Second, we propose multiple new regularizers to improve the quality. Third, we propose a technical improvement to the heuristic layer selection proposed in StyleGAN-NADA that is more elegant and robust.

In summary, we make the following contributions:

1. We greatly reduce the mode collapse/overfitting problem which often occurs in one-shot and few-shot domain adaptation.
2. Our results look more similar to the target domain images with fewer artifacts. These results are also faithful to the identities of the input images and able to capture fine details.
3. Our attribute-aware domain adaptation provides more freedom to control the "similarity", which makes a large number of downstream applications possible, e.g., pose adaptation, lighting adaptation, expression adaptation, texture adaptation, interpolation, and layer mixing, using state of the art image editing frameworks.
4. Our framework makes it possible for users to edit input images in different domains such as modern art like anime characters or classical art like the Mona Lisa.

2 Related Work

Domain adaptation. Domain adaptation is the task of adapting the model to different domains. Different works in this domain (Bousmalis et al. (2016; 2017); Na et al. (2020); Wang & Breckon (2020); Kang et al. (2019)) try to learn diverse domain independent representations using the source domain to make predictions in the target domains. More importantly, generating diverse representations of images by combining natural language supervision has been of interest to the computer vision and NLP research communities. Recently, OpenAI's Contrastive Language-Image Pretraining (CLIP) (Radford et al. (2021)) work established that transformer nets, and large datasets, could generate very transferable visual models. In CLIP, both images and text are represented by high dimensional semantic-embedding vectors which can then be used for zero-shot learning.

GAN-based domain adaptation. In the GAN domain, various models and training strategies have been proposed for the zero-shot/ few-shot domain adaptation tasks (Bousmalis et al. (2017); ZHANG et al. (2018); Li et al. (2020); Liu et al. (2019)) by leveraging from the adversarial training. More specifically, relevant to our work, the domain adaptation methods (Patashnik et al. (2021); Gal et al. (2021); Jang et al. (2021); Song et al. (2021)) that build upon the StyleGAN (Karras et al. (2019; 2020b;a)) demonstrate impressive visual quality and semantic interpretability in the target domain. These methods can be broadly classified into few-shot and single-shot domain adaptation methods.

A notable few-shot method, StyleGAN-ADA (Karras et al. (2020a)) proposes an adaptive discriminator augmentation method to train the StyleGAN on limited data. Another work, DiffAug (Zhao et al. (2020)), applies differentiable transformations to the real and generated images for robust training. Discriminator related approach, FreezeD (Mo et al. (2020)) freezes lower layers of the discriminator to achieve domain adaptation. Toonify (justinpinkney/toonify) interpolates between the generators to generate samples from a novel domain. A more recent work (Ojha et al. (2021)), reduces overfitting on limited data by preserving the relative similarities and differences in the instances of source domain using cross domain correspondence.

Among the single-shot methods, StyleCLIP (Patashnik et al. (2021)) extended CLIP to perform GAN-based image editing. StyleCLIP uses the CLIP embedding vector to traverse the StyleGAN manifold to make a generated image's CLIP embedding similar to the target vector, while remaining close to the input in latent space. A downside to this approach is that these edits are unable to shift the domain of a GAN outside its manifold. Another approach, StyleGAN-NADA (Gal et al. (2021)) creates a new GAN using refinement learning. The original and target domain are represented by CLIP text embeddings. The difference of the embeddings represents a direction used to shift the domains. Although in the accompanying source-code (rinongal/StyleGAN NADA), they use a

bootstrap-estimate of the mean CLIP image embedding of the original domain, and use a reference image or its CLIP image embedding to represent the new domain.

Latent space interpretation and semantic editing. GAN interpretation and understanding of the latent space has been a topic of interest since the advent of the GANs. Some notable works in this domain (Bau et al. (2018; 2019); Härkönen et al. (2020); Shen et al. (2020); Tewari et al. (2020a)) have led to many GAN-based image editing applications. More recent studies into the activation space of the StyleGAN have demonstrated the semantic and spatial properties of the GANs can be exploited for downstream tasks like unsupervised and few-shot part segmentation (Zhang et al. (2021); Tritrong et al. (2021); Abdal et al. (2021a); Collins et al. (2020); Bielski & Favaro (2019)), extracting 3D models of the objects (Pan et al. (2021); Chan et al. (2020)) and other semantic image editing applications (Zhu et al. (2021); Tan et al. (2020); Wu et al. (2020); Patashnik et al. (2021)).

Image embedding is one of the approaches used to study the interpretability of the GANs. To enable the semantic editing of a given image using GANs, one needs to embed/project the image into its latent space. Image2StyleGAN (Abdal et al. (2019)) embeds images into the extended StyleGAN space called $W+$ space. Some followup works (Zhu et al. (2020a); Richardson et al. (2020); Tewari et al. (2020b)) introduce regularizers and encoders to keep the latent code faithful to the original space of the StyleGAN. Improved version of Image2StyleGAN (II2S) (Zhu et al. (2020b)) uses $P - norm$ space to regularize the embeddings for high-quality image reconstruction and image editing. We use this method to embed real images into the StyleGAN and show that our domain adaptation preserves the properties of the original StyleGAN in Sec 4.

Image editing is another tool to identify the concepts learned by a GAN. In the StyleGAN domain, recent works (Härkönen et al. (2020); Shen et al. (2020); Tewari et al. (2020a); Abdal et al. (2021b)) try to extract meaningful linear/ non-linear paths in the latent space. Among them is InterfaceGAN (Shen et al. (2020)) which finds the linear directions of editing in a supervised manner. On the other hand, GANSpace (Härkönen et al. (2020)) extracts unsupervised linear directions for editing using PCA in the W space. Another framework, StyleRig (Tewari et al. (2020a)), maps the latent space of the GAN to a 3D model. StyleFlow (Abdal et al. (2021b)) extracts non-linear paths in the latent space to enable sequential image editing. In this work, we will use StyleFlow to test the semantic editing of our domain adapted images.

3 METHOD

Our approach involves fine-tuning a GAN trained for some original domain A, e.g. FFHQ faces, to adapt it to a new related domain B. In our approach, the images in domain A and the images in domain B are related to each-other by a common latent code. Any image which can be generated or embedded in domain A can be transferred to a corresponding and similar image in domain B. We use the CLIP embeddings as a semantic-space in order to model the difference between domains A and B, and we use StyleGAN (Karras et al. (2018; 2020b)) as the image generator because of its high performance on several domains. A key to our approach is to preserve directions within and across domains as illustrated in Fig. 2. Before fine-tuning the GAN for domain A (to obtain the GAN for domain B), we determine a domain-gap direction. This direction, called v^{ref}, which is a vector in CLIP embedding space, points towards a reference image in domain B from its corresponding image I_A in domain A. Let I_B indicate an input reference image in domain B and I_A its corresponding image in domain A. Finding I_A in domain A for a given image in domain B is a significant limitation in the current state of the art Gal et al. (2021) as they use the mean of domain A. The mean of domain A is a very crude approximation for I_A. Instead, we propose an inverse domain adaption step, by projecting the image I_B into the domain A to find a sample that is a lot more similar and specific to the reference image than the mean of domain A. In principle, this problem is also a domain adaption problem similar to the problem we are trying to solve, just in the inverse direction. The major difference is that we have a pre-trained GAN available in domain A. We use the II2S GAN-inversion method (Zhu et al. (2020b)) in order to find a latent code for an image similar to I_B that is plausibly within domain A. II2S uses a hyperparameter, λ, which can be increased in order to generate latent codes in higher density regions of the $W+$ latent space. The effect of this parameter is shown in Fig. 1. The value suggested in the original work was $\lambda = 0.001$, however, low values of lambda allow II2S to find latent codes that are too far away from the latent-codes produced by the mapping network of the original GAN and thus produce images that are less plausible to have come

from domain A, underestimating the gap between domains. In the context of domain shift we find it is useful to use $\lambda = 0.01$ as illustrated in Fig. 1. The result is a latent code w^{ref} in $W+$ space which is shifted towards a high-density portion of the domain A. Then the image generated from that code, I_A, is an image in domain A that corresponds with I_B. We use the CLIP image-embedding network E_I to find

$$v^{\text{ref}} = E_I(I_B) - E_I(I_A). \tag{1}$$

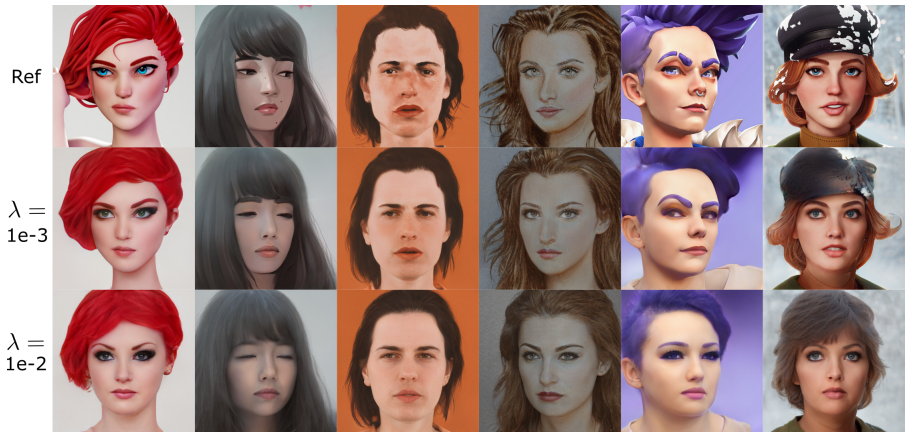

Figure 1: An illustration showing how II2S embeds I_B in the the original StyleGAN domain A, shown for two different values of λ. Five different reference images from other domains are shown in the top row. The value recommended by Zhu et al. (2020b) is shown in the second row, and the value used in this work is shown in the third row. Although there is some subjectivity involved, we believe that the large value $\lambda = 1e-2$ is needed for II2S to find images that plausibly could belong to the domain A, which in this case is FFHQ faces.

Training As illustrated in Fig. 1, we use II2S to find an image I_A which we consider to be similar to I_B still plausibly within a domain A. In principle, it is possible that II2S finds I_A so that I_B is similar enough to be considered the same, in which case the two domains overlap. However, we are concerned with the cases where the domains are different, and the vector v^{ref} indicates the direction of a gap between domain A and domain B. We use refinement learning to train a new generator, G_B, so that images generated from G_B are shifted parallel to v^{ref} in CLIP space, relative to images from G_A. The desired shift is indicated by the red arrows in Fig. 2. During training, latent codes w are generated using the mapping network of StyleGAN2. Both G_A and G_B are used to generate images from the same latent code, but the weights of G_A are frozen and only G_B is updated during training. The goal of refinement learning is that G_B will preserve semantic information that is *within* domain A but also that it will generate image shifted *across* a gap between domains. The overall process of training is illustrated in Fig. 3.

During training, our goal is to shift CLIP embeddings from domain A in a direction parallel to v^{ref}. We use the vector v^{samp} to represent the current domain shift of the network G_B during training, on a single sample. We have

$$v^{\text{samp}} = E_I(G_B(w)) - E_I(G_A(w)) \tag{2}$$

as a cross-domain vector for corresponding images generated from the same w latent code using the two generators. The reference and sample shifts are parallel when the cosine-similarity score between the two vectors is high, so we use the loss

$$L_{\text{clip_across}} = 1 - \text{sim}(v^{\text{ref}}, v^{\text{samp}}), \tag{3}$$

where $\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$ is the cosine similarity score. This loss term is minimized when the domain shift vectors are parallel.

During training, it is important that the reference image I_B matches the generated image, $G_B(w^{\text{ref}})$, both in a semantic sense, as measured by the similarity of the CLIP embedding, and also in a visual

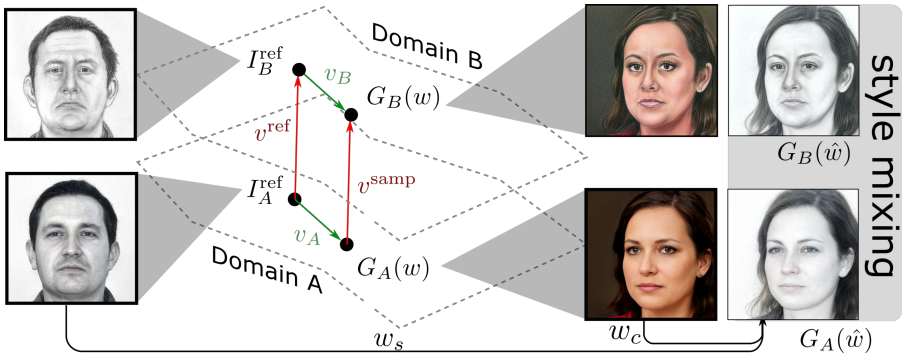

Figure 2: The vectors in the CLIP image embedding space, E_I, which control domain adaptation. Each domain is depicted here as a dashed outline; the vectors v^{ref} and v^{samp} cross between the two domains and are used to refine a generator for domain B. Corresponding images should be shifted in the same direction. The vectors v_A and v_B model important semantic differences within each domain that should also be preserved by domain transfer. $G_A(w)$ and $G_B(w)$ are corresponding images for an arbitrary latent-code w encountered during training. Style mixing (shown on the right) shifts a part of the latent code towards the reference image effecting the result in both domains.

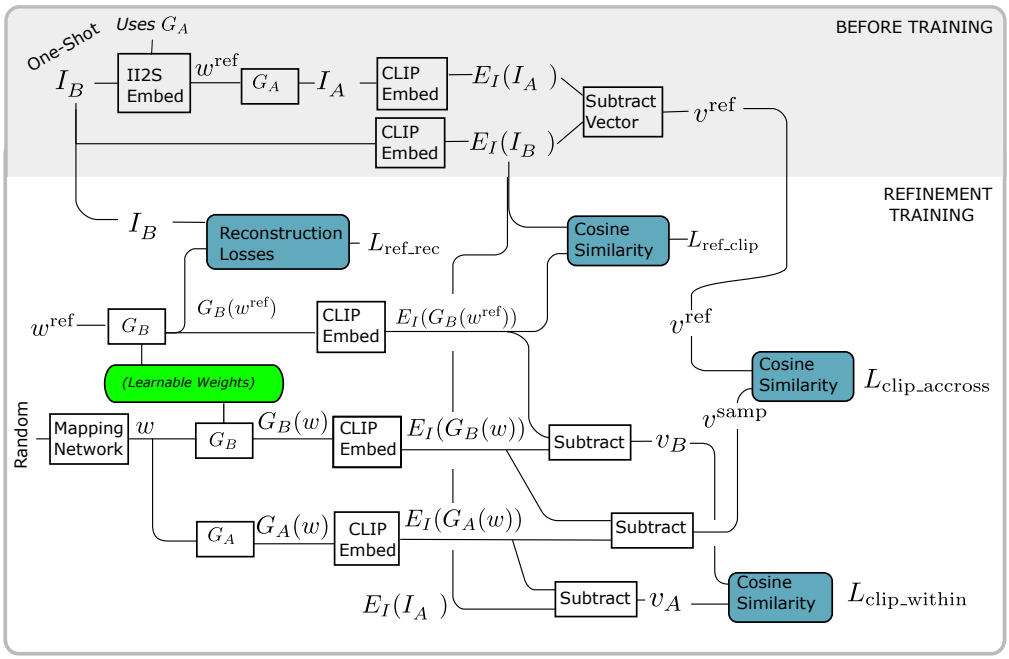

Figure 3: A process diagram for domain transfer. White rectangles indicate calculations, computed values are shown on the connecting lines. The four loss-calculations are indicated by blue rectangles, and the learnable weights are indicated in green.

sense. We accomplish this using two losses: $L_{\text{ref_clip}}$ and $L_{\text{ref_rec}}$. The first loss measures the change in the CLIP-embeddings of the original and reconstructed reference image,

$$L_{\text{ref_clip}} = 1 - \text{sim}\left(E_I\left(I_B\right), E_I\left(G_B(w^{\text{ref}})\right)\right), \tag{4}$$

ensuring that the G_B can reconstruct the embedding direction. Unlike $L_{\text{clip_accross}}$, this loss term is not based on a *change* in embeddings between the two domains, instead it guides G_B by aligning it to a global direction in CLIP space.

The second loss term is a reconstruction loss based on perceptual and pixel-level accuracy,

$$L_{\text{ref_rec}} = L_{\text{PIPS}}\left(I_B, G_B(w^{\text{ref}})\right) + L_2\left(I_B, G_B(w^{\text{ref}})\right) \tag{5}$$

where L_{PIPS} is the perceptual loss from Zhang et al. (2018), and L_2 is the squared euclidean difference between pixels. The purpose of this loss is to ensure that the visual, and not just the semantic, qualities of the image are preserved. This is necessary in addition to $L_{\text{ref_clip}}$ because, while the CLIP embeddings do capture many visual qualities of the image, there are still many perceptually distinct images that could produce the same CLIP embedding.

There is a tendency for GANs to reduce the variation during training, especially in low shot domains. We combat this by preserving the semantic information that is *not* related to the domain gap. A semantic change that is not related to the change in domains should not be affected by G_B. Therefore, the vector connecting the reference and sample images within the domain A should be parallel to the corresponding vector in domain B. Let $v_A = E_I(G_A(w)) - E_I(I_A)$ be a vector connecting a sample image with latent-code w to the reference image in the CLIP space. This vector represents semantic changes that are *within* the manifold of domain A, and we want the matching semantic changes to occur within the domain B. Let $v_B = E_I(G_B(w)) - E_I(I_B)$ denote the corresponding vector in domain B. We introduce the loss

$$L_{\text{clip_within}} = 1 - \text{sim}(v_A, v_B), \tag{6}$$

which is minimized when the two within-domain changes are parallel.

The final loss is then

$$L = L_{\text{clip_across}} + L_{\text{clip_within}} + L_{\text{ref_clip}} + L_{\text{ref_rec}}. \tag{7}$$

Together, these four loss terms guide the refinement process for G_B. Among these losses, $L_{\text{clip_across}}$ was proposed by StyleGAN-NADA (Gal et al. (2021)). The other losses are novel contributions of this work.

Style Mixing After the training step, the generator G_B generates images that are semantically similar to the reference image I_B. However, we have observed that the visual style may not be sufficiently similar. We attribute this to the idea that the target domain may be a *subset* of the manifold of the new generator G_B. This issue was addressed in StyleGAN-NADA (Gal et al. (2021)) using a second latent-mining network in order to identify a distribution of latent codes within the domain of G_B that better matches the reference image. Our approach exploits the structure of latent codes in $W+$ space. Latent vectors in $W+$ space can be divided into 18 blocks of 512 elements, each impacting a different layer of StyleGAN2. Empirically, the latter blocks of the $W+$ code have been shown to have more effect on the style (e.g. texture and color) of the image whereas the earlier layers impact the coarse-structure or content (Zhu et al. (2021)) of the image. We partition the latent code in the image into $w = (w_C, w_S)$ where w_C consists of the first m blocks of the $W+$ latent code that capture the content of the image, and w_S consists of the remaining blocks and captures the style. Then we transfer the style from a reference image using linear interpolation, to form $\hat{w} = (w_C, \hat{w}_S)$ where

$$\hat{w}_S = (1 - \alpha)w_S + \alpha(w_S^{\text{ref}}). \tag{8}$$

Consider the distribution of images generated from random w drawn according to the distribution of latent codes from the mapping network of StyleGAN2. If $\alpha = 0$, then the distribution of images $G_B(\hat{w})$ include the reference image, but encompass a wide variety of other fine visual styles. If $\alpha = 1$, then the images $G_B(\hat{w})$ will still have a diverse content, but they will all very closely follow the visual style of I_B. An important application of this method is in conditional editing of real photographs. To achieve that, first we take a real input image I_{real} and invert it on domain A generator G_A using II2S in order to find a $W+$ latent code w_{real}. Then $G_B(w_{real})$ generates a corresponding image in domain B. We can then compute $\hat{w}_{real}$ by interpolating the style codes (Eq. 8) so that the final image $G_B(\hat{w_{real}})$ is similar to I_{real} but has both content and the visual style shifted towards domain B.

4 RESULTS

In this section, we will show qualitative and quantitative results of our work. While the most closely related work StyleGAN-NADA (Gal et al. (2021)) is unpublished at the time of submission we compare to it as the main competitor. It is the only existing single shot method. Moreover it demonstrates impressive improvements over the state of the art and even beats many SOTA few-shot methods considering the visual quality. As our method can still significantly improve upon the results shown in StyleGAN-NADA, this underlines the importance of our idea in reducing overfitting.

Training and Inference Time. Given a reference image, the training time for our method is about 15 minutes for 600 iterations on a single Titan XP GPU using ADAM as an optimizer with the same settings as Gal et al. (2021). We use a batch size of 4. At inference time, there are different applications. In a basic operation, GAN generated images can be transferred with a single forward pass through a GAN generator network, which works in less than a second. Considering a more advanced operation, where existing photographs are embedded into a GAN latent space using, the additional embedding time has to be considered. This embedding time is a few seconds using and about two minutes using Zhu et al. (2020b).

Visual Evaluation. In Figure 4, we show a comparison of our results on faces against the two most relevant competing methods – StyleGAN-NADA (Gal et al. (2021)) and few-shot-domain-adaptation (Ojha et al. (2021)). The results show that our method remains faithful to the original identity of the embedded images in Domain A, while the other two methods either suffer from collapsing to narrow distributions with changed content features (for example the identity of a person) or severe overfitting. We show many additional visual results in the appendix, including results on cars and dogs and results for fine tuning the domain adaptation.

User Study. We also perform a user study by collecting 187 responses from mechanical Turk in order to compare the visual quality and the domain transfer capabilities of our framework compared to the competing methods. When asked which method generates higher quality images from doman B, 73% of users preferred our approach to StyleGAN-NADA, and 77% selected ours over Few-shot (Ojha et al. (2021)). When asked which method is better and maintaining the similarity to a corresponding source image in domain A, we found that 80% of the responses chose our approach over StyleGAN-NADA, and 91% preferred our approach to Few-shot. Our method beats the competing works in terms of the quality of the generated image, and the similarity of the generated image to the source image from domain A. Other methods produce images that are more similar I_B, but that can also be an indication of overfitting / mode collapse.

Ablation study. We perform an ablation study to evaluate each component of our framework. In Fig. 5, we show the effect of II2S embedding, different losses and style mixing/interpolation on the output of the domain B. Notice that after the final style mixing stage the images maintain the content of the domain A and styles are mixed from the reference image using the style interpolation.

Image editing capabilities. Another important aspect of our method is that we are able to preserve the semantic properties of the original StyleGAN (domain A) in domain B. We can make edits to the images in domain B via the learned generator G_B without retraining the image editing frameworks on the new domain. Fig. 6 shows image editing capabilities of the new domain B. We test the editing capabilities of both reference and the images embedded in domain A. We use StyleFlow edits such as lighting, pose, gender etc. to show the fine grained edits possible in the new domain.

Limitations Our method has several limitations. Some of these limitations are inherent due to the challenging nature of the problem of single-shot domain adaptation. Other limitations can be addressed in future work. First, when we find the initial image in domain A that corresponds to the input in domain B, we do not attempt to control for the semantic similarity. Future work should encourage the images to have similar semantics. Second, sometimes using in-the-wild reference images for domain adaptation can result in artifacts in the final image. We attribute this problem to the fact that not every style can be represented by the original StyleGAN as such images can be out of the latent distribution in original StyleGAN (domain A). Third, also relevant to the original

Figure 4: Comparison of our framework with state-of-the-art frameworks for StyleGAN domain adaptation. Here we compare with StyleGAN-NADA (Gal et al. (2021)) and few-shot-gan-adaptation (Ojha et al. (2021)) methods. The interpretation of rows and columns is same as Fig. 8. Notice that our method is able to match the styles of the reference images, while StyleGAN-NADA fails to maintain the content of the images in domain A (for example the identity of a person is lost). On the other hand, the few-shot-gan-adaptation method suffers from severe mode collapse.

Figure 5: Ablation study of the losses and style mixing used in our framework. From left to right: the reference image I_A and several images from domain A, the baseline approach (StyleGAN-NADA), adding II2S instead of using the mean of domain A, adding $L_{\text{ref_clip}}$, $L_{\text{clip_within}}$, and then using style mixing. The top row shows reconstructions of the reference image I_B using G_B.

distribution of the StyleGAN, embeddings into the StyleGAN work best when the objects are transformed to the canonical positions (for example face transformations same as FFHQ). Extreme poses of the objects in the reference images sometimes fail to extract the relevant styles.

5 Conclusions

We propose a novel method for single shot domain adaption. The main achievement of this work is to obtain results of unprecedented quality while significantly reducing overfitting observed in previous

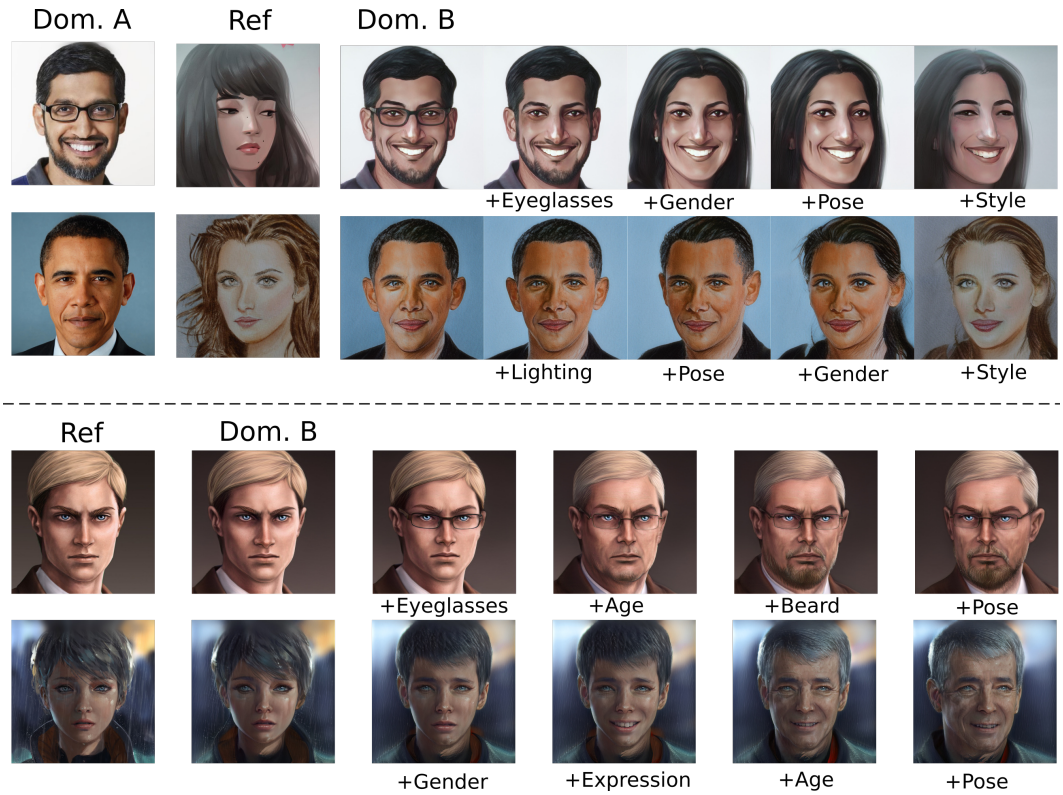

Figure 6: Image editing capabilities of the new domain B using StyleFlow Abdal et al. (2021b). The top figure shows the editing results of the embedded real image I_{real} transferred to domain B and the bottom figure shows the editing results of the reference images I_B in domain B. Notice that our method preserves the semantic properties of the original StyleGAN.

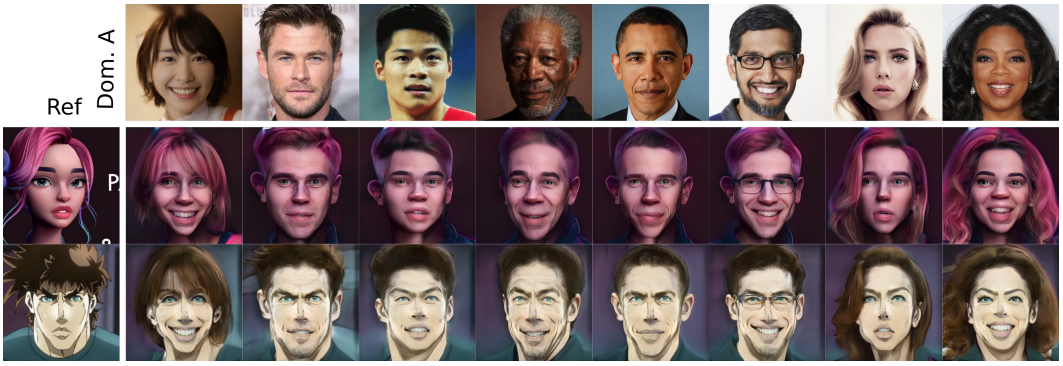

Figure 7: Some failure cases of our method. In these examples we can observe that the identity of the face is compromised a bit stronger than in typical examples of our method.

work. The technical key components of our work is a method to model the domain gap as vector in CLIP embedding space including several extensions for fine-grained attribute-based control. We also introduce several new regularizers and a style mixing approach. In future work, we would like to extend to few shot domain adaption, revisit text-based control using the insights from our work, and propose a solution for stylized rendering of 3D models.

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, Peihao Zhu, Niloy Mitra, and Peter Wonka. Labels4free: Unsupervised segmentation using stylegan, 2021a.

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 2021b. ISSN 0730-0301. doi: 10.1145/3447648. URL https://doi.org/10.1145/3447648.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks, 2018.

David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4502–4511, 2019.

Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. *arXiv preprint arXiv:1905.12663*, 2019.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks, 2016.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. doi: 10.1109/cvpr.2017.18. URL http://dx.doi.org/10.1109/CVPR.2017.18.

Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis, 2020.

Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of gans, 2020.

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

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

Wonjong Jang, Gwangjin Ju, Yucheol Jung, Jiaolong Yang, Xin Tong, and Seungyong Lee. Stylecarigan. *ACM Transactions on Graphics*, 40(4):1–16, Aug 2021. ISSN 1557-7368. doi: 10.1145/3450626.3459860. URL http://dx.doi.org/10.1145/3450626.3459860.

justinpinkney/toonify. Toonify. https://github.com/justinpinkney/toonify.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.00503. URL http://dx.doi.org/10.1109/CVPR.2019.00503.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

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.

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

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020b.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *CoRR*, abs/2106.12423, 2021.

Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.01348. URL http://dx.doi.org/10.1109/cvpr42600.2020.01348.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. doi: 10.1109/iccv.2019.01065. URL http://dx.doi.org/10.1109/ICCV.2019.01065.

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans, 2020.

Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation, 2020.

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, 2021.

Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d {gan}s know 3d shape? unsupervised 3d shape reconstruction from 2d image {gan}s. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=FGqiDsBUKL0.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery, 2021.

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.

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.

rinongal/StyleGAN NADA. Stylegan-nada. https://github.com/rinongal/StyleGAN-nada.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

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 Transactions on Graphics (Proc. SIGGRAPH)*, jul 2021.

Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. Michigan: Multi-input-conditioned hair image generation for portrait editing. *ACM Transactions on Graphics (TOG)*, 39(4):1–13, 2020.

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6142–6151, 2020a.

Ayush Tewari, Mohamed Elgharib, Mallikarjun BR, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. volume 39, December 2020b. doi: 10.1145/3414685.3417803.

Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. Repurposing gans for one-shot semantic part segmentation, 2021.

Qian Wang and Toby Breckon. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (04):6243–6250, Apr 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i04.6091. URL http://dx.doi.org/10.1609/AAAI.V34I04.6091.

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. *arXiv preprint arXiv:2011.12799*, 2020.

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

Ruixiang ZHANG, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/4e4e53aa080247bc31d0eb4e7aeb07a0-Paper.pdf.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort, 2021.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training, 2020.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision*, pp. 592–608. Springer, 2020a.

Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2020b.

Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks, 2021.

A Appendix: Additional Results

Visual Evaluation of Style Transfer. We provide additional visual evaluation of the results by transferring style to existing photographs embedded into the StyleGAN latent space. In Fig. 8 we show results for faces. The input photographs are in the top row and the style images in the first column. We can see that the results take on the style of the reference image, even though the reference image is far outside the original GAN's latent space. Also, we can notice that overfitting is successfully limited, as each result maintains several important aspects of the input image. In Fig. 11 we show results for cars and dogs on the same task. This shows that our method is consistent across different StyleGAN objects/datasets.

Controlling the Domain Gap. Our method provides a way to control the domain gap between the domain A and domain B by explicitly controlling both the content and the style of the images sampled from or embedded in domain A. We can define which parts should be copied from the input image and which parts from the reference image. For the style interpolation case, Fig. 9 shows that we can control the levels of style from the reference image which is not possible with any of the competing methods. We also show the content interpolation in Fig. 10 where we can change the content of the image based on the reference image while remaining faithful to the style of the image in domain A. Both these operations give users more control over the content generation and editing.

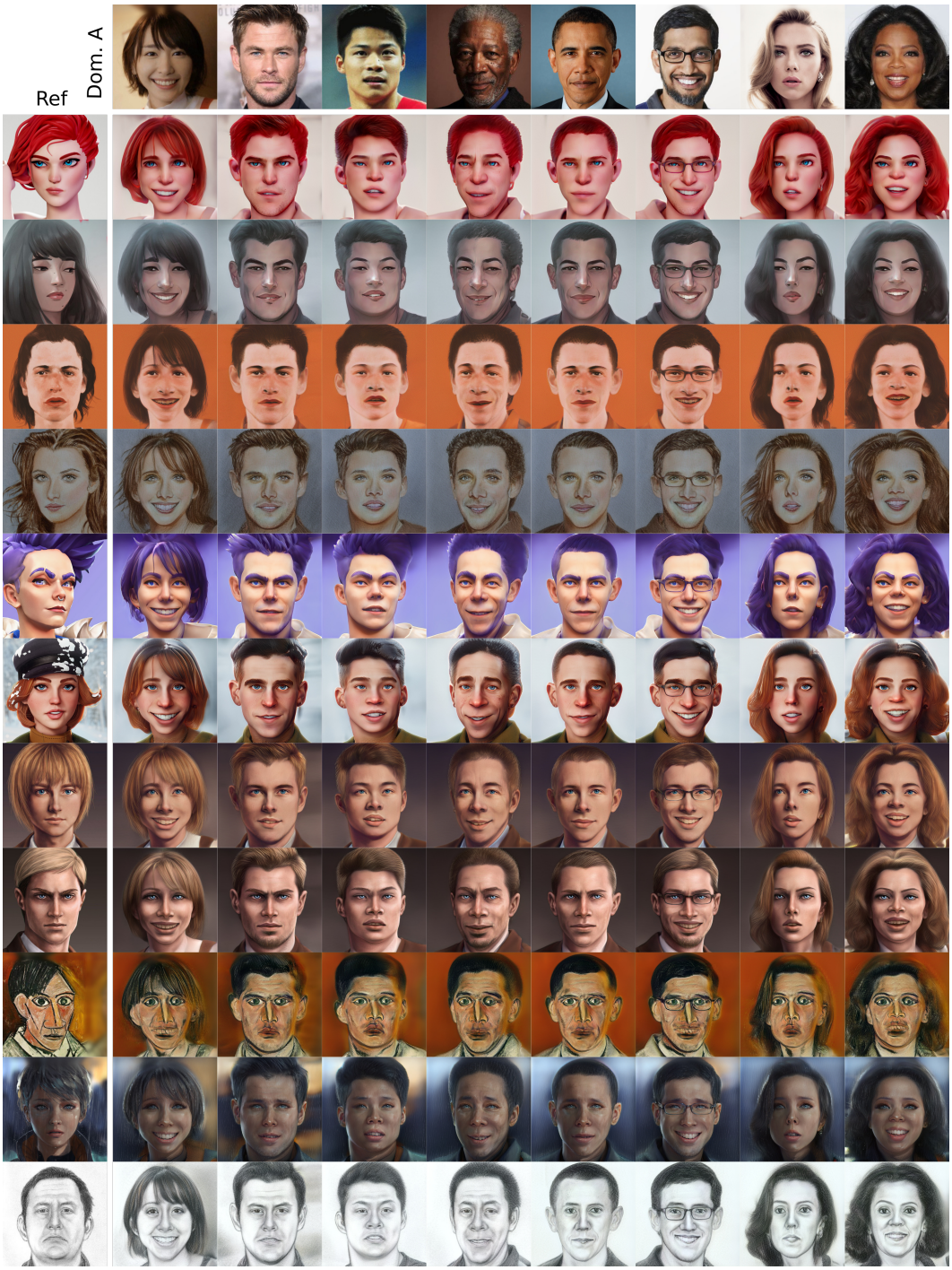

Figure 8: Style transfer results obtained by our method after style interpolation in domain B. The top row represents the real images embedded in the latent space of G_A (domain A) whose latent codes are then used by G_B (domain B). The first column represents the reference images I_B which are input to our domain adaptation framework.

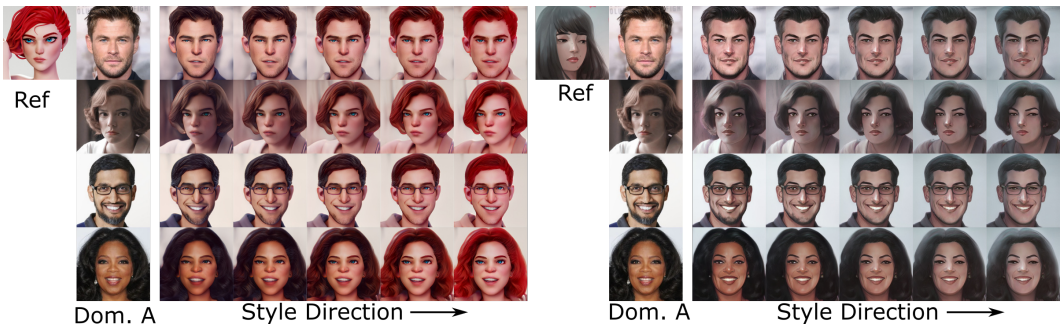

Figure 9: Style interpolation results achieved by our framework. Unlike the competing methods, our method has an explicit control over the styles in the domain B. Each sub figure shows a reference image and images embedded in domain A. Notice that we can control the styles depending on the user input.

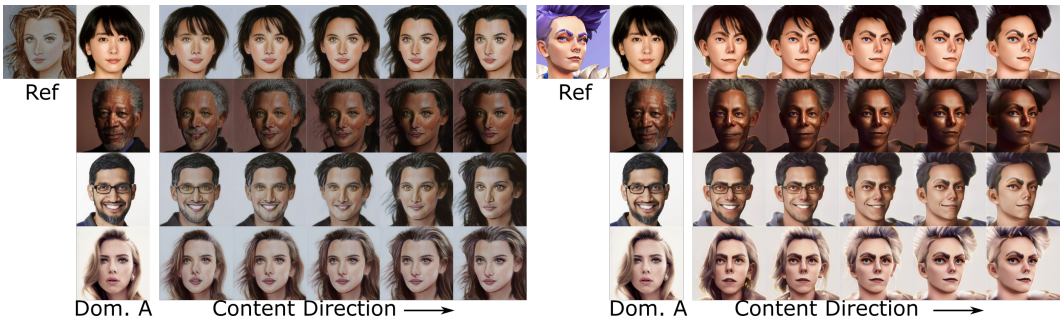

Figure 10: Content interpolation results achieved by our framework. Each sub figure shows a reference image and images embedded in domain A. Notice that we can control the content of the images in domain A using the reference image while keeping the style faithful to the original image.

Figure 11: Our style transfer results on cars and dogs. The interpretation of rows and columns is same as Fig. 8.