NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Applied Mathematics and Informatics"

BACHELOR'S THESIS

Research Project on the Topic:

Developing effective StyleGAN encoders for domain adaptation

Submitted by the Student:

group #БПМИ202, 4th year of study Sedov Sergey

Approved by the Supervisor:

Meshchaninov Viacheslav
Research Fellow
Faculty of Computer Science, HSE University

Co-supervisor:

Alanov Aibek
Research Fellow
Faculty of Computer Science, HSE University

Moscow 2024

Contents

Annotation

StyleGAN models have shown significant advances in the plain domain adaptation settings, which implies changing domains on stochastically generated data, e.g. painting styles of generated faces. For the real-data applications of StyleGAN models, Encoders are developed, transforming the information from initial images into the latent space of StyleGANs. Latents are then substituted into the generative models, allowing the application of various methods for the real-data tasks, with domain adaptation being one of them. As research on domain adaptation and Encoders frequently develops separately, methods from both of these fields may not merge with each other in an effective way, so the precise interrelation between them is crucial in terms of possible performance. In particular, we consider state-of-the-art domain adaptation methods of StyleGAN-2 that were developed separately with the Encoder approach. We then research the possibilities in changing the existing Encoder architectures to improve their combination, showing better results in the real-data adaptation tasks.

Аннотация

Модели StyleGAN показывают значительные успехи в задачах доменной адаптации случайных сгенерированных изображений, например в задачах смены стилей рисовки сгенерированных лиц. Для применения моделей StyleGAN на существующих данных разрабатываются Энкодеры - отдельные модели, преобразующие информацию из первоначальных изображений в латентное пространство StyleGAN. Подставление латентных представлений в генератор StyleGAN'а позволяет эффективно применять методы для задач реальных данных, в том числе методы доменной адаптации. Поскольку исследования Энкодеров и Доменной Адаптации StyleGAN зачастую развиваются отдельно друг от друга, методы из обеих областей могут недостаточно подходить для совместного применения, поэтому эффективное объединение этих методов играет важную роль с точки зрения конечных результатов. В частности, мы рассматриваем современные методы доменной адаптации модели StyleGAN-2, которые были разработаны отдельно от Энкодеров. Мы исследуем возможные изменения в архитектуре Энкодеров в целях улучшения их взаимодействия с методами доменной адаптации на задачах редактирования существующих изображений.

Keywords

Deep Learning, Computer Vision, StyleGAN, Domain Adaptation, StyleGAN Encoders

1 Introduction

Through the recent years, Generative Adversarial Networks have been widely developed in various Computer Vision tasks. Comparing to other generative approaches such as Variational Auto-Encoders and Diffusion Models, adversarial training of generator together with the appropriate choice of discriminator architecture leads to significantly higher quality of the generated images. And vice versa, GANs suffer from the lack of diversity: generator's main goal is to output precisely accurate faces so that discriminator could not classify them as artificial ones, therefore training process does not predispose the capability of various experimental images generation. A well-known extreme of this behavior among GAN models is the mode-collapse phenomenon, when generators tie to unique mode of the distribution over images and do not possess any generalization ability over the training data. However, contemporary approaches deal with such problems fairly good by applying diverse regularizations. In other words, current usage limitations of GAN models could be formulated as follows: while not being able to surpass training dataset in creativity, they accumulate the precise knowledge of provided data, which could be used both for generation of new samples and for applications in real-data editing tasks.

One of the milestones of the area of GAN research is the emergence of StyleGAN [9], [10], [8] models, starting from 2019, which have proposed several crucial modifications for the image generation tasks since then. Besides being one of state-of-the-art solutions for computer vision generation tasks in terms of precision quality, this model family is remarkable in terms of interpretability and controllability of the generation process, leading towards successes in various applications beyond common settings. One of the rapidly developing areas connected with these models is the research on encoders, which invert information real images into the latent spaces of StyleGANs. Such approaches allow the reconstruction of real images using the built-in StyleGAN knowledge and may be further adapted towards diverse editing and domain shifting purposes. However, domain adaptation approaches are usually effectively developed separately, on original stochastic image generation tasks. Therefore, in terms of practical applications on real images, the joint usage of StyleGAN Encoders and corresponding domain adaptation methods must be effective. Moreover, domain adaptation approaches could be developed together with encoder architectures, leading the way for the mutual research of these areas. In this paper we aim to analyze the behaviour of

different encoder methods in domain adaptation settings in order to formulate several hypotheses on devising their collaborative evolution.

2 Literature Review

StyleGAN models

One of the initial breakthroughs of these models is the way that they incorporate the stochastical component into the generator architecture. Authors switch from the common straightforward convolutional backbone, which gradually transforms sampled noise into images. Following the idea of separating duties between different layers of generator network, they make use of the Adaptive Instance Normalization [7] - AdaIN operation. Basically, AdaIN normalizes layer's inputs and rescales them channel-wise with the received mean and std coefficients, which makes each convolutional layer of the generator work with its own rescaled feature maps. These mean and std coefficients appear to be an excellent place for stochasticity in the generator model. Authors add the fully-connected Mapping Network, which learns to transform sampled noise inputs z to the latent codes w. After applying different learned affine transforms to w for each block, they acquire the aforementioned rescaling coefficients.

While such modification may already improve the quality of generated images, authors make an important finding that sampled z, which then transforms into latent w and rescaling coefficients, is in fact a sufficient source of randomness by its own. They remove the default noise generator input as it interferes with randomness in AdaIN, so the initial backbone learns the determined weights and all meaningful information for particular generated images is contained in the rescaling coefficients, previously taken from latent w's. Authors however argue that there are some eternally random features of images, for instance, the same hairstyle may look differently from time to time. Therefore, they add true noise inputs transformed only with affine layers to all backbone blocks, making it somewhat similar to latent injection - different parts of backbone have their own transformed randomness, while previously it was passed only to the first layer of network.

As already mentioned, all meaningful varying information, determining specific features of each generated image contains in latent w's, therefore mapping network becomes a crucial part of the architecture. The essential problem arising here was the controllability of these latents, as we aim to build an interpretable generator with its layers having different purposes. Training such generator would allow us to work with latents directly, extrapolating features from one image to another in a predictable way. Followed by these conclusions, authors state the importance of the

disentanglement property for the latent space. In fact, we seek GAN latent space to consist of linear subspaces that independently control certain features of images. Authors come up with new disentanglement metrics, that help them to choose from a variety of modifications and provide the evidence of the w latents' usefulness.

Perceptual path length controls the smoothness of transition in latent space between images. It moves in latent space ($\mathcal{Z}$ or $\mathcal{W}$) from first image latent to the other, estimating the difference in generated images for each ε-step. If the latent space is disentangled, then such steps will show up as small and smooth changes in the generated images and the PPL metric will be low. Another metric calculates the linear separability score by making use of SVM and pretrained classifier neural network. It computes the conditional entropy $H(Y_i|X_i)$ for 40 various image attributes, where X_i are the SVM prediction and Y_i are the neural network predictions. Such approach reveals the amount of additional information needed to determine attribute classes, comparing to linear SVM classification. The final separability score for all attributes is computed as $exp(\sum_i H(Y_i|X_i))$ and provides another view on the disentanglement property of latent spaces.

StyleGAN-2 [10] model brings several significant improvements: switching from AdaIN operation to weight demodulations (weaker renormalizations on statistical assumptions), introducing path length and lazy regularizations, rejecting progressive growing in favor of residual architectures. Either way, the essential concept of style latent space stays the same, serving as a convenient mechanism for various image related tasks.

StyleGAN Encoders

As StyleGAN models showed great abilites in image manipulation tasks, plain generators are not capable to interact with real images by themselves. GAN Encoders are one of the contemporary approaches to deal with real images, initially solving the image-inversion task: reconstruction of such latents from real images, that generated images would be as close as possible to the real ones. In fact, this setting includes both dimension reduction problem and the interrelation with the pretrained StyleGAN latent spaces. Actually, in image inversion tasks Encoders may not succeed in predicting a unified $w \in \mathcal{W}$ latent, which is then transformed with affine layers into 18 latents for each block of StyleGAN. The common approach is to map each image to $\mathcal{W}+$ space of these 18 latents, resulting in $18 \cdot 512$ dimensions.

pSp Encoder

One of the most common StyleGAN Encoders is the Pixel-to-Style-to-Pixel Encoder or pSp [12]. Its architecture is similar to U-Net, dissassembling the feature maps into lower resolutions for first layers of StyleGAN and assembling them back to higher resolutions for the deeper layers. Thus, the pSp Encoder produces 18 shift predictions from the average $\bar{w}$, being trained using MSE loss on image inversion, LPIPS loss that deals better with feature extraction and cosine similarity loss of generated and original images using the feature maps of the pretrained ArcFace network [4].

Encoder-for-Editing

Another bright example among StyleGAN Encoders is the work on Encoder-for-Editing simply named as E4E [13]. This work aims to match distributions of encoder-inferred style codes with original ones by lowering the variance of inferred ones. Authors design a convolutional encoder, training it to predict main "support" latent and a set of offsets to it for each StyleGAN block. They argue that previous works pursue lower distortion ratios of the final images, while the perceptual quality and editability of them is also important. Followed by this idea, they train an encoder together with an additional discriminator that works on the space of latents, regularizing the sum of predicted offsets on the other hand. Authors remark that such approach leads to lower variance among latents, making them closer to the original $\mathcal{W}+$ space. However, they also observe distortion-editability and distortion-perception trade-offs in image inversion tasks, which means that excessive regularization towards $\mathcal{W}+$ may lead to lower distortion qualities, while enhancing the editability abilities similar to original style codes.

Feature-Style-Encoder

While Feature-Style-Encoder or FSE [14] is one of the best StyleGAN Encoders to this day, this work is also important for the analysis of latent-only approaches to inversion tasks. In fact, the key idea of authors was to train an another encoder head for replacing the whole feature-maps of K-th StyleGAN layer G^K with the feature prediction F, besides providing latents predictions. As such encoder really shows better results, it either means that:

- Contemporary encoder methods do not work with style-codes in the best possible way, leaving the gap which could be fullfilled by introducing stochasticity by directly modyfing the backbone weights.
- Style-codes could not be considered containing the universal image information, i.e. the

pretrained StyleGAN model does not store generalized absolute information about faces into the backbone weights while leaving all stochasticity to latents. It is being excessively biased towards its training data, on the contrary with human brains, that may perceive faces in various representations.

In terms of image editing tasks, authors utilize the following procedure: if we would like to change image style-codes from w to $\tilde{w}$, we could add the difference in them on the level of generator feature maps - instead of replacing $G^K(w)$ with F, we would replace it with $F + G^K(\tilde{w}) - G^K(w)$. Such approach may be beneficial in terms of domain adaptation settings, which we would describe further on.

StyleRes Encoder

Finally, StyleRes Encoder [11] devises the idea of backbone weights interference. This time, the encoder is splitted into 3 parts. The E0 part has a regular duty of predicting the latent codes from images. Afterwards, these latent codes are fed to the first half of the freezed StyleGAN generator's backbone, let us name it G1. The trainable E1 part works both on the activations of the final G1 layer and the intermediate E0 layer. In fact, it's aim is to build interaction between encoder's features and generator ones, transmitting outputs towards the second half of the generator, G2. However, authors move further in the improvement of encoder-generator interaction, by processing the second branch on mixed latents - E0 predictions are mixed with the original StyleGAN latents from Mapping Network which works on the noise samples. Mixed latents are then passed through G1 once more to E2, the last part of the encoder, which works on activations of E1 and G1-mixed similar to E1 that works on E0 and G1. Finally, outputs of E2 are passed to the second half of StyleGAN generator. This intricate pipeline combines real image details with original noisy latents, incorporating everything into the generator backbone. While E1 is aiming to reconstruct the image details the best way inside the backbone, E2 follows the idea of adding randomness into the last layers of generator.

Domain Adaptation

We have reviewed several approaches to building StyleGAN Encoders in detail, though we have not spoken of domain adaptation settings yet. The purpose of such a narrative is to show the possible inconsistencies between these research areas, as domain adaptation frameworks usually develop separately with the encoders. In domain adaptation research it turned up that StyleGAN models possess a great internal knowledge of image structure, they are capable to rapidly adapt

towards new domains of images, even in one-shot settings. Moreover, recent research searches for the localized modifications of original StyleGAN model, showing results that are comparable with full fine-tuning approach. An important aspect of one-shot domain adaptation capabilities is that it may be performed on the text descriptions of the desired domain as well as on the original domain images.

HyperDomainNet

HypedDomainNet [2] proposed adapt generator towards new domain by inserting domain-modulated styles alongside with the original StyleGAN latents. Such approach requires training of a small fraction of model parameters, while preserving the consistency with the backbone architecture and showing comparable results with full-finetuning method of adaptation. However, in the future work authors study the necessity of adding domain modulations in parallel with the original latent-modulation.

StyleDomain

StyleDomain paper [1] introduced several approaches to one-shot domain adaptation in the similar domains settings (e.g. various painting styles of images) and few-shot adaptation for dissimilar domains (e.g. dogs' and cats' faces). Following the hypothesis that image-specific information is contained in the latent style codes and in the real-data applications encoders provide input latents (before the affine layers) or latent style codes (after the affine layers) to StyleGAN model, authors firstly consider simple setting called StyleSpace, which includes learnable offsets to style codes that adapt towards the new domain. This approach shows surprisingly good results for one-shot domain adaptation on similar domains. It matches the quality of whole StyleGAN generator fine-tuning, confirming the aforementioned hypothesis in some sense. However, its performance on dissimilar domains is worse than full methods' one, which implies the bias of generator backbone towards the initial domain.

Authors research several different ways of performing domain adaptation, expanding the set of trainable parameters from the style offsets. As latents are mapped into the backbone generator network with affine transforms, the consistent idea is to make these affine layers learnable. Though this idea includes much more learnable parameters, it does not improve the adaptation results much. As we expect our backbone to be biased towards the initial domain (e.g. towards dog faces, lacking the ability to generalize well among cat faces), we face the necessity of adapting its structure too. Authors propose to additionally train the intermediate StyleGAN block, which would adapt

the most important layer of style codes into the generator feature maps. While such approach does not increase the number of trainable parameters much, it effectively combines learnable affine weights with initial generator bias. Being called Affine+, it surpasses the perfomance of full finetuning approach even at dissimilar domains settings.

In our work we will focus on the improvements of StyleSpace, so we will consider this method in finer details. The style offsets were trained using different losses for the image-based and text-based domain adaptation tasks which obviously leads to different performance between them. We will make a comparison between these two settings in the Domain Adaptation Comparison section further on, while focusing on theory in this one.

Image-based one-shot domain adaptation

So, the image-based one-shot StyleSpace was trained with losses proposed in the DiFa [16] paper: global loss, local loss and selective cross-domain consistency loss.

$$\mathcal{L}_D\big(G_\theta(w)\big) = \mathcal{L}_{global}\big(G_\theta(w)\big) + \lambda_{local}\mathcal{L}_{local}\big(G_\theta(w)\big) + \lambda_{scc}\mathcal{L}_{scc}\big(G_\theta(w)\big) \tag{1}$$

These losses extensively use the pre-trained CLIP image encoder E_I. The **global loss** is based on the alignment of differences between CLIP-embeddings of stochastically generated images ΔI_{sample} and domain direction ΔI_{domain}. The last one represents the cosine similarity of domain-image I_{target} CLIP embedding v_{domain} and the average CLIP embedding over the source domain v_{source}:

$$\begin{gathered}
\mathcal{L}_{global}\big(G_\theta(w)\big) = 1 - \frac{\Delta I^T_{sample} \cdot \Delta I_{domain}}{\|\Delta I_{sample}\| \cdot \|\Delta I_{domain}\|} \\
\Delta I_{domain} = v_{domain} - v_{source}, \quad v_{domain} = E_I\big(I_{target}\big), \quad v_{source} = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[G_{\theta_0}(z)] \\
\Delta I_{sample} = E_I(G_\theta(z)) - E_I(G_{\theta_0}(z)), \quad z \sim \mathcal{N}(0, I)
\end{gathered} \tag{2}$$

Here the trained model is G_θ, while the freezed original StyleGAN generator is G_{θ_0}. In other words, it is being trained on the randomly sampled z to result into the same "domain shift" as the original domain image does, with the "domain shift" being evaluated by CLIP encoder.

Regarding the **local loss**, it's being an extensive regularizer on cosine similarities between intermediate CLIP-encoder feature-maps. It encourages the final model G_θ to inherit image features even with significant domain divergencies. Finally, the **selective cross-domain consistency loss** aims to preserve domain-invariant image features while supplementing the diverse domain-adaptations of various images. It computes sampled latents from domain-adapted and source generators $w_{domain} = G_\theta(z),\ w_{source} = G_{\theta_0}(z)$ and dynamically updates Δw based on them. As

during training G_θ gradually pays more attention to the finer differences between domain and source images, this loss punishes high deviations between w_{domain} and w_{source} if the previously accumulated Δw is low:

$$\mathcal{L}_{scc} = \|\text{mask}(\Delta w, \alpha) \cdot (w_{domain} - w_{source})\|_1 \tag{3}$$

Text-based one-shot domain adaptation

In the text-based one-shot domain adaptation settings, StyleSpace was trained only on the StyleGAN-NADA [5] loss, eventually similar to the DiFa global loss:

$$\begin{aligned} \mathcal{L}_{direction}\big(G_\theta(w)\big) &= 1 - \frac{\Delta I^T \cdot \Delta T}{\|\Delta I\| \cdot \|\Delta T\|} \\ \Delta I &= E_I\big(G_\theta(w)\big) - E_I\big(G_{\theta_0}(w)\big), \\ \Delta T &= E_T(T_{domain}) - E_T(T_{source}) \end{aligned} \tag{4}$$

Where E_I and E_T are CLIP image and text encoders, T_{domain} and T_{source} are text-descriptions of the desired and the source domain of the generator G_{θ_0}. In other words, the text-based domain adaptation imply the gap in CLIP encoder's understanding of text, the gap between CLIP image and text encoder knowledge and the absence of other regularizations from DiFa paper.

Unifying Motivation

Though the results of modern domain adaptation methods look promising, there are yet several problems arising even without change of the pretrained generator backbone. For instance, we could not state for sure that full finetuning of the generator provides the best possible results, this procedure might be ineffective too. Nevertheless, one of the most evident problems is the applications to the real-data domain adaptation tasks: non-ideal encoder latents may not relate well with learned offsets and affine transformations from StyleDomain, for instance. Moreover, contemporary encoders tend to work with insides of the generator backbone, so the behaviour of domain adaptation methods may differ significantly. In this work we aim to research the possibilities for encoder improvements in domain adaptation settings.

3 Methodology

During the preparatory stage of the work, several StyleGAN domain adaptation methods (MindTheGap, HyperDomainNet, StyleDomain) were examined. However, as this work would focus on

improvements in interactions between StyleDomain methods and encoders, the literature review part pays more attention to the various directions of the Encoder research, although ideas from other domain adaptation approaches may be worth mentioning in the future too.

Restricting to the first set of ideas on the research, interaction of StyleSpace and Affine+ methods with E4E and FSE would be explored in more detail. Such choice could be motivated by differences in the encoder purpose approaches, framing the overall question of generalization ability of generator. In the following sections, first thoughts on the combinations of these approaches are provided. Nevertheless, pSp and StyleRes Encoders mentioning is justified by such important ideas they brought to the field as: symmetrical encoder architectures, necessity of randomness in the encoder and the different view of interaction between encoder and generator backbone. Careful rethinking of these ideas may lead towards new interpretations of encoders' duty in their interaction with domain adaptation methods.

Experiments setup

In terms of testing the quality of generated images and precisely affecting the training of our encoder, it is important to have a list of factors that determine the performance of various methods. We may utilize common metrics and losses used among various StyleGAN Encoder and domain adaptation publications:

- SSIM, PSNR, MSE to measure image reconstruction
- LPIPS [15] for perceptual quality for Encoders
- Input Identity (ID) metric from pSp paper [12] as the inversion measure for Encoders
- FID [6] to measure the distribution shift for Encoders
- Quality and Diversity metrics from HyperDomainNet [2], based on the CLIP-embeddings cosine-similarities

Talking about human evaluation, there is a list of aspects that we should care about:

- The overall quality of domain adaptation regarding the main object of the image, preservation of basic attributes of this object, i.e. face shape
- Consistency of domain shifts on the flat surfaces, as the evidence of surface understanding
- Image inversion quality, the measure of lost knowledge

- Changes of background details while switching domains
- Consistency of secondary objects on the picture, as our network may separate them from background while still process them incorrectly comparing to the main objects
- Randomness of various features, e.g. hairstyle or eye direction - our models may feel free to change such features even in the restricted real-data editing regime

4 Encoders Comparison

The first step was the comparison of pure Encoder-4-Editing and Feature-Style-Encoder methods in the similar domain adaptation settings. Pretrained E4E was used on inference in StyleDomain paper as one of the most common encoders in the field. By design, it provides latents similar to the original $\mathcal{W}+$ space, so together with StyleSpace method it may be considered as a consistent baseline for the real-data domain adaptation tasks. Moreover, it was trained to predict latent offsets reducing the variance of outputs, in some sense similar to StyleSpace. However, FSE shows much better results on the image inversion tasks, which serves as the evidence of its capability to capture more precise knowledge from images. Contrary to E4E, FSE leans on the interrelation between predicted latents and replaced backbone feature map, which may lead to the shift in the latents distribution. As their interaction with StyleSpace offsets may not produce the desired results, it is important to test the capability of combining this methods. Moreover, such test gives us the insight into the reliance of the StyleSpace offsets on the generator backbone feature maps.

In order to properly compare these two encoder architectures, we have tested three possible utilizations of Feature-Style-Encoder. The first one includes plain usage of predicted styles, just as Encoder-for-Editing does. The second one also substitutes the predicted feature map replacing the output of original generator's 5-th layer. Finally, the third application makes use of generator-shift: in StyleSpace's case of style-codes modification from w to $\tilde{w}$, instead of replacing $G^K(w)$ with F, we are replacing it with $F + G^K(\tilde{w}) - G^K(w)$. By proceeding with our experiments, we frequently face the so-called distortion-editability or quality-editability trade-off, firstly observed in the E4E paper. While the addition of feature-maps and generator-shifts improves the quality of face inversion and preserves main face features better, the particular domain style-changes are less obvious. For instance, we provide a simple inference example in Figures 4.1 and 4.2 on the similar domains adaptation tasks, while examining them much further in the next sections.
It is evident that such features as face shape and and personal traits are preserved better with FSE. However, original hair and face colors are preserved too, which shows less adaptivity towards new

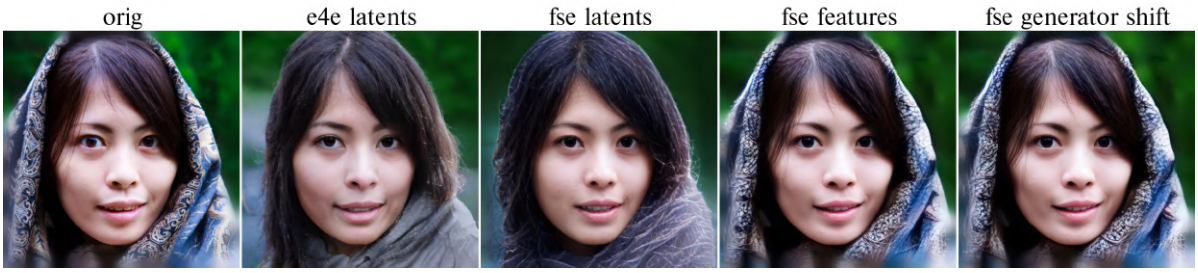

Figure 4.1: Original image and its inversions using: E4E latents, FSE latents, FSE latents and feature map, FSE latents and feature map with applied generator-shift.

Figure 4.2: Original domain image of Murasaki Nora Asuya used for StyleSpace training and domain adaptations applied using: E4E latents, FSE latents, FSE latents and feature map, FSE latents and feature map with applied generator-shift.

domain. Disabling feature extraction in FSE leads to smoothed faces and less consistent make-up, though the model is becoming more capable of editing the hair-color. Latent-based-only methods (second and third columns) tend to work with face-landscapes much more freely. Overall, domain adaptation looses some personal features of the image, for instance changing the eyebrows shape. Looking at the first results of generator-shift usage, its effect is not obvious at all. Comparison between L2-norms of FSE predicted feature map F and $G^K(\tilde{w}) - G^K(w)$ showed that the value of the latter one is 100 times less. Such observation shows that StyleSpace method does not change latent w to $\tilde{w}$ that much for the first layers of generator. Application of generator-shifted FSE to Dissimilar Domains settings will possibly bring much more effect, as objects from dissimilar domains fundamentally differ in their nature. Such differences are regularly embedded into first-layers style codes, therefore StyleSpace should allow massive offsets among them, which will occur in $G^K(\tilde{w}) - G^K(w)$.

Encoders inversion comparison

Intending to understand subtle encoders differences, we have selected a few dozen of photos considering various aspects of face and background reconstructions. In this section we present sample of this comparison in Figure 4.3 and point out the major categories of observed differences. Moreover, their examination may be important for the further enhancement of StyleGAN encoders, as

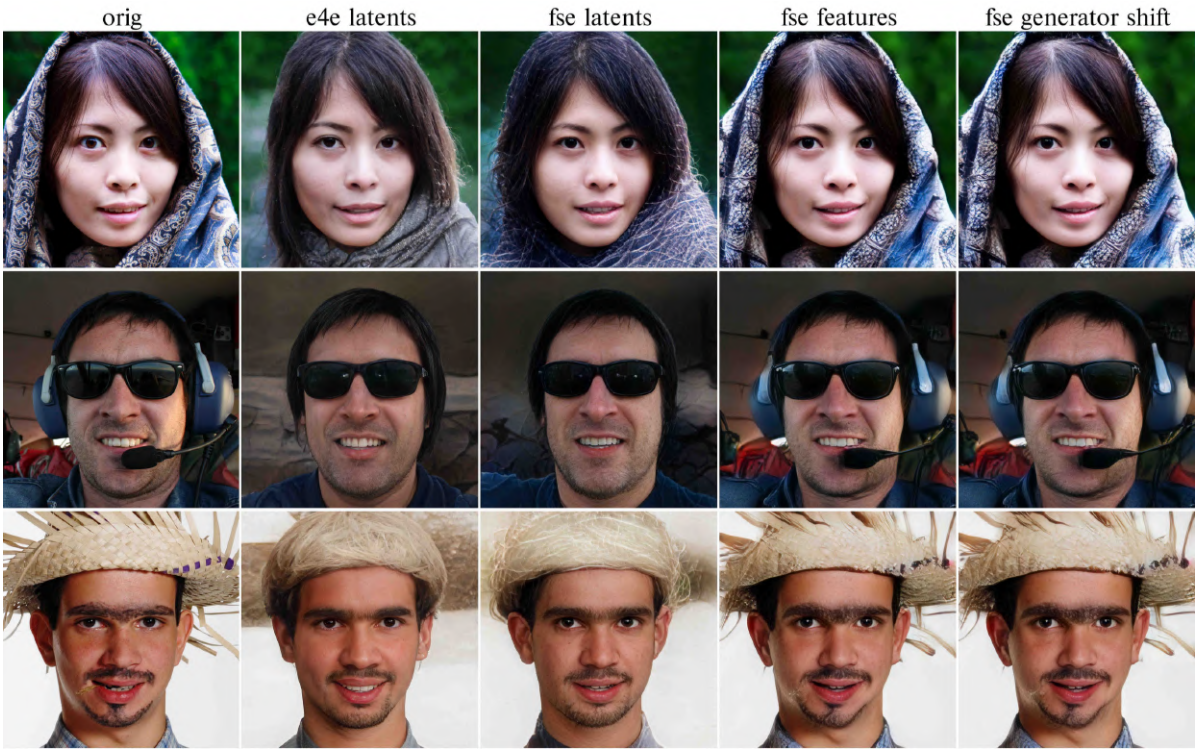

Figure 4.3: Inversion comparison example

these observations may essentially highlight the specifics of human faces perception.

- FSE preserves exterior objects much better than E4E, however it might treat them improperly. For instance, mixing headset and haircut in the inversion, or dealing with the shadow as if it were an extension of the corresponding object.

- Extending the first point, the incorrect hair treatment is one of the most eye-catching disadvantages of E4E. It oversimplifies hair style or mixes hair with external objects, not only changing the background of photo, but transforming the personal features too.

- Humans pay especial attention to the eyes region, precisely perceiving emotions through the eyes depth and facial expressions around them. Though FSE did some improvements, both FSE and E4E lower the expressiveness of human eyes, as even the subtle FSE deviations may result in the different perception of emotions. Nevertheless, FSE recovers original eyelid direction much better, apparently using the backbone feature map for such purpose. To summarize this point in terms of skin reconstruction, we need to pay attention to the consistency of twinkles appearing around eyes and mouth regions. Comparisons shown that as FSE abilities to reconstruct precise skin patterns had grown, under thorough examination it may still lead to the intractable results.

- Pose reconstruction. One of the biggest advantages of FSE besides background objects

reconstruction is its ability to deal with various angles of faces, whereas for E4E it's the common struggle.

- Shades and lighting treatment. While FSE performs much better with shades and even has the surprising ability to distinguish between external lighting and face texture, it still have a tendency to make places with bright external light reflection dimmer. This results into face shapes being slightly flattened under the complicated lighting conditions. It also applies to the aforementioned lack of eye-depth reconstruction, apparently being a common regularization consequence.

Even though we may have criticized some features of FSE inversions, their quality still stays close to ideal. So these observations tend to point out future challenges for the encoders development. On the other hand, they point out our limits in terms of domain adaptation, stating the huge gap between E4E and FSE inversions. As we will try to move towards effective FSE application in domain adaptation tasks, we will tackle the FSE's quality-editability trade-off. At this point we should measure the preservation of typical FSE improvements after the domain adaptation of given images. However, the starting point in such research is the detailed domain adaptation of pure E4E and FSE applications.

Encoders domain adaptation comparison

While in case of pure inversion quality FSE obviously outperforms E4E, in the domain adaptation tasks the situation is reversed. The contemporary domain adaptation methods work on StyleGAN latent spaces, so their influence on the backbone feature maps after the FSE substitution is most likely to be insufficient. In this section we aim to measure this insufficiency, firstly in terms of our own perception, and then by evaluating CLIP-based quality and diversity metrics. Images generated by FSE with generator-shift could be seen as an intermediate stage in the distortion-editability trade-off. It preserves face shape much better than plain latents from E4E and FSE, but transmits strong domain shifts. For instance, persons's chin may be tightened and skin may be smoothed out among several domains, but it would be still possible to recognize the personal traits. An example of domain adaptation is provided in Figure 4.4.

In the domain adaptation settings it may be harder to come up with the unique strategy to list the methods' drawbacks, as the set of domain features is not strictly defined. While somebody may consider face shape preservation to be necessary within the domain adaptation application, others seek for massive domain changes, transforming personal features too. We have tried finding

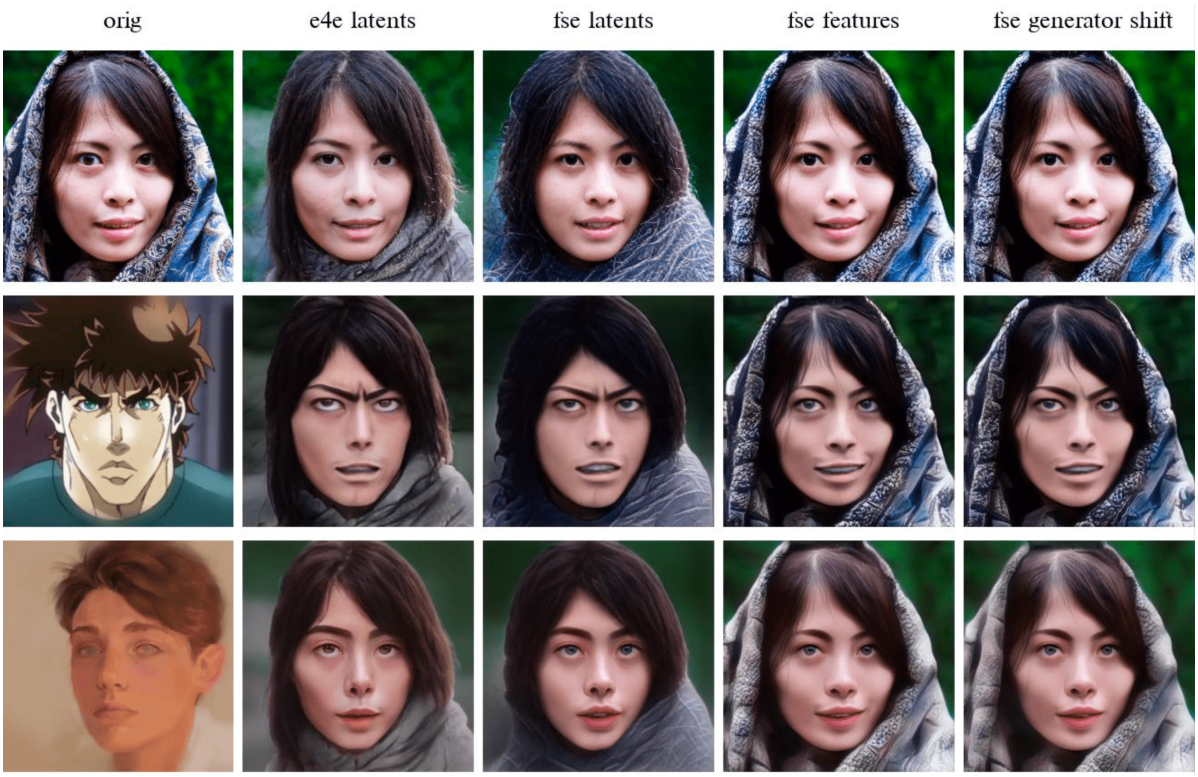

Figure 4.4: Domain adaptation comparison example

balance between such points of view, so here is the list of terms, in which we have compared these methods:

- The original face shape should be preserved, however its texture and depth-landscape may vary over domains. For instance, in the anime domains diverse face undulations should get straightened, transforming into faceted 2d textures. Comparing the effect of StyleDomain on the E4E and FSE inversions, we can state that E4E deals with this point better. However, the E4E inversion itself is smoothened and frequently lies far from the original face.

- Skin color tones should be consistent and ideally change towards the specified domain even stronger than StyleDomain with E4E. On some domains skin color tones tend to be fragile, which is not that critical with E4E application, but with FSE they become clearly discontinuous. On the other hand, sometimes the skin color effect is just different, most likely because latent vectors being turned in the slightly different direction. We will explore latent distributions from other perspective further on.

- Hair and eyes colors - one of the most adaptable features. In the FSE usage examples hair color adaptation is much weaker. We may treat this point as the marker of the degree of the domain adaptation success, as this face feature is consistent and obvious at once.

- The same regarding the eyes color adaptation. However, models tend to even overestimate

the importance of this feature, without paying proper attention to others. Another common problem is the eye-rolling phenomenon, as the domain adaptation methods frequently roll eyelids up.

- The background and exterior objects treatment, especially interaction with headwear. In this case FSE obviously wins E4E and we aim to preserve its performance in the future.

Text-based domain adaptation

As we have discussed in the Literature Review section, the text-based StyleSpace model was trained using only the StyleGAN-NADA [5] direction loss. Text guidance possesses powerful capabilities in terms of domain adaptation of generated images. For instance, easily guided diffusion models have made a big step forwards in terms of flexibility of image generation in the recent years. In the case of GAN models, text-guidance relies on CLIP encoders text-to-image capabilities. On one hand, text-based domain adaptation is able to create a more coherent domain representation lying further from the original image. On the other hand, CLIP text embeddings may misunderstand the text meaning, guiding this representation in not quite correct direction. For instance, the StyleSpace trained with domain text description "Dali painting" besides transforming image colors in the correct way attempts to draw mustache similar to the Dali's one. This clearly is the evidence of misunderstanding of the text description - CLIP text embedding guided the model to make images closer towards Dali's personal appearance, not only towards his painting style. Such occasions happen rarely with other artists' styles in the sampled Figure 4.5 being represented correctly. At least if we would agree that Frida Kahlo made a lot of self-portraits and her face is the first association with "Frida Kahlo Painting" for us too.
Considering the adaptation gap between E4E and different FSE applications, it still stays huge. In the shown domains face shape transformation does not seem to be that necessary, however in such domains as "The Thanos" it is much more preffered. Overall, it could be seen that skin features adaptation is wicker with FSE features too. Besides that FSE still suffers from insufficient background adaptation as it is mostly in the feature map and the common insufficiency in "necessary" face shape transformations.

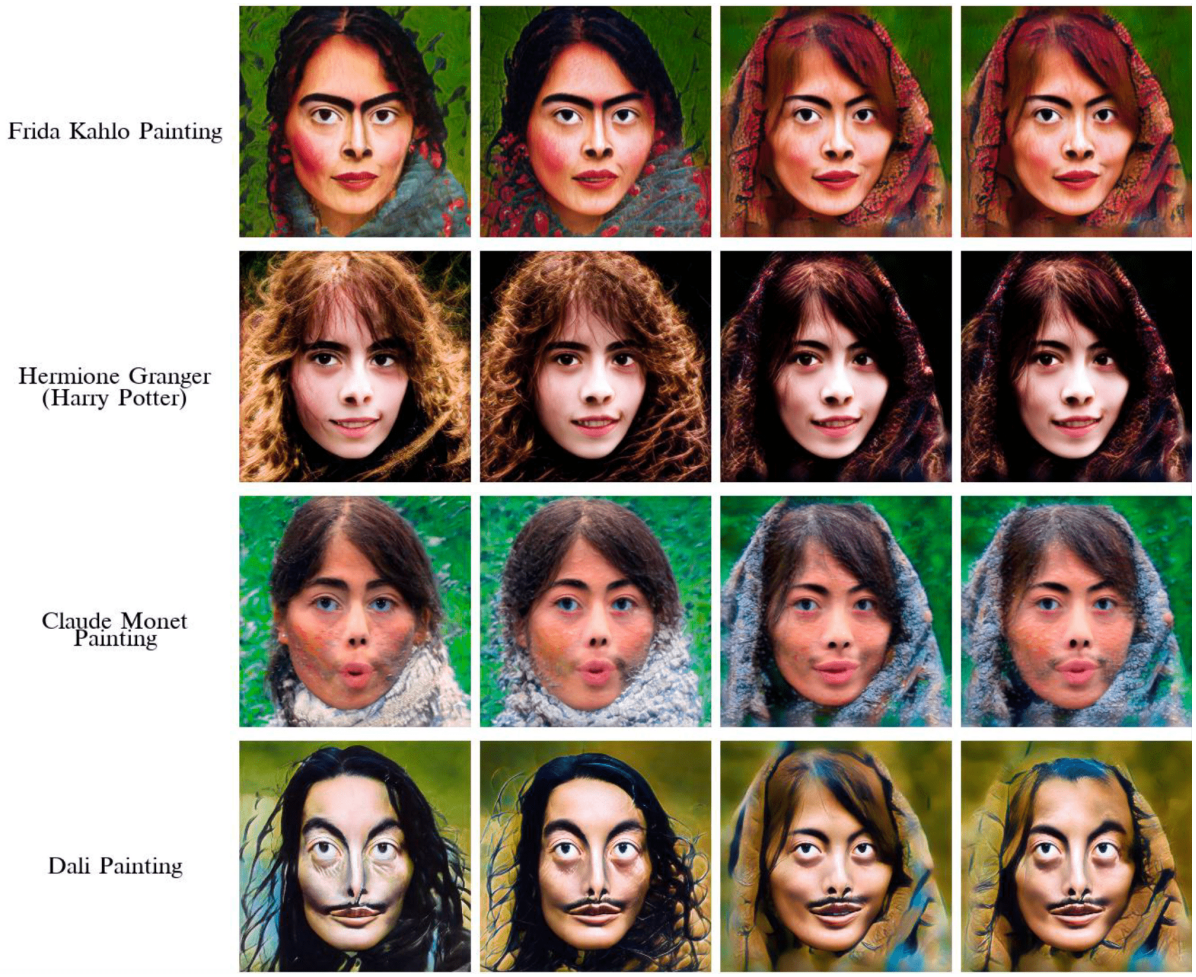

Figure 4.5: Text-based domain adaptation example

Another interesting aspect of text-based domain adaptation was mentioned during the additional celebrities comparison that is not fully included in this work. We took black-and-white photo of Leonardo DiCaprio represented in Figure 4.6 for the domain adaptation comparison, and tested it on every image and text domain. In the case of image-based domain adaptation a common behaviour of StyleSpace is to adapt hair and face colors only, besides necessary face shape transformation. When it comes to text-based domain adaptation, it tends to adapt all colors presented in image towards the required style, as it may be preferred in terms of particular painting style adaptations. However, this observation highlights the fundamental difference in the image and text based domain styles: while the first ones tend to affect image only locally, the second ones usually adapt the image completely. This becomes the evidence of those extended DiFa [16] losses that has been applied only to the image-based domain offsets. Regarding various domains of human faces representations, we may argue that sometimes localization is preferred, sometimes the domain adaptation effect should take the whole picture into account.

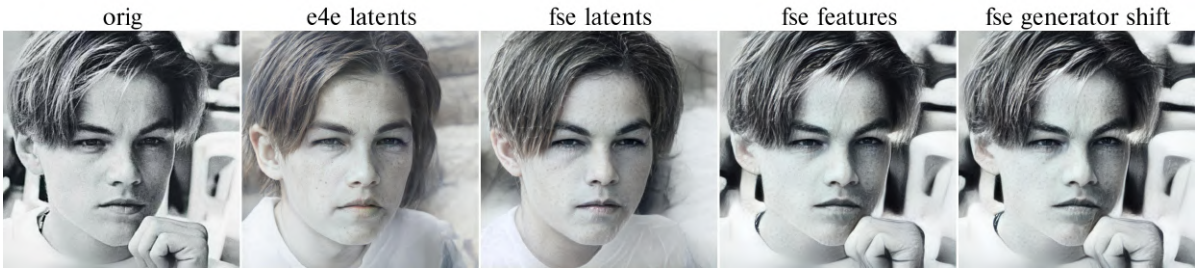

Figure 4.6: Black-and-white Leonardo DiCaprio

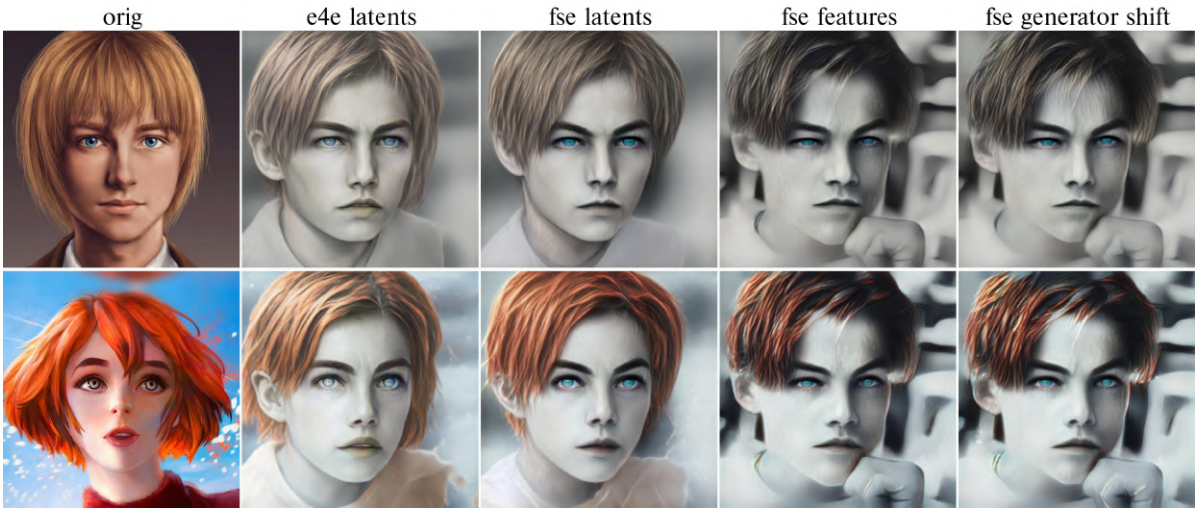

Figure 4.7: DiCaprio adaptation on image domains

Figure 4.8: DiCaprio adaptation on text domains

All together, development of effective collaboration between text descriptions and domain images seems to be the possible outcome. This conclusion implies structured information on preferred localized changes for CLIP encoder i.e. "blue eyes". However, any relative information about the domain image i.e. "the same face color" could not be trained with CLIP text embeddings, so other applications could be researched in the future work.

Quality and Diversity metrics

We have evaluated these 4 methods and the original random faces generated from the StyleGAN-2 Mapping Network latents on the CLIP-based Quality and Diversity metrics. These metrics were firstly proposed in the HyperDomainNet paper, Appendix A.3.3 [2]:

$$Quality = \frac{1}{N}\sum_{i=1}^{N}\langle E_T(domain\_text),\ E_I(image_i)\rangle$$

$$Diversity = \frac{2}{N(N-1)}\sum_{i<j}^{N}\left[1 - \langle E_I(image_i),\ E_I(image_j)\rangle\right]$$

Basically, quality metric represents the average cosine similarity between CLIP image-embeddings of domain adapted images and CLIP text-embedding of the domain description. And diversity - the degree of cosine similarities between CLIP image-embeddings of domain adapted images. In the Table 4.1 we provide the averaged metrics over all domains. It reinforces the CLIP embeddings interpretation abilities in terms of insufficiency of domain adaptation with Feature-Style-Encoder. Therefore, we have used quality metric over each domain in particular to split them into 3 groups: which behave differently with E4E and FSE, similarly or somewhere in the middle.

	Quality		Diversity	
	ViT-B-16	ViT-B-32	ViT-B-16	ViT-B-32
Orig	0.251	0.248	0.217	0.231
E4E latents	0.252	0.249	0.202	0.219
FSE latents	0.250	0.248	0.208	0.225
FSE features	0.237	0.235	0.292	0.320
FSE generator shift	0.244	0.241	0.266	0.289

Table 4.1: Table of averaged Quality and Diversity metrics over all considered Similar Domains for original stochastically generated images and various encoder choices with CLIP ViT-B-16 and ViT-B-32 models respectively. Domain adaptation was performed using StyleDomain StyleSpace latents' offsets.

The Table 4.2 illustrates ViT-B-32 based metrics for image and text domains separately. This table provides us with evidence of the huge gap between the StyleSpace domain adaptation on image and text domains evaluation. We have previously discussed the main reasons for this differences, but its important to note that the metric itself is based on the CLIP embeddings, therefore it behaves slightly differently on the image and text domains too. In terms of domain categorization, this gap suggests selecting 3 groups among image and text domains separately.

	Quality ViT-B-32		Diversity ViT-B-32	
	Image	Text	Image	Text
Orig	0.248	0.313	0.231	0.208
E4E latents	0.249	0.308	0.219	0.202
FSE latents	0.249	0.308	0.225	0.201
FSE features	0.234	0.276	0.320	0.294
FSE generator shift	0.241	0.288	0.289	0.261

Table 4.2: Table of averaged Quality and Diversity metrics using ViT-B-32 CLIP encoder separated for Image and Text Similar Domains. Domain adaptation was performed using StyleDomain StyleSpace latents' offsets.

Examining latent distributions

In this section we will examine the differences between per-channel distributions of E4E and FSE latents, notably comparing them to the original ones that are stochastically generated with the Mapping Network. As StyleSpace was trained on the original random latents, it is important to justify the same behaviour with considered encoders or analyze the occurring deviations.

We have conducted our experiments on the FFHQ small dataset consisting of 3000 photos from the same domain as the original 70k FFHQ. On the presented plots in Figure 4.9 we consider 18 averaged latents of E4E (blue) and FSE (orange) encoders over FFHQ small. It is important to note that here we use latents after learned style affine transforms, so they should be probably named as styles. This smoothed scatterplot represents channel-values for each modulated convolutional layer of the backbone. On the x axis lies the original Mapping Network distribution that has been averaged over 3000 sampled $z \sim \mathcal{N}(0, I)$, while on the y axis lie the distributions of E4E and FSE latents.

Ideally we should observe averaged per-channel values lying along the $y = x$ line, while in fact per-channel distributions of both E4E and FSE are shifted with specific coefficient $y = \alpha \cdot x$ on the majority of layers. We can split modulated layers into three groups:

1 First layers surprisingly have low covariance, especially the 1-st layer (enumerated from 0) demonstrating the independency of FSE distribution from the original one. The application of domain adaptation offsets to this layers may lead to unintended results, occasionally witnessed in FSE applications with generator shift.

2 Middle layers, more precisely layers 5-11, have latents that seem to be well-calibrated both for E4E and FSE. A possible interpretation is that encoders pass the fundamental knowledge of images differently to the first layers, while calibrating it towards the same representations in the middle layers.

3 Upper layers tend to follow the aforementioned pattern: E4E and FSE per-channel values are usually scaled by some constant $y = \alpha \cdot x$. These layers are usually responsible for various color manipulations, so they play the sufficient role in domain adaptation settings.

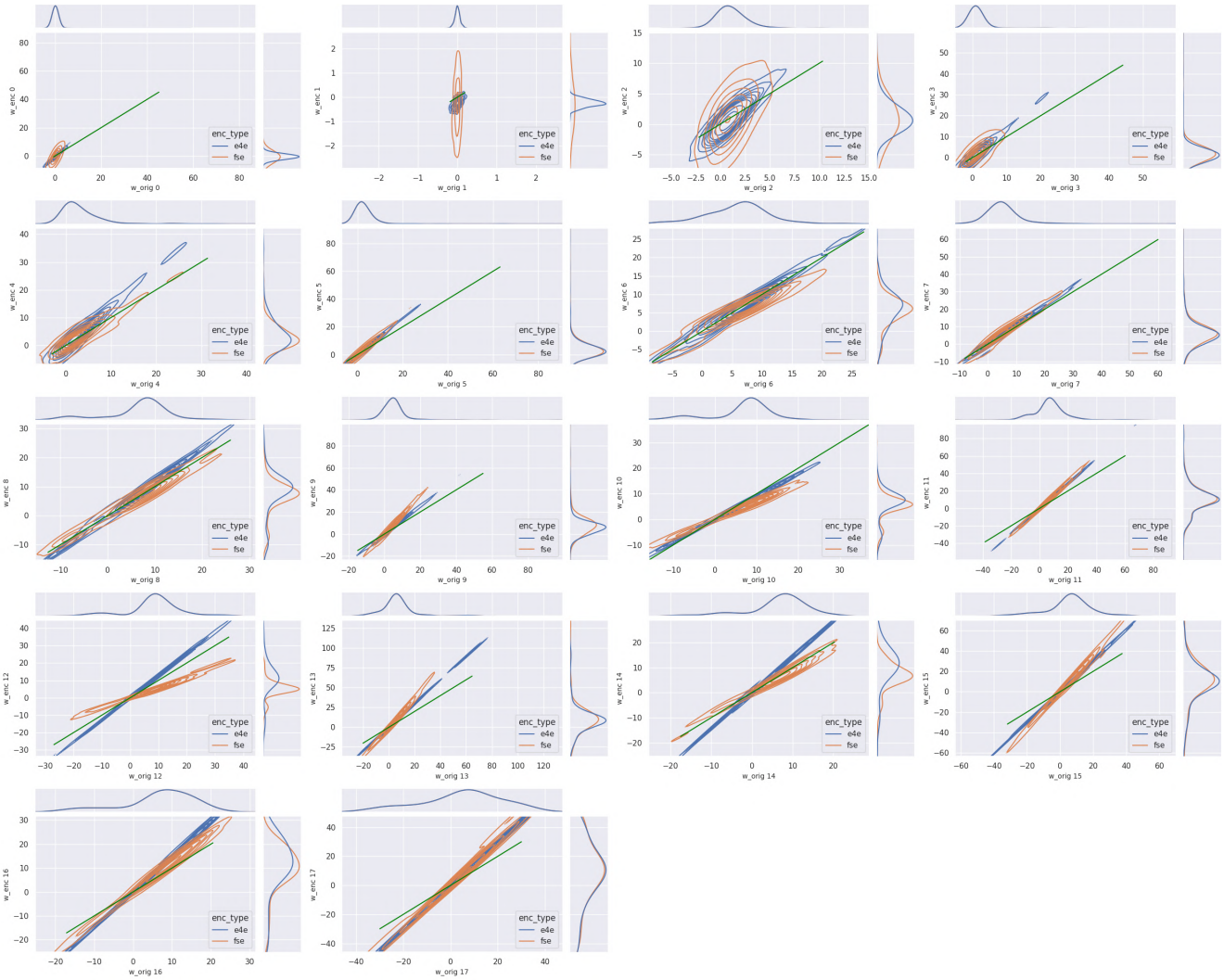

Figure 4.9: Encoders latents per-channel distribution relationship with original random ones. Each subplot shows distributions for the particular modulated StyleGAN-2 convolutional block. The green line represents the ideal distribution of original latents.

To further measure the effect of StyleSpace domain offsets, we have calculated cosine similarities between latents with or without offsets. While taking latents from the original Mapping Network of from different encoders, the final metric takes differences between their cosine similarities. It is designed to measure the difference between domain adaptation effect on each i-th layer with encoder latents or with original ones:

$$EncoderLatentsDomainAdaptability^{(i)} = \langle w_{enc}^{(i)},\ w_{enc}^{(i)} + \Delta w_{domain}^{(i)} \rangle - \langle w_{orig}^{(i)},\ w_{orig}^{(i)} + \Delta w_{domain}^{(i)} \rangle \quad (5)$$

Figure 4.10 presents calculated metric over FFHQ small. The huge difference in metrics on the

0-th layer	stoch_sim: 1.000	fse_sim: 1.000	e4e_sim: 1.000	fse_diff: -0.000	e4e_diff: -0.000
1-th layer	stoch_sim: 0.609	fse_sim: 0.944	e4e_sim: 0.758	fse_diff: 0.334	e4e_diff: 0.148
2-th layer	stoch_sim: 0.996	fse_sim: 0.996	e4e_sim: 0.995	fse_diff: 0.000	e4e_diff: -0.000
3-th layer	stoch_sim: 0.998	fse_sim: 0.999	e4e_sim: 0.999	fse_diff: 0.000	e4e_diff: 0.001
4-th layer	stoch_sim: 0.998	fse_sim: 0.997	e4e_sim: 0.998	fse_diff: -0.001	e4e_diff: 0.000
5-th layer	stoch_sim: 0.999	fse_sim: 0.999	e4e_sim: 0.999	fse_diff: 0.000	e4e_diff: 0.000
6-th layer	stoch_sim: 0.998	fse_sim: 0.996	e4e_sim: 0.997	fse_diff: -0.002	e4e_diff: -0.001
7-th layer	stoch_sim: 0.998	fse_sim: 0.998	e4e_sim: 0.998	fse_diff: -0.000	e4e_diff: -0.000
8-th layer	stoch_sim: 0.998	fse_sim: 0.996	e4e_sim: 0.998	fse_diff: -0.002	e4e_diff: -0.000
9-th layer	stoch_sim: 0.997	fse_sim: 0.999	e4e_sim: 0.998	fse_diff: 0.001	e4e_diff: 0.001
10-th layer	stoch_sim: 0.997	fse_sim: 0.991	e4e_sim: 0.992	fse_diff: -0.006	e4e_diff: -0.005
11-th layer	stoch_sim: 0.998	fse_sim: 0.999	e4e_sim: 0.999	fse_diff: 0.001	e4e_diff: 0.001
12-th layer	stoch_sim: 0.997	fse_sim: 0.984	e4e_sim: 0.997	fse_diff: -0.013	e4e_diff: 0.000
13-th layer	stoch_sim: 0.998	fse_sim: 0.999	e4e_sim: 0.999	fse_diff: 0.001	e4e_diff: 0.001
14-th layer	stoch_sim: 0.994	fse_sim: 0.989	e4e_sim: 0.996	fse_diff: -0.005	e4e_diff: 0.002
15-th layer	stoch_sim: 0.997	fse_sim: 0.999	e4e_sim: 0.998	fse_diff: 0.001	e4e_diff: 0.001
16-th layer	stoch_sim: 0.995	fse_sim: 0.996	e4e_sim: 0.997	fse_diff: 0.001	e4e_diff: 0.001
17-th layer	stoch_sim: 0.997	fse_sim: 0.998	e4e_sim: 0.998	fse_diff: 0.001	e4e_diff: 0.001

Figure 4.10: Cosine Similarities between latents with or without domain offsets per each StyleGAN modulated conv layer. First columns represent the pure cosine similarities for original latents, FSE latents or E4E latents. The last columns represent differences between cosine similarities of encoders' latents comparing to original ones.

first levels supports the corresponding distribution plot - FSE and E4E latents behave strongly differently to the original ones on the first layer. Moreover, 10-th, 12-th and 14-th layers do have higher deviations on the previous plot too.

After noticing such consistent differences between encoders latents distributions, we have tried to diminish their effect by learning unified per-layer coefficients representing offsets power. More specifically, we have learned simple Linear Regression models to determine the aforementioned $\alpha^{(i)}$ per each layer in the $w_{enc}^{(i)} = \alpha^{(i)} \cdot w_{orig}^{(i)}$ hypothesis. As statistics of encoder latents may differ widely between images, $\alpha^{(i)}$ are learned on the mean statistics over FFHQ small. These learned coefficients do not necessarily represent the slopes on the Figure 4.10 as they take low-scale deviations into account too. Assuming the StyleGAN's latent space disentanglement, i.e. that it is unwarped, we further scale domain offsets on each layer with $1/\alpha^{(i)}$. In other words, if encoder's latents on the particular layer tend to be twice as large as the original ones, we double up the offsets power, implying the same domain adaptation direction on the different scale of latent coefficients.

We have tested this hypothesis in the previously considered Similar Domains adaptation settings presenting the results in Figure 4.11. Overall, results could be summarized as follows:

- This simple enhancement frequently improves the quality of domain adaptation on E4E and FSE with feature extraction. Applications with FSE latents-only or with generator-shift may collapse due to the reasons described further on.
- One of its strong sides is the consistency of skin color tone adaptation, which improves significantly over the huge portition of considered domains.

However, domain adaptation with FSE latents and generator-shift may collapse. Note that FSE

with feature extraction does not collapse as generator-shift application does. This suggests that the critical changes happen on the first-layer FSE latents. Therefore, we have tested the offsets enhancements with StyleSpace indomain adaptation, as indomain adaptation does not shift first layer latents. For instance, we may compare the "Modigliani Painting" and "Modigliani Painting (indomain)" rows in the Figure 4.11. Indeed, FSE latents demonstrate the consistency in indomain application settings. Overall, the domain adaptation with FSE becomes consistent without first-layer offsets, though its capabilities in terms of face shape transformations are lowered. This observation is the important evidence of the FSE latent space's complexity and curvature on the corresponding first StyleGAN layers which we have observed in the Figure 4.9.

Figure 4.11: Example of enhanced domain adaptation with learned coefficients for each encoder.

Improving the gap

In this section we present the main contribution of this work aimed to balance the aforementioned gap in domain adaptation quality between E4E and FSE. Similar to the upcoming paper of Denis Bobkov, Vadim Titov, Aibek Alanov, and Dmitry Vetrov that presents StyleFeatureEditor [3], we train the additional encoder to adapt FSE feature tensor similarly to the corresponding $G_5(w_{E4E}) \rightarrow G_5(w_{E4E} + \Delta w_{domain})$. In this paper authors designed FeatureEditor encoder to deal with image editing tasks using editing direction and E4E latents. In our work we have trained similar encoder on the domain adaptation tasks, which implies less local changes and seemingly negotiable measurements of the adaptation effect. Besides implementing this model for the domain adaptation settings, we have came up with several possible calibrations and improvements of this pipeline for such tasks.

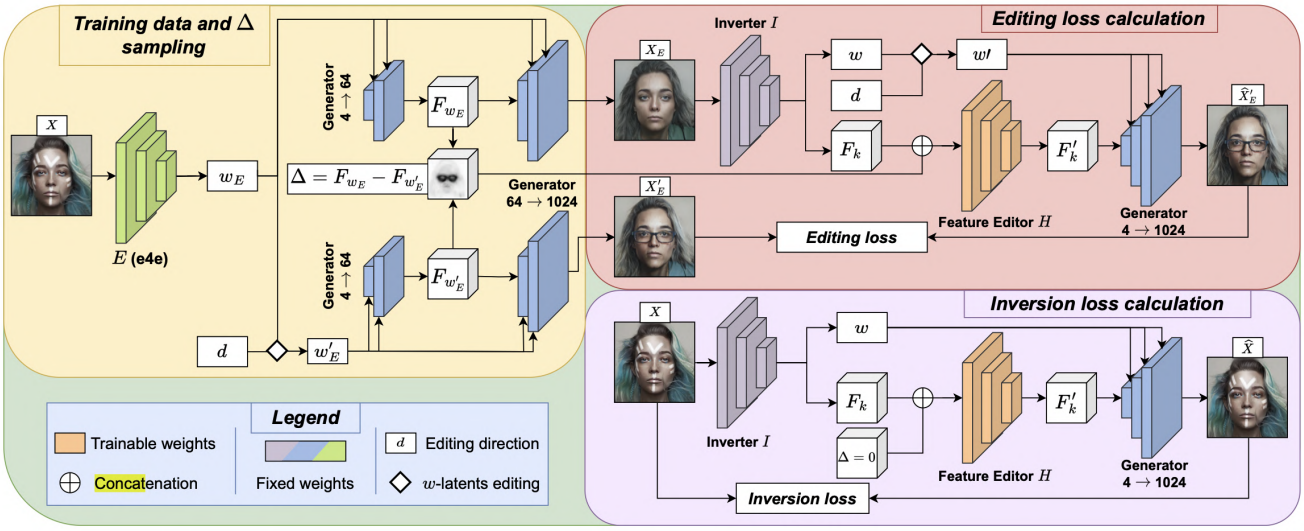

Figure 4.12: Architecture of StyleFeatureEditor from the corresponding article

The architecture constists of 3 parts:

- Applying StyleSpace domain offsets to E4E latents, we inference pretrained StyleGAN-2 generator, selecting $\Delta = F_{w_E} - F_{w'_E}$ from intermediate features of the generator's 5-th layer. This delta represents the plain domain changes in terms of E4E affected feature tensors.
- The E4E-inverted image goes through FeatureStyle Encoder which produces FSE latents and feature-map. Here the FeatureEditor Encoder (in our case named FeatureShift Encoder) is added, which purpose is to transform FSE feature map towards the desired domain, while taking into consideration the concatenated Δ from E4E features. The overall purpose of this part is to train the FeatureEditor Encoder to completely recover the domain adaptation effect on the E4E-inverted image.

- Finally, the third part of the described pipeline includes training FeatureEditor Encoder on the inversion task, which means preserving profound inversion quality of FSE with $\Delta = 0$.

Inference pipeline includes just the first steps with only difference is that we no longer need to double-invert the E4E-inverted image loosing all necessary details. We apply FeatureEditor with E4E domain Δ on the original image inverted with FSE, getting the final domain adapted image.
Dataset: For the training of our model we took 10k images from FFHQ dataset, that have been processed over the first steps of the drawn pipeline. FeatureShift is trained on deltas formed by 20 image and 50 text Similar Domains from StyleDomain paper, supporting all of them at once.
Model: FeatureShift a.k.a FeatureEditor. We have primarly used E4E as the initial encoder that guides the domain adaptation. In one of our side experiments we have replaced original encoder with FSE in the latent-only settings. This was motivated by them showing similar results regarding the metrics, while being more consistent with FSE feature maps. Moreover, in the text-based domain adaptation comparison we have observed reasonably stronger domain adaptation of FSE image latents than E4E ones, meaning that they are better suited for the less-local changes. In the referenced paper authors pretrain the Inverter too, finding out that predicting 9-th layer generator features suits the editing tasks setup better. In our case the Inverter is the pretrained FSE, so we have not conducted such experiments, working only with 5-th layer features.
FeatureShift is the iResnet model consisting of 6-layers, that transforms channel-concatenated tensors of Features and Domain Delta having 1024 channel all together. Each 2 blocks of iResnet lower the channel number $1024 \rightarrow 768 \rightarrow 512$. Our impementation does not restrict FeatureShift from being applied on different layers of generator backbone, so possible replacements of FeaturStyleEncoder as the Inverter could be considered in the future.
Losses:

- **Editing a.k.a domain loss** pulls the model towards the e4e domain inversions
  - MSE loss with $\lambda_{MSE} = 1.0$
  - LPIPS loss on 256 and 128 resolutions with $\lambda_{LPIPS} = 0.8$
  - ID loss with $\lambda_{ID} = 0.1$
  - Adversarial loss on the trainable domain adaptation discriminator haven't shown better results in the editing tasks, but our implementation supports this option too.
- **Inversion loss** pulls the model towards the original image reconstruction
  - MSE loss with $\lambda_{MSE} = 1.0$

  - LPIPS loss on 256 and 128 resolutions with $\lambda_{LPIPS} = 0.8$
  - ID loss with $\lambda_{ID} = 0.1$
  - Adversarial loss with $\lambda_{Adv} = 0.01$. Here the original StyleGAN discriminator stays frozen and is applied only after the sufficient number of training iterations.

- As one of the possible calibrations due to the wider nature of domain adaptation comparing to editing task we have considered adding **domain inversion loss**. It is applied to the generated domain adapted image and the original ones, encouraging them to be initially similar in terms of face shape. As per-pixel MSE loss only suffers the quality of domain adaptation, we have used only LPIPS and ID with lower coefficients than the previous ones:
  - LPIPS loss on 256 and 128 resolutions with $\lambda_{LPIPS} = 0.2$
  - ID loss with $\lambda_{ID} = 0.025$

So, the overall loss is: $\mathcal{L} = \mathcal{L}_{domain} + \mathcal{L}_{inversion} + \big[optional\big]\ \mathcal{L}_{domain\_inversion}$.

As sometimes training on both domain adaptation and inversion tasks from randomly initialized FeatureShift may crash, we have pretrained it on the inversion task only for the first 10000 iterations which equals to 12 epochs over our dataset.

Experiments

Besides coming up with the main pipeline of training, we have conducted a series of experiments with FeatureShift, testing several hypothesis:

- Does pretraining on the inversion-only task help adapting to various domain deltas? It helped stabilizing training runs, for learning FeatureShift on the small number of domains. However, our main goal was to build unified encoder that is applicable over all 70 domains, and in such settings model's gradients explode after switching from inversion task with $\Delta F = 0$ to various domain-specific ΔF.

- Does domain inversion loss affect the quality-editability trade-off? It seems to be the reasonable mechanism to adapt the quality-editability trade-off in the desired direction.

- Whether applying E4E as an original encoder is sufficient in terms of desired adaptability trade-off, or replacing with FSE brings sufficient desired changes to the final results?

- Is adversarial loss necessary in our setup or its effect is barely distinctible? During training of original StyleFeatureEditor authors observed that adversarial training on editing tasks does not bring remarkable changes to the final quality of the model.

During our first stages of experiments we have also tested regularization on E4E intermediate feature maps both for the domain adaptation and for inversion task and on fse feature maps for the inversion task only. We have experimented with common hyperparameters to some extent, for instance showing that weight-decay may serve as an sufficient regularizer for the non-stable domain training. However, in our case of training on Similar Domains its effect hurts the quality slightly while not being necessary as correctly setuped training procedures succeeded in terms of their robustness.

Results

In order to compare FeatureShift + FSE performance to other encoders, in Figure 4.3 we measured Quality and Diversity metrics over the list of image and text domains that perform differently in terms of the gap between E4E and FSE performance discussed above. Regarding the provided

	Quality ViT-B-32		Diversity ViT-B-32	
	Image	Text	Image	Text
Orig	0.773	0.351	0.241	0.169
E4E latents	0.774	0.347	0.224	0.151
FSE latents	0.756	0.345	0.236	0.152
FSE features	0.640	0.286	0.330	0.286
FSE generator shift	0.682	0.318	0.303	0.208
FeatureShift on E4E	0.700	0.326	0.295	0.201
FeatureShift on FSE	0.729	0.337	0.266	0.170

Table 4.3: Table of averaged Quality and Diversity metrics over "different" Similar Domains in terms of E4E and FSE comparison for original stochastically generated images and various encoder choices including FeatureShift, metrics are measured with CLIP ViT-B-32 encoder model. Domain adaptation was performed using StyleDomain StyleSpace latents' offsets.

metrics, FeatureShift successes being another way to balance out original image reconstruction and domain adaptability. Notably, FeatureShift with FSE latents perform quite differently to FeatureShift with E4E ones. In side-by-side comparison FeatureShift with FSE latents achieves slightly more enjoyable results. However, this metrics does not necessary imply the consistency of generated images. FSE with generator shift is considered to be an another balanced solution, though sometimes facing inconsistencies in the generated images struggling with hair reconstruction in text-based domains for example. Obviously, it does not mean that FeatureShift is protected from such behaviour, in some cases it may inherit common problems from E4E or from FSE too.

That is the main reason for us to propose several regularizers of FeatureShift balancing ability. For each wide category of domains it will be possible to find the appropriate balance if the original implementation inherits problems of E4E or FSE too frequently. Finally, in Figure 4.13 we provide a domain adaptation example, where FeatureShift evidently acts different to E4E and FSE.

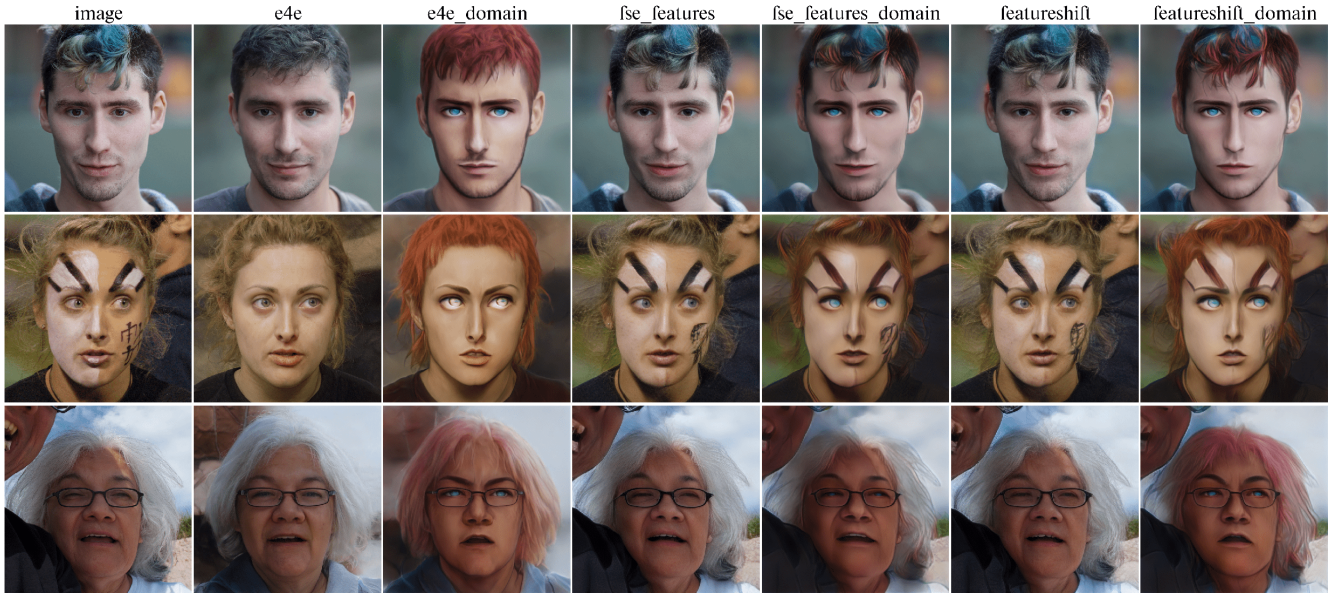

Figure 4.13: FeatureShift comparison with E4E and FSE on the Murasaki Nora Asuya domain

Future work

In this section we provide ideas and their motivation complementing the observations from our work, possible proceedings or more wide-scale concerns about domain adaptation research.

Latent distributions

Treating first-layer latents: The proposed method of offsets enhancements together with analysis of per-channel latents distributions in Figure 4.9 revealed the curvature of first-layer FSE latents. As encoders interaction with StyleGAN generator backbone is becoming more complex, their effect on the first layers that fundamentally build object shape is still restricted by latents only. Therefore, the correct treatment of first-layers latents may be strongly beneficial, especially in the Different Domains settings, where objects' shapes and structures may transform widely. In other words, the further research may be aimed towards the correct enhancements of domain adapted latents, regarding the aforementioned huge differences in their interactions with backbones features comparing to the original StyleGAN latents.

Learning on similar faces only: On the other hand, the proposed offsets enhancements are general, being trained on the per-layer mean latents statistics for the whole FFHQ small dataset. The idea itself suggests that such enhancement coefficients may be found for the particular modes of human faces distribution. In other words, application of clusterization algorithms on the processed latents may help gathering statistics among the corresponding face cluster only.

Latent correlations

One of the hypotheses to test further on is the necessity of reducing variance of offsets in encoder training as it was done in e4e. Some information from images may require significant changes in the particular style codes, especially in terms of domain shifts. For instance, we may encourage encoder or domain adaptation methods to save the correlation between style codes by design, bringing wider opportunities for the knowledge about the domain shift to be stored in them, while preserving the relationship between style codes. This idea could be added as a regularization term to StyleSpace training procedure:

$$\mathcal{L}_{corr} = \sum_i \sum_{j \neq i} \left[(s_i^{new})^T s_j^{new} - (s_i^{old})^T s_j^{old} \right]^2 =$$
$$= \sum_i \sum_{j \neq i} \left[\Delta s_i^T (\Delta s_i + 2 \cdot s_j^{old}) \right]^2 \rightarrow \min_{\Delta s}$$

We may also follow two ways to implement this idea using encoders:

- Training the general encoder, we may force the distribution of correlations between predicted latents to stay the same as average statistics over the original latents, similar to E4E.
- Merging the encoder approach with domain adaptation methods, we may attempt to regularize the correlations in a more precise way - by making use of domain transferring methods such as II2S. Processing new-domain image through the trainable encoder, we regularize outputs' correlations to stay the same, as in the outputs of frozen encoder that is fed with the domain-transferred image. This idea of domain-transferring is also applicable to the StyleSpace offsets regularization. Training on encoder outputs with such regularization will adapt learnable offsets towards the encoder latent distribution.

Extending text-based domain adaptation

We have already mentioned that this is the wide topic in terms of further improvements of existing methods. Such improvements may start from simple replacement of CLIP encoders to their more

robust versions. In particular we frequently face the word order problem with short text-domain descriptions and contemporary CLIP-based algorithms show some improvements in such tasks. As an alternative, we may consider combining several CLIP-directions in our losses, if we would prefer to add longer and more detailed descriptions to text-domains.

Training a unified domain encoder

While we have shown that contemporary domain encoder methods may support a lot of domain adaptation directions at once, there arises the idea of renouncing the particular domain division. Ideally, we would like to guide an image A to look alike an image B in some terms. These terms may be predefined by image similarities or may also be guided in the more profound way by Large Language Model. Here is a possible design of such unified domain encoder:

- Each input image is supposed to have its own domain shift, even though we assign the "main" domain - FFHQ images for instance. We need this main domain definition for the sake of adversarial training of our encoder.
- We may imagine training FSE-like domain encoder which outputs: latents, latent offsets, backbone features and backbone feature offsets.
  - Latents and features (w, F) are trained to adversarially on the main-domain discriminator and being slightly regularized by the inversion loss on the original image. In such settings the encoder will isolate image mapping into the original faces domain.
  - Latent offsets and feature offsets $(\Delta w, \Delta F)$ are trained to inverse the "domain" image itself with $(w + \Delta w, F + \Delta F)$.
  - In order to apply domain adaptation i.e. to make an image A look alike an image B we may replace $(\Delta w_A, \Delta F_A)$ with $(\Delta w_B, \Delta F_B)$ or combine them somehow differently by training an additional part of encoder similarly to FeatureEditor.

Encoder-Generator architectures coherence

One of the important features of pSp Encoder is the U-Net shape of its architecture, compelling us to mention this work not only as a fundamental one for this field. We may consider the overall pipeline of image inversion tasks to be symmetrical, as Encoders are basically trying to extract the same latent features from images that StyleGAN is working on. The general idea lies in transmitting the deep knowledge of generator into the encoder layers, using it as a predefined weights of encoder for training. Such operation could have been done in the following ways:

- Inverting the feature maps on each layer of generator in order to learn the weights of transposed convolutions (similar to convolutions in each StyleGAN block). This approach requires usage of different pretrained encoder, therefore some noise or bias will be added from the unperfect encoder inversion. Renormalization coefficients may also be taken as the average latents of StyleGAN model. Such encoder would produce latent codes as intermediate statistics of its layer activations, while being also trained to match the generator's first layer feature map

- Deconvolutional operation that is defined as the inverse of convolution. It could be done using Fourier Transform, however it imposes non-zero restrictions on the convolution weights, so small shifting of generator feature maps may be required.

The overall idea could be described as follows: making the encoder architecture splitted in the same way as StyleGAN does by storing the sample-specific information in separated latents could benefit building the generalized knowledge in encoders too. Such encoder would gradually disassemble image knowledge into latents, applying StyleGAN generator aftwards to assemble it backwards with necessary changes. This statement seems to meet the initial idea over StyleGAN encoders, however in practice they are mostly developed in the one-sided manner.

5 Conclusion

StyleGAN models are known for their remarkable knowledge of image structure, presenting state-of-the-art precision in image generation tasks. Since the initial discoveries of style-modulation in generator architectures, encoder approaches have been developing for their various applications on the existing-data tasks. On the other hand, research on domain adaptation of StyleGANs have shown significant improvements over the last years with contemporary methods being able to rapidly adapt towards new data domains. In particular, comparing Encoder-for-Editing with Feature-Style-Encoder results, we have analyzed the effect of different encoders on the domain adaptation tasks in terms of distortion-editability trade-off, discovering distinctive strengths for each of three different FSE applications. We have considered several encoder architectures and have examined their combinations with StyleSpace domain adaptation method introduced in the recent StyleDomain paper [1]. During our research of latent distributions, we have proposed a simple domain enhancement method that frequently improves domain adaptation quality even for E4E baseline. We have successfully adapted architecture from the upcoming StyleFeatureEditor paper [3] to the domain adaptation settings, testing several ideas on its proper regularization in

terms of distortion-editability trade-off balance. Trained domain encoder is capable of achieving better results than considered E4E and FSE approaches in a wide number of scenarios, while inheriting some of encoders' problems on the other hand. We have also proposed several hypotheses for the further research on the intersections of domain adaptation and encoders research.

References

[1] Aibek Alanov, Vadim Titov, Maksim Nakhodnov, and Dmitry Vetrov. "StyleDomain: Efficient and Lightweight Parameterizations of StyleGAN for One-shot and Few-shot Domain Adaptation". In: (2023). arXiv: 2212.10229 [cs.CV].

[2] Aibek Alanov, Vadim Titov, and Dmitry Vetrov. "HyperDomainNet: Universal Domain Adaptation for Generative Adversarial Networks". In: (2023). arXiv: 2210.08884 [cs.CV].

[3] Denis Bobkov, Vadim Titov, Aibek Alanov, and Dmitry Vetrov. "The Devil is in the Details: StyleFeatureEditor for Detail-Rich StyleGAN Inversion and High Quality Image Editing". 2024.

[4] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (Oct. 2022), pp. 5962–5979. ISSN: 1939-3539. DOI: 10.1109/tpami.2021.3087709. URL: http://dx.doi.org/10.1109/TPAMI.2021.3087709.

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

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: (2018). arXiv: 1706.08500 [cs.LG].

[7] Xun Huang and Serge Belongie. "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization". In: (2017). arXiv: 1703.06868 [cs.CV].

[8] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Alias-Free Generative Adversarial Networks". In: (2021). arXiv: 2106.12423 [cs.CV].

[9] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: (2019). arXiv: 1812.04948 [cs.NE].

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and Improving the Image Quality of StyleGAN". In: (2020). arXiv: 1912.04958 [cs.CV].

[11] Hamza Pehlivan, Yusuf Dalva, and Aysegul Dundar. "StyleRes: Transforming the Residuals for Real Image Editing with StyleGAN". In: (2022). arXiv: 2212.14359 [cs.CV].

[12] 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: (2021). arXiv: 2008.00951 [cs.CV].

[13] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. "Designing an Encoder for StyleGAN Image Manipulation". In: (2021). arXiv: 2102.02766 [cs.CV].

[14] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. "Feature-Style Encoder for Style-Based GAN Inversion". In: (2022). arXiv: 2202.02183 [cs.CV].

[15] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: (2018). arXiv: 1801.03924 [cs.CV].

[16] Yabo Zhang, Mingshuai Yao, Yuxiang Wei, Zhilong Ji, Jinfeng Bai, and Wangmeng Zuo. "Towards Diverse and Faithful One-shot Adaption of Generative Adversarial Networks". In: (2022). arXiv: 2207.08736 [cs.CV].