Fix the Noise: Disentangling Source Feature for Controllable Domain Translation

Dongyeun Lee[1,2] Jae Young Lee[1] Doyeon Kim[1] Jaehyun Choi[1]
Jaejun Yoo[3] Junmo Kim[1]
[1]KAIST [2]Klleon AI Research [3]UNIST

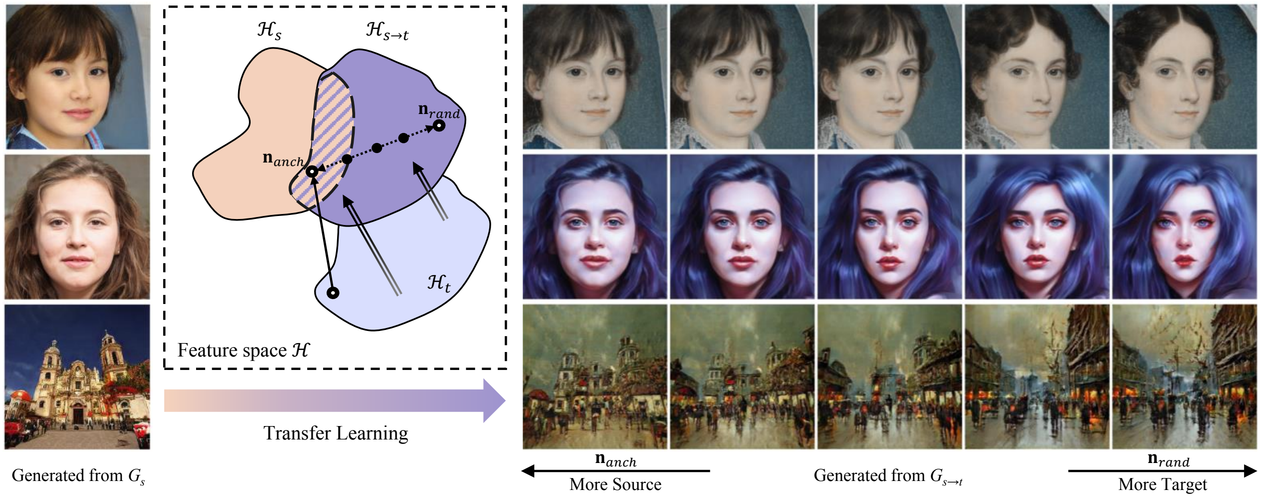


Figure 1. Given a source model G_s, our method can smoothly control the degree of source domain features in a fine-tuned model $G_{s\to t}$. Samples in each row are generated from the same latent code $\mathbf{z} \in \mathcal{Z}$ by G_s and $G_{s\to t}$. Here, $\mathcal{H}_s$, $\mathcal{H}_{s\to t}$, and $\mathcal{H}_t$ denote feature spaces of the source, our model, and simply fine-tuned target model, respectively. Our approach explicitly guides the model to preserve the source features by using the anchor point n_{anch}, which allows a flexible and smooth cross-domain control via $G_{s\to t}$.

Abstract

Recent studies show strong generative performance in domain translation especially by using transfer learning techniques on the unconditional generator. However, the control between different domain features using a single model is still challenging. Existing methods often require additional models, which is computationally demanding and leads to unsatisfactory visual quality. In addition, they have restricted control steps, which prevents a smooth transition. In this paper, we propose a new approach for high-quality domain translation with better controllability. The key idea is to preserve source features within a disentangled subspace of a target feature space. This allows our method to smoothly control the degree to which it preserves source features while generating images from an entirely new domain using only a single model. Our extensive experiments show that the proposed method can produce more consistent and realistic images than previous works and maintain precise controllability over different levels of transformation. The code is available at LeeDongYeun/FixNoise.

1. Introduction

Image translation between different domains is a long-standing problem in computer vision [8, 9, 13, 20, 22, 24, 35, 52, 62]. Controllability in domain translation is important since it allows the users to set the desired properties. Recently, several studies have shown promising results in domain translation using a pre-trained unconditional generator, such as StyleGAN2 [27], and its fine-tuned version [29, 30, 42, 48]. These studies implemented domain translation by embedding an image from the source domain to the latent space of the source model and by providing the obtained latent code into the target model to generate a target domain image. To preserve semantic correspondence between different domains, previous works commonly focused on the hierarchical design of the unconditional generator. They used several techniques like freezing [30] and swapping [42] layers or both [29]. In these approaches, users can control the degree of preserved source features by setting the number of freezing or swapping layers of the target model differently.

However, one of the notable limitations of the previous methods is that they cannot control features across domains in a single model. Imagine morphing between two images x_0 and x_1. Previous methods approximated midpoints between x_0 and x_1 by either building a new hybrid model by converting weights or training a new model. In these approaches, each intermediate point is drawn from the output distribution of different models, which would produce inconsistent results. Moreover, getting an additional model for each intermediate point (image) also increases the computational cost. Another common limitation of these layer-based methods is that their control levels are discrete and restricted to the number of layers, which prevents fine-grain control.

In this paper, we introduce a new training strategy, FixNoise, for cross-domain controllable domain translation. To control features across domains in a single model, we argue that the source features should be preserved but disentangled with the target in the model's inherited space. To this end, we focus on the fact that the noise input of StyleGAN2, which is added after each convolution, expands the functional space composed of the latent code expression. In other words, the feature space could be seen as a set of subspaces corresponding to each random noise. To preserve the source features only to a particular subset of the feature space of the target model, we fix the noise input when applying a simple feature matching loss. The disentangled feature space allows our method to fine-grain control the preserved source features only in a single model without limited control steps through linear interpolation between the fixed and random noise. The extensive experiments demonstrate that our approach can generate more consistent and realistic results than existing methods on cross-domain feature control and also show better performance on domain translation qualitatively and quantitatively.

2. Related work

Domain translation aims to synthesize a target domain image conditioned on a source domain image. Early works [22, 35, 52, 62] successfully solved domain translation by jointly training the encoder for the source and the decoder for the target. Afterward, several works have extended this framework to multi-domain and multi-modal settings [8, 9, 20, 31, 36, 63]. On top of this framework, many studies have been conducted in diverse applications such as style transfer [6, 19, 44–46], cartoonization [7, 28, 32, 39, 49, 53], caricature generation [14, 33, 47], and makeup transfer [5, 11, 16, 23]. However, the joint training framework has a weakness in terms of scalability. Since they train the network according to the source/target setting initially given, the entire framework has to be newly trained if it becomes a different setting, such as adding a new target domain.

Domain translation using unconditional GANs. Recently, several methods [29, 30, 42, 48] have introduced a new approach to domain translation by leveraging a pre-trained unconditional generator, such as StyleGAN2, of the source domain and that of the target domain which is fine-tuned from the source generator. The new framework consists of a two-step approach for domain translation. First, a latent code is obtained by embedding a source domain image to the latent space of the source generator by optimization [1, 2, 10, 27, 38] or encoder [3, 12, 18, 43, 50, 51, 55, 61]. Then, a target domain image is generated by forwarding the given latent code to the target generator. The success of this two-step approach is further explained by the observation of StyleAlign [54] that $\mathcal{W}$ space of the two models is similar, which is in line with the assumption of several methods in the joint training framework mentioned above [20, 35, 36]. The two-step framework only requires different domain generators and the latent inversion method for domain translation. It indicates that there is no need to train the entire framework for different settings. Thus, the framework that utilizes the pre-trained unconditional generator is stable and superior to the joint training framework in terms of scalability.

In the two-step approach, previous methods introduced several techniques to encourage correspondence between images from different domains. Layer-swap [42] generated a target domain image with coarse spatial characteristics of the source domain by combining low-resolution layers of the source model and high-resolution layers of the target model. By adjusting the number of layers to be swapped, their method can control the degree of remaining source features. Freeze G [30] obtained a similar effect by freezing weights of initial layers of the generator during transfer learning. UI2I StyleGAN2 [29] froze mapping layers to ensure exactly the same $\mathcal{W}$ space between the source and target models and combined it with Layer-swap. AgileGAN [48] tried to preserve the source domain features by early stopping. Some recent studies [56, 57] introduce an exemplar-based task in a limited data setup. However, the exemplar-based task requires latent optimization that is substantially time-consuming [43] for the entire dataset, which is not practically applicable to large datasets.

3. Method

In StyleGAN2, the model synthesizes an output image x from a latent code $\mathbf{z} \in \mathcal{Z}$ and a noise input $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is expressed as

$$x = G(\mathbf{z}, \mathbf{n}). \tag{1}$$

Given a source domain model G_s, our goal is to train a target domain model $G_{s \to t}$ initialized with the source model weights while preserving the source domain features. In Sec. 3.1, we briefly discuss the space in which to preserve features and introduce a simple but effective feature matching loss. In Sec. 3.2, we propose FixNoise that ensures disentanglement between the two domain features in the feature space of the target model. In Sec. 3.3, we introduce cross-domain feature control using noise interpolation.

3.1. Which feature to preserve?

Remark that StyleGAN2 [27] contains two types of feature spaces: an intermediate feature space that consists of feature convolution layer outputs and an RGB space that consists of RGB outputs transformed from an intermediate feature by tRGB layers. We choose to preserve the intermediate feature space, which is denoted as $\mathcal{H}$, for the following reasons. First, it has recently been found that the feature convolution layers change the most among layers during transfer learning [54]. This observation indicates that the source features mostly vanish in $\mathcal{H}$ which is the output space of the feature convolution layers. Second, matching features of the source and target models in $\mathcal{H}$ enables the subsequent tRGB layers to learn the target distribution. Consequently, preserving $\mathcal{H}$ space when training enables $G_{s\to t}$ to maintain coarse features of the source while learning fine features of the target. Furthermore, images generated from the model trained with such preservation go beyond simple color filtering effects applied on source images.

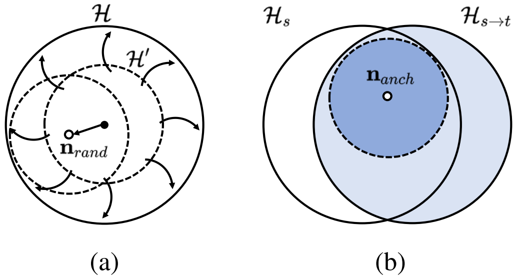
From the same latent code $\mathbf{z} \in \mathcal{Z}$, we encourage the target model $G_{s\to t}$ to have similar features as those of the source model G_s in $\mathcal{H}$ using a simple feature matching loss

$$\mathcal{L}_{fm} = \mathbb{E}_{\mathbf{z}}\left[\frac{1}{L}\sum_{l=0}^{L}\left(G_s^l(\mathbf{z},\mathbf{n}_s) - G_{s\to t}^l(\mathbf{z},\mathbf{n}_{s\to t})\right)^2\right], \quad (2)$$

where L denotes the number of feature convolution layers. Note that $\mathbf{n}_s$ and $\mathbf{n}_{s\to t}$ are independently sampled noise inputs for each model, respectively. Recall that losses that utilize the intermediate features of a network are widely used in GANs literature, such as perceptual loss [24, 59]. However, the main difference between $\mathcal{L}_{fm}$ and the perceptual loss is in which space the features are matched. The perceptual loss encourages the source and target models to have similar features in the feature space of the external network which is unrelated to image generation, whereas our loss encourages them to have similar intermediate features internally. With the loss $\mathcal{L}_{fm}$, we can encourage the target model to have a shared feature space with the source model internally.

3.2. Disentangled feature space using FixNoise

The loss $\mathcal{L}_{fm}$ enforces the entire feature space of the target model $\mathcal{H}_{s\to t}$ to be the same as that of the source model when we naively apply the loss. This may disturb $G_{s\to t}$ to learn diverse target features that do not exist in the source domain. Even if the target features are learned, the degree of preserved source features cannot be controlled if the source and target features are entangled in the feature space of the target model. Instead of applying the loss $\mathcal{L}_{fm}$ to the entire feature space $\mathcal{H}_{s\to t}$, we introduce an effective strategy, FixNoise, that does not disturb target feature learning and allows the different domain features to be disentangled from each other in the feature space of the target model. Our method begins with an assumption that both can be achieved if the source features are mapped in a particular subspace in $\mathcal{H}_{s\to t}$.



(a) (b)

Figure 2. An illustration of FixNoise. (a) The black dot indicates $\mathbf{0}$ noise corresponding to $\mathcal{H}'$. Randomly sampled noise expands $\mathcal{H}'$ to $\mathcal{H}$. (b) Anchored subspace is denoted by a dotted line. Source features are only mapped to the anchored subspace of $\mathcal{H}_{s\to t}$.

As Eq. 1, a final output image is generated from two input components: the latent code $\mathbf{z} \in \mathcal{Z}$, and the per-pixel Gaussian noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ which creates stochastic variation such as curls of hair, eye reflection, and background detail. If the noise $\mathbf{n} = \mathbf{0}$, a feature $\mathbf{h}' \in \mathcal{H}'$ is deterministically generated from a latent code $\mathbf{z}$, where $\mathcal{H}'$ is a subspace of $\mathcal{H}$ corresponding to $\mathbf{n} = \mathbf{0}$. As depicted in Figure 2, when each randomly sampled noise $\mathbf{n}_{rand}$ is added to $\mathbf{h}'$, it shifts $\mathcal{H}'$ to a space that corresponds to each $\mathbf{n}_{rand}$. This shift of subspace by each noise input consequently expands $\mathcal{H}'$ to $\mathcal{H}$. It signifies that the feature space $\mathcal{H}$ consists of subspaces corresponding to each random noise $\mathbf{n}_{rand}$. To ensure that the source features are only mapped to a particular subspace of $\mathcal{H}_{s\to t}$, we fix the noise to a single predefined value when $\mathcal{L}_{fm}$ is applied. By substituting $\mathbf{n}_s$ and $\mathbf{n}_{s\to t}$ to $\mathbf{n}_{anch}$ in Eq. 2, the feature matching loss $\mathcal{L}_{fm}$ with FixNoise is described as

$$\mathcal{L}'_{fm} = \mathbb{E}_{\mathbf{z}}\left[\frac{1}{L}\sum_{l=0}^{L}\left(G_s^l(\mathbf{z},\mathbf{n}_{anch}) - G_{s\to t}^l(\mathbf{z},\mathbf{n}_{anch})\right)^2\right], \quad (3)$$

where $\mathbf{n}_{anch}$ denotes the fixed noise. We refer to the fixed noise as *anchor point* $\mathbf{n}_{anch}$ and the corresponding subspace as *anchored subspace*. $\mathbf{n}_{anch}$ is sampled from the Gaussian distribution same as $\mathbf{n}_{rand}$ and fixed for the whole training process. The anchor point $\mathbf{n}_{anch}$ gives the model explicit guidance to preserve the source feature in $\mathcal{H}_{s\to t}$.

To learn the target features over the entire feature space $\mathcal{H}_{s\to t}$, we use randomly sampled noise $\mathbf{n}_{rand}$ when apply-

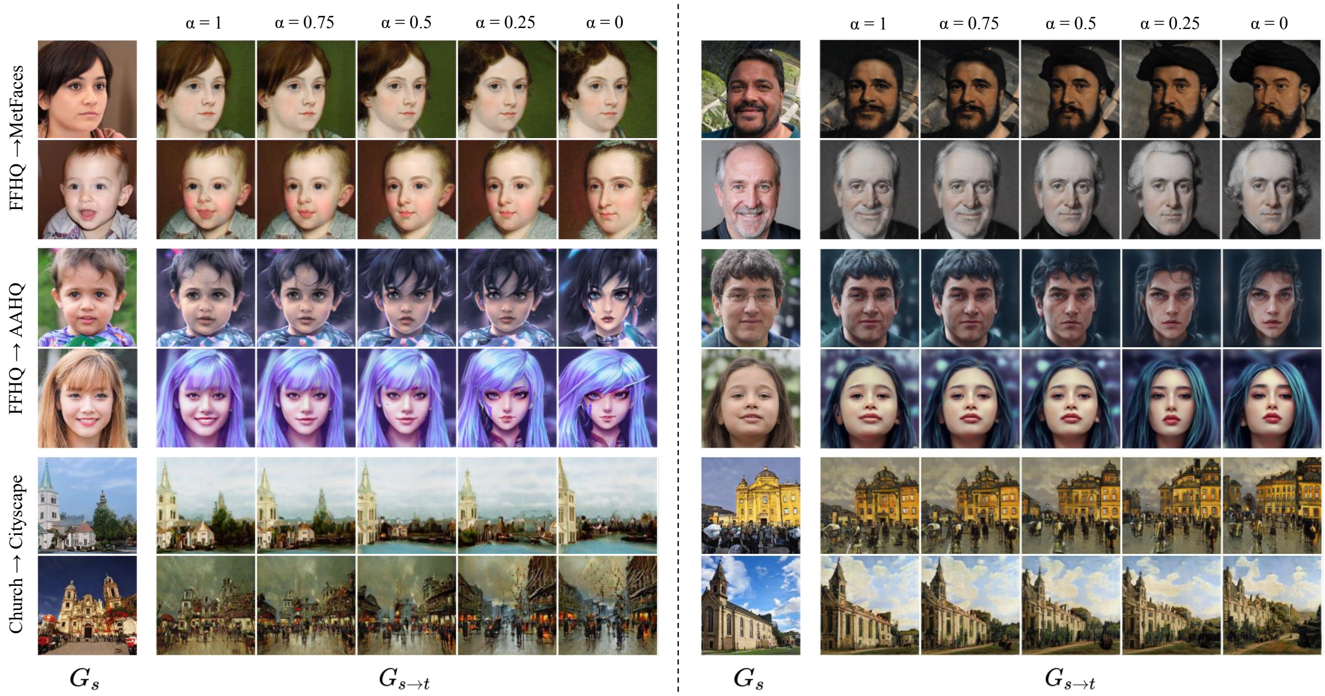


Figure 3. Noise interpolation results on different settings. The interpolation weight α is presented above each column.

ing the adversarial loss [15]

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{z}}\big[-\log D(G_{s\to t}(\mathbf{z}, \mathbf{n}_{rand}))\big], \tag{4}$$

where D denotes a discriminator.

By combining Eq. 3 and Eq. 4, our objective function for $G_{s\to t}$ is described as

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda \mathcal{L}'_{fm}, \tag{5}$$

where λ denotes loss weight. Through this, the source features are only mapped in the anchored subspace by $\mathcal{L}'_{fm}$, while the target features are freely adapted to the entire $\mathcal{H}_{s\to t}$ by $\mathcal{L}_{adv}$. We believe that common features of the two domains are embedded in the anchored subspace, and features that exist only in the target are embedded in the remainder space of $\mathcal{H}_{s\to t}$. To sum up, the disentanglement between the different domain features can be achieved in $\mathcal{H}_{s\to t}$ by the anchor point.

3.3. Cross-domain feature control

As described in Sec. 3.2, we achieved disentanglement between the two domains within the feature space of the target model. To be specific, we preserve the source features only in the anchored subspace of $\mathcal{H}_{s\to t}$ that corresponds to the fixed noise $\mathbf{n}_{anch}$. On the other hand, the target features are learned to the entire $\mathcal{H}_{s\to t}$ that corresponds to all random noise $\mathbf{n}_{rand}$. What should be noted here is that only noise input disentangles the two domain features in $\mathcal{H}_{s\to t}$. This enables a smooth transition between images by linear interpolation of the anchor point and random noise:

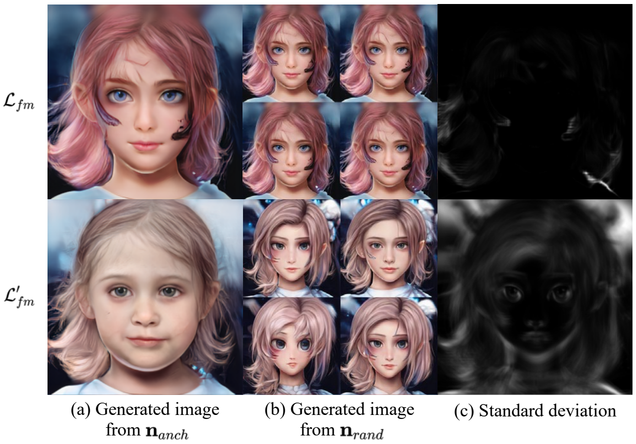
$$\mathbf{n}_{interp} = \alpha \cdot \mathbf{n}_{anch} + (1-\alpha) \cdot \mathbf{n}_{rand}, \tag{6}$$

$$x_{interp} = G_{s\to t}(\mathbf{z}, \mathbf{n}_{interp}), \tag{7}$$

where $\mathbf{n}_{interp}$ is interpolated noise and α represents the interpolation weight. This property is in line with recent work [37] that enables smooth transition across domains by interpolation of latent code. The main assumptions of their work are that the smooth transition by the latent interpolation can be achieved if (i) the margin between the different domains in latent space is small, and (ii) the entire latent distribution is Gaussian. The fact that the fixed and random noise are sampled from Gaussian distribution already satisfies their two assumptions. Thus, our approach, which utilizes Gaussian noise to disentangle different domain features, enables a gradual transition between the two domains as shown in Figure 3.

4. Experiments

Datasets. We conduct experiments for several source and target settings considering the spatial similarity between the source and target domains. For the similar domain setting, we transfer FFHQ [26] to MetFaces [25] and AAHQ [34]. For the distant domain setting, we transfer LSUN Church [58] to WikiArt Cityscape [21]. All experiments are conducted on 256 × 256 resolution images.



(a) Generated image from $\mathbf{n}_{anch}$ (b) Generated image from $\mathbf{n}_{rand}$ (c) Standard deviation

Figure 4. Visualizing the effect of the noise input when FixNoise is applied. Images in the same training setting are generated from the same latent code $\mathbf{z}$. The first and second row correspond to results from G_t trained by applying $\mathcal{L}_{fm}$ (Eq. 2) and $\mathcal{L}'_{fm}$ (Eq. 3), respectively. (a) Generated images from the anchor point $\mathbf{n}_{anch}$. (b) Generated images from the random noise $\mathbf{n}_{rand}$. Varying the noise has global effects such as identity or structure when FixNoise is applied (zoom-in is recommended). (c) Standard deviation of each pixel over 100 different random noise inputs. The FixNoise strategy makes the noise affect images more coarsely.

Implemention detail. We build upon the base configuration in the official Pytorch [41] implementation of StyleGAN2-ADA[1] [25]. Official pre-trained weights for the source model and discriminator trained on FFHQ[2] and LSUN Church[3] are used. We use a non-saturating adversarial loss [15] for $\mathcal{L}_{adv}$ and set $\lambda = 0.05$ for all experiments. Following StyleGAN2 [27], we additionally use style mixing regularization [26] and path length regularization [27] with $\mathcal{L}_{total}$. The discriminator objective function follows StyleGAN2-ADA. Like $G_{s\to t}$, the discriminator D is also initialized with the weights of the source model's discriminator. Adaptive discriminator augmentation [25] is used to prevent discriminator overfitting. The batch size is set to 64. FFHQ → MetFaces, FFHQ → AAHQ, and LSUN Church → WikiArt Cityscape are trained for 2000K, 12000K, and 5000K images, respectively.

4.1. Analysis of FixNoise

Effect of the noise input. In the original StyleGAN model [26, 27], the latent code affects global aspects such as identity and pose, whereas the noise input affects inconsequential stochastic variation (*e.g.* curls of hair, eye reflection, and background detail) which is a localized effect. However, when we apply FixNoise during transfer learning of the model, we find that the noise gives more diverse effects to the images. Figure 4 shows how FixNoise changes the effect of the noise input on the generated images. We can observe that the noise input also affects global aspects when FixNoise is applied, whereas, without FixNoise, the noise only affects stochastic aspects. The observation is due to the discrepancy between the source and target domains. The discrepancy between different domain features includes not only the local but also global aspects. As mentioned above, the source and target features are disentangled in the feature space $\mathcal{H}_{s\to t}$, and the only factor that disentangles the two domain features is the noise input. Thus, the noise input, which is responsible for the disentanglement between two different domain features, is given the role to control some global aspects.

Source	FFHQ				Church	
Target	MetFaces		AAHQ		Cityscape	
α	LPIPS	FID	LPIPS	FID	LPIPS	FID
1	**0.412**	40.37	**0.316**	31.65	**0.521**	27.64
0.75	0.432	37.59	0.366	22.70	0.557	20.59
0.5	0.451	30.17	0.381	14.60	0.626	17.37
0.25	0.481	23.27	0.410	13.65	0.653	12.53
0	0.536	**19.68**	0.510	**5.10**	0.679	**11.49**

Table 1. Quantitative comparison with different interpolation weights. We report the best FID, and measure LPIPS using the same network snapshot.

Noise interpolation. We evaluate our effectiveness on cross-domain feature control in different training settings. As shown in Figure 3, the source domain features are well preserved in the images generated from the anchored subspace ($\alpha = 1$), whereas they are lost in the rest of space in $\mathcal{H}_{s\to t}$ ($\alpha = 0$). This indicates that FixNoise successfully disentangles the source and target features in $\mathcal{H}_{s\to t}$. The fact that the features of both domains are embedded in a single space $\mathcal{H}_{s\to t}$ enables a smooth transition between the source and target features through interpolation between the anchor point $\mathbf{n}_{anch}$ and other randomly sampled noise $\mathbf{n}_{rand}$. This allows us to control the degree of preserved source features. Further, we quantitatively examine the effects of the noise interpolation. LPIPS [59] and FID [17] are used to capture distance with the source and target distribution, respectively. For LPIPS, we randomly sample 2000 latent codes $\mathbf{z} \in \mathcal{Z}$ and measure the distance between images generated by G_s and $G_{s\to t}$ from the same $\mathbf{z}$. We use target domain images for FID measurement. Low LPIPS indicates that the generated images are similar to the source images, while low FID indicates that the distribution of the generated images is close to the target data distribution. As shown

[1] https://github.com/NVlabs/stylegan2-ada-pytorch

[2] http://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorch/pretrained/transfer-learning-source-nets/ffhq-res256-mirror-paper256-noaug.pkl

[3] http://nvlabs-fi-cdn.nvidia.com/stylegan2/networks/stylegan2-church-config-f.pkl

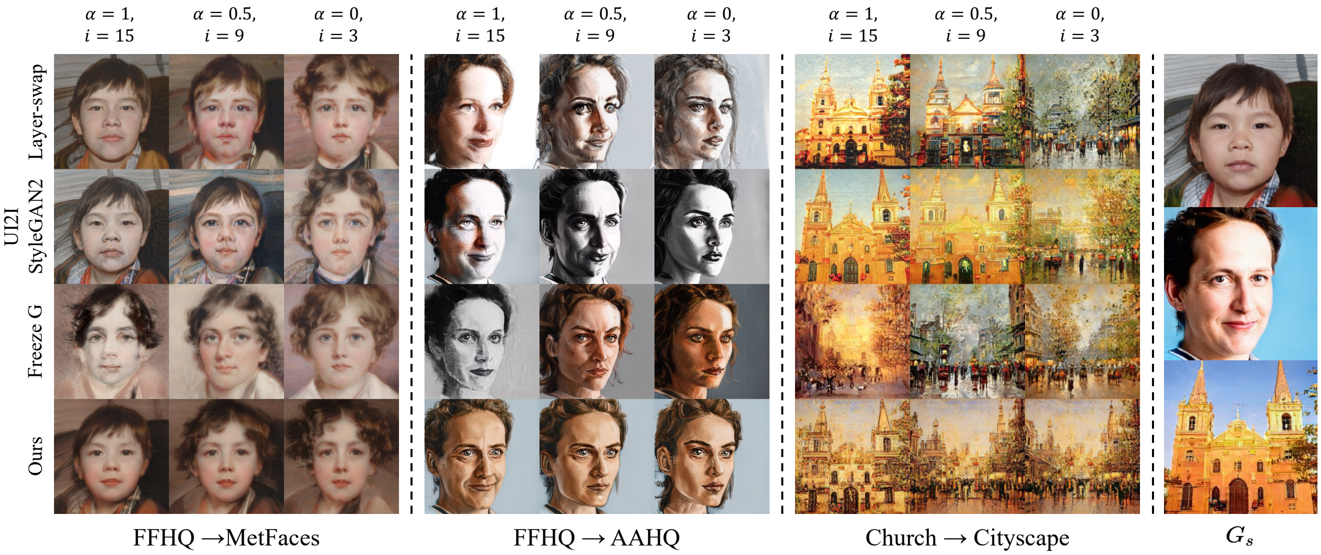


Figure 5. Qualitative comparison on controlling preserved source features. Our results show a consistent transition between the source and target features.

Source	FFHQ						Church		
Target	MetFaces			AAHQ			Cityscape		
	PS	FID	KID $(\times 10^3)$	PS	FID	KID $(\times 10^3)$	PS	FID	KID $(\times 10^3)$
Layer-swap	0.641	68.31	34.69	0.574	38.03	28.30	0.604	52.02	38.46
UI2I StyleGAN2	0.649	79.54	45.38	0.594	51.10	40.77	0.596	64.49	50.03
Freeze G	0.496	24.12	5.41	0.465	7.93	2.82	0.446	12.84	3.55
Ours ($\alpha = 1$)	**0.828**	40.37	14.88	**0.835**	31.65	22.86	**0.709**	27.64	16.28
Ours ($\alpha = 0$)		**19.68**	**3.31**		**5.10**	**1.55**		**11.49**	**3.03**
StyleGAN2-ADA	✗	19.04	2.74	✗	4.32	1.22	✗	11.04	2.75

Table 2. Quantitative comparison with unconditional GANs based methods. We report the best FID, and measure PS and KID using the same network snapshot.

in Table 1, our method obtained the lowest FID and highest LPIPS when $\alpha = 0$, and vice versa when $\alpha = 1$. Thus, we could infer that the generated images lose their source features and approach the target distribution as α decreases. The result demonstrates that our method can control features across domains just by modifying the noise weight.

4.2. Comparison

We evaluate the proposed method with two different approaches: unconditional GANs based methods and conventional domain translation methods. The detailed evaluation metrics are described in the supplemental material.

Comparison with unconditional GANs based method. We compare our approach with unconditional GANs based approaches for domain translation including Freeze G [30], Layer-swap [42] and UI2I StyleGAN2 [29] that combines freeze FC with Layer-swap. These methods including ours have constraints on source feature preservation. In order to explore how the constraints of each method affect target distribution learning, we additionally train StyleGAN2-ADA [25] under the same source and target settings. Note that StyleGAN2 has 21 layers including constant input for 256×256 resolution images. When freezing or swapping layer $i = 0$, $G_{s\rightarrow t}$ is fine-tuned without any constraints, and when $i = 21$, $G_{s\rightarrow t}$ is mere G_s.

The qualitative comparison with previous leading methods is shown in Figure 5. For a qualitative comparison, we use the interpolation weight $\alpha = 1, 0.5, 0$ for ours, and $i = 15, 9, 3$ for baselines to match a similar preservation level, respectively. In FFHQ → MetFaces and FFHQ → AAHQ settings, inconsistent transitions occur in the baselines. In particular, changes in human identity are observable in the results of Layer-swap. Several unnatural color transitions are observed in Layer-swap and UI2I StyleGAN2 due to

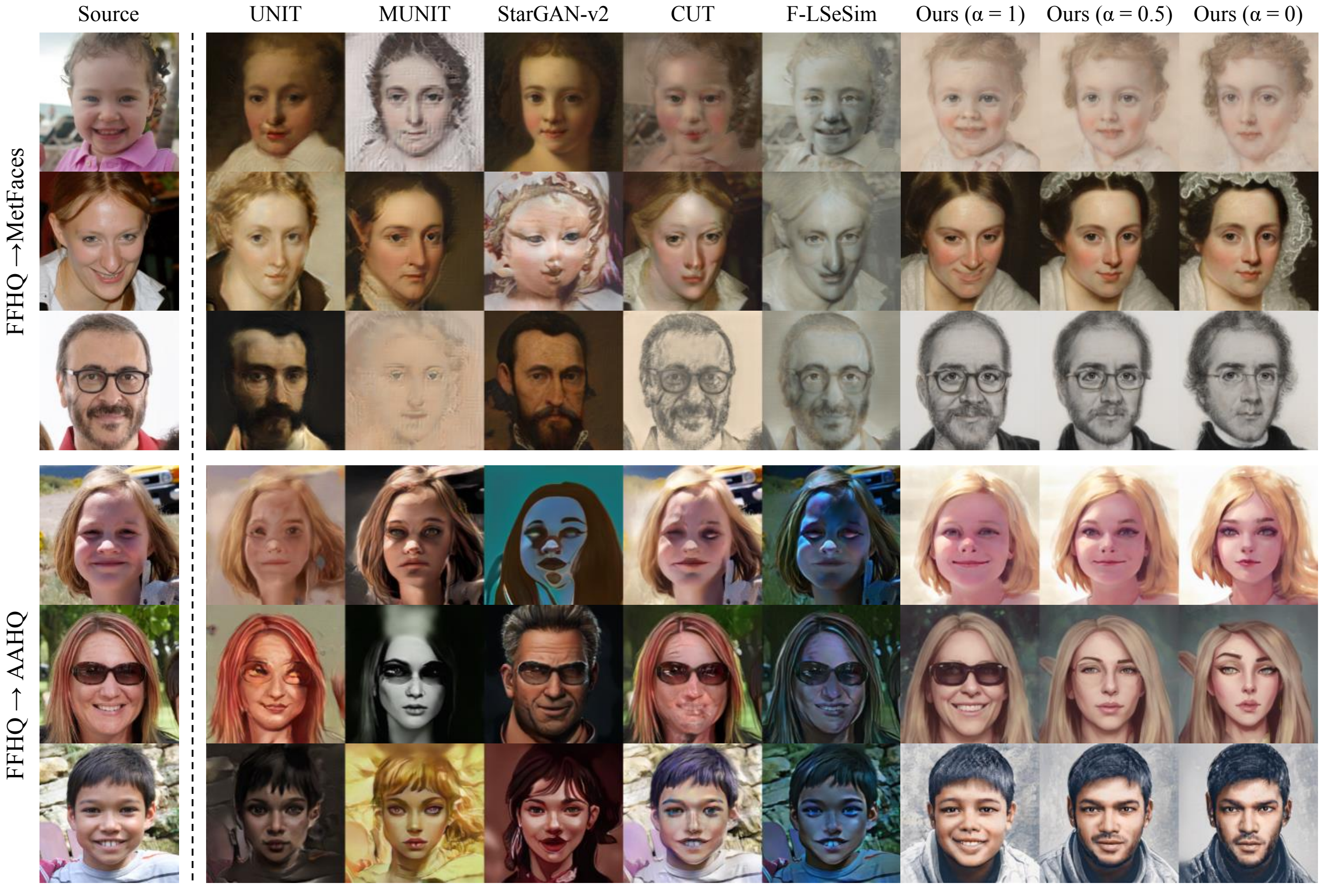


Figure 6. Qualitative comparison with domain translation methods.

simply combining two different models. Although the inconsistency and color transition problems are less important in Church → Cityscape, the feature control steps in the baselines are restricted to the number of layers, which interferes with a fine-grain transition. In addition, to control the source features, previous methods require new models by swapping layers or training, which is not suitable for practical application. In contrast, our method that generates the most realistic results enables smooth transition in a single model, which is easily applicable to diverse tasks.

The quantitative comparison is shown in Table 2. We use a modified version of Perceptual Smoothness (PS) [37] to measure the smoothness of interpolation between different domain features. FID [17] and KID [4] are adopted to evaluate generation quality and diversity. For PS, we use the interpolation weight $\alpha = 1, 0.75, 0.5, 0.25, 0$ for ours, and $i = 15, 12, 9, 6, 3$ for baselines to get interpolated images from the same $\mathbf{z} \in \mathcal{Z}$, respectively. For FID and KID, we use $i = 15$ for baselines following [29]. The PS of our method notably outperforms the competing methods, which implies that our approach is more precise and consistent in controlling features across domains. Moreover, when $\alpha = 0$, our method achieved the highest FID and KID. Compared to StyleGAN2-ADA which does not have any constraints on source preservation, our method shows similar performance while the other methods show significant performance drops. It indicates that, in contrast to previous methods, our proposed method hardly interferes with the learning of the target distribution. Although Freeze G obtained a better FID and KID than ours when $\alpha = 1$, they require additional models for each control level and show inconsistent results as shown in Figure 5. In short, our approach can produce the most coherent and fine-grain interpolation results while generating the most realistic images.

Comparison with domain translation method. We additionally compare our method to recent domain translation methods including UNIT [35], MUNIT [20], StarGAN-v2 [9], CUT [40], F-LSeSim [60] by combining ours with the inversion method. It has recently been observed that an optimization method to $\mathcal{Z}$ space shows the best performance in domain translation among inversion methods to several spaces (*e.g.* $\mathcal{Z}+, \mathcal{W}$, and $\mathcal{W}+$) [54]. They observed that inversion to $\mathcal{W}$ or $\mathcal{W}+$ space yields good reconstruction, but causes color artifacts to target images due to the

Source	FFHQ			
Target	MetFaces		AAHQ	
	FID	KID $(\times 10^3)$	FID	KID $(\times 10^3)$
UNIT	42.69	22.66	20.12	14.78
MUNIT	93.77	81.73	21.82	16.73
StarGAN-v2	37.61	17.88	19.20	**10.44**
CUT	55.52	34.04	20.29	12.26
F-LSeSim	71.07	47.74	47.10	38.90
Ours ($\alpha = 1$)	53.80	31.45	34.90	23.46
Ours ($\alpha = 0$)	**27.14**	**10.29**	**18.53**	11.75

Table 3. Quantitative comparison with domain translation methods. We report the best FID, and measure KID using the same network snapshot.

changes in mapping function ($\mathcal{Z}$ to $\mathcal{W}$) when training the target model. Thus, following StyleAlign [54], we modify the optimization method from StyleGAN2 [27] to embed source images into $\mathcal{Z}$ space of the source model.

Figure 6 shows the qualitative comparison on the domain translation task. Competing methods except for StarGAN-v2 commonly fail to generate realistic images, and particularly include remarkable artifacts in generated images. Results of StarGAN-v2 show better visual quality than the other competing methods, however, some of them are unnatural and fail to preserve the human identity. Although CUT and F-LSeSim successfully preserve the identity of the source images, they generate source images with simple filtering effects and did not adapt well to the target domain. Compared to the competing methods, our approach generates the most realistic and well-adapted images while preserving the source features. In addition to visual quality, the most notable property of our approach compared to previous methods is that the preserved source features can be controlled in only a single model.

The quantitative comparison is shown in Table 3. We randomly sample 20K images from the source domain and generate a single image from each image. When $\alpha = 1$, our method got higher FID and KID than UNIT and StarGAN-v2. However, since FID and KID measure the distance from the target distribution, it is natural that the FID and KID are high when the source features are strongly preserved. We emphasize that the advantage of our method is that we can control the degree of the preserved source features. When the source features are less preserved ($\alpha = 0$), our method greatly outperforms other competing methods except for KID of FFHQ → AAHQ. Also, compared to StarGAN-v2, which achieved the best performance among competing methods, our method is not only qualitatively good, but also preserves the features of the source images much better.

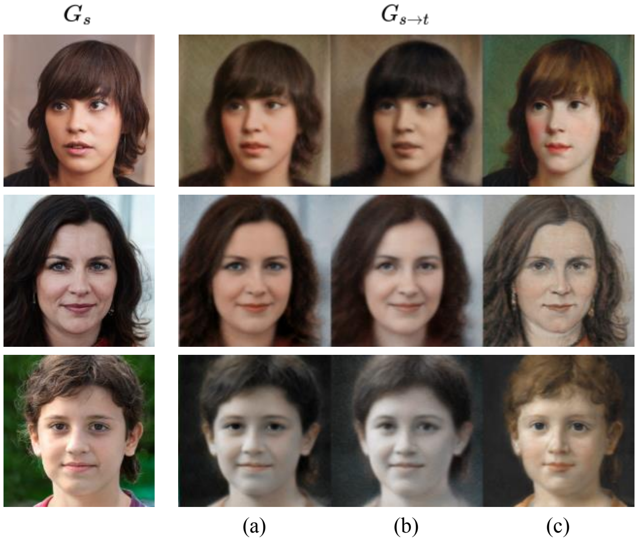


Figure 7. Visualizing the effects of the feature matching loss in different spaces: (a) image space, (b) RGB space, (c) intermediate feature space $\mathcal{H}$ (ours).

4.3. Ablation study on feature matching loss

In FFHQ → MetFaces setting, we study in which space features are appropriate to preserve. We conduct an experiment by applying a feature matching loss in intermediate feature space $\mathcal{H}$ (ours), RGB space, and image space. Figure 7 shows a qualitative comparison on applying the loss in the different spaces. Results of the loss applied in the image and RGB space show well-preserved source features, however, they are not adapted to the target domain. On the other hand, when the loss is applied to the intermediate feature space (ours), a target model $G_{s\to t}$ successfully learns features of the target domain while preserving the source features. The feature matching loss in $\mathcal{H}$ allows tRGB layers to learn the target distribution.

5. Conclusion

In this paper, we proposed a new training strategy, FixNoise, for cross-domain controllable domain translation. By focusing on the fact that the noise input of StyleGAN2 expands the functional space composed of the latent expression, our approach successfully disentangles the source and target features in the feature space of the target model. Consequently, through the noise interpolation, our method can coherently control the degree of the source features in a single model without limited steps. Furthermore, experimental results show that the proposed method remarkably outperforms the previous works in terms of image quality and consistent transition. We believe that our methods can be applied to various fields that utilize multi-domain features. Additional future work and limitations are described in the supplemental material.

References

[1] 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 (ICCV)*, October 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 2

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing, 2021. 2

[4] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 7

[5] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 40–48, 2018. 2

[6] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1897–1906, 2017. 2

[7] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9465–9474, 2018. 2

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 1, 2

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 1, 2, 7

[10] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018. 2

[11] Han Deng, Chu Han, Hongmin Cai, Guoqiang Han, and Shengfeng He. Spatially-invariant style-codes controlled makeup transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6549–6557, 2021. 2

[12] Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 1

[14] Julia Gong, Yannick Hold-Geoffroy, and Jingwan Lu. Autotoon: Automatic geometric warping for face cartoon generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 360–369, 2020. 2

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 4, 5

[16] Qiao Gu, Guanzhi Wang, Mang Tik Chiu, Yu-Wing Tai, and Chi-Keung Tang. Ladn: Local adversarial disentangling network for facial makeup and de-makeup. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10481–10490, 2019. 2

[17] 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. *Advances in neural information processing systems*, 30, 2017. 5, 7

[18] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. *arXiv preprint arXiv:2203.07932*, 2022. 2

[19] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2

[20] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 1, 2, 7

[21] Mohammed Innat. Wiki-art : Visual art encyclopedia. www.kaggle.com/ipythonx/wikiart-gangogh-creating-art-gan, 2020. Accessed Jan. 2022. 4

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1, 2

[23] Wentao Jiang, Si Liu, Chen Gao, Jie Cao, Ran He, Jiashi Feng, and Shuicheng Yan. Psgan: Pose and expression robust spatial-aware gan for customizable makeup transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5194–5202, 2020. 2

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 1, 3

[25] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 4, 5, 6

[26] 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*, pages 4401–4410, 2019. 4, 5

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2, 3, 5, 8

[28] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019. 2

[29] Sam Kwong, Jialu Huang, and Jing Liao. Unsupervised image-to-image translation via pre-trained stylegan2 network. *IEEE Transactions on Multimedia*, 2021. 1, 2, 6, 7

[30] Bryan Lee. Freeze g. `http://github.com/bryandlee/FreezeG`, 2020. Accessed Jan. 2022. 1, 2, 6

[31] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018. 2

[32] Bing Li, Yuanlue Zhu, Yitong Wang, Chia-Wen Lin, Bernard Ghanem, and Linlin Shen. Anigan: Style-guided generative adversarial networks for unsupervised anime face generation. *IEEE Transactions on Multimedia*, 2021. 2

[33] Wenbin Li, Wei Xiong, Haofu Liao, Jing Huo, Yang Gao, and Jiebo Luo. Carigan: Caricature generation through weakly paired adversarial learning. *Neural Networks*, 132:66–74, 2020. 2

[34] Mingcong Liu, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, and Wen Zheng. Blendgan: Implicitly gan blending for arbitrary stylized face generation. *Advances in Neural Information Processing Systems*, 34, 2021. 4

[35] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017. 1, 2, 7

[36] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019. 2

[37] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10785–10794, June 2021. 4, 7

[38] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[39] Ori Nizan and Ayellet Tal. Breaking the cycle-colleagues are all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7860–7869, 2020. 2

[40] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020. 7

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[42] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 1, 2, 6

[43] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 2

[44] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *proceedings of the European conference on computer vision (ECCV)*, pages 698–714, 2018. 2

[45] Falong Shen, Shuicheng Yan, and Gang Zeng. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8061–8069, 2018. 2

[46] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[47] Yichun Shi, Debayan Deb, and Anil K Jain. Warpgan: Automatic caricature generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10762–10771, 2019. 2

[48] 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 (TOG)*, 40(4):1–13, 2021. 1, 2

[49] Hao Su, Jianwei Niu, Xuefeng Liu, Qingfeng Li, Jiahe Cui, and Ji Wan. Mangagan: Unpaired photo-to-manga translation based on the methodology of manga drawing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2611–2619, 2021. 2

[50] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), jul 2021. 2

[51] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[52] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 1, 2

[53] Xinrui Wang and Jinze Yu. Learning to cartoonize using white-box cartoon representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8099, 2020. 2

[54] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned styleGAN models. In *International Conference on Learning Representations*, 2022. 2, 3, 7, 8

[55] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021. 2

[56] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Pastiche master: Exemplar-based high-resolution portrait style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2022. 2

[57] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Vtoonify: Controllable high-resolution portrait video style transfer. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 2

[58] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3, 5

[60] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 7

[61] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1, 2

[63] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017. 2