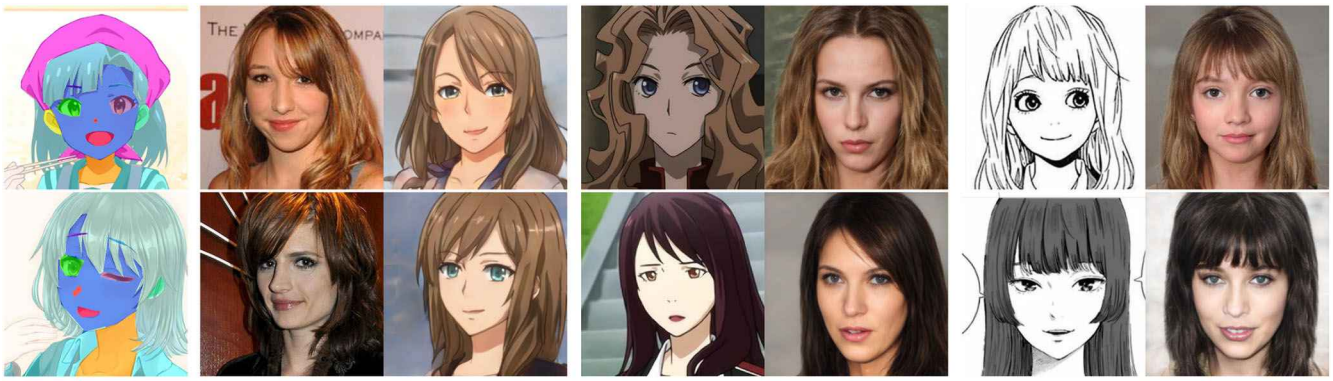
# Parsing-Conditioned Anime Translation: A New Dataset and Method

ZHANSHENG LI, South China University of Technology, China and Singapore Management University, Singapore
YANGYANG XU, The University of Hong Kong, China and South China University of Technology, China
NANXUAN ZHAO, Adobe Research, USA
YANG ZHOU, South China University of Technology, China and Singapore Management University, Singapore
YONGTUO LIU, University of Amsterdam, Netherlands and South China University of Technology, China
DAHUA LIN, The Chinese University of Hong Kong, China
SHENGFENG HE, Singapore Management University, Singapore



(a) Proposed Dataset (b) Portrait-to-Anime (c) Anime-to-Portrait (d) Manga-to-Portrait

Fig. 1. Samples of our Danbooru-Parsing dataset (a), the first anime portrait parsing dataset. This dataset opens up new opportunities for multiple challenging translation tasks to produce high-quality results. We thus use this dataset to train a novel anime translation model, supporting unsolved tasks like portrait-to-anime (b), anime-to-portrait (c), and manga-to-portrait (d).



Anime is an abstract art form that is substantially different from the human portrait, leading to a challenging misaligned image translation problem that is beyond the capability of existing methods. This can be boiled down to a highly ambiguous unconstrained translation between two domains. To this end, we design a new anime translation framework by deriving the prior knowledge of a pre-trained StyleGAN model. We introduce disentangled encoders to separately embed structure and appearance information into the same latent code, governed by four tailored losses. Moreover, we develop a FaceBank aggregation method that leverages the generated data of the StyleGAN, anchoring the prediction to produce in-domain animes. To empower our model and promote the research of anime translation, we propose the first anime portrait parsing dataset, *Danbooru-Parsing*, containing 4,921 densely labeled images across 17 classes. This dataset connects the face semantics with appearances, enabling our new constrained translation setting. We further show the editability of our results, and extend our method to manga images, by generating the first manga parsing pseudo data. Extensive experiments demonstrate the values of our new dataset and method, resulting in the first feasible solution on anime translation.

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Image processing;**

Additional Key Words and Phrases: Generative adversarial networks, image-to-image translation, image editing

**ACM Reference format:**
Zhansheng Li, Yangyang Xu, Nanxuan Zhao, Yang Zhou, Yongtuo Liu, Dahua Lin, and Shengfeng He. 2023. Parsing-Conditioned Anime Translation: A New Dataset and Method. *ACM Trans. Graph.* 42, 3, Article 30 (April 2023), 14 pages.
https://doi.org/10.1145/3585002

Z. Li and Y. Xu contributed equally to this research.
This project is supported by the National Natural Science Foundation of China (No. 61972162); Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2023B1515020097); Guangdong International Science and Technology Cooperation Project (No. 2021A0505030009); Guangdong Natural Science Foundation (No. 2021A1515012625); and Guangzhou Basic and Applied Research Project (No. 202102021074).
Authors' addresses: Z. Li and Y. Zhou, South China University of Technology, Guangzhou, China, 510006 and Singapore Management University, Singapore, 178903; emails: {lzs452552, matrixgle19}@gmail.com; Y. Xu, The University of Hong Kong, Hong Kong, China and South China University of Technology, Guangzhou, China, 510006; email: cnnlstm@gmail.com; N. Zhao, Adobe Research, San Jose, USA, 95110-2704; email: nanxuanzhao@gmail.com; Y. Liu, University of Amsterdam, 1012 CN Amsterdam, Netherlands and South China University of Technology, Guangzhou, China, 510006; email: smanlyt@mail.scut.edu.cn; D. Lin, The Chinese University of Hong Kong, Hong Kong, China; email: dhlin@ie.cuhk.edu.hk; S. He (corresponding author), Singapore Management University, Singapore, 178903; email: shengfenghe@smu.edu.sg.


0730-0301/2023/04-ART30 $15.00
https://doi.org/10.1145/3585002

## 1 INTRODUCTION

Anime, originated from Japanese animation, has gained popularity across the world with its unique style. Many people have grown up with anime and then pass it on to their children, generation by generation. Turning a portrait selfie into an anime character is thus in high demand not only for anime production but also for entertainment, especially for people who want to personalize their own portrait in favorite anime styles. However, a high-quality anime translation needs to capture both accurate anime style and portrait features, which is not an easy task even for experienced designers.

In this paper, we aim to develop an automatic anime translation method, allowing the general public to generate customized anime portraits* in high quality. Since it is hard to obtain a large amount of portrait-anime paired data, we formulate our problem as an unsupervised image-to-image translation task [Huang et al. 2018; Kim et al. 2020; Liu et al. 2017, 2019; Zhu et al. 2017]. This kind of task aims to establish a mapping function between source and target domains, such as from sketch to image, from summer to winter, and from horse to zebra, and so on. Though stunning results have been achieved due to the development of **generative adversarial networks (GANs)**, the results are often limited to appearance-level translation. In addition, since people are sensitive to facial features, the generated artifacts are easily noticeable and the results are less visually pleasing.

As large-scale generative models like StyleGAN [Karras et al. 2019, 2020] evolve, they have shown great potential in high-quality portrait synthesis. Several works are using the prior knowledge of a pre-trained StyleGAN for image-to-image translation on portrait [Huang et al. 2021; Richardson et al. 2021]. In particular, pSp [Richardson et al. 2021] learns to invert the images of the source domain to the latent space of a StyleGAN pre-trained on the target domain. Toonify [Pinkney and Adler 2020] and the concurrent works of UI2I [Huang et al. 2021] and StyleCariGAN [Jang et al. 2021] further train two StyleGANs for two different domains, respectively, and then integrate these two networks to realize image-to-image translation. Overall, because of the rich and diverse prior knowledge encoded in the large-scale generative models, the above attempts achieve stable and high-quality editing and translation performance, compared to training a translation model from scratch. However, anime translation cannot be solved easily with existing frameworks using unconstrained direct mapping (e.g., pSp) or layer swapping (e.g., UI2I and Toonify). This is because of the large domain gap between anime and natural images. Thus, anime translation involves extensive abstraction simplification and large deformation on local structures, which is extremely challenging. On the other hand, rather than simply keeping anime style, original facial features also need to be preserved in our task.

To address the above challenges, we propose a novel framework, called *StyleAnime*, by bridging the domain gap through a parsing map. *StyleAnime* can translate a human portrait into an anime and vice versa. The underlying principle is to convert a source domain image conditioned by its parsing map into a latent code of a StyleGAN pre-trained on the target domain. To support our framework and further promote the research of anime analysis, we contribute the first anime portrait parsing dataset, named as *Danbooru-Parsing*. This dataset contains 1,521 professionally annotated anime portraits over 17 classes of portrait components. A comprehensive quantitative experiment is conducted to evaluate several state-of-the-art methods over these annotated images. The best performed method is selected to augment *Danbooru-Parsing* to 4,921 images for usage. This new dataset not only makes our framework become possible, it can also initiate a new anime face parsing task.

Simply relying on the parsing condition is not enough for translating two domains with such a large gap. To this end, we take two specially designed components for solving this problem. We first introduce a disentangled encoder that discriminatively encodes structure and appearance features from the parsing map and source image, respectively. In this way, the translation can preserve both the input facial features and anime style in the target domain. Our encoder is trained under the self-supervision of the anime images, meanwhile it can be adapted on the portraits. We tailor several losses to handle structure and appearance consistency and the domain adaptation problems. To ensure the quality and editability of the resulted latent codes, we propose a FaceBank aggregation approach to generate and integrate an in-domain codebook of StyleGAN. Extensive experiments demonstrate that our method can produce plausible translation results with appropriate structure deformations and appearance consistency. Furthermore, we extend our dataset and method to several novel applications, including parsing-based anime editing, manga portrait parsing, manga-to-portrait translation, and video translation. Note that our method is based on the pre-trained StyleGAN that trained on the Danbooru Dataset, which includes mostly female anime characters. Thus, all experiments are performed on female anime translation. However, our method is general and extendable to male anime translation by simply adopting a StyleGAN that trained on a gender-balanced dataset.

In summary, our contributions are three-fold:

- We contribute the first anime face parsing dataset, which consists of 4,921 anime-parsing pairs. It provides the explicit guidance of structural deformations. This valuable dataset will benefit not only anime translation tasks, but enable more possibilities for anime analysis.
- We propose the novel disentangled encoders that map the portrait and corresponding parsing map to the latent space of StyleGAN. Several losses are designed to handle color consistency and domain adaptation problems.
- Extensive experiments show that our method is the first feasible anime translation solution. Our method can be easily extended to novel applications including parsing-based editing, appearance-based editing, and manga translation. Besides, our model can also work on videos.

## 2 RELATED WORK

### 2.1 Image-to-Image Translation

**Image-to-image translation (I2I)** aims to translate an image from the source domain to the target domain, and our problem of anime translation can fall in this line of research. Pix2Pix [Isola et al. 2017] designs a conditional GAN for the image to image

*We use *anime* and *anime portrait* interchangeably in the context of our paper.

translation, showing expressive results on diverse tasks, such as from photo to Monet and from winter to summer. Pix2PixHD [Wang et al. 2018] introduces a multi-scale generator and discriminator to synthesize high-resolution images. However, both of these works require paired data as supervision. To explore learning under the unsupervised setting, CycleGAN [Zhu et al. 2017] proposes a cycle-consistent constraint that encourages reconstructing images after a cycle translation. Although CycleGAN [Zhu et al. 2017] can preserve identity well, it cannot deal with the structural deformations (e.g., mouth, nose). UNIT [Liu et al. 2017] learns a joint distribution of different domains based on a shared latent code assumption. MUNIT [Huang et al. 2018] further decomposes the latent space into shared content and independent style space for generating diverse outputs. A similar disentanglement idea also emerges in DRIT [Lee et al. 2018]. Though the above works have gained huge progress on the appearance-level translation, they often fail on the translation tasks with structure gap, such as portrait-to-anime translation. For capturing the most discriminative structure areas, UGATIT [Kim et al. 2020] and AniGAN [Li et al. 2021] design generators that simultaneously transform local shapes and appearance. However, since there is no explicit guidance on the shape transformation, they cannot synthesize the plausible structure transformation results. Instead, we share a similar spirit with disentangled image editing [Lee et al. 2020; Park et al. 2020; Tan et al. 2020; Wu et al. 2021], to disentangle the structure and appearance through encoders by introducing parsing maps for providing explicit guidance on structure deformation. We can translate results consistently on both anime styles and facial features.

## 2.2 Portrait Generation and Stylization

Recently, many generative models are proposed for synthesizing the high-quality images, such as PGGAN [Karras et al. 2018], BigGAN [Brock et al. 2019], and StyleGAN [Karras et al. 2019, 2020]. Thanks to StyleGAN, high-resolution human portraits can be generated in a natural way that cannot be distinguished from real photos easily. Moreover, StyleGAN has a style-based architecture and its layer-wise representation disentangles the structure and appearance information [Karras et al. 2019, 2020]. This feature powers real applications on portrait stylization. Toonify [Pinkney and Adler 2020] and Huang et al. [2021] introduce layer swapping in pre-trained StyleGANs for achieving facial stylization, and they can obtain high-quality stylized results. However, the layer swapping lacks explicit guidance to preserve the visual features of input portraits. An alternative StyleGAN-NADA [Gal et al. 2021] leverages the semantic power of large scale **Contrastive-Language-Image-Pre-training (CLIP)** [Radford et al. 2021] models to shift a generative model to new domains driven by texts. Instead of warping and deforming facial appearance through control points [Cao et al. 2018; Shi et al. 2019], directly generating a portrait with exaggerated structure and style becomes possible [Jang et al. 2021] with the help of StyleGAN. The quality of generated caricature enhances because of the explicit modeling on dense and detailed shape deformation, learned from unpaired photo-caricature data. In addition, Song et al. [2021] and Jang et al. [2021] impose attribute consistency between the input portrait and the stylized one. It is worth noticing that most of the above methods are based on the same assumption that the stylized results share the same latent code with the input portrait. We argue that this is a too strong assumption for our portrait-to-anime translation due to the large domain gap. In contrast, we design disentangled encoders with elaborated losses for producing more plausible and high-quality anime results.


Fig. 2. Analysis of annotation consistency. The first row shows animes that need to be labeled. The second and third rows show annotated parsing maps. The two parsing maps of each anime are annotated by the same person at different times (i.e., a three-month interval). And the last row shows the differences between the two annotations of each anime. The pixel-wise consistency accuracy is shown at the bottom of each example.

# 3 DANBOORU-PARSING DATASET

In this section, we describe the creation of Danbooru-Parsing dataset, which contains around 5k anime-parsing pairs with manual and automatic annotations jointly. To ensure the diversity and quality of our dataset, we choose to select anime images from the largest anime dataset called Danbooru2020 [Anonymous et al. 2021]. Danbooru2020 contains over 4.2 million unprocessed animes, and each anime contains also the body part. Since we only want the portrait part, we first randomly selected 10,000 anime images from Danbooru2020 and then cropped the portraits out by an anime-face detector [Nagadomi and Youkaichao 2014]. After filtering out the monochrome and low-resolution images, we obtained 1,521 valid animes with the resolution of $256 \times 256$ for manual annotation. This filtered dataset covers portraits in diverse poses, facial features, and artistic styles.

## 3.1 Manual Annotation

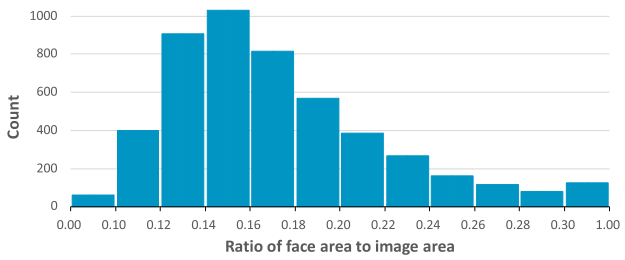
We annotated the selected anime portraits on the Labelbox.[†] We follow CelebaMask-HQ dataset [Lee et al. 2020] by annotating each

[†]https://labelbox.com/product/platform.

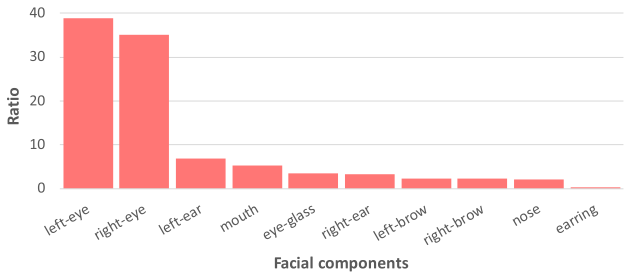
Table 1. Comparison of State-of-the-Art Segmentation Methods on Anime Parsing

| Method | Background | Skin | Left-Brow | Right-Brow | Left-Eye | Right-Eye | Eye-Glass | Left-Ear | Right-Ear | Earring |
|---|---|---|---|---|---|---|---|---|---|---|
| FCNet [Long et al. 2015] | 77.45 | 85.72 | 17.67 | 7.57 | 84.22 | 83.91 | 0.50 | 59.56 | 26.39 | 1.09 |
| EHANet [Luo et al. 2020] | 79.40 | 85.99 | 22.77 | 17.28 | 83.92 | 84.43 | 0.03 | 50.36 | 29.77 | 0.94 |
| BiSeNet [Yu et al. 2018] | 80.04 | 86.09 | 19.50 | 15.98 | 84.73 | 84.81 | 2.15 | **62.46** | 28.18 | **3.48** |
| Deeplabv3 [Chen et al. 2017] | 82.51 | 85.82 | 22.05 | 21.44 | 84.48 | 84.04 | 0.96 | 61.92 | 28.23 | 2.64 |
| Deeplabv3+ [Chen et al. 2018] | 82.63 | **86.56** | **25.94** | **21.91** | 84.74 | 84.47 | 2.41 | 57.69 | 30.76 | 1.37 |
| DANet [Fu et al. 2019] | **83.48** | 86.55 | 22.74 | 19.68 | **85.01** | **85.13** | **4.77** | 57.62 | **31.32** | 1.83 |
| **Method** | **Nose** | **Mouth** | **Neck** | **Necklace** | **Cloth** | **Hair** | **Hat** | | **Overall Acc.** | **mIoU** |
| FCNet [Long et al. 2015] | 46.12 | 70.59 | 65.51 | 19.01 | 61.18 | 86.21 | 40.65 | | 87.75 | 49.02 |
| EHANet [Luo et al. 2020] | 48.50 | 72.07 | 70.41 | 18.27 | 64.92 | 86.91 | 43.62 | | 88.79 | 50.56 |
| BiSeNet [Yu et al. 2018] | 47.16 | 72.11 | 72.83 | 24.03 | 66.08 | 88.14 | 45.95 | | 89.38 | 51.98 |
| Deeplabv3 [Chen et al. 2017] | **50.35** | 71.26 | 74.08 | 26.58 | 72.34 | 88.73 | 53.54 | | 90.44 | 53.59 |
| Deeplabv3+ [Chen et al. 2018] | 49.72 | 71.60 | 74.08 | 26.09 | 72.20 | 89.30 | 55.25 | | 90.68 | 53.92 |
| DANet [Fu et al. 2019] | 45.14 | **72.99** | **74.62** | **31.49** | **75.14** | **89.67** | **55.37** | | **91.10** | **54.27** |

We report pixel-wise accuracy (%) on different component classes and on the whole image (overall). We also report class-wise mean intersection over union (mIoU, %). The best results are marked in **bold**.




(a) Distribution of face area




(b) Distribution of facial components

Fig. 3. Statistics of our Danbooru-Parsing dataset. We show the distribution of face area in (a) and the distribution of facial components in (b).

anime with 17 component classes by the professional experts, including "skin", "nose", "eyes", "eyebrows", "ears", "mouth", "hair" and so on. Note that we also include class "hat" to indicate hair accessories (e.g., hairband and hair ring) as it is a common feature of anime portrait. During the labeling process, we neglected the regions outside of our pre-defined classes.

To examine whether our annotations are reliable and consistent, we follow ADE20K dataset [Zhou et al. 2017] by re-annotating 50 randomly selected images by the same annotator after three months. On average, 93.77% of the pixels got the same labels as the initial annotation, which demonstrates that our annotations have strong consistency. We show four examples with annotation accuracy in Figure 2. The example in the first column has lower consistency due to the omissions and labeling errors in classes 'cloth', 'neck', and 'hat'. The consistencies of the other three examples are higher, with only misalignment on the edges.

## 3.2 Automatic Annotation

Though manually labeling can achieve annotations in the highest accuracy, it is time-consuming and labor-intensive, which limits the scalability of our dataset. In this subsection, we aim to train a parsing model for automatic annotation. To find a sophisticated model, we compare existing segmentation models on our manually annotated samples. More specifically, we divide our annotated dataset into 1,200 and 321 samples for training and testing, respectively. We test a set of state-of-the-art face parsing models, including FCNet [Long et al. 2015], Deeplabv3 [Chen et al. 2017], Deeplabv3+ [Chen et al. 2018], BiSeNet [Yu et al. 2018], DANet [Fu et al. 2019], and EHANet [Luo et al. 2020]. All the tested models are pre-trained on the CelebAMask-HQ dataset [Lee et al. 2020] and fine-tuned on our training split of 1,200 anime-parsing pairs. After training, we measure the pixel-wise accuracy and **mean intersection over union (mIoU)** on the test split. We average accuracy across all pixels, and mIoU across different parsing classes.

The results are shown in Table 1. We can see that all methods can achieve around 50% mIoU and 90% pixel-wise overall accuracy. The earliest work FCNet [Long et al. 2015] gets a 87.75% overall accuracy and a 49.02% mIoU, but cannot detect small anime components effectively, such as right brow. In addition, DANet [Fu et al. 2019] obtains the best results among these methods on both metrics. In this way, we use this well-trained DANet for automatic annotations over 3,400 animes provided by UGATIT [Kim et al. 2020]. We also undergo a second pass on these annotations to correct wrong predictions. Finally, we get 4,921 anime-parsing pairs in total as our Danbooru-Parsing dataset.

## 3.3 Dataset Statistics

To better understand our Danbooru-Parsing dataset, we show some statistics in Figure 3. Figure 3(a) shows the distribution of face area over the whole image, and facial components include 'left-eye', 'right-eye' 'eye-glass', 'left-brow', 'right-brow', 'left-ear', 'right-ear', 'earring', 'nose', 'mouth'. We can see that the majority of the face area falls in the range of (0.1,0.3]. Figure 3(b) shows the distribution of facial components. Note that 'left-eye' and

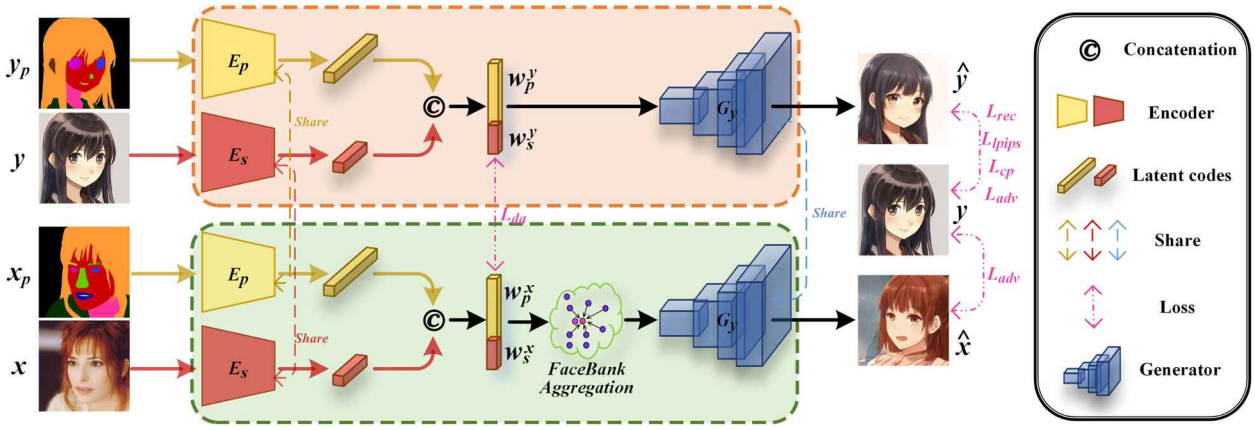



Fig. 4. Overview of our conditional portrait-to-anime translation. We disentangle the structure and appearance by feeding the parsing maps $x_p$, $y_p$ and portraits $x$, $y$ into different encoders $E_p$ and $E_s$, respectively. The generated latent codes control the different layers of a pre-trained StyleGAN generator $G_y$ for a better disentanglement. During training, the latent code is sent into the generator directly, as shown in the orange region. A set of calibrated losses are added to train our model in a self-supervised manner. During testing, as shown in the green region, the latent code undergoes a FaceBank aggregation before decoding to maintain a high-quality anime result.

'right-eye' have higher percentages and 'mouth' has a lower percentage. This is consistent with the characteristics of anime images on big eyes and small mouths.

## 4 STYLEANIME

Given an image $x$ in the source domain $X$ (i.e., portrait), our goal is to learn a mapping function to translate $x$ to its counterpart $y$ in the target domain $Y$ (i.e., anime). The large structural deformation of the translation induces serious artifacts. Previous works, such as MUNIT [Huang et al. 2018] and DRIT [Lee et al. 2018], handle the domain barrier by learning two independent mapping functions. One is for domain-invariant content space, and the other is for domain-specific attribute space. However, both of them take a single source image as the input, which lacks the explicit separation signal between structure and appearance. This limits the translations within the appearance level. In our work, we seek structure guidance from a parsing map $x_p$, also providing an editing handle for users to adjust the generated anime easily.

Contrasting to the previous methods that learn the mapping function from scratch, we use a StyleGAN [Karras et al. 2019, 2020] pre-trained on the target domain as a prior bank. As the state-of-the-art portrait generation model, StyleGAN can generate high-quality portraits that humans cannot distinguish from the real ones. With the high capacity, the layer-wise representation of StyleGAN's $W+$ latent space can disentangle the structure and appearance information apart [Abdal et al. 2019; Karras et al. 2019; Richardson et al. 2021]. The former part of the $W+$ latent code controls the structure of output images through shallow layers of the generator, and the latter part of the $W+$ latent code controls the appearance of output images through deep layers. This is a strong prior for conquering the learning ambiguity of structure and appearance in the portrait-to-anime translations.

The overview of our conditional portrait-to-anime translation is shown in Figure 4. Taking a portrait $x$ and its corresponding parsing map $x_p$ as inputs, our disentangled encoders generate latent codes to control the anime $\hat{x}$ generation through a pre-trained StyleGAN decoder $G_y$. During training, a set of elaborated losses are imposed to learn anime translation in a self-supervised way. During testing, a FaceBank aggregation is added to enhance the quality of output animes. We then introduce each critical component in detail.

### 4.1 Disentangled Encoders

As mentioned above, different from previous works that take a single source image as the input, our model also inputs the parsing map $x_p$ for providing the explicit guidance signal on the structure. Instead of fusing both inputs directly through a single encoder, our disentangled encoders contain two: one is the parsing encoder $E_p$ that maps the parsing map $x_p$ to its latent code $w_p$, and the other is the style encoder $E_s$ that encodes the style information of source image $x$ into code $w_s$.

We use the modifications of pSp [Richardson et al. 2021] for disentangled encoders. pSp is designed to match each input image to the $W+$ latent space accurately with a feature pyramid through three levels: coarse, medium, and fine. The lower layer features are not only fed to the higher layer, but also sent into a map2style block for generating the styles in the latent space. For the parsing encoder, we only keep features from the coarse and medium levels extracted from the parsing map for controlling the structure of output images. For the style encoder, we only keep the features from the fine level by discarding the output features from map2style blocks of coarse and medium levels. In this way, the structure and appearance information is encoded by two respective encoders. This disentanglement operation separates the structure and appearance effectively, and also provides a more flexible editing knob. Note that these two encoders are trained to have specific focuses on parsing map and color distribution regardless of the input image domain. Thus, we can apply the same encoders to the parsing map and image from another domain (e.g., portrait in Figure 4) during testing to realize translation.

Particularly, the code $w_p \in R^{10*512}$ generated from the parsing map controls the former layers of StyleGAN (i.e., 1st~10th layers in

our implementation). Each layer is controlled by a $512 - d$ vector, respectively. The code $w_s \in R^{4*512}$ encoding the style and color consistency controls the latter layers of generator (i.e., 11th∼14th layers). As demonstrated in Section 5.6, our disentangled encoders relieve the learning ambiguity effectively.

## 4.2 Loss Functions

Our disentangled encoders aim to map the source images to the latent space of the pre-trained StyleGAN. However, without pair-wise data, this mapping function becomes unclear. How to design loss functions become the key to solving this problem. Previous works often rely on cycle-consistency by reconstructing images through a cycle translation. It requires learning two mapping functions simultaneously, increasing the difficulty of learning. In our work, we take the images of the target domain as a bridge. The disentangled encoders are trained by reconstructing the target images through self-supervision, while adapting to the source images. Hence, our losses can be divided into two groups: the first ones are the self-supervision losses on target images, and the other is the domain adaptation loss on the source images.

*4.2.1 Reconstruction Loss.* Given an input image $y$ in the target domain with its parsing map $y_p$, our model should reconstruct the input image in an auto-encoder way. We use the pixel-wise $L_2$ loss for reconstruction, formulated as:

$$\hat{y} = G_y(\texttt{cat}(E_p(y_p), E_s(y))), \tag{1}$$

$$\mathcal{L}_{rec} = ||y - \hat{y}||_2, \tag{2}$$

where $\texttt{cat}(\cdot)$ denotes the concatenation operation between two latent codes, and $G_y(\cdot)$ is the StyleGAN generator pre-trained on the target domain.

Following the work [Richardson et al. 2021], we also use the LPIPS loss [Zhang et al. 2018] for a better image quality preservation, which can be presented as:

$$\mathcal{L}_{lpips} = ||F(y) - F(\hat{y})||_2, \tag{3}$$

where $F(\cdot)$ denotes denotes the perceptual feature extractor.

*4.2.2 Color-preservation Loss.* Different from portraits, animes always have glorious colors. With only reconstruction loss as the supervision signal, the generated colors are degraded because of the learn-to-average nature of $L_2$. To make sure that the reconstructed images still preserve the same color distribution as their inputs, inspired by BeautyGAN [Li et al. 2018], we introduce **histogram matching (HM)** into the color preservation [Gonzalez et al. 2009]. However, directly counting color histogram for calculating loss is not differentiable. Instead, we create a guidance image $HM(\hat{y}, y)$ through histogram mapping, by preserving both the content of $\hat{y}$ and the color distribution of $y$. Particularly, $HM(\hat{y}, y)$ transforms the image $\hat{y}$ so that the output has the same color histogram with $y$ and content with $\hat{y}$, which can be used as a guidance signal for preserving the color information. Furthermore, to put more emphasis on the portrait features, we constrain the histogram matching within three components, i.e., hair, skin, and eyes. We use the parsing map for more fine-grained color preservation. Specifically, for each component $c$, we define the color preservation loss as follows:

$$\mathcal{L}_c = y_p^c \odot ||\hat{y} - HM(\hat{y} \odot y_p^c, y \odot y_p^c)||_2, \tag{4}$$

where $y_p^c$ denotes the binary parsing maps of the specific component, and $\odot$ denotes element-wise multiplication.

Finally, the proposed color-preservation loss is formulated as:

$$\mathcal{L}_{cp} = \alpha_h \mathcal{L}_{hair} + \alpha_s \mathcal{L}_{skin} + \alpha_e \mathcal{L}_{eyes}, \tag{5}$$

where $\alpha_h$, $\alpha_s$, and $\alpha_e$ denote the trade-off weights to balance each term of the color-preservation loss.

*4.2.3 Domain Adaptation Loss.* So far, the above losses are designed only on images in the target domain. To make our disentangled encoders work on the source domain, we propose a domain adaptation loss for narrowing the domain gap, achieving a better generalization ability.

As shown in Figure 4, our domain adaptation loss is applied on the $W+$ latent space. We align the distribution of source images' latent codes with the targets' codes by a code discriminator. In particular, we take the codes produced by target images as the real samples, and those produced by the source images as the fake ones. This discriminator is trained with encoders in an adversarial manner, which can be presented as:

$$\mathcal{L}_{da}^{E} = - \mathop{\mathbb{E}}_{x_p \sim p_{X_p}, x \sim p_X} [\log(D_1(\texttt{cat}(E_p(x_p), E_s(x))))], \tag{6}$$

$$\begin{aligned} \mathcal{L}_{da}^{D_1} = &- \mathop{\mathbb{E}}_{x_p \sim p_{X_p}, x \sim p_X} [\log(1 - D_1(\texttt{cat}(E_p(x_p), E_s(x))))] \\ &- \mathop{\mathbb{E}}_{y_p \sim p_{Y_p}, y \sim p_Y} [\log(D_1(\texttt{cat}(E_p(y_p), E_s(y))))], \end{aligned} \tag{7}$$

where $p_X$, $p_{X_p}$, $p_Y$ and $p_{Y_p}$ denote the distribution of the source image, source parsing map, target image, and target parsing map, respectively. $D_1(\cdot)$ is a discriminator for the generated latent codes, regarding real samples as 1s, and fake samples as 0s. The discriminator is trained with the disentangled encoders in a min-max way. After training, both the appearance encoder $E_s$ and structure encoder $E_p$ can be well generalized on the source images.

*4.2.4 Adversarial Loss.* To make sure that the reconstructed and the translated images are realistic enough, we also propose an adversarial loss that is applied to the generated images. In particular, as shown in Figure 4, our model generates two fake images $\hat{x}, \hat{y}$ based on the source input and the target input. Both of them are aligned with the real target image $y$ by image discriminator $D_2$. The discriminator is trained with a dual-supervised encoder in an adversarial manner, which can be presented as:

$$\begin{aligned} \mathcal{L}_{adv}^{E} = &- \mathop{\mathbb{E}}_{y_p \sim p_{Y_p}, y \sim p_Y} [\log(D_2(G_y(\texttt{cat}(E_p(y_p), E_s(y)))))] \\ &- \mathop{\mathbb{E}}_{x_p \sim p_{X_p}, x \sim p_X} [\log(D_2(G_y(\texttt{cat}(E_p(x_p), E_s(x))))))], \end{aligned} \tag{8}$$

$$\begin{aligned} \mathcal{L}_{adv}^{D_2} = &- \mathop{\mathbb{E}}_{y_p \sim p_{Y_p}, y \sim p_Y} [\log(1 - D_2(G_y(\texttt{cat}(E_p(y_p), E_s(y)))))] \\ &- \mathop{\mathbb{E}}_{x_p \sim p_{X_p}, x \sim p_X} [\log(1 - D_2(G_y(\texttt{cat}(E_p(x_p), E_s(x))))))] \\ &- \mathop{\mathbb{E}}_{y \sim p_Y} [\log(D_2(y))] + \frac{\gamma}{2} \mathop{\mathbb{E}}_{y \sim p_Y} || \nabla_y D_2(y)||_2^2. \end{aligned} \tag{9}$$

**Total loss.** We get our final loss function for training the disentangled encoders:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{lpips} + \lambda_3 \mathcal{L}_{cp} + \lambda_4 \mathcal{L}^E_{da} + \lambda_5 \mathcal{L}^E_{adv}, \quad (10)$$

where $\{\lambda_i\}$ denote the weight factors for balancing loss terms.

### 4.3 FaceBank Aggregation

Our model can already generate satisfying results most of the time after training with the elaborated losses. However, since there are no ground truths during training and the domain gap between portraits and animes is large, sometimes noticeable artifacts can be found during testing. The artifacts come out when the generated latent code from the source input does not lie on the $W+$ latent space. To mitigate these artifacts, following the spirit of DeepFaceDrawing [Chen et al. 2020], we adopt a FaceBank aggregation strategy during testing. We first generate a large number of $W+$ latent codes (i.e., 50,000) by a StyleGAN pre-trained on the target domain, to form a FaceBank, denoted as $S_{bank} = \{w_0, w_1, w_2, \ldots, w_n\}$. These latent codes are naturally lied on the $W+$ latent space, and inherit the linear property of $W+$ space for interpolation.

During the testing, for each predicted latent code $w^x$, we search its top $k$ nearest neighbor codes in the FaceBank $S_{\text{bank}}$, measured by the Euclidean distance, formulated as:

$$S_{\text{kNN}}(w^x) = [w^x_1, w^x_2, \ldots, w^x_k], w^x_i \in S_{\text{bank}}, i \in [1, k]. \quad (11)$$

After obtaining $k$ nearest neighbor codes, we design an anchor code $w^x_{\text{anchor}}$ as a weighted combination of these neighbor codes. For each neighbor, we calculate the weight $\beta_i$ by minimizing the Euclidean distance between the anchor code $w^x_{\text{anchor}}$ and the latent code $w^x$, formulated as:

$$w^x_{\text{anchor}} = \sum_{i=1}^{k} \beta_i \cdot w^x_i, s.t. \sum_{i=1}^{k} \beta_i = 1, \quad (12)$$

$$\beta^*_i = \arg\min_{\beta_i} \left\| w^x - w^x_{\text{anchor}} \right\|_2^2. \quad (13)$$

Since the anchor code $w^x_{\text{anchor}}$ is a combination of the neighbors in the $W+$ latent space, it is more likely to be lied in $W+$ space. Hence, we interpolate between the latent code $w^x$ and the reliable $w^x_{\text{anchor}}$ for narrowing the domain gap, which can be presented as:

$$w^x_{\text{agg}} = \alpha \cdot w^x + (1 - \alpha) \cdot w^x_{\text{anchor}}, \quad (14)$$

where $\alpha$ is the scale factor for interpolating two codes. Feeding the interpreted code $w^x_{\text{agg}}$ to the target generator $G_y$ can yield the final result fitting more into the target domain. In addition, by controlling the $\alpha$ value, we can obtain multi-interpolated results.

## 5 EXPERIMENTS

In this section, we first describe the experimental settings and compare our methods with state-of-the-art methods in related domains both qualitatively and quantitatively. Ablation studies have been conducted to examine the effectiveness of each model design choice. We use our Danbooru-Parsing dataset as the anime dataset for training and evaluation. For portrait, we select 4,921 females randomly from the CelebaMask-HQ [Lee et al. 2020], a dataset with well-annotated parsings. All images are resized to a resolution of $256 \times 256$. To show the generality of our model, we test on both portrait-to-anime and anime-to-portrait tasks in most of the experiments. Note that we train two sets of encoders for portrait-to-anime and anime-to-portrait tasks, respectively.

### 5.1 Implementation Details

We take the StyleGAN2 [Karras et al. 2020] pre-trained on the FFHQ dataset [Karras et al. 2019] as our portrait generator due to its generation ability. Meanwhile, we also fine-tune it on our Danbooru-Parsing dataset with a learning rate of 0.002 as the anime generator. The fine-tuning process contains 90,000 iterations, which takes around two days on a single Nvidia GeForce RTX 2080Ti GPU.

We follow e4e [Tov et al. 2021] using a 4-layer MLP network as our latent code discriminator, and the discriminator of StyleGAN2 [Karras et al. 2020] is employed as the image discriminator. We jointly train the encoders and discriminators with the fixed pre-trained generator. Moreover, we chose Adam [Kingma and Ba 2015] optimizer with the learning rate of $1e^{-4}$ for encoders and discriminators. Especially, in our experiment, the $\lambda$ values are set as $\lambda_1 = 2.5$, $\lambda_2 = 2$, $\lambda_3 = 0.1$, $\lambda_4 = 0.1$, and $\lambda_5 = 0.1$.

### 5.2 Compared Methods

We chose 11 state-of-the-art image translation works as our competitors, they include: CycleGAN [Zhu et al. 2017], UNIT [Liu et al. 2017], MUNIT [Huang et al. 2018], DRIT++ [Lee et al. 2020], UGATIT [Kim et al. 2020], CouncilGAN [Nizan and Tal 2020], ACLGAN [Zhao et al. 2020], AnimeGANv2 [Chen et al. 2019], ReStyle [Alaluf et al. 2021], UI2I [Huang et al. 2021]/Toonify [Pinkney and Adler 2020], and StyleGAN-NADA [Gal et al. 2021] (i.e., SG-NADA). Particularly, ReStyle, UI2I/Toonify, and SG-NADA also use the pre-trained StyleGAN generators for image translation. Note that we use the reference-based SG-NADA in the comparison, the results of text-based SG-NADA can be found in the supplement. For non-StyleGAN-based methods, we retrain their models using our dataset for a fair comparison. For StyleGAN-based methods, all methods use the same pre-trained models as ours. Please refer to the supplementary document for more details of these methods. Note that different from our method, none of them employ the structure information explicitly.

### 5.3 Evaluation Metrics

For the quantitative comparisons, we use two metrics: **Fréchet Inception Distance** (**FID**) [Heusel et al. 2017] and **Learned Perceptual Image Patch Similarity (LPIPS)** [Zhang et al. 2018], for evaluating the realism and quality of outputs translated by different methods. FID computes the Wasserstein-2 distance between the distribution of generated and target images and the lower the better. As for the LPIPS, we follow UI2I [Huang et al. 2021] that uses it for evaluating the diversity of generated images and semantic consistency between the source inputs and the generated results. For measuring the diversity, we calculate the LPIPS distance between two images randomly selected from the generated results. We denote this metric as **LPIPS-d** and the higher the better. For measuring the semantic consistency, we calculate the LPIPS between sources and the generated images, which is represented as **LPIPS-s** and the lower the better for this metric.

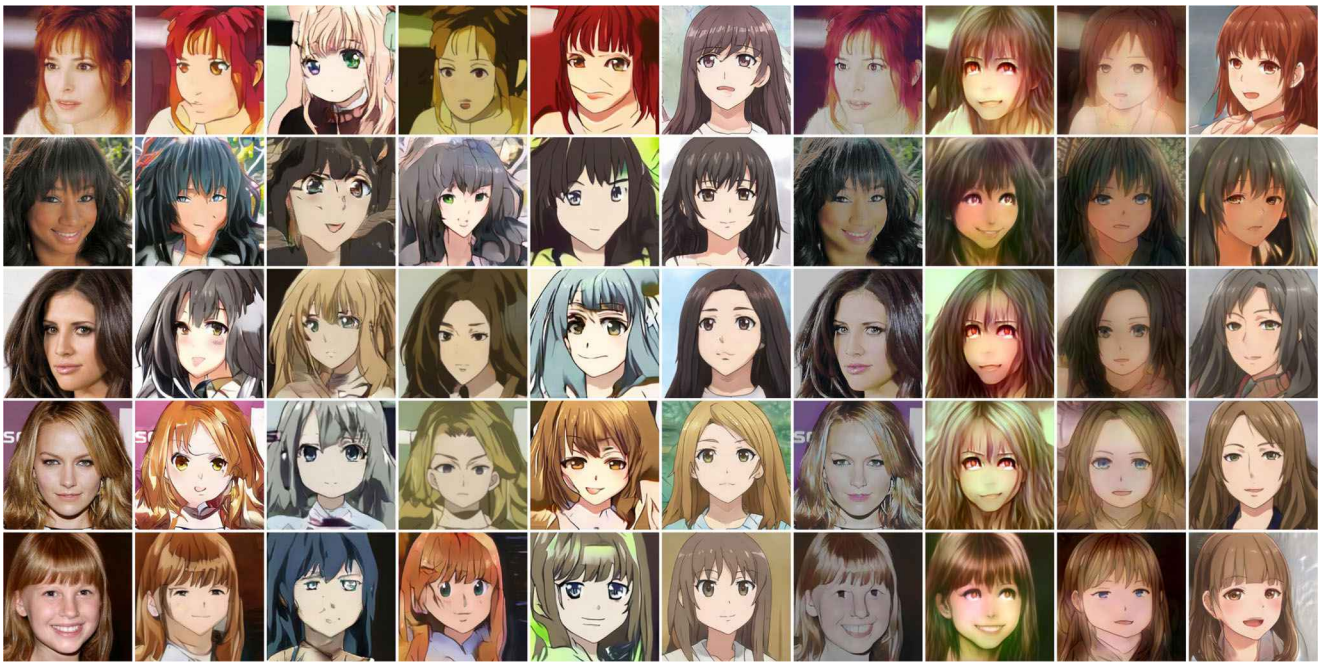



Fig. 5. Qualitative comparison of our method with state-of-the-art methods on portrait-to-anime task. Note that results from other methods have serious artifacts either on structure deformation or appearance consistency.

### 5.4 Qualitative Comparison

In this section, we first present the qualitative comparisons on the portrait-to-anime task. Different from the traditional I2I translation, portrait-to-anime is a much more difficult task since it needs large structure deformations and appearance variations. The comparison results can be seen in Figure 5 (comparison with UNIT [Liu et al. 2017], CouncilGAN [Nizan and Tal 2020], and ACLGAN [Zhao et al. 2020] can be found in the supplementary materials due to the limited space). We can see that most of the competitors cannot synthesize plausible results. Though much progress has been gained by CycleGAN [Zhu et al. 2017], MUNIT [Huang et al. 2018], and DRIT++ [Lee et al. 2020] on the common image translations, they fail on the tough portrait-to-anime task that needs both structure deformation and style transfer. Particularly, the results produced by CycleGAN [Zhu et al. 2017] preserve appearance information with the input but fail to achieve plausible anime structure.

Except for the unsatisfactory structure, MUNIT [Huang et al. 2018] cannot preserve the color-consistency of the source input. Cooperated with the attention module, UGATIT [Kim et al. 2020] is more reasonable on the structure deformation than previous works, but the translated results cannot handle the deformation of the local regions, and the quality of outputs is not high. As shown in the last row of Figure 5, the result produced by UGATIT [Kim et al. 2020] presents the blurry facial components with noticeable artifacts. AnimeGANv2 [Chen et al. 2019] is an unsupervised method that learns an anime filter, and therefore it can only change the rendering style. UI2I [Huang et al. 2021] also uses a pre-trained StyleGAN generator and the shared latent space assumption. It can produce plausible results, but loses the semantic consistency and identity information with the sources, especially on the profile faces. Moreover, their synthesized results are monotonous, as they share the same facial components. Another StyleGAN-based method Restyle [Alaluf et al. 2021] can handle the structure deformation more than other methods, but it generates a misty appearance. SG-NADA [Gal et al. 2021] focuses more on text-driven I2I, and a pre-trained CLIP [Radford et al. 2021] is too general to provide a specific representation of anime style. Thanks to our disentangled encoders, we can provide explicit guidance to the structural deformation. Another interesting observation is that UNIT [Liu et al. 2017], MUNIT [Huang et al. 2018], and DRIT++ [Lee et al. 2020] also utilize the idea of structure and appearance disentanglement, but their image-level disentanglement cannot work for anime translation. Our latent-space disentanglement presents a reasonable and consistent shape deformation with the source inputs. In addition, our generated results are more visually pleasing with harmonious colors. More importantly, our method can capture the unique characters of the input, even the finest expressions, and the translated faces are plausible and diverse.

The qualitative comparison on anime-to-portrait task can be seen in Figure 6. Same as the portrait-to-anime task, we can see that most of the generated portraits show weird faces, which indicates that they cannot handle the difficult structure deformations, as well as the appearance transformations. UI2I [Huang et al. 2021] generates a plausible result, however, their faces cannot be

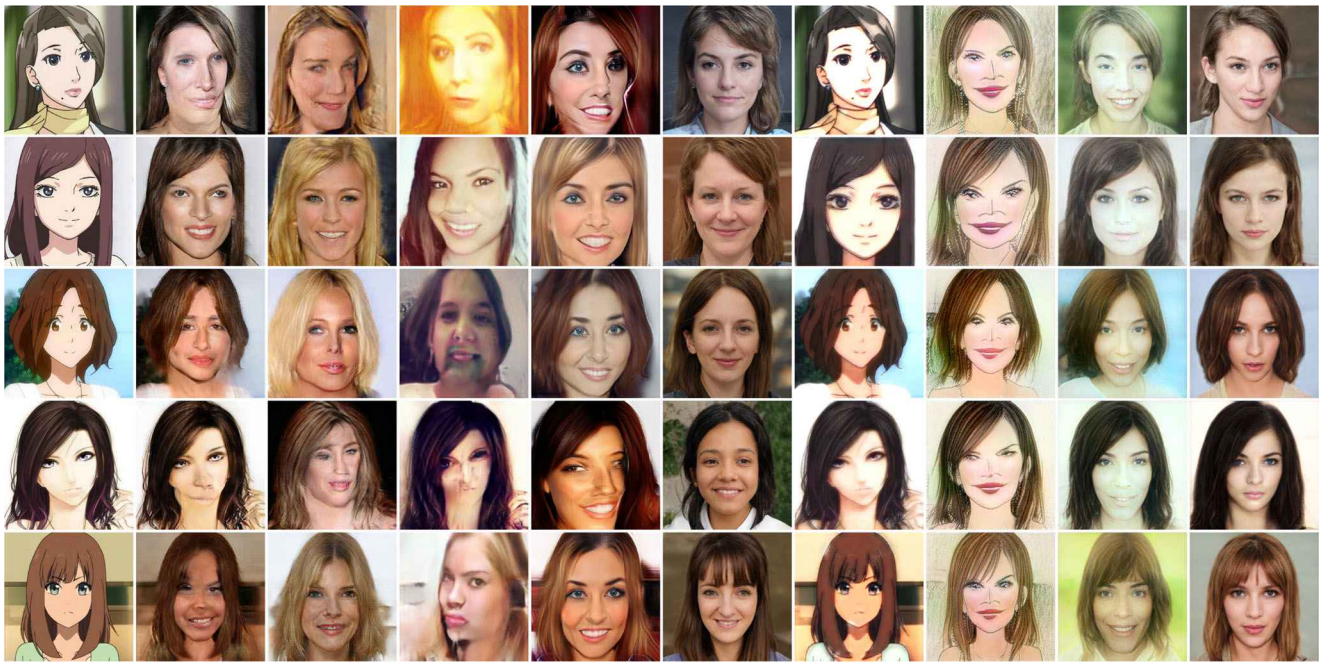



Fig. 6. Qualitative comparison of our method with state-of-the-art methods on anime-to-portrait task.

well aligned with the source animes, like the bangs and expressions. In opposite, our generated results not only present the believable faces but also can preserve the semantic information with the source animes. It is worth mentioning that the portrait and anime images are aligned with different criteria, such that the sample distributions are different (e.g., anime heads tend to be smaller). This problem is well addressed by the involved domain-adaptation and adversarial losses in our method.

## 5.5 Quantitative Comparison

The quantitative comparison on portrait-to-amine and anime-to-portrait are shown in Table 2, respectively. We can see that our model outperforms most of the other methods on three metrics by a large margin. AnimeGANv2 [Chen et al. 2019] presents the best performance on LPIPS-s metric. This is because AnimeGANv2 applies an image filter to the input and does not change the shapes of the objects (see Figures 5 and 6). Our performance on the LPIPS-s metric indicates that our model has a better translation ability and produces semantic consistent results. We believe that this improvement attributes to our parsing encoder, which provides explicit guidance on the translation process. In addition, our superiority on the LPIPS-d and FID metrics indicates that our model can capture the distribution of target images and synthesis of diverse images. To demonstrate our robustness in real-world scenarios that ground truth parsing maps are unavailable, we replace the input ground truth parsing maps by the predicted parsing maps (using DANet [Fu et al. 2019]), denoted as "Ours-Pred" in Table 2. We can see it achieves a close performance to using ground truth inputs, because of (1) parsing results are sufficiently accurate, and (2) subtle differences will not affect the generation quality (qualitative results can be found in the supplement).

Table 2. Quantitative Comparison of our Method with State-of-the-Art Methods on Portrait-to-Anime and Anime-to-Portrait Tasks

| Methods | Portrait-to-Anime | | | Anime-to-Portrait | | |
|---|---|---|---|---|---|---|
| | **FID↓** | **LPIPS-s↓** | **LPIPS-d↑** | **FID↓** | **LPIPS-s↓** | **LPIPS-d↑** |
| CycleGAN | 101.8 | 0.442 | 0.580 | 118.9 | 0.597 | 0.529 |
| UNIT | 105.5 | 0.542 | 0.576 | 114.9 | 0.576 | 0.561 |
| MUNIT | 101.8 | 0.558 | 0.585 | 105.6 | 0.553 | 0.521 |
| UGATIT | 105.6 | 0.455 | 0.553 | 147.2 | 0.602 | 0.570 |
| DRIT++ | 99.3 | 0.590 | 0.567 | 106.9 | 0.559 | 0.457 |
| CouncilGAN | 109.3 | 0.623 | 0.575 | 129.8 | 0.678 | 0.494 |
| ACLGAN | 118.1 | 0.496 | 0.466 | 141.3 | 0.544 | 0.517 |
| UI2I | 105.9 | 0.597 | 0.432 | 127.8 | 0.634 | 0.400 |
| AnimeGANv2 | 153.9 | **0.301** | 0.571 | 172.2 | **0.255** | 0.588 |
| SG-NADA | 176.3 | 0.645 | 0.401 | 146.5 | 0.682 | 0.449 |
| ReStyle | 132.3 | 0.497 | 0.498 | 119.4 | 0.550 | 0.504 |
| Ours | **91.9** | 0.421 | **0.603** | **81.4** | 0.505 | **0.610** |
| Ours-Pred | 91.8 | 0.421 | 0.599 | 81.4 | 0.502 | 0.610 |

↑ denotes the higher the better and vice visa. The best results are marked in **bold**. "Ours-Pred" indicates the results using predicted parsing maps as input.

## 5.6 Ablation Studies

In this subsection, we analyze the efficacy of the main components in the network architectures and the losses design of the proposed method. We first set a baseline called **Vanilla Encoder**, by directly using the architecture and training scheme of pSp [Richardson et al. 2021] as our encoder. It takes the parsing map of an anime as input, and the model learns to generate its ground truth anime

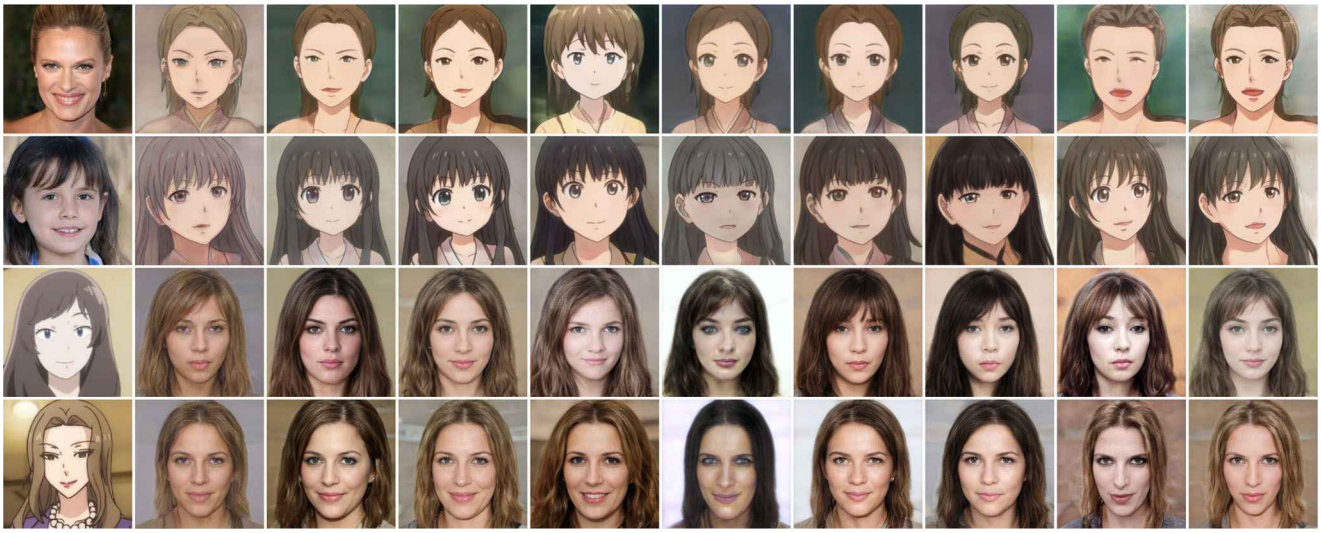



Fig. 7. Ablation studies on network architectures (i.e., "Vanilla Encoder", "w/o $\mathcal{WS}$", "w/o $\mathcal{WS}-\mathcal{P}$", "w/o $\mathcal{PE}$", and "w/o FaceBank") and loss function designs (i.e., "w/o $\mathcal{L}_{cp}$", "w/o $\mathcal{L}_{da}$", and "w/o $\mathcal{L}_{adv}$"). Each of the components make an essential contribution to the final quality of the results.

under a supervised setting (similar to GAN inversion, i.e., parsing branches only and the green regions is removed in Figure 4). We further compare our full model with seven another variants: (1) **w/o $\mathcal{WS}$**, we apply different parsing encoders and style encoders for anime and portrait domains (i.e., not shared during training and testing); (2) **w/o $\mathcal{WS}-\mathcal{P}$**, we use two individual parsing encoders for two domains and a shared style encoder; (3) **w/o $\mathcal{PE}$**, by removing the parsing encoder. We map the source input to the target one using a single appearance/image encoder; (4) **w/o $\mathcal{L}_{cp}$**, by removing the color-preservation loss; (5) **w/o $\mathcal{L}_{da}$**, by removing the domain adaption loss; (6) **w/o $\mathcal{L}_{adv}$**, by removing the adversarial loss; and (7) **w/o FaceBank**, by removing FaceBank aggregation during the testing. We show qualitative and quantitative results on both portrait-to-anime and anime-to-portrait tasks.

**Qualitative Comparison.** The qualitative comparisons of different variants are shown in Figure 7. Though the Vanilla Encoder can produce plausible results, without the elaborated disentangled encoders and loss functions, it cannot preserve the appearance of the source inputs (e.g., the hair color). The variant "w/o $\mathcal{WS}$" trains two individual encoders for two domains, the source encoders can be only trained with unsupervised losses. Without supervision from the target domain (e.g., reconstruction of target inputs), the structures of the outputs cannot be consistent with the inputs. For example, the hairstyle around the forehead in $1^{st}$ row is inconsistent with the input portrait. The same inconsistency problem also emerges in the results of variant "w/o $\mathcal{WS}-\mathcal{P}$", since its source parsing encoder also only trained without supervision from the target domain. In contrast, our results preserve more structural consistency with the inputs by sharing the parsing encoders, demonstrating that our disentangled encoders are domain-agnostic that can be applied across domains. By comparing two variants with our full model, we draw the conclusion that the parsing encoder sharing strategy plays a vitally important role in our model. That is because the domain gap between portraits and animes mainly exists in the structure, and our parsing encoder can narrow it effectively. The images translated by the variant "w/o $\mathcal{L}_{cp}$" present the gray and dim styles compared with the real animes, showing that our color-preservation loss can narrow the color-domain gap effectively. The same efficacy also occurred on the "$\mathcal{L}_{da}$" and "$\mathcal{L}_{adv}$". Without these two losses, the translated results cannot yield vivid animes. The translated results produced by the variant of "w/o FaceBank" may contain the artifacts in small regions, such as the region of hair in the example of the first row, and region of the right eye in the example of the second row. Instead, our FaceBank aggregation can eliminate these artifacts effectively. With our whole model, the generated results can preserve both structure and appearance consistent with the input portraits. Moreover, vivid details (e.g., the highlights on the hairs) are also presented in the final results like those that appeared in the real animes. The above findings and conclusions drawn from the portrait-to-anime task are also fit in the anime-to-portrait results as shown in the last two rows of Figure 7.

**Quantitative Comparison.** The quantitative comparisons of different variants can be seen in Table 3. From the results, we can see that each component and loss can boost the quantitative performance on two tasks. The Vanilla Encoder presents the worst quantitative results on three metrics. "w/o $\mathcal{WS}$" achieves the second-worst overall performance, mainly due to the shape and structure inconsistency as shown in Figure 7. The variant of "w/o $\mathcal{PE}$" achieves the highest value on LPIPS-s metric in two tasks, which evidences that the parsing encoder plays vital importance on the semantic consistency. In addition, the color-preservation loss, adversarial loss, and domain adaption loss influence the FID metric a lot, which indicates that these three losses can align the distribution of translated results to the real targets. With the help of FaceBank aggregation, our model further improves the quantitative performances as desired.

Table 3. Quantitative Results of Ablation Studies on Portrait-to-Anime and Anime-to-Portrait Tasks

| Variants | Portrait-to-Anime | | | Anime-to-Portrait | | |
|---|---|---|---|---|---|---|
| | FID↓ | LPIPS-s↓ | LPIPS-d↑ | FID↓ | LPIPS-s↓ | LPIPS-d↑ |
| Vanilla Enc. | 162.4 | 0.514 | 0.536 | 138.7 | 0.587 | 0.441 |
| w/o $\mathcal{WS}$ | 147.0 | 0.535 | 0.508 | 123.4 | 0.595 | 0.432 |
| w/o $\mathcal{WS}-\mathcal{P}$ | 129.6 | 0.521 | 0.514 | 116.2 | 0.574 | 0.439 |
| w/o $\mathcal{PE}$ | 119.1 | 0.564 | 0.490 | 106.5 | 0.612 | 0.443 |
| w/o $\mathcal{L}_{cp}$ | 128.4 | 0.509 | 0.582 | 100.4 | 0.584 | 0.568 |
| w/o $\mathcal{L}_{adv}$ | 128.3 | 0.522 | 0.565 | 109.7 | 0.553 | 0.453 |
| w/o $\mathcal{L}_{da}$ | 126.9 | 0.481 | 0.560 | 112.4 | 0.570 | 0.511 |
| w/o FaceBank | 93.7 | 0.433 | 0.591 | 92.2 | 0.521 | 0.588 |
| Full | **91.9** | **0.421** | **0.603** | **81.4** | **0.505** | **0.610** |

↑ denotes the higher the better and vice visa. The best results are marked in **bold**.

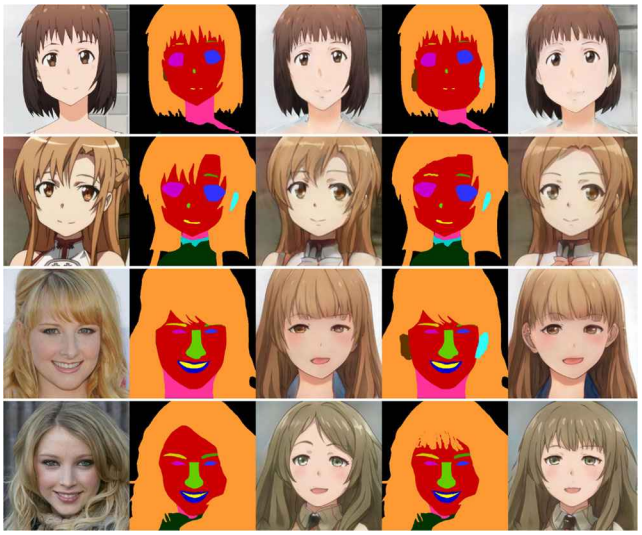



Fig. 8. Real anime editing based on the parsing maps of portraits and animes. When the local components in the parsings are modified (e.g., ears in the 1st and 3rd rows, hairstyle in 2nd and 4th rows), the resulted animes are adjusted accordingly.

## 6 APPLICATIONS

Not limiting to the portrait-to-anime and anime-to-portrait translations, our model can support various other applications, including parsing-based editing, manga-to-portrait translation, video translation, manga-to-anime translation, appearance-based editing, and semantic-based editing (the last three applications can be found in the supplement). Artists often edit the anime to achieve the desired goals. With our model, designer can adjust the resulted anime in structure and appearance individually.

### 6.1 Parsing-based Real Image Editing

The parsing branch of our disentangled encoders embeds the parsing map into the latent space. As a result, we can edit the resulted animes through the parsing maps. Note that the input to our parsing encoder is not limited to portrait parsing map, and parsing maps of existing animes are also acceptable. Given an input real anime, we can encode it using our disentangled encoders to obtain corresponding latent code, then we follow PTI [Roich et al. 2022] to fine-tune the anime generator for the faithful reconstruction of anime image (see 1st and 2nd rows of Figure 8). When taking a portrait input, we can also edit the corresponding parsing map to change the resulted animes (see 3rd and 4th rows of Figure 8). We can see that the resulted animes are modified based on the parsing maps. For example, in the 1st row of Figure 8), the face becomes rounder and the ears appear after the parsing map is edited. Similar observations can be found in the other samples, such as the changes in haircut (2nd and 4th rows of Figure 8). Besides, our modification is bounded to the local regions by maintaining other unchanged facial features.

### 6.2 Manga-to-Portrait Translation

Manga, *aka* black-and-white Japanese comics, becomes one of the essential pop arts around the world. Many readers want to hallucinate the manga characters in the real world but failed to do so. Here, our model can be extended to manga translation. Our StyleAnime also supports various manga translation tasks after manga data processing (details can be found in the supplement).

In this application, we need a reference image for providing the style code (also known as style-mixing in Abdal et al. [2019]). To the best of our knowledge, no previous works target the manga-to-portrait translation. We compare our method with two state-of-the-art sketch-to-portrait works, that is, DeepFaceDrawing [Chen et al. 2020] and pSp encoder [Richardson et al. 2021]. For a fair comparison, we also provide the same reference image to pSp encoder. The results are shown in Figure 9. The facial features are not preserved by DeepFaceDrawing, such as the haircut, facial expression, and bangs. The results of both previous methods lack vivid details and those from pSp cannot preserve the face shapes with the input mangas. However, our model can synthesize high-quality portraits, aligning with the facial features of the input manga images.

### 6.3 Video Translation

Our StyleAnime achieves satisfied translations on static images. However, when applying our method to videos, the translated videos present incoherent results between neighboring frames (such as flickering). The reason is that spatial information may be lost because of the abstracted nature of the latent space. In this subsection, we propose two new losses for eliminating the discontinuity of the video-based translation.

*6.3.1 Latent Code Smoothing Loss.* The translation is controlled by the latent codes completely. For obtaining the coherent translation video, we first proposed a latent code smoothing loss that enforces the smoothness of the predicted latent codes in a video. Particularly, given sequential frames $\{x_0, x_1, \ldots, x_n\}$, we can get corresponding latent codes $\{w_0, w_1, \ldots, w_n\}$. The smooth loss is defined as:

$$\mathcal{L}_{lcsc} = \left\| \frac{1}{2}(w_{i+1} + w_{i-1}) - w_i \right\|_2 . \tag{15}$$

This loss enforces the smoothness of neighbor latent codes, which guarantees the temporal coherence of translated frames.

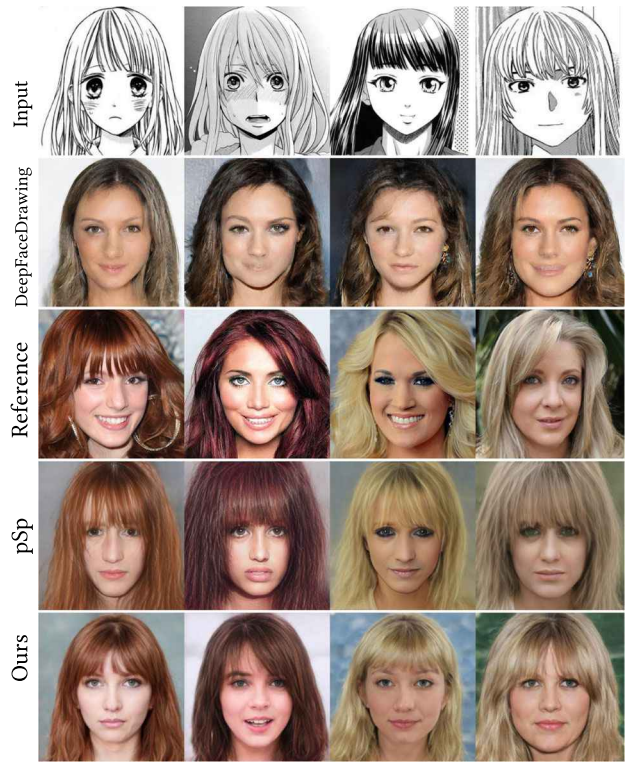



Fig. 9. Comparison on manga-to-portrait translation. The appearances of our results and pSp are assigned according to the reference images.

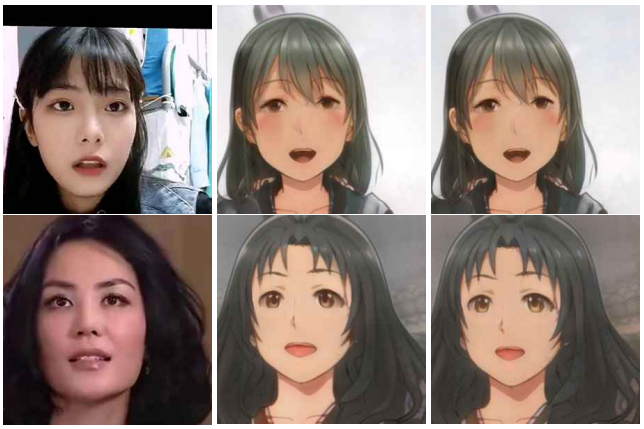



Fig. 10. Portrait-to-Anime translation on video. Non-Smooth denotes that our video continuity losses are disabled. This figure contains *animated videos*, which are best viewed using Adobe Acrobat. Video results can be found in the supplement.

*6.3.2 Frame Interpolation Loss.* Besides the latent code smoothing loss, we further propose a frame interpolation loss that applies on the translated frames directly. Given a triplet neighboring translated frame $\{\hat{x}_{t-1}, \hat{x}_t, \hat{x}_{t+1}\}$, we perform the frame interpolation based on $\hat{x}_{t-1}$ and $\hat{x}_{t+1}$ by a pre-trained frame interpolation model, and minimize the distance between interpolation results and $\hat{x}_t$ to guarantee the frame coherency, which can be presented as:

$$\mathcal{L}_{fic} = ||M(\hat{x}_{i-1}, \hat{x}_{i+1}) - \hat{x}_i||_2, \tag{16}$$

where $M(\cdot,\cdot)$ is the pre-trained anime frame interpolation model, and we use AnimeInterp [Siyao et al. 2021] in this paper.

*6.3.3 Portrait-to-Anime on Video.* For each input video, we fine-tune our StyleAnime model on this video using the above two losses with original loss on images in Equation 10 as:

$$\mathcal{L}_{video} = \lambda_6 \mathcal{L}_{lcsc} + \lambda_7 \mathcal{L}_{fic}, \tag{17}$$

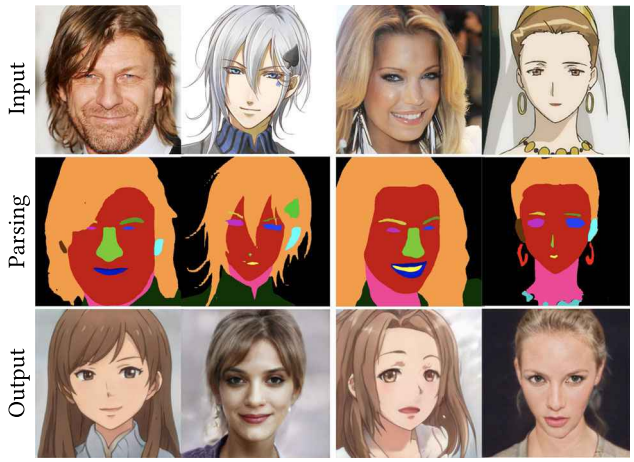
$$\mathcal{L}_{final} = \mathcal{L} + \mathcal{L}_{video}, \tag{18}$$

where $\lambda_6$ and $\lambda_7$ denote the weights for balancing loss terms and they are both set as 1. We set the learning rate as $1e^{-4}$ and iterations as 5,000. As shown in Figure 10, we obtain the portrait2anime results on real videos. The results show that our method not only demonstrates visually pleasing results on individual frames, but also can maintain the smoothness and continuity when applied to video.

## 7 CONCLUSION

In this paper, we resolve the troublesome portrait-to-anime translation by introducing the anime-parsing dataset and proposing the disentangled style encoders by absorbing the prior knowledge of a pre-trained StyleGAN model. The anime-parsing dataset connects the anime semantics with styles, which provides the explicit separate signal between structures and appearances to our disentangled encoders. The encoders are trained under our tailored losses, which handle structure and appearance consistency meanwhile domain adaptation problems. Furthermore, we develop a FaceBank aggregation method that aligns the generated latent codes with the distribution of a pre-defined latent space. Extensive experiments demonstrate the values of our new dataset and method, resulting in an excellent solution on portrait-to-anime translation. Furthermore, our model also supports various interesting applications, including parsing or appearance-based anime editing, and manga-translation task.

**Limitations.** One limitation of our work is that we cannot translate the male portrait or anime correctly, as shown in Figure 11(a). We can see that the predicted parsing maps by our model are accurate, which demonstrates our parsing model is gender-agnostic. However, the model still translates a male input into a female anime/portrait. This limitation is caused by the lack of large-scale male anime images for training the base StyleGAN. Besides, as shown in Figure 19 in the supplement, a similar training dataset limitation of profile faces prevents our method from generating portraits with large angles. These two problems and other data bias-related problems (e.g., age) can be solved in the future by proposing a large attribute-balanced anime dataset. On the other hand, our model cannot translate the small or unusual accessories accurately. As shown in Figure 11(b), the earrings, necklaces, and hood are missed in the translated results. We believe that because those unusual accessories cannot be captured by the pre-trained StyleGAN generator. Thus, we cannot obtain a precise latent code in its latent space. Our global constraints added on structure and




(a) Male subjects (b) Accessories

Fig. 11. Examples of failure cases. Although our parsing model is gender-agnostic, our translation cannot handle the male portrait/anime shown in (a) due to the data deficiency of the base StyleGAN training. Similar problem occurs for the small or unusual accessories in (b).

appearance can also lead to some minor defects, such as asymmetric eye sizes and salient color preservation in anime translation. This might be mitigated by adding an adaptive symmetrical or locality-aware constraint, and we leave this problem in the future work.

## REFERENCES


Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to embed images into the StyleGAN latent space?. In *ICCV*. 4432–4441.

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021. ReStyle: A residual-based StyleGAN encoder via iterative refinement. In *ICCV*. 6711–6720.

Anonymous, Danbooru community, and Gwern Branwen. 2021. Danbooru2020: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset. (January 2021). https://www.gwern.net/Danbooru2020.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*.

Kaidi Cao, Jing Liao, and Lu Yuan. 2018. CariGANs: Unpaired photo-to-caricature translation. *ACM TOG* 37, 6 (2018), 1–14.

Jie Chen, Gang Liu, and Xin Chen. 2019. AnimeGAN: A novel lightweight GAN for photo animation. In *ISICA*. 242–256.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI* 40, 4 (2017), 834–848.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*. 801–818.

Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. 2020. DeepFaceDrawing: Deep generation of face images from sketches. *ACM TOG* 39, 4 (2020), 72–1.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *CVPR*. 3146–3154.

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

Rafael C. Gonzalez, Richard E. Woods, and Barry R. Masters. 2009. Digital image processing. (2009).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*. 6629–6640.

Jialu Huang, Sam Kwong, and Jing Liao. 2021. Unsupervised image-to-image translation via pre-trained StyleGAN2 network. *IEEE TMM* (2021).

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *ECCV*. 172–189.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*. 1125–1134.

Wonjong Jang, Gwangjin Ju, Yucheol Jung, Jiaolong Yang, Xin Tong, and Seungyong Lee. 2021. StyleCariGAN: Caricature generation via StyleGAN feature map modulation. *ACM TOG* 40, 4 (2021), 1–16.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*. 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of StyleGAN. In *CVPR*. 8110–8119.

Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. 2020. U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. MaskGAN: Towards diverse and interactive facial image manipulation. In *CVPR*. 5549–5558.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. 2018. Diverse image-to-image translation via disentangled representations. In *ECCV*. 35–51.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. 2020. Drit++: Diverse image-to-image translation via disentangled representations. *IJCV* 128, 10 (2020), 2402–2417.

Bing Li, Yuanlue Zhu, Yitong Wang, Chia-Wen Lin, Bernard Ghanem, and Linlin Shen. 2021. AniGAN: Style-guided generative adversarial networks for unsupervised anime face generation. *IEEE TMM* (2021).

Tingting Li, Ruihe Qian, Chao Dong, Si Liu, Qiong Yan, Wenwu Zhu, and Liang Lin. 2018. BeautyGAN: Instance-level facial makeup transfer with deep generative adversarial network. In *ACM MM*. 645–653.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *NeurIPS*. 700–708.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-shot unsupervised image-to-image translation. In *ICCV*. 10551–10560.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. 3431–3440.

Ling Luo, Dingyu Xue, and Xinglong Feng. 2020. EHANet: An effective hierarchical aggregation network for face parsing. *Applied Sciences* 10, 9 (2020), 3135.

Nagadomi and Youkaichao. 2014. lbpcascade animeface. (2014). https://github.com/nagadomi/lbpcascade_animeface/.

Ori Nizan and Ayellet Tal. 2020. Breaking the cycle-colleagues are all you need. In *CVPR*. 7860–7869.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. 2020. Swapping autoencoder for deep image manipulation. *NeurIPS* 33 (2020), 7198–7211.

Justin N. M. Pinkney and Doron Adler. 2020. Resolution dependent GAN interpolation for controllable image synthesis between domains. In *NeurIPS Workshop*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. 8748–8763.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in style: A StyleGAN encoder for image-to-image translation. In *CVPR*. 2287–2296.

Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. 2022. Pivotal tuning for latent-based editing of real images. *ACM TOG* 42, 1 (2022), 1–13.

Yichun Shi, Debayan Deb, and Anil K. Jain. 2019. Warpgan: Automatic caricature generation. In *CVPR*. 10762–10771.

Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. 2021. Deep animation video interpolation in the wild. In *CVPR*.

Guoxian Song, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. 2021. AgileGAN: Stylizing portraits by inversion-consistent transfer learning. *ACM TOG* 40, 4 (2021), 1–13.

Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. 2020. MichiGAN: Multi-input-conditioned hair image generation for portrait editing. *ACM TOG* 39, 4 (2020), 95–1.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and D. Cohen-Or. 2021. Designing an encoder for StyleGAN image manipulation. *ACM TOG* 40, 4 (2021), 1–14.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional GANs. In *CVPR*. 8798–8807.

Keyu Wu, Lingchen Yang, Hongbo Fu, and Youyi Zheng. 2021. iHairRecolorer: Deep image-to-video hair color transfer. *Science China Information Sciences* 64, 11 (2021), 1–15.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. BiseNet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*. 325–341.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. 586–595.

Yihao Zhao, Ruihai Wu, and Hao Dong. 2020. Unpaired image-to-image translation using adversarial consistency loss. In *ECCV*. 800–815.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *CVPR*. 633–641.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*. 2223–2232.

Received 19 September 2021; revised 7 February 2023; accepted 13 February 2023