









Stylized Face Sketch Extraction via Generative Prior with Limited Data

Kwan Yun*¹  Kwanggyoon Seo*¹  Chang Wook Seo*¹  Soyeon Yoon¹ 
Seongcheol Kim¹  Soohyun Ji²  Amirsaman Ashtari¹  Junyong Noh¹ 

¹KAIST, Visual Media Lab ²Technical University of Munich

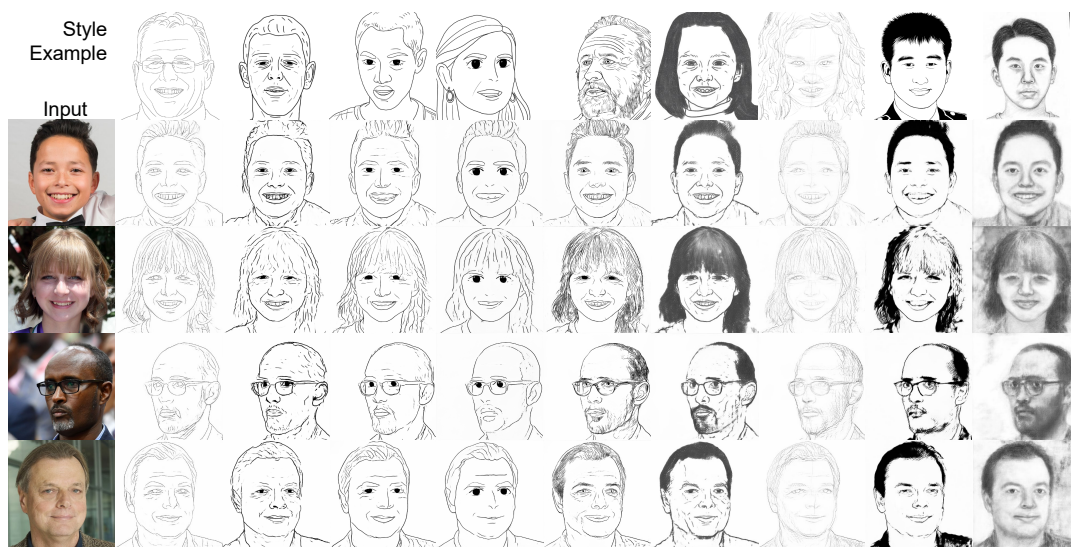


Figure 1: Sketches extracted in nine different styles from the input images. Each style is trained with 16 pairs of data.

Abstract

Facial sketches are both a concise way of showing the identity of a person and a means to express artistic intention. While a few techniques have recently emerged that allow sketches to be extracted in different styles, they typically rely on a large amount of data that is difficult to obtain. Here, we propose StyleSketch, a method for extracting high-resolution stylized sketches from a face image. Using the rich semantics of the deep features from a pretrained StyleGAN, we are able to train a sketch generator with 16 pairs of face and the corresponding sketch images. The sketch generator utilizes part-based losses with two-stage learning for fast convergence during training for high-quality sketch extraction. Through a set of comparisons, we show that StyleSketch outperforms existing state-of-the-art sketch extraction methods and few-shot image adaptation methods for the task of extracting high-resolution abstract face sketches. We further demonstrate the versatility of StyleSketch by extending its use to other domains and explore the possibility of semantic editing. The project page can be found in https://kwanyun.github.io/stylesketch_project.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence; Computer vision; Computer vision;**

1. Introduction

Sketching is the first step in artistic creation. It is a fundamental task for abstracting and expressing artistic ideas, which pre-visualizes

the main structure and content of the final image. Despite the simple construction based on lines, sketches can have unique styles; some artists prefer to use lines with varied thicknesses depending on the locations of the constituent lines while others like to emphasize specific areas such as eyes with filled-in black color.

* These authors contributed equally to this work

Facial sketches can play a more important role than sketches that depict general objects. Facial sketch can show the identity of a person in a concise way while expressing artistic intention. One recent line of research has attempted to reconstruct realistic human face images from sketches for various applications such as crime investigation, character design, educational training, and so on [CSG*20]. In many of these applications, performance has been noticeably improved by using generative models such as generative adversarial networks (GANs) and diffusion probabilistic models [WLW*22, LCWZ19, LKL*20, CSG*20, SLC*22, PHS*18, LCL*22]. However, these learning-based methods require a large number of paired sketch and photo images for training to produce adequate results.

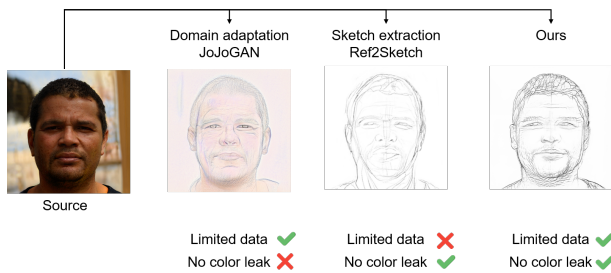


Figure 2: Data-Artifact Tradeoff: Domain adaptation methods may require less data but often exhibit color bleeding [CF22]. Image translation-based sketch extraction methods avoid such artifacts, but require a large dataset [ASK*22]. Our approach, using deep features from generative models while training from scratch, avoids artifacts and can be trained with a limited number of data.

Instead of drawing sketches from scratch to obtain the dataset, many researchers have investigated how to extract synthetic sketches from a given image. Automatic techniques such as filter-based edge detection methods [Can86, Win11] or data-driven sketch extraction [lll17, XLY*21, SSIS16, ASK*22, CDI22] methods have been proposed. Each type of sketch extraction method has its own unique characteristics. Filter-based edge detection methods are efficient, yet the results often consist of noisy and dotted lines. This is because their purpose is to find the precise edges instead of expressing sketch-like drawings. While learning-based sketch extraction methods can generate artistic style sketches based on training data, they again require a large dataset to train the system.

To address this challenge, we propose StyleSketch, which can extract high-resolution artistic sketches from a given face image after being trained with a limited number of data. StyleSketch leverages the deep generated features from the pretrained StyleGAN [KLA*20a] as a building block. The deep features have been shown to have meaningful information for downstream tasks [ZLG*21, LLK*22, PZZ*22, AZMW21a]. Using these deep features as input, we trained a new sketch generator with deep fusion modules.

StyleSketch is designed specifically for a sketch extraction task. A dedicated sketch generator is trained on the features generated from the pretrained StyleGAN. This makes StyleSketch different from existing domain adaptation methods. Domain adaptation methods either fine-tune [CF22, XLW*22, ZAFW21, OLL*21] or

employ adaptation modules to manipulate the weights of the pre-trained generator [KKK*22, YJLL22]. While these approaches can be applied to many style related tasks, their stylized sketch extraction results are often unsatisfactory due to mode collapse or color bleeding. StyleSketch avoids these issues by designing a dedicated sketch generator that takes deep generated features of a pre-trained generator as input. In addition, thanks to the rich semantic features of deep generated features, StyleSketch does not require a large training dataset, unlike other sketch extraction methods [YLLR19, YXL*20, YLLR20, CDI22, ASK*22, SAN23]. The differences of our method are highlighted in Fig. 2

In addition to the newly proposed model architecture, we designed a specialized discriminator and loss functions for each part of the semantic region because the sketch style may differ from one semantic part to another. We also included a scheduler to guide the training process for faster convergence of the model. After training the sketch generator, StyleSketch can perform sketch extraction by inverting the image into the latent code of the StyleGAN [AQW20, RAP*21, TAN*21]. The latent code is then transformed to deep features that are fed to the sketch generator to produce an artistic sketch image. In addition to sketch extraction, StyleSketch can perform related tasks such as photorealistic image generation from a face sketch and semantic editing of sketch images.

For validation, we construct a high-resolution dataset of paired face sketches, **SKSF-A** featuring the same identity in seven different styles drawn by artists. There are several face image datasets that include the corresponding sketch drawings; however previous face sketch datasets consist of sketches with only one or at most a few drawing styles and the size of the dataset is typically small. Our newly constructed SKSF-A dataset consists of 938 images with seven different drawing styles and contains attributes for photo information. Our novel dataset is of higher resolution (1024×1024) compared to existing face datasets which are typically lower than $779.62 \pm 15.05 \times 812.10 \pm 13.92$ in resolution [YLLR19, ZWT11, WT08]. With this SKSF-A and previous face sketch datasets [YLLR19, WT08], we show that our method can be trained with a limited number of paired data and performs equally or even better compared to state-of-the-art methods trained with more data. Example results are presented in Fig. 1.

Our contributions can be summarized as follows.

- We propose **StyleSketch**, a novel method that utilizes the pre-trained StyleGAN features to extract sketches from face images enabling the network to be trained with a limited number of data.
- StyleSketch utilizes a dedicated sketch generator that allows to aggregate the generated features effectively using a deep fusion module. During the training of the sketch generator, we use a two-stage learning process for improved convergence.
- In addition to sketch extraction, StyleSketch can perform the task of semantics editing of input sketches having diverse styles.
- We introduce **SKSF-A**, a high-resolution dataset of paired face sketches in seven distinct styles drawn by professional artists. SKSF-A contains 134 identities and 938 sketches in total. SKSF-A is the most diverse and highest-resolution sketch dataset.

2. Related work

2.1. Sketch Extraction

The simplistic form of sketch extraction is edge detection. Most early studies of edge detection are based on techniques of finding the edge from the sudden change of color or brightness [Can86, Win11]. While these methods are simple and powerful without learning from a large corpus of data, the generated results sometimes consist of noisy dots and scattered lines. To reproduce natural expression of drawings in the extracted sketches, learning-based approaches have been proposed. These methods are capable of detecting object boundaries or extracting lines in particular styles [XLY*21, XT15, III17, LLM*19, LLW17]. Learning To Generate Line Drawings (Learn-to-draw) [CDI22] improved previous methods by utilizing the depth and semantic meanings of a color image to extract a higher quality sketch. More recently, Ref2sketch [ASK*22] was introduced which allows a sketch to be extracted in a style of the reference sketch using paired training while Semi-Ref2sketch [SAN23] utilizes contrastive learning for semi-supervised training.

Unlike general purpose sketch extraction methods, some methods aim to extract sketches for a certain domain of images. Due to the need for face sketches in many applications, there have been approaches specifically designed for facial sketch generation [YLLR19, YLLR20, YXL*20]. These methods can generate higher quality sketch results compared to general sketch extraction methods due to the use of domain-specific networks and losses.

Because of the nature of learning-based image-to-image translation methods, the aforementioned techniques require a large number of data to train the model. Obtaining authentic sketch drawings of various styles with a large number of paired color images is very difficult and expensive, leading to data scarcity and consequent difficulty in training a sketch extraction model. To address this challenge, our method is designed to train the network with a few paired data by utilizing a deep generative prior in the face domain.

2.2. Few-shot Domain Adaptation

While many attempts have been made to train generative models from scratch using a large corpus of data [KLA19, BDS18, HJA20, RBL*22, CZB*23], several methods have shown promising results by leveraging a pretrained deep generative model to perform domain transfer with a limited number of data. The current approaches typically involve training a base model on a large number of source domain data and then fine-tuning it in a closely related target domain. One line of the approach utilizes additional modules in the generator to produce a target domain image [NH19, WGGB*20, YJLL22, SRL*23, ZA23]. Another approach employs regularization to avoid mode collapse that is often observed in many few-shot training approaches [OLL*21, XLW*22, MCS20, LZLS20]. More recently, there have been attempts to use the semantic power of a vision-language model CLIP [RKH*21] that can guide the training process to transfer the source to the target domain while preserving the source distribution [GPM*22, ZAFW21]. Optimizing the strokes with CLIP in vector representation has also been proposed [VPB*22]. Instead of fine-tuning the generator, DynaGAN [KKK*22] trains a hypernetwork using contrastive learning

to modulate the weight of the generator for training with a limited number of stylized data while preventing mode collapse.

These domain adaptation methods generally work well when the joint probability distribution of the target data is the same as or similar to the original source distribution [HLYJ20, ZW20, CFHR17, FVRA21]. However, when the target domain is far from the original domain, it is challenging to learn and generate the target, such as sparse line drawings of a face where most regions have no information while real faces are pre-trained in the original domain. Recently, there have been attempts to personalize diffusion-based models by fine-tuning the denoising network [RLJ*23] or finding a textual embedding given a concept or style image [GAA*22]. We designed a model and a training scheme that can mimic the perceptual style of the artistic sketch images while preserving the geometric details and identity of the original photo.

2.3. Leveraging StyleGAN Prior for the Downstream Task

In addition to generating images in an unsupervised manner, StyleGAN [KLA*20a] can be further extended to various downstream tasks. pSp [RAP*21] is a generic image-to-image translation framework that employs a frozen StyleGAN. The method inverts input images to the latent space of pretrained StyleGAN, which enables the generation of corresponding photorealistic images. In addition, GAN inversion methods [AQW20, RAP*21, TAN*21] can perform tasks such as image/video editing [SGT20, AZMW21b, YNGH21, TMG*22, XAH22, SOL*22] and video generation [TRC*21, FTET21].

Another line of research has explored the deep geometric features generated by pretrained GAN models to exploit rich semantic information. DatasetGAN [ZLG*21] uses a few paired labeled data to train a semantic segmentation model. Labels4free [AZMW21a] uses a background and foreground generator to train an additional segmentation network in an unsupervised manner, and GANgealing [PZZ*22] utilizes GAN supervised training for dense visual alignment. These methods are possible because the features generated by StyleGAN hold valuable information that is structurally aligned with the task. Building upon this concept, we leverage the features generated from StyleGAN and put emphasis on the connectivity and style of the sketch edge.

3. Methods

3.1. Overview

StyleSketch extracts sketches from a given input face image through the process described below.

1. First, a paired sketch and image dataset are prepared and the RGB image is projected into the pretrained latent space of StyleGAN G . With the projected latent code w^+ , we hijack the deep features F from StyleGAN. F are extracted at the end of each style block resulting in 18 features total in different sizes.
2. The deep features and their corresponding sketch I_{sketch} are used to train the sketch generator G_{sketch} . G_{sketch} is trained in a supervised manner with additional part-based loss functions using a segmentation network. After training G_{sketch} , we can simultaneously generate both photo-realistic images from the origi-

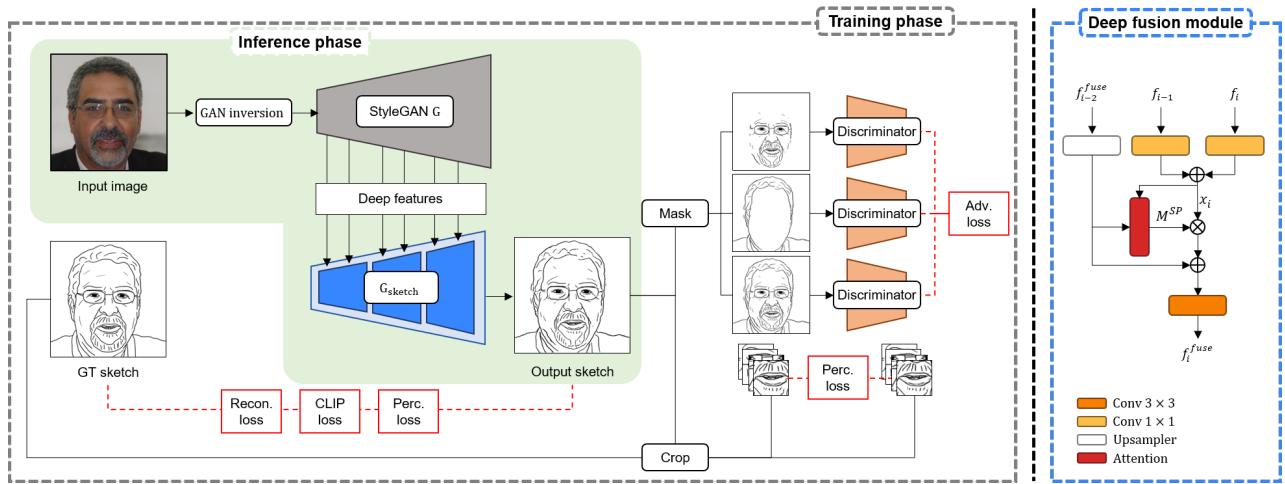


Figure 3: Overview of StyleSketch. To extract a sketch from a face image, we first train a sketch generator G_{sketch} that accepts deep features as input. G_{sketch} is constructed with the deep fusion module represented on the right side of the figure. After training, we extract sketches by performing GAN inversion of the input image followed by feeding the deep features into G_{sketch} .

nal pretrained StyleGAN and sketches from G_{sketch} of the same identity and structure.

3. To extract a sketch from an image at inference, we project the image into the latent space of pretrained StyleGAN and use the projected latent code to synthesize the sketch using G and G_{sketch} .

In the following subsections, we will describe the data construction (Sec. 3.2), the structure of sketch generator G_{sketch} (Sec. 3.3), its loss functions (Sec. 3.4), and finally the proposed sketch extraction method (Sec. 3.6).

3.2. Data Construction

Our goal is to train G_{sketch} using a limited number of paired data by leveraging the semantics of a deep generative prior which is known to embed the knowledge of the human face. To use such prior, we first need to project the input image to the latent space of StyleGAN using a pretrained GAN inversion encoder [TAN*21] and a few optimization steps [AQW20]. This latent code w^+ is then passed through StyleGAN, and we hijack the intermediate generated deep features as $F = \{f_1, \dots, f_n\}$ where n is a total number of features. We use the extracted deep features F and their corresponding sketch I_{sketch} as paired data to train the sketch generator G_{sketch} .

3.3. Sketch Generator

Our sketch generator G_{sketch} was built with convolution and attention blocks, as shown in Fig. 3. G_{sketch} takes F as input and synthesizes the corresponding sketch \hat{I}_{sketch} , which can be written as follows:

$$\hat{I}_{sketch} = G_{sketch}(F). \quad (1)$$

Unlike other learning-based sketch extraction methods in which a network takes an image as input [ASK*22, CDI22, YLLR19], our method accepts multiple deep features that have different spatial

resolutions and channel numbers. The width and height of the deep features increase gradually from 4×4 to 1024×1024 while the channel number decreases from 512 to 3. Because of the multiple inputs that have differences in spatial resolution and channel number, the deep features are gradually fused from coarse to fine levels. Our deep fusion module accepts two features (f_i, f_{i-1}) from the $(i, i-1)^{th}$ layer of pretrained StyleGAN, which have the same spatial resolution, along with previously fused features f_{i-2}^{fuse} . The channel of (f_i, f_{i-1}) is decreased using a 1×1 convolution layer for computational efficiency and the features are concatenated to obtain x_i . Because f_{i-2}^{fuse} has a lower spatial resolution compared to that of (f_i, f_{i-1}) , we first resize the features using an upsampling layer \uparrow .

A spatial attention map M^{SP} is synthesized from the fused feature representation $[f_{i-2}^{fuse}]$ and subsequently multiplied by x_i . M^{SP} serves as the gating coefficient for input information x_i . The resulting output from this step is concatenated with $[f_{i-2}^{fuse}]$ and passed through a convolutional layer to produce f_i^{fuse} . The deep fusion module can be written as follows:

$$\begin{aligned} x_i &= Conv^{1 \times 1}(f_i) \oplus Conv^{1 \times 1}(f_{i-1}) \\ f_i^{fuse} &= Conv^{3 \times 3}((M^{SP} \otimes x_i) \oplus [f_{i-2}^{fuse}]), \end{aligned} \quad (2)$$

where $Conv^{3 \times 3}$ and $Conv^{1 \times 1}$ are the convolution layers, \otimes is element-wise multiplication, and \oplus is concatenation in the channel dimension.

3.4. Objective Functions

To train the sketch generator G_{sketch} , we use four loss functions: reconstruction loss \mathcal{L}_1 , perceptual loss \mathcal{L}_{perc} , clip loss \mathcal{L}_{clip} , and adversarial loss \mathcal{L}_{adv} . When computing \mathcal{L}_{perc} and \mathcal{L}_{adv} , we additionally use a segmentation model [YWP*18] to mask out or crop parts of the face so that each semantic area can have a different style as performed in previous methods [YLLR19, CSG*20].

The most basic loss for an image-to-image translation task is to use a reconstruction loss. We use L1 distance between the ground truth sketch image I_{sketch} and \hat{I}_{sketch} , which can be written as follows:

$$\mathcal{L}_{recon} = \|I_{sketch} - \hat{I}_{sketch}\|_1. \quad (3)$$

We use \mathcal{L}_{recon} in the first stage of learning to make the network converge faster and perform better. This will be further explained in the last paragraph of this section.

The perceptual loss function \mathcal{L}_{perc} guides the synthesized \hat{I}_{sketch} to closely resemble the ground-truth I_{sketch} using a pretrained VGG16 [SZ14], which is shown to be effective in following the fine details of the target style [JAFF16, XXM*19]. The loss is formulated as follows:

$$\begin{aligned} \mathcal{L}_{perc} = & \sum_l \|\phi_l(I_{sketch}) - \phi_l(\hat{I}_{sketch})\|_1 \\ & + \frac{1}{N} \sum_C \sum_l \|\phi_l(C(I_{sketch})) - \phi_l(C(\hat{I}_{sketch}))\|_1, \end{aligned} \quad (4)$$

where ϕ_l refers to the activation map of the l^{th} layer of a pretrained VGG16, and C represents the cropping operation for each facial part such as eyes, nose, and lips, using a segmentation model [YWP*18]. N refer to the total number of cropped face features.

In addition to the perceptual loss, we use a visual-text embedding CLIP [RKH*21] to capture high-level semantics for sketches, as reported in Learn-to-draw [CDI22]. Thus, we compute the cosine similarity of the CLIP embedding between I_{sketch} and \hat{I}_{sketch} as follows:

$$\mathcal{L}_{clip} = 1 - \langle CLIP(I_{sketch}), CLIP(\hat{I}_{sketch}) \rangle, \quad (5)$$

where $\langle \cdot, \cdot \rangle$ is the cosine distance function.

Lastly, we jointly train multiple discriminators with G_{sketch} so that the generator accurately reflects the trained sketch style. By incorporating the discriminators, G_{sketch} is enforced to generate a sketch image that belongs to the style observed in the ground truth sketch dataset. We use a conditional patch adversarial loss [IZZE17] as follows:

$$\begin{aligned} \mathcal{L}_{adv} = & \sum_D \left[\mathbb{E}_{I, I_{sketch}} [\log D(M_D \otimes I, M_D \otimes I_{sketch})] \right. \\ & \left. + \mathbb{E}_{I, \hat{I}_{sketch}} [\log(1 - D(M_D \otimes I, M_D \otimes \hat{I}_{sketch}))] \right], \end{aligned} \quad (6)$$

where M^D is a binary mask that indicates where the discriminator D should focus generated from segmentation [YWP*18]. We further jointly augment I , I_{sketch} , and \hat{I}_{sketch} using the non-leaking augmentation method proposed in StyleGAN-ada [KAH*20a].

The final objective function can be written as follows:

$$\begin{aligned} G_{sketch}^* = & \arg \min_{G_{sketch}} \max_D \lambda_{recon} \mathcal{L}_{recon} + \lambda_{perc} \mathcal{L}_{perc} \\ & + \lambda_{clip} \mathcal{L}_{clip} + \lambda_{adv} \mathcal{L}_{adv}, \end{aligned} \quad (7)$$

where λ_{recon} , λ_{perc} , λ_{clip} , and λ_{adv} are the weight for each loss term.

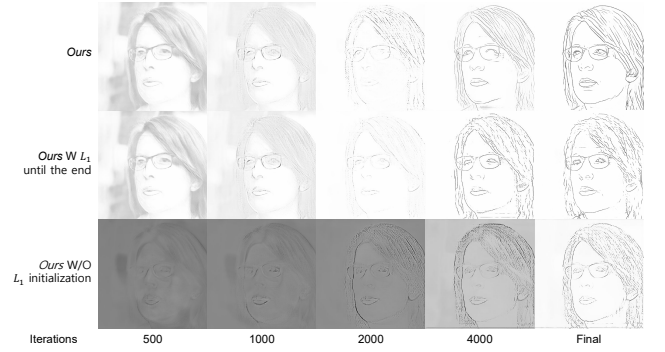


Figure 4: Results produced according to the iterations performed with and without the initial stage. Training without L1 loss requires a large number of iterations to optimize the output into the sketch domain.

3.5. Two-stage Learning

Despite its wide use as a loss function, the L1 distance, which is L_{recon} in our method, can be problematic for sketch extraction because it focuses more on pixel-wise differences than perceptual information [ASK*22, XXM*19]. While L_{recon} enables capturing of the overall structure of the sketch faster in the initial training steps, it can degrade the final quality when it is used throughout training, leading to noisy dotted patterns. To this end, we adopt a two-stage training approach that utilizes L_{recon} in the initial stage only. As shown in Fig. 4, with two-stage learning, our model converges faster in the early stage of training and ultimately leads to higher quality sketches. More experiments and their results can be found in Sec. 5.4.

3.6. Inference

Once the sketch generator G_{sketch} has been trained, we can extract a sketch from a given face image by performing GAN inversion, as explained in Sec. 3.2. Specifically, the input image is transformed to a latent code w^+ that will be fed to the pretrained StyleGAN G for the generation of deep features F . G_{sketch} then takes F as input to generate the corresponding sketch \hat{I}_{sketch} . The process is shown in the green box of Fig. 3.

4. SKSF-A Dataset

There exist several face image datasets that include the corresponding sketch drawings. We briefly describe the details of these datasets based on the descriptions of the FS2K paper [FHZ*22]. See Table 1 for the summarized details.

Among widely used face sketch datasets of a single drawing style, CUFS [WT08] consists of 606 color photo and sketch pairs, emphasizing shading effects created by pencil drawings. CUFSF [ZWT11], an extension of CUFS, is designed for evaluating face sketch synthesis methods and offers 1,194 pairs with varied contrast and background colors. APDrawing [YLLR19] provides high-quality face drawings, featuring 140 pairs expanded to contain 910 additional synthetic sketches through augmentations.

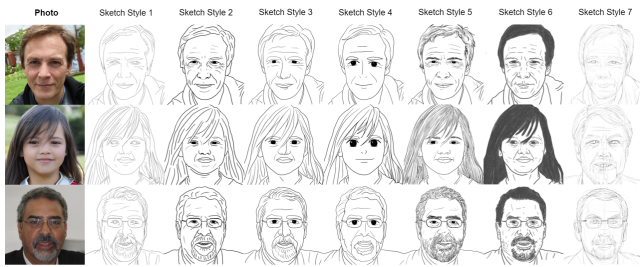
Table 1: Comparison with existing authentic face sketch datasets.

Dataset	Total	Style	Resolution	Attributes	Paired	Public
SKSF-A(ours)	938	7	1024 × 1024	✓	✓	✓
CUFS [WT08]	606	1	200 × 250	×	✓	✓
CUFSF [ZWT11]	1,194	1	779.62 × 812.10	×	✓	✓
DisneyPortrait [BSM*13]	672	1	-	×	×	✓
APDrawing [YLLR19]	140	1	512 × 512	×	✓	✓
UPDG [YLLR20]	952	3	-	×	×	×
FS2K [FHZ*22]	2,104	3	299.74 × 273.56	✓	✓	✓

These datasets [WT08, ZWT11, YLLR19] include paired authentic sketches, enabling easy evaluation of image-to-image translation models.

The Disney portrait dataset [BSM*13] includes 672 sketches of a single style and in four levels of abstraction. UPDG [YLLR20] comprises 952 sketches in three styles. Different from the aforementioned datasets [WT08, ZWT11, YLLR19], Disney portrait dataset [BSM*13] and UPDG do not provide corresponding photo images, making it challenging to compare the images generated from image-to-image translation models with their corresponding ground truths. The most recent and largest FS2K dataset [FHZ*22] features 2,104 pairs of photos and sketches in three different styles created by different artists. The sketches underwent slight changes for artistic expression. Attributes such as hair style and facial feature colors are also provided. Unfortunately, the provided images are of low resolution and only one style is provided for each identity, which restricts its usability.

Focusing on resolution, style variety, and a paired setting, we build a dataset of authentic sketch style faces (SKSF-A) which consists of face sketches in seven different styles. The seven different styles represent complex and diverse details such as clear lines, rough lines, or filled-in colors. Because the SKSF-A dataset is drawn on FFHQ [KLA19] aligned face, it can be easily utilized with methods trained with the FFHQ dataset. See Fig. 5 for examples.

**Figure 5:** Overview of our authentic face sketch dataset (SKSF-A). The dataset consists of photo and sketch pairs while the sketch has one of 7 different styles.

4.1. Face Photos

For free licensing of our SKSF-A dataset and its convenient distribution, we generated synthetic face photo images by utilizing GAN-based methods. Among various realistic face image generation methods [SSG22, HBB22, LZG*21, WZH*22, KLA*20b, KAH*20b], we utilized StyleGAN2 with the weights pretrained by

the FFHQ [KLA19] dataset to generate high quality face images. From the generated images, we selected 134 images without any artifacts on the face. To utilize these images as a learning-based method dataset, we divided the SKSF-A dataset into a training set of 80 images and a validation set of 54 images. In the experiments, our method uses 16 randomly selected images from the 80-image training set unless otherwise noted.

4.2. Face Sketches

Two artists participated in drawing the sketches for the SKSF-A dataset. These artists have no color-blindness or color-weakness. As shown in Fig. 5, the sketches represent diverse styles. Sketches of the sixth and seventh styles were drawn by the second artist and the others were drawn by the first artist. The artists drew all of the sketches with a pen tablet device and drawing software [CEL12].

5. Experiments

5.1. Implementation Details

We implemented StyleSketch and trained the sketch generator on a computer with 2 Nvidia RTX-3090 GPUs. All experiments were conducted on a computer with an Nvidia RTX-3090 GPU. In all of our experiments, StyleGAN2 pretrained on FFHQ [KLA*20b] was used unless noted otherwise. For training G_{sketch} , the learning rate was fixed to 0.00014. The parameters used in Eq. 7 were fixed at $\lambda_{adv} = 1$, $\lambda_{clip} = 120$, $\lambda_{perc} = 1.2$ for both the initial stage and the second stage, and $\lambda_{L1} = 200$ for the initial stage. Out of a total of 7200 iterations, the first 1600 iterations were considered the initial stage. The sketch generator requires a total of 9 hours for training and 0.43 seconds for inference. We used 16 pairs of face and sketch images for training unless noted otherwise.

5.2. Comparison with Baselines

For quantitative evaluation, we compared our method with seven different baselines. Four baselines are domain adaptation methods for stylization while three methods are sketch extraction methods. Mind the gap (MTG) [ZAFW21] and JoJoGAN [CF22] are one shot domain adaptation methods; the former fine-tunes the StyleGAN [KLA*20b] with bidirectional CLIP information and the latter achieves the same with augmentation. RSSA [XLW*22] and few-shot image generation (Few-shot) [OLL*21] are few shot adaptation methods that utilize a pretrained StyleGAN to manipulate face images. RSSA uses relaxed spatial structure alignment while Few-shot [OLL*21] utilizes a cross-domain distance consistency loss. These features help prevent overfitting when training with a limited number of data. All four domain adaptation methods were trained in an unpaired setting with proper alignment. However, this approach sometimes failed to align sparse sketch styles. Therefore, we aligned the sketches using our paired method for training.

Ref2sketch [ASK*22] is a general purpose sketch extraction model which allows line drawings of various styles to be extracted in one model by imitating the styles of a reference input. Learning to generate line drawing (Learn-to-draw) [CDI22] is also a general purpose sketch extraction model that utilizes depth and semantic

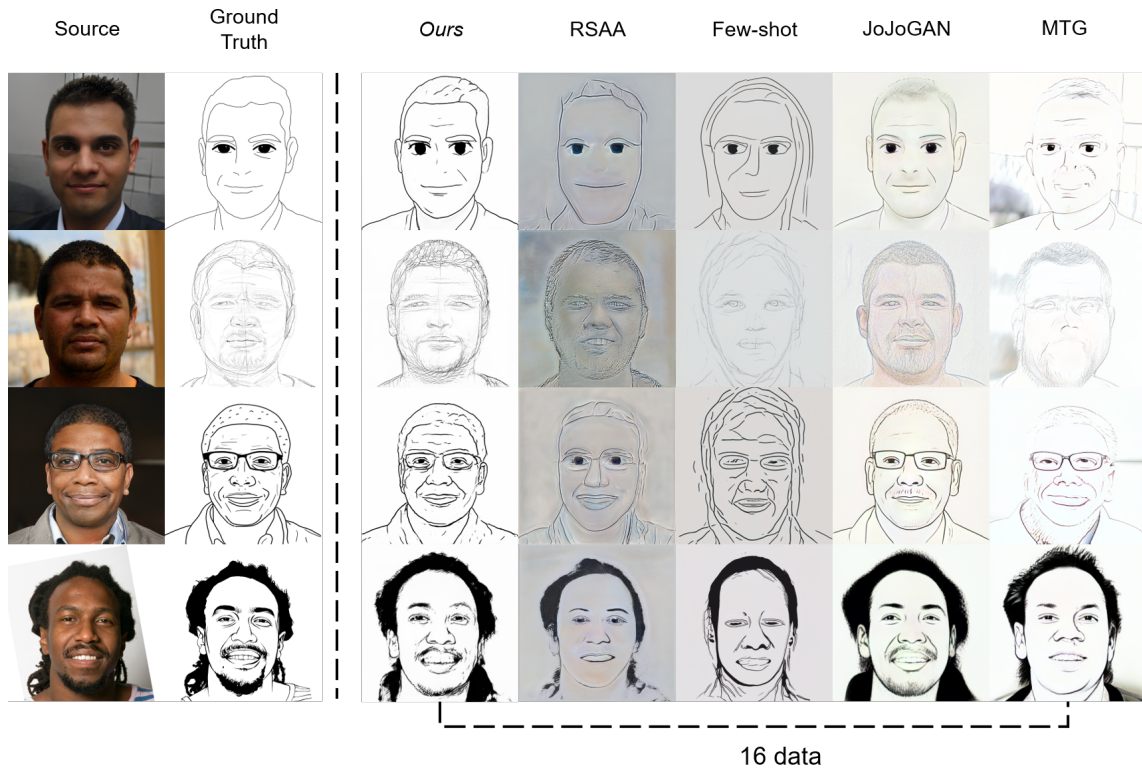


Figure 6: Comparison with domain adaptation baseline methods. All methods are trained with 16 data. Our method produced the best quality results among all the methods while other methods suffer from identity shift or color bleeding.

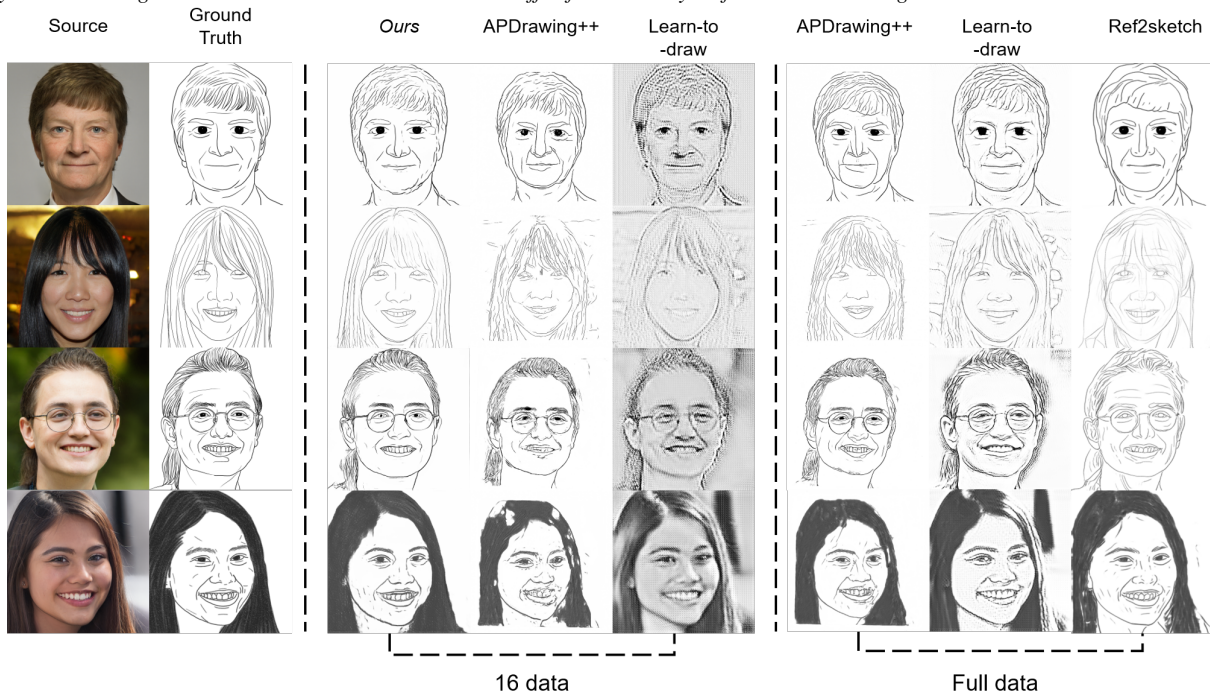


Figure 7: Comparison with sketch extraction baseline methods. Methods on the left are trained with a limited number of data (16) and on the right are trained with the full data (from 70 to 88 while 560 for Ref2sketch). Our method produced the best quality results among the methods trained with a limited number of data. The quality of our results is also comparable to that of the results produced by the models trained with the full data.

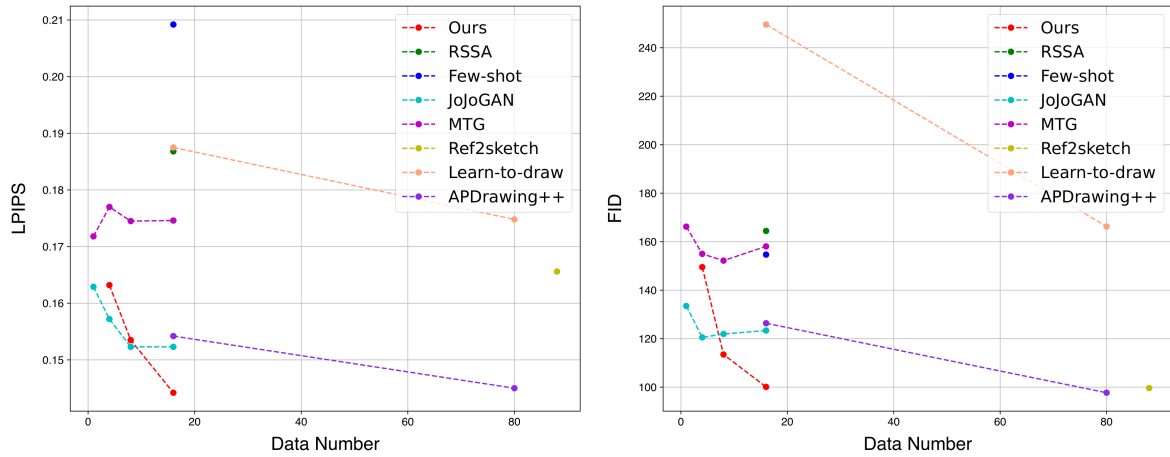


Figure 8: Quantitative results. The left and right plots show the LPIPS and FID scores, respectively.

meaning of images to enhance the quality of output. These methods are not specialized on generating facial sketches; therefore, we also chose APdrawing++ [YXL*20] as another baseline. This method also utilizes a pretrained model related to face image features and is specially designed to generate face sketch images.

For our analysis, we used three different face sketch datasets: SKSF-A, APdrawing [YLLR19], and CUFS [WT08]. SKSF-A consists of seven different styles, and we trained the models separately for each style, except for Ref2sketch, which was designed to train various styles within a single model. To ensure a fair comparison with the state-of-the-art method APdrawing++, we aligned all testing images using five facial landmarks and transformed them to match the control points predefined by APdrawing++. Additionally, we trained Learn-to-draw and APdrawing++ using two different datasets: one with a limited number of data, similar to our method (16 images), and the other with the full dataset (70-88 images) as originally provided. Few-shot was trained only with a limited number of data reflecting the specific characteristic of the model. Similarly, Ref2sketch was trained with the full dataset. Other settings such as optimization and the number of iterations were identically employed according to the descriptions in the paper and code of each method. While *Ours* and other domain adaptation methods can extract sketches in high resolution of 1024×1024 , other methods do so in lower resolutions. Specifically, APdrawing++ produces results in 512×512 , and Ref2Sketch and Learn-to-draw produce results in 256×256 .

As evaluation metrics, we used LPIPS [ZIE*18] for perceptual similarity and FID [HRU*17] for distribution similarity. Because FID requires a large number of test sets, we used it as a secondary indicator, following the approach taken in previous sketch extraction research [YLLR19, YXL*20, ASK*22, SAN23]. Figs. 6 and 7 show the example results, and Fig. 8 shows evaluation scores. *Ours* achieved the best scores when a limited number of data were used, outperforming the baseline methods including the ones trained with

full data. We provide the tabulated complete results of all nine styles in Table 2 of the supplementary material. More examples from the methods [YJLL22, SAN23, GAA*22, RLJ*23, ZA23] that could not generate sketches in a trained style or produced severe artifacts are presented in Sec. 4.3 of the supplementary material.

5.3. Perceptual Study

We performed a perceptual study to evaluate the output sketches based on human perception with different sketch extraction methods. We recruited 53 participants for a survey that compared generated sketches with test images from three different datasets: SKSF-A, CUFS, and APdrawing. Because we did not prioritize any particular groups of individuals, our assessments should not be influenced by the details of any demographic distribution. A total of 20 comparison tasks for randomly chosen results were presented to each participant. Among the 20 comparisons, 16 comparison data were from the SKSF-A dataset that consists of 7 different styles, 3 comparison data were from the APdrawing dataset, and 1 comparison data was from the CUFS dataset. The participants were presented with a ground truth target sketch image along with four generated sketches for each comparison. Note that all of the baseline methods from sketch extraction methods [ASK*22, YXL*20, CDI22] were trained using the full dataset, whereas our method was trained using only a limited number of data.

The participants were asked to choose the sketch that most closely resembled the target sketch image. There was no time limit imposed on the participants during this process. An example of survey is depicted in Fig. 9, and the resulting scores are listed in Table 2. The outcome of this perceptual study demonstrates that our method trained with a limited number of data produced results that were dominantly selected by the participants as that most similar to the target sketch image despite the fact that the competing baselines were trained with the full data.

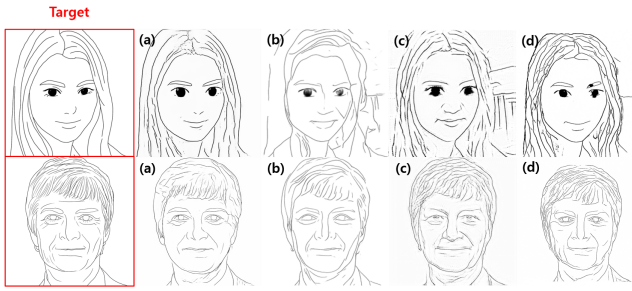


Figure 9: Comparison sample of our perceptual study. The sketch results shown above were produced by our method and three baseline methods: (a) Ours, (b) Ref2sketch, (c) Learn-to-draw, and (d) APDrawing++. The order in which these results are displayed was randomly selected for each comparison.

Table 2: Results from the user perceptual study. The percentage indicates the selected frequency.

Method	User Score
Ours	42.92%
Ref2sketch	24.43%
Learn-to-draw	11.33%
APDrawing++	21.32%

5.4. Ablation study

To conduct an ablation study, we trained our network with different loss conditions, training schemes, and architecture designs. Specifically, we trained our network without L1 loss initialization, with L1 loss until the end, without attention layers in the generator, with one adversarial loss, and without clip loss and utilizing only the partial features from StyleGAN. Because we did not compare the results with those from APdrawing++ for ablation, we did not align the images to the landmark control points that were pre-defined by APdrawing++, unlike the comparisons performed with baselines. See Table 3 for the resulting scores which are averaged over all nine styles. Table 3 in the supplementary material provides a comprehensive presentation of individual quantitative results for all nine styles.

In the ablation study that focused on initialization, Ours clearly outperformed Ours without L1 initialization and Ours with L1 until the end, which is also verified visually in Fig. 4. Ours without L1 initialization required a significantly larger number of iterations to generate images in the sketch domain compared to Ours with L1 until the end or Ours. While Ours and Ours with L1 until the end were able to produce images that can be considered in the sketch domain within approximately 2000 iterations, Ours without L1 initialization failed to do so even after 4000 iterations. This difficulty in optimizing the output into the sketch domain leads to poor image quality in the final result.

Ours without attention and Ours with one adversarial loss resulted in cut lines or deleted information in the hair and mustache regions. Additionally, when Ours used only the first half of the 18 available features, it generated results with missing information. Ours that used a deep fusion module without f_{i-1} , last features, and middle ten features generated reasonable sketches, albeit with degraded quality, especially in the first style. Overall, Ours outper-

Table 3: Ablation study using LPIPS and FID scores, average of nine styles.

Methods	LPIPS↓	FID↓
Ours	0.1772	81.39
Ours W/O L_1 initialization	0.1837	94.98
Ours W L_1 until the end	0.1885	113.45
Ours W/O attention	0.1805	92.80
Ours W one adversarial loss	0.1861	94.06
Ours W/O L_{clip}	0.1855	96.01
Ours deep fusion module W/O f_{i-1}	0.2064	120.15
Ours W first half features	0.2452	192.92
Ours W last half features	0.1974	98.40
Ours W middle ten features	0.1929	102.40

formed other alternatives in terms of the measured scores, which is reflected in the visual results, as shown in Fig. 10. The results produced by Ours show superior quality in terms of the completeness of the generated lines or faithful reconstruction of local features, while the results of the alternatives suffer from missing lines and noise.

6. Application

6.1. Sketch generation in different domain

Although our explanation has focused on the extraction of sketches from face images, StyleSketch can easily extend to handle images from various other domains such as dogs or outdoor scenes as long as related deep features can be obtained from a pre-trained StyleGAN. We tested StyleSketch with StyleGAN pre-trained with AFHQ-dog [CUIH20], LSUN-horse, and LSUN-church [YSZ*15]. We first modified the loss functions for both \mathcal{L}_{adv} and \mathcal{L}_{perc} . While methods to parse facial parts are available, there are not many methods for segmenting parts for churches or animal faces; thus, we modified \mathcal{L}_{adv} to use three discriminators, each of which is specialized for the background, foreground, and full sketch. In addition, we did not use the part-based \mathcal{L}_{perc} and instead used the full image to compute \mathcal{L}_{perc} . With these modifications, we were able to train the sketch generator with the same number of limited data pairs which consequently enables us to perform sketch extraction, as shown in Fig. 11. Sketches in each column reflect the style of the corresponding training data.

6.2. Sketch to face image generation

Photo-realistic images can be generated from a face sketch by inverting the given sketch to the latent space of pre-trained StyleGAN. Finding the correct latent code for a given sketch is a non-trivial task. We first tested an optimization-based method [AQW20] for the purpose. While the optimized code was able to generate a sketch similar to the given sketch, the latent code did not lie inside the valid space of pre-trained StyleGAN. Thus, we opted to train an encoder that inverts a sketch image to a latent code. For this, we first generated a huge number of paired face and sketch image data with StyleGAN G and the sketch generator G_{sketch} . Following the training strategy of pSp [RAP*21], the trained encoder with

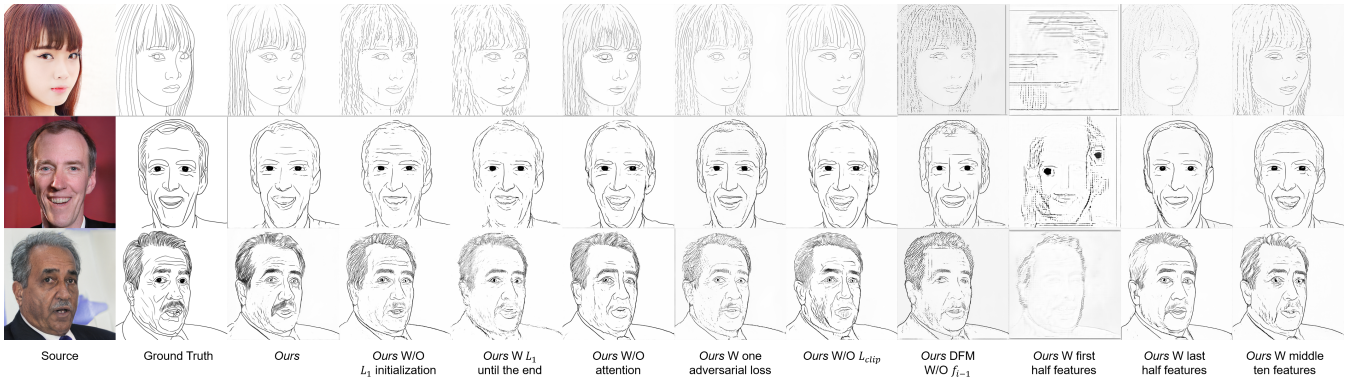


Figure 10: Visual examples of the ablation study. *Ours* generates robust results to different styles and draws the hair and mustache more accurately compared to the alternatives. Without L_1 initialization and with L_1 until the end, incomplete lines or artifacts on the face are generated. Without attention, the mustache and wrinkles disappear. With one adversarial loss, the hair area is not drawn the same as the ground truth. Without clip, noticeable artifacts on the face and hair appear. Utilizing only partial features cannot properly generate the first style.

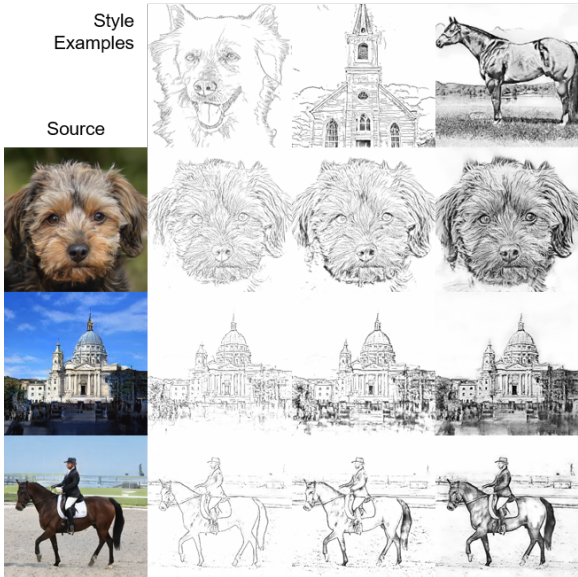


Figure 11: Results of StyleSketch in different domains. For each style, G_{sketch} was trained with 16 data pairs of images generated from StyleGAN and the corresponding sketches extracted using a previous method [SAN23].

the pretrained StyleGAN can successfully generate photo-realistic face images given an input sketch, as shown in Fig. 12. Because our method can generate pairs of face images and sketches in different styles, the trained encoder can invert the sketch, matching the identity and characteristics of both the sketch and the original source image.

We emphasize that the ability of the encoder to successfully convert diverse style sketches to photo-realistic images is due to the use of images generated by StyleSketch. Furthermore, because the sketch does not convey the color information, it can be multi-modal. Therefore, the predicted latent code can be mixed with a

desired style code to generate photo-realistic face images of various styles.

6.3. Semantic editing

As in image manipulation tasks, users may want to edit their sketches to achieve semantically different results, such as increasing the degree of smiling or facing a different direction. While previous methods focus only on extracting the sketch from an image, StyleSketch can easily extend to perform semantic editing using the property of the pretrained latent space. To achieve this, we extract the latent code from the sketch image, as explained in Sec. 6.2. The extracted latent code is manipulated by adding a known semantic latent direction [SGTZ20, HHL20] followed by generating the edited image using G_{sketch} . Examples of edited sketches are shown in Fig. 13. We observed that while the identity and style of the sketch are preserved, semantics such as age, pose, and expression can be effectively manipulated.

7. Limitations and Future Work

StyleSketch is capable of extracting high-resolution sketches from an input face image after training with a limited number of data samples. Because StyleSketch are trained with few data, it occasionally produces blurry sketches or noisy results when discrepancy between the training and test images exists. In addition, because StyleSketch utilizes deep features from a pretrained StyleGAN, the quality of inversion is crucial for effective training. If the inversion is not performed correctly, G_{sketch} can start learning incorrectly from the beginning. This can also pose a problem at inference, because the same inversion is performed during training and inference. This challenge originates from the GAN model and its inversion techniques. Because a distortion-perception trade-off exists when inverting an image to w^+ latent space [TAN*21], a method that avoids this trade-off can reduce the dependency on the quality of inversion. For example, estimating the deep features directly during inversion, as proposed in the Feature-Style Encoder

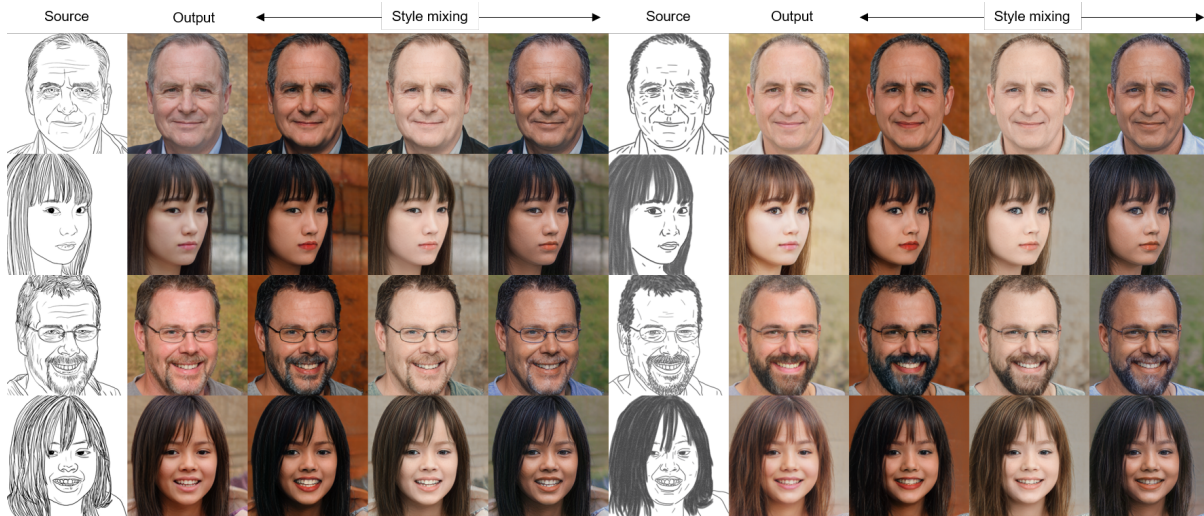


Figure 12: Synthesizing multiple photo-realistic faces from sketch images by training pSp [RAP*21] with StyleSketch synthetic images.

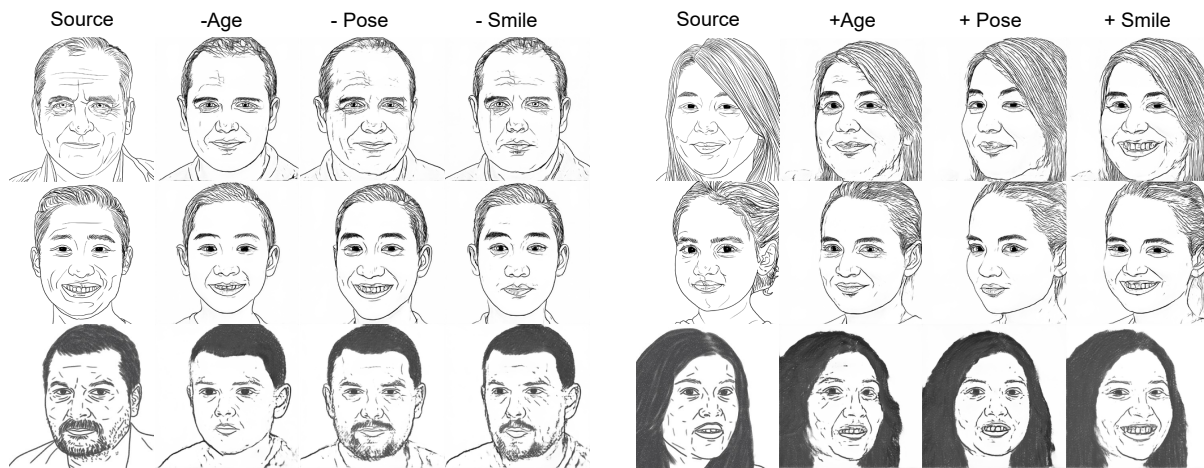


Figure 13: Semantic editing of face sketches. Given a sketch as input, semantic editing can be applied.

[YNGH22], and fusing them with the generated features could be a possible solution. We leave this as an intriguing direction for future research.

Another possible direction is to utilize distillation by training a new network to skip the inversion. Using paired sketch and face images generated from StyleSketch, an image-to-image translation network can be trained with these synthetic data [VIK20]. Through distillation, the newly trained network will not be influenced by inversion as it directly utilizes the input image.

8. Conclusion

In this study, we present a novel approach for extracting high-resolution stylized sketches from a source face image after training the network with a limited number of paired data. Unlike previous approaches, which directly extract sketches from images, thus providing limited information that makes training deep neural net-

works with a small number of paired data difficult, our method boosts the information from the inversion and deep feature extraction process. In addition, StyleSketch incorporates spatial and channel features through the implementation of a newly proposed deep fusion module. We also introduced a two-stage learning technique that is specifically designed to enable the model to be trained for sketch extraction.

We validated the effectiveness of our method through a comprehensive set of both quantitative and qualitative evaluations. Although our explanations focused on sketch extractions in the face domain, we also showed that our method can be extended into other domains such as dog and building images using limited data. Leveraging our method, sketch-based applications such as face image generation from sketches and semantic editing is also possible without a large dataset. We believe that our method can be extensively utilized across various industries, through diverse applications, and can stimulate further research.

Acknowledgement

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2023 (Project Name: Development of Universal Fashion Creation Platform Technology for Avatar Personality Expression, Project Number: RS-2023-00228331, Contribution Rate: 100%).

References

- [AQW20] ABDAL R., QIN Y., WONKA P.: Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8296–8305. [2](#), [3](#), [4](#), [9](#)
- [ASK*22] ASHTARI A., SEO C. W., KANG C., CHA S., NOH J.: Reference based sketch extraction via attention mechanism. *ACM Transactions on Graphics (TOG)* **41**, 6 (2022), 1–16. [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
- [AZMW21a] ABDAL R., ZHU P., MITRA N. J., WONKA P.: Labels4free: Unsupervised segmentation using stylegan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 13970–13979. [2](#), [3](#)
- [AZMW21b] ABDAL R., ZHU P., MITRA N. J., WONKA P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.* **40**, 3 (2021). [3](#)
- [BDS18] BROCK A., DONAHUE J., SIMONYAN K.: Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018). [3](#)
- [BSM*13] BERGER I., SHAMIR A., MAHLER M., CARTER E., HODGINS J.: Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG)* **32**, 4 (2013), 1–12. [6](#)
- [Can86] CANNY J.: A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 6 (1986), 679–698. [2](#), [3](#)
- [CDI22] CHAN C., DURAND F., ISOLA P.: Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 7915–7925. [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
- [CEL12] CELSYS: Clip studio paint. <https://www.clipstudio.net/>, 2012. [6](#)
- [CF22] CHONG M. J., FORSYTH D.: Jojogan: One shot face stylization. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI* (2022), Springer, pp. 128–152. [2](#), [6](#)
- [CFHR17] COURTY N., FLAMARY R., HABRARD A., RAKOTOMANONJY A.: Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems* **30** (2017). [3](#)
- [CSG*20] CHEN S.-Y., SU W., GAO L., XIA S., FU H.: Deepfacedrawing: Deep generation of face images from sketches. *ACM Transactions on Graphics (TOG)* **39**, 4 (2020), 72–1. [2](#), [4](#)
- [CUYH20] CHOI Y., UH Y., YOO J., HA J.-W.: Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8188–8197. [9](#)
- [CZB*23] CHANG H., ZHANG H., BARBER J., MASCHINOT A., LEZAMA J., JIANG L., YANG M.-H., MURPHY K., FREEMAN W. T., RUBINSTEIN M., ET AL.: Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704* (2023). [3](#)
- [FHZ*22] FAN D.-P., HUANG Z., ZHENG P., LIU H., QIN X., VAN GOOL L.: Facial-sketch synthesis: A new challenge. *Machine Intelligence Research* **19**, 4 (2022), 257–287. [5](#), [6](#)
- [FTET21] FOX G., TEWARI A., ELGHARIB M., THEOBALT C.: Stylevideogan: A temporal generative model using a pretrained stylegan, 2021. [3](#)
- [FVRA21] FARAHANI A., VOGHOEI S., RASHEED K., ARABNIA H. R.: A brief review of domain adaptation. *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020* (2021), 877–894. [3](#)
- [GAA*22] GAL R., ALALUF Y., ATZMON Y., PATASHNIK O., BERMANO A. H., CHECHIK G., COHEN-OR D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022). [3](#), [8](#)
- [GPM*22] GAL R., PATASHNIK O., MARON H., BERMANO A. H., CHECHIK G., COHEN-OR D.: Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)* **41**, 4 (2022), 1–13. [3](#)
- [HBB22] HUMAYUN A. I., BALESTRIERO R., BARANIUK R.: Polarity sampling: Quality and diversity control of pre-trained generative networks via singular values. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10641–10650. [6](#)
- [HHLP20] HÄRKÖNEN E., HERTZMANN A., LEHTINEN J., PARIS S.: Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems* **33** (2020), 9841–9850. [10](#)
- [HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33** (2020), 6840–6851. [3](#)
- [HLYJ20] HAN T., LIU C., YANG W., JIANG D.: Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application. *ISA transactions* **97** (2020), 269–281. [3](#)
- [HRU*17] HEUSEL M., RAMSAUER H., UNTERTHINER T., NESSLER B., HOCHREITER S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017). [8](#)
- [IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), pp. 1125–1134. [5](#)
- [JAFF16] JOHNSON J., ALAHI A., FEI-FEI L.: Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14* (2016), Springer, pp. 694–711. [5](#)
- [KAH*20a] KARRAS T., AITTALA M., HELLSTEN J., LAINE S., LEHTINEN J., AILA T.: Training generative adversarial networks with limited data. In *NIPS* (2020). [5](#)
- [KAH*20b] KARRAS T., AITTALA M., HELLSTEN J., LAINE S., LEHTINEN J., AILA T.: Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems* **33** (2020), 12104–12114. [6](#)
- [KKK*22] KIM S., KANG K., KIM G., BAEK S.-H., CHO S.: Dynagan: Dynamic few-shot adaptation of gans to multiple domains. In *SIGGRAPH Asia 2022 Conference Papers* (2022), pp. 1–8. [2](#), [3](#)
- [KLA19] KARRAS T., LAINE S., AILA T.: A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference* (2019). [3](#), [6](#)
- [KLA*20a] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference* (2020). [2](#), [3](#)
- [KLA*20b] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8110–8119. [6](#)

- [LCL*22] LIU F.-L., CHEN S.-Y., LAI Y., LI C., JIANG Y.-R., FU H., GAO L.: Deepfacevideoediting: Sketch-based deep editing of face videos. *ACM Transactions on Graphics* 41, 4 (2022), 167. 2
- [LCWZ19] LI Y., CHEN X., WU F., ZHA Z.-J.: Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Multimedia* (2019), pp. 2323–2331. 2
- [LKL*20] LEE J., KIM E., LEE Y., KIM D., CHANG J., CHOO J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 5801–5810. 2
- [LLK*22] LI D., LING H., KIM S. W., KREIS K., FIDLER S., TORRALBA A.: Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 21330–21340. 2
- [lll17] LLLYASVIEL: sketchkeras. <https://github.com/lllyasviel/sketchKeras>, 2017. 2, 3
- [LLM*19] LI M., LIN Z., MECH R., YUMER E., RAMANAN D.: Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2019), IEEE, pp. 1403–1412. 3
- [LLW17] LI C., LIU X., WONG T.-T.: Deep extraction of manga structural lines. *ACM Transactions on Graphics (TOG)* 36, 4 (July 2017), 117:1–117:12. 3
- [LZG*21] LIN J., ZHANG R., GANZ F., HAN S., ZHU J.-Y.: Any-cost gans for interactive image synthesis and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 14986–14996. 6
- [LZLS20] LI Y., ZHANG R., LU J., SHECHTMAN E.: Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780* (2020). 3
- [MCS20] MO S., CHO M., SHIN J.: Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964* (2020). 3
- [NH19] NOGUCHI A., HARADA T.: Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 2750–2758. 3
- [OLL*21] OIHA U., LI Y., LU J., EFROS A. A., LEE Y. J., SHECHTMAN E., ZHANG R.: Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 10743–10752. 2, 3, 6
- [PHS*18] PORTENIER T., HU Q., SZABO A., BIGDELI S. A., FAVARO P., ZWICKER M.: Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972* (2018). 2
- [PZZ*22] PEEBLES W., ZHU J.-Y., ZHANG R., TORRALBA A., EFROS A. A., SHECHTMAN E.: Gan-supervised dense visual alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 13470–13481. 2, 3
- [RAP*21] RICHARDSON E., ALALUF Y., PATASHNIK O., NITZAN Y., AZAR Y., SHAPIRO S., COHEN-OR D.: Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 2287–2296. 2, 3, 9, 11
- [RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10684–10695. 3
- [RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SAstry G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763. 3, 5
- [RLJ*23] RUIZ N., LI Y., JAMPANI V., PRITCH Y., RUBINSTEIN M., ABERMAN K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 22500–22510. 3, 8
- [SAN23] SEO C. W., ASHTARI A., NOH J.: Semi-supervised reference-based sketch extraction using a contrastive learning framework. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–12. 2, 3, 8, 10
- [SGTZ20] SHEN Y., GU J., TANG X., ZHOU B.: Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 9243–9252. 3, 10
- [SLC*22] SEO J., LEE G., CHO S., LEE J., KIM S.: Midms: Matching interleaved diffusion models for exemplar-based image translation. *arXiv preprint arXiv:2209.11047* (2022). 2
- [SOL*22] SEO K., OH S. W., LU J., LEE J.-Y., KIM S., NOH J.: Style-portraitvideo: Editing portrait videos with expression optimization. 165–175. 3
- [SRL*23] SOHN K., RUIZ N., LEE K., CHIN D. C., BLOK I., CHANG H., BARBER J., JIANG L., ENTIS G., LI Y., ET AL.: Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983* (2023). 3
- [SSG22] SAUER A., SCHWARZ K., GEIGER A.: Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 1–10. 6
- [SSISI16] SIMO-SERRA E., IIZUKA S., SASAKI K., ISHIKAWA H.: Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11. 2
- [SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). 5
- [TAN*21] TOV O., ALALUF Y., NITZAN Y., PATASHNIK O., COHEN-OR D.: Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14. 2, 3, 4, 10
- [TMG*22] TZABAN R., MOKADY R., GAL R., BERMANO A., COHEN-OR D.: Stitch it in time: Gan-based facial editing of real videos. In *SIGGRAPH Asia 2022 Conference Papers* (2022), pp. 1–9. 3
- [TRC*21] TIAN Y., REN J., CHAI M., OLSZEWSKI K., PENG X., METAXAS D. N., TULYAKOV S.: A good image generator is what you need for high-resolution video synthesis. In *International Conference on Learning Representations* (2021). 3
- [VIK20] VIAZOVETSKYI Y., IVASHKIN V., KASHIN E.: Stylegan2 distillation for feed-forward image manipulation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII* 16 (2020), Springer, pp. 170–186. 11
- [VPB*22] VINKER Y., PAJOUHESHGAR E., BO J. Y., BACHMANN R. C., BERMANO A. H., COHEN-OR D., ZAMIR A., SHAMIR A.: Clipasso: Semantically-aware object sketching. *ACM Trans. Graph.* 41, 4 (jul 2022). 3
- [WGGB*20] WANG Y., GONZALEZ-GARCIA A., BERGA D., HER-RANZ L., KHAN F. S., WEIJER J. V. D.: Minegan: effective knowledge transfer from gans to target domains with few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 9332–9341. 3
- [Win11] WINNEMÖLLER H.: Xdog: advanced image stylization with extended difference-of-gaussians. In *Proceedings of the ACM SIGGRAPH/eurographics symposium on non-photorealistic animation and rendering* (2011), pp. 147–156. 2, 3
- [WLW*22] WU S., LIU W., WANG Q., ZHANG S., HONG Z., XU S.: Reffacenet: Reference-based face image generation from line art drawings. *Neurocomputing* 488 (2022), 154–167. 2
- [WT08] WANG X., TANG X.: Face photo-sketch synthesis and recognition. vol. 31, IEEE, pp. 1955–1967. 2, 5, 6, 8

- [WZH*22] WANG Z., ZHENG H., HE P., CHEN W., ZHOU M.: Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262* (2022). 6
- [XAH22] XU Y., ALBAHAR B., HUANG J.-B.: Temporally consistent semantic video editing. *arXiv preprint arXiv:2206.10590* (2022). 3
- [XLW*22] XIAO J., LI L., WANG C., ZHA Z.-J., HUANG Q.: Few shot generative model adaption via relaxed spatial structural alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11204–11213. 2, 3, 6
- [XLY*21] XIANG X., LIU D., YANG X., ZHU Y., SHEN X.: Anime2sketch: A sketch extractor for anime arts with deep networks. <https://github.com/Mukosame/Anime2Sketch>, 2021. 2, 3
- [XT15] XIE S., TU Z.: Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1395–1403. 3
- [XXM*19] XU X., XIE M., MIAO P., QU W., XIAO W., ZHANG H., LIU X., WONG T.-T.: Perceptual-aware sketch simplification based on integrated vgg layers. *IEEE transactions on visualization and computer graphics* 27, 1 (2019), 178–189. 5
- [YJLL22] YANG S., JIANG L., LIU Z., LOY C. C.: Pastiche master: exemplar-based high-resolution portrait style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 7693–7702. 2, 3, 8
- [YLLR19] YI R., LIU Y.-J., LAI Y.-K., ROSIN P. L.: Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10743–10752. 2, 3, 4, 5, 6, 8
- [YLLR20] YI R., LIU Y.-J., LAI Y.-K., ROSIN P. L.: Unpaired portrait drawing generation via asymmetric cycle mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8217–8225. 2, 3, 6
- [YNGH21] YAO X., NEWSON A., GOUSSEAU Y., HELLIER P.: A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 13789–13798. 3
- [YNGH22] YAO X., NEWSON A., GOUSSEAU Y., HELLIER P.: Feature-style encoder for style-based gan inversion. *arXiv preprint arXiv:2202.02183* (2022). 11
- [YSZ*15] YU F., SEFF A., ZHANG Y., SONG S., FUNKHOUSER T., XIAO J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365* (2015). 9
- [YWP*18] YU C., WANG J., PENG C., GAO C., YU G., SANG N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 325–341. 4, 5
- [YXL*20] YI R., XIA M., LIU Y.-J., LAI Y.-K., ROSIN P. L.: Line drawings for face portraits from photos using global and local structure based gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 10 (2020), 3462–3475. 2, 3, 8
- [ZA23] ZHANG L., AGRAWALA M.: Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023). 3, 8
- [ZAFW21] ZHU P., ABDAL R., FEMIANI J., WONKA P.: Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. *arXiv preprint arXiv:2110.08398* (2021). 2, 3, 6
- [ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 586–595. 8
- [ZLG*21] ZHANG Y., LING H., GAO J., YIN K., LAFLECHE J.-F., BARRIUSO A., TORRALBA A., FIDLER S.: Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 10145–10155. 2, 3
- [ZW20] ZHANG W., WU D.: Discriminative joint probability maximum mean discrepancy (djp-mmd) for domain adaptation. In *2020 international joint conference on neural networks (IJCNN)* (2020), IEEE, pp. 1–8. 3
- [ZWT11] ZHANG W., WANG X., TANG X.: Coupled information-theoretic encoding for face photo-sketch recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2011), pp. 513–520. 2, 5, 6