SketchInverter: Multi-Class Sketch-Based Image Generation via GAN Inversion

Zirui An[*]
Beihang University

Jingbo Yu[*]
Beihang University

Runtao Liu
Johns Hopkins University

Chuang Wang
Beihang University

Qian Yu[†]
Beihang University

Abstract

This paper proposes the first GAN inversion-based method for multi-class sketch-based image generation (MC-SBIG). MC-SBIG is a challenging task that requires strong prior knowledge due to the significant domain gap between sketches and natural images. Existing learning-based approaches rely on a large-scale paired dataset to learn the mapping between these two image modalities. However, since the public paired sketch-photo data are scarce, it is struggling for learning-based methods to achieve satisfactory results. In this work, we introduce a new approach based on GAN inversion, which can utilize a powerful pretrained generator to facilitate image generation from a given sketch. Our GAN inversion-based method has two advantages: 1. it can freely take advantage of the prior knowledge of a pretrained image generator; 2. it allows the proposed model to focus on learning the mapping from a sketch to a low-dimension latent code, which is a much easier task than directly mapping to a high-dimension natural image. We also present a novel shape loss to improve generation quality further. Extensive experiments are conducted to show that our method can produce sketch-faithful and photo-realistic images and significantly outperform the baseline methods.

1. Introduction

Human free-hand sketch is an intuitive and powerful visual expression. In recent years, sketch has received increasing attention from both computer vision and computer graphics communities, and many sketch-related tasks have been investigated, such as recognition [16, 57], sketch parsing [39, 40], sketch-based retrieval for 2D images [32, 47, 49, 56] or 3D shapes [7, 12, 55, 59], and sketch-based 2D image generation [10, 18, 19, 33, 34, 54] or 3D shape generation [22, 35, 48]. Among these tasks, **S**ketch-**B**ased **I**mage **G**eneration (SBIG) is popular given its wide applications in animation, fashion, and education. With the development of the Generative Adversarial Network (GAN) [20] and its variants [24, 37, 62, 63], this task has embraced significant improvements.

Object-level SBIG aims to generate a object-level photo-realistic image based on a sketch automatically. Specifically, this task can be categorized into two types: single-class and multi-class. For the single-class setting, a model is designed to handle a specific target class. For example, *DeepFaceDrawing* [8] and *DeepFacePencil* [30] use sketches as soft constraints to control face image generation; the work [33] proposes a two-stage approach to generate fashion images (e.g., shoes) from sketches. In contrast, multi-class SBIG focuses on producing images of multiple classes, such as EdgeGAN [18] and SketchyGAN [10].

Generating photos[*] based on sketches is very challenging due to the large domain gap between photos and sketches, resulting from two intrinsic characteristics: (1) sketches are abstract and deformed, and (2) sketches lack colors and most texture information. In contrast, photos are faithful to objects in the real world. Therefore, to synthesize an image for a given sketch, a model needs to rectify shape deformation in a sketch and fill in missing colors and textures, which requires lots of prior knowledge. Learning prior knowledge is non-trivial: existing datasets are barely sufficient for learning single-class SBIG models but cannot support multi-class. This problem explains why the synthesis quality of multi-class SBIG models [10, 18, 19, 34, 54] is much worse than single-class models [8, 30, 33]. To alleviate the domain gap between photos and sketches, EdgeGAN [18] learns a joint semantic embedding for these two domains. The recent work [54] adopts CycleGAN [62] as the baseline and proposes an open-domain optimization strategy; thus it can generalize to open-domain classes when training data is relatively limited. Unfortunately, the quality of the images generated by these methods is still far from satisfactory.

[*] These authors contributed equally to this work.
[†] Corresponding author.

[*] In this paper, photos and natural images are used interchangeably.

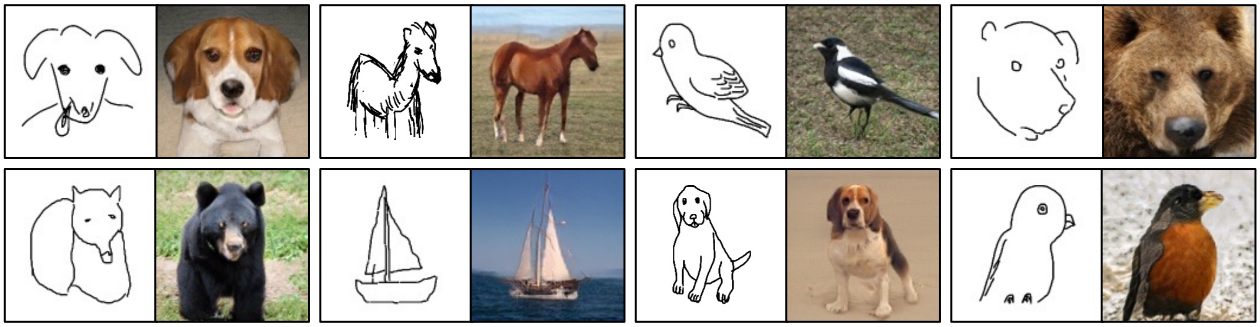

Figure 1. Natural images generated by our proposed method, *SketchInverter*, based on free-hand sketches. Sketches are randomly sampled from the SketchyCOCO dataset [18] and Sketchy database [47] test set.

Although the large-scale sketch-photo datasets are scarce, it is worth noting that large-scale photo datasets are available (e.g., ImageNet), which can be used to train a well-performed multi-class image generator. The straightforward idea is to leverage the prior knowledge learned by a powerful image generator. GAN Inversion is a commonly-used method for image editing, where a pretrained GAN model is employed to produce images. At the same time, the input vector is dynamically tuned (for optimize-based) or learned (for learning-based) so that the generated images can match the expectation. The GAN Inversion method guarantees high quality of synthesized images since it utilizes a pretrained GAN model. Intuitively, the prior knowledge of a pretrained GAN model also significantly reduces the domain gap between sketches and photos.

In this work, we for the first time introduce GAN inversion to multi-class SBIG, aiming to obtain high-quality photos for given sketches (as shown in Fig. 1). As shown in Fig. 2(a), we adopt the learning-based GAN inversion approach. Specifically, with an image generator pretrained on a large-scale image dataset, we additionally train a sketch encoder to map a sketch into the latent space of the image generator (as shown in Fig. 2(b)). Compared with existing multi-class SBIG models, such design has two advantages: (1) it realizes the task in two steps, first converting the input sketch to a latent code and then generating a realistic image based on the latent code. With a pretrained image generator, the model can focus on the first step. (2) It leverages the prior knowledge of the pretrained image generator, ensuring the quality of the synthesized photos.

Specifically, we choose to invert a well-trained GAN model, i.e., BigGAN [6] in our experiments due to its impressive performance in image generation. Unlike the conventional GAN inversion-based methods, our task requires generating photos of multiple classes. Many existing GAN inversion works focus on the single-class setting, while only a few works consider additional conditions, such as ICGAN [38]. ICGAN uses two encoders: one maps an input image to the corresponding latent code, and the other maps it to a class label. However, the generation results can deviate from the target category if the input sketch is ambiguous or the inverted class label is incorrect. Therefore, we design a conditional encoder and take the class label as a condition during the sketch encoding and photo generation. To encourage the generated images to match the input sketches, we introduce a shape loss that minimizes the difference between the input sketch and the contour of a generated photo.

In addition, we build a synthetic dataset to train our model, including paired data of latent codes, images, and sketches. Although our model is trained on synthetic data, extensive experiments show that our model can generalize to real data. The visual quality of generated images is significantly better than other existing works. The contributions of this work are summarized as follows:

- We, for the first time, introduce GAN inversion for object-level multi-class sketch-based image generation. The prior knowledge learned by a pretrained GAN model can significantly reduce the domain gap between sketches and photos.
- We design a conditional encoder to map sketches with class labels to latent space and generate multi-class images through a pretrained GAN model; we additionally propose a shape loss to encourage the generated images to match the input sketches.
- Extensive experiments have been conducted to show that our method can outperform other baseline methods by a noticeable margin.

2. Related Works

Sketch-Based Image Generation. Sketch-based image generation aims to generate a photo-realistic image from a given sketch. Early works such as Sketch2-Photo [9] and PhotoSketcher [17] chose to compose a new photo from photos retrieved for a given sketch. In recent years, with the development of Generative Adversarial Networks(GANs) [20], an increasing number of works adopt

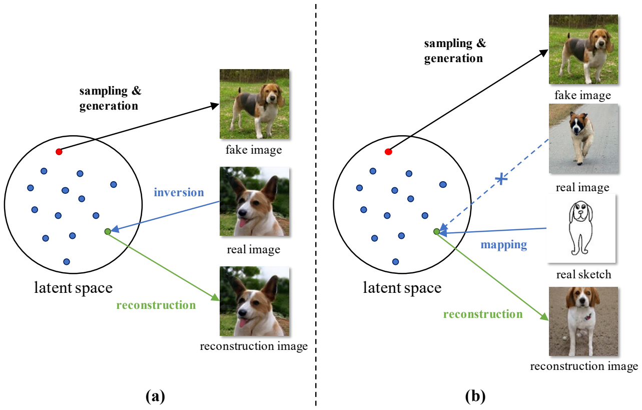

Figure 2. (a) Typical GAN inversion processes: inverting an image to the latent space and reconstructing it. (b) Our method of sketch-based image generation: mapping a sketch to the latent space of a generator and generating an image. Through the reconstruction we train our model to learn this mapping on sketch-photo datasets.

GANs to learn pixel-wise translation from sketches to photos directly. Based on the type of generated images, this task can be divided into three categories: (i) Face image generation [8, 30]; (ii) Scene-level image generation [18]; (iii) object-level image generation [10, 18, 19, 33, 34, 54].

There are two types of object-level sketch-to-image generation: (i) single-class image generation: [33] proposes a two-stage model for the unsupervised sketch-to-photo generation with reference images in a single class. They adopt CycleGAN [62] as a baseline and train in a self-supervision fashion. (ii) multi-class image generation: [10] uses a generator with Masked Residual Unit blocks to generate images from 50 classes. [18] learns images and edge maps jointly into a shared latent space where vectors can encode high-level attribute information from cross-modality data. [54] adopts CycleGAN [62] as the baseline and proposes an open-domain optimization strategy. [19] uses outlines to represent free-hand sketches and generates photos from partial strokes with two-stage generators. ContextualGAN [34] turns sketch-to-image generation into an image completion problem: the network learns the joint distribution of sketches and image pairs and acquires the result by iteratively traversing the manifold. The recent work [51] uses a sketch instance as the supervision to fine-tune a well-trained generator (StyleGAN [26]), aiming to generate sketch-faithful images. Note that our method is different from theirs, our goal is to train an auto-encoder that can map a collection of multi-class sketches to the latent space and generate photos through a well-trained generator [6].

GAN Inversion. GAN inversion is a task that aims to find the corresponding latent code to recover the input image for a fixed well-trained GAN model. GAN inversion enables multiple downstream tasks such as image manipulation, image interpolation, image restoration, 3D reconstruction, and multi-modal learning, which gains increasing attention from the community.

There are three existing types of inversion approaches [53]. (i) Learning-based GAN inversion [3, 38, 61]: this method first generates a collection of images with randomly sampled latent codes and then uses the images and codes as inputs and supervisions, respectively, to train an encoder that maps images to codes. (ii) Optimization-based GAN inversion [1, 2, 11, 31, 36, 42, 50]: this method deals with a single instance at one time by directly optimizing the latent code to minimize the pixel-wise reconstruction loss of the generated image. (iii) Hybrid GAN Inversion [3, 4, 21, 60, 61]: this method combines the methods above by using the encoder to generate an initial latent code for the later optimization. The GAN model to be inverted can be either (i) conditional [38] or (ii) non-conditional. Most existing GAN inversion model belong to (ii).

Cross-modal Image Translation. Currently, the computer vision community pays more and more attention to using large-scale pre-training models to learn cross-modal image translation. [43, 44] use pre-training CLIP [41] to achieve text-based surrealist image generation. NUWA [52] is a multi-modal pre-training model that can generate new data or manipulate existing visual data for various visual tasks. We do not compare to NUWA as it might be unfair due to the difference in the scale of the datasets. The proposed model only takes 16,000 sketch-photo pairs for training while NUWA uses 2.9 million text-image pairs.

3. Method

Our model, SketchInverter, aims to generate photos of multiple classes based on given sketches. It is built on the learning-based GAN inversion, which first learns the mapping from the sketches to the latent space of a conditional GAN (cGAN) model and then utilizes the pretrained cGAN to generate photos that are faithful to the input sketches.

Three designs are specifically presented for this task, including (1) A novel conditional GAN inversion encoder is proposed to map an input sketch into the latent space with a class label as the condition. (2) A novel shape loss is introduced to ensure the faithfulness of the generated image. (3) A synthetic dataset is constructed for training to address the paired data scarcity issue.

3.1. Overall Architecture

The overall architecture of our model is illustrated in Fig. 3. Our model aims to map a given sketch $\mathbf{s}$ to the corresponding latent code $\mathbf{z}$, and then generate a natural image $\mathbf{x}$ that matches $\mathbf{s}$ in some aspects, e.g., pose and orientation. Specifically, our model takes a sketch $\mathbf{s}$ and its corresponding label $\mathbf{y}$ as input and maps them to a latent code $\mathbf{z}$ via the sketch encoder $\mathbf{E}$. Next, a fixed pretrained generator $\mathbf{G}$ generates the corresponding photo given the latent code

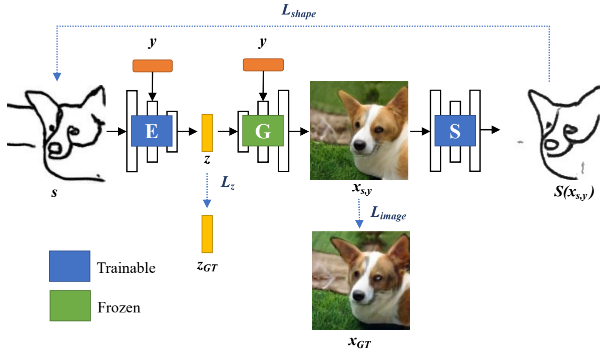

Figure 3. Our framework for multi-class sketch-based image generation. It contains three networks: a conditional encoder $\boldsymbol{E}$, a pretrained generator $\boldsymbol{G}$, and a image-to-sketch network $\boldsymbol{S}$. Model blocks in blue are trainable and green are fixed. Black arrows indicate feed-forward, and blue arrows indicate the supervisions of encoder $\boldsymbol{E}$. $\mathbf{s}$, $\mathbf{z}_{GT}$ and $\mathbf{x}_{GT}$ are ground-truth pairs.

and class label. The ground-truth latent code and image are $\mathbf{z}_{GT}$ and $\mathbf{x}_{GT}$. During training, besides the reconstruction loss, a shape loss is applied to improve the faithfulness of generated images, which compares the differences between a fake sketch $\mathbf{s}_{\mathbf{rec}}$ and the original input sketch $\mathbf{s}$. The fake sketch is converted from the generated photo by a sketch-photo translation network $\mathbf{S}$.

In our proposed model, BigGAN is used as the image generator $\mathbf{G}$. It is a conditional generative adversarial network (cGAN) and is pretrained on the ImageNet [14] dataset. For the original BigGAN, it takes a random noise vector and a class label as the input and outputs an image.

3.2. Conditional Sketch Encoding

Given a sketch, the encoder maps it to the latent space of the pretrained image generator. For multi-class SBIG, a sketch with different class labels should be mapped into different places of the latent space. However, for most learning-based GAN inversion methods, the encoders simply map an image to a latent code without any condition (as shown in Fig. 4(a)). The later work cGAN inversion [38] uses two encoders to predict the latent code and the class label for a given image (see Fig. 4(b)). Unfortunately, our experiments (see Sec. 4.4) show that this design performs poorly for our task as free-hand sketch is abstract and colorless, and thus the encoder often fails to predict the class labels correctly.

In this work, for the first time, we propose a conditional encoder $\mathbf{E}$ for learning-based GAN inversion method to predict a *class-conditioned* latent code $\mathbf{z}_{\mathbf{y}} = \mathbf{E}(\mathbf{s}, \mathbf{y})$. As shown in Fig. 4(c), the conditional encoder includes 6 residual blocks and the category information of free-hand sketch is injected into the encoder via class-conditional BatchNorm [13, 15]. The detailed architecture is provided in Supplementary. Next, the latent code $\mathbf{z}_{\mathbf{y}}$ will be the input of the pretrained image generator $\mathbf{G}$.

To train this conditional sketch encoder, we apply the L1 loss between the ground-truth latent code $\mathbf{z}_{GT}$ and the predicted latent code $\mathbf{z}_{\mathbf{y}}$:

$$\min_{\Theta_E} \mathcal{L}_z = \left\|\mathbf{z}_{GT} - E\left(\mathbf{s}, \mathbf{y}\right)\right\|_1 \tag{1}$$

3.3. Image Generation

Our proposed model uses the BigGAN network as the image generator $\mathbf{G}$. So given the latent code $\mathbf{z}_{\mathbf{y}}$ and class label $\mathbf{y}$, image generator will produce the coresponding image, $\mathbf{x}_{\mathbf{s},\mathbf{y}} = \mathbf{G}(\mathbf{z}_{\mathbf{y}}, \mathbf{y}) = \mathbf{G}(\mathbf{E}(\mathbf{s}, \mathbf{y}), \mathbf{y})$. Since it is pretrained on the ImageNet dataset, our proposed model can leverage its prior knowledge and generate high-quality images. During training, the parameters of the BigGAN are fixed. Besides, image reconstruction loss and shape loss are used during training to ensure the quality and faithfulness of the generated images.

3.3.1 Image Reconstruction Loss

We adopt an image reconstruction loss to guarantee that the generated image is similar to the target image (Eq. 2). Like previous GAN inversion works, we calculate pixel-wise distance and perceptual-wise distance between $\mathbf{x}_{\mathbf{s},\mathbf{y}}$ and $\mathbf{x}_{GT}$. We additionally introduce LPIPS [58] loss based on the feature extracted by AlexNet [28] since LPIPS loss has been proven to preserve better image quality [21, 45] than perceptual loss [25].

$$\begin{aligned}\min_{\Theta_E} \mathcal{L}_{image} &= \left\|\mathbf{x}_{GT} - G\left(E\left(\mathbf{s}, \mathbf{y}\right), \mathbf{y}\right)\right\|_1 \\ &+ \lambda_{LPIPS}\left\|F\left(\mathbf{x}_{GT}\right) - F\left(G\left(E\left(\mathbf{s}, \mathbf{y}\right), \mathbf{y}\right)\right)\right\|_1\end{aligned} \tag{2}$$

where $\mathbf{F}$ denotes a pretrained AlexNet [28].

3.3.2 Shape Loss

Only the reconstruction loss cannot guarantee that the generated image and the input sketch are similar in pose and orientation. We suppose that this may due to the loss in the image domain that can not fully promise the shape and details of generating images. To make the content of the generated image more aligned with the input sketch, the overall learned mapping should be cycle-consistent and we introduce a shape loss as supervision. The shape loss is applied between the input sketch and the fake sketch generated by a trainable photo-to-sketch translation network, $\mathbf{S}$. $\mathbf{S}$ and $\mathbf{E}$ are jointly optimized by the shape loss:

$$\min_{\Theta_S} \min_{\Theta_E} \mathcal{L}_{shape} = \left\| \mathbf{s} - S\left(\mathbf{x}_{GT}\right) \right\|_1 + \left\| F\left(\mathbf{s}\right) - F\left(S(\mathbf{x}_{GT})\right) \right\|_1 + \left\| \mathbf{s} - S\left(G\left(E\left(\mathbf{s}, \mathbf{y}\right), \mathbf{y}\right)\right) \right\|_1 + \left\| F\left(\mathbf{s}\right) - F\left(S\left(G\left(E\left(\mathbf{s}, \mathbf{y}\right), \mathbf{y}\right)\right)\right) \right\|_1 \tag{3}$$

L1 norm distance is used as it achieves the best performance in our preliminary experiments.

3.3.3 Full Objective

The overall objective of our model is:

$$\mathcal{V}(E, S) = \lambda_z \mathcal{L}_z + \lambda_{image} \mathcal{L}_{image} + \lambda_{shape} \mathcal{L}_{shape} \tag{4}$$

where $\lambda_z, \lambda_{image}$ and λ_{shape} control the weights of different loss terms. In the ablation study section, we compare different variants of full objective and show that each term contributes to the model's performance.

3.4. Synthetic Dataset

To map sketches into the latent space of pretrained BigGAN properly, we need large amounts of paired sketches and images. The images of existing multi-class sketch-photo datasets [18, 47] are limited and lack enough diversity so that they cannot cover the generation space of a pretrained BigGAN.

To address the data issue, we build a synthetic dataset composed of pairs of images, latent codes, and sketches. Specifically, we first select 16 categories from ImageNet 1,000 classes, e.g., birds, dogs, planes, sailboats. Then we sample a collection of latent codes $\mathbf{z}_{GT}$ from prior distribution $\boldsymbol{p}$. Next, we obtain the images $\mathbf{x}_{GT}$ through pretrained generator $\boldsymbol{G}$. Finally, the corresponding sketches $\mathbf{s}$ of these images are obtained using a pretrained photo-to-sketch network [29]. This synthetic dataset consists of 12,000 paired latent codes $\mathbf{z}_{GT}$, images $\mathbf{x}_{GT}$, and sketches $\mathbf{s}$.

3.5. Training Strategy

In the previous sections, we introduce the loss, Eq. 4, under the setting of training on synthetic data only. To explore whether we can obtain better results by utilizing the real dataset, such as SketchyCOCO [18] dataset, we design and compare three training strategies: (i) Training on our synthetic dataset and directly testing on the real dataset. (ii) Training on our synthetic dataset and fine-tuning on the real dataset. (iii) Training from scratch on the mix of our synthetic dataset and the real dataset. In Sec. 4.4, we compare the results of different training strategies. Note that when training or fine-tuning on the real dataset where the ground-truth latent codes are not available, the optimization objective is:

$$\mathcal{V}(E, S) = \lambda_{image} \mathcal{L}_{image} + \lambda_{shape} \mathcal{L}_{shape} \tag{5}$$

4. Experiments

4.1. Experimental Setup

Datasets and Evaluation Protocal. We train our proposed model, SketchInverter, and baselines on our synthetic dataset. We evaluate them on real free-hand sketch-photo datasets, i.e., Sketchy Database [47] and Sketchy-COCO [18]. Note that when comparing with baselines, all models are trained **only** on the synthetic dataset.

- **Our Synthetic Dataset.** We collected this dataset following the description in Sec. 3.4.
- **Sketchy Database [47].** This dataset includes pairs of images and sketches We choose 8 classes that overlap with our synthetic dataset from the original 125 categories and split them into train and test sets.
- **SketchyCOCO [18].** This dataset includes 14 object classes, We choose 4 classes related to our synthetic dataset and split them into train and test set.

Baseline Methods. We compare our method with three baseline methods, including Pix2pix [24], EdgeGAN [18], and AODA [54].

- **Pix2pix [24].** Pix2pix is proposed for the task of image-to-image translation. Following [18], the model is trained under two modes. The first mode is denoted as **Pix2pix-Sep**, in which 16 models are trained separately for each class. The second mode is denoted as **Pix2pix-Mix**, where only a single model is trained for all 16 classes.
- **EdgeGAN [18].** EdgeGAN is proposed for sketch-based image generation. We train this model using paired sketches and photos of our synthetic dataset. We name the setting of training with sketches instead of edge maps as **EdgeGAN-S**.
- **AODA [54].** AODA proposes a framework that jointly learns sketch-to-photo and photo-to-sketch mappings, This model is also trained on our synthetic dataset.

Implementation Details and Evaluation Metrics. We train SketchInverter for 200 epochs on our synthetic dataset. The learning rate is set to be 0.001. The latent code z is 128-dim and the size of sketches and photos is 128×128. We use Adam[27] optimizer and the batch size is set to be 128. Further implementation details are provided in Supplementary.

We use the following five metrics to evaluate the quality, diversity, and faithfulness of the images generated by different methods, including Fréchet Inception Distance (FID) [23], Kernel Inception Distance (KID) [5], Inception Score (IS) [46], Learned Perceptual Image Patch Similarity (LPIPS) [58], and Classification Accuracy (Acc).

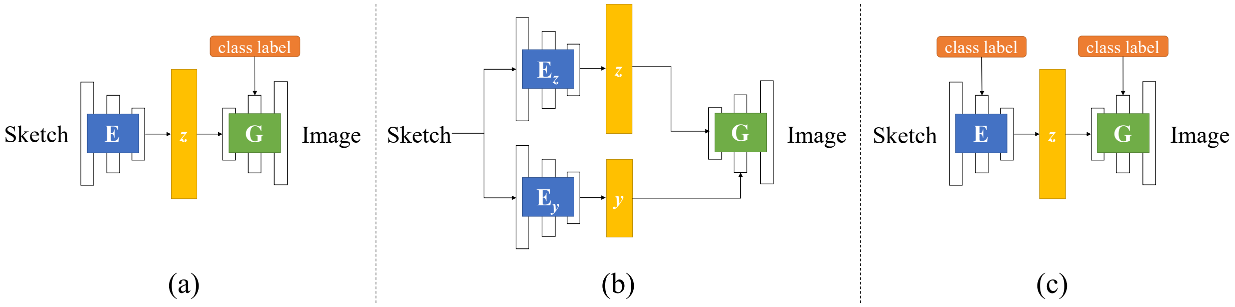

Figure 4. (a) Using one non-conditional encoder to map sketches to latent codes, and generating images based on class labels. (b)Using two non-conditional encoders to map sketches to latent codes and class labels respectively, and using both of them to generate images. $\mathbf{E}_z$ and $\mathbf{E}_y$ output latent codes and class labels, repectively. (c) Our method SketchInverter: using a conditional encoder to map sketches with class labels to latent codes, and generates images through a generator.

Table 1. Comparison of baselines and our method. Our method outperforms other baselines in all metrics.

	Sketchy Database					SketchyCOCO			
Model	FID↓	KID↓	IS↑	LPIPS↓	Acc↑	FID↓	KID↓	IS↑	Acc↑
Pix2Pix-Sep [24]	107.59	0.043	9.04	0.67	0.94	170.06	0.081	6.94	0.125
Pix2Pix-Mix [24]	207.65	0.17	4.97	0.7	0.174	219.39	0.17	4.89	0.123
EdgeGAN-S [18]	182.2	0.017	6.37	0.66	0.81	218.67	0.12	5.54	0.903
AODA [54]	305.8	0.21	4.6	0.68	0.021	315.46	0.2	3.1	0.11
SketchInverter (Ours)	**56.71**	**0.012**	**9.63**	**0.55**	**0.988**	**121.04**	**0.024**	**7.15**	**0.995**

4.2. Qualitative Results

Figure 5 shows representative results of our proposed method, SketchInverter, and four baseline methods. The sketches (1st column) and their corresponding photos (2nd column) are from the Sketchy database. Due to space limitations, we show more visualization results achieved on the SketchyCOCO dataset in Supplementary. It is clear to see that our approach (Fig. 5(c)) can produce significantly higher-quality photos than others.

Handling different classes by a single model is very challenging. As shown in the last three columns, the baseline methods are struggling to generate realistic photos of different classes. Pix2pix-Sep (Fig. 5(d)) works relatively better as it is trained for individual classes. In contrast, SketchInverter is a multi-class model and can handle the task well, even outperforms Pix2pix-Sep. For either animal classes like birds and dogs or other classes like jack-o-lanterns, SketchInverter can generate photos with proper color, texture, and shape. Such superiority is achieved by adopting a pretrained image generator, which allows our model to utilize its prior knowledge. It is worthy to note that our approach has the potential to generate higher-resolution or higher-quality natural images by switching to a more advanced image generator.

In terms of faithfulness, i.e., whether the objects in generated photos are aligned with the sketched objects or not, our method also performs the best among all methods. It should be noted that although some sketch-photo pairs in real datasets are not totally aligned, like the bird example in the second row of Fig. 5, our generated photos are more faithful to the input sketches while maintaining realism. More results are shown in Supplementary.

4.3. Quantitative Results

As shown in Table 1, our method outperforms other baseline methods in terms of all metrics. Pix2pix-Sep is the only baseline that is trained separately for each class. Its performance is much better than that of Pix2pix-Mix, which implies that the in-domain gaps among classes are non-trivial. Our method outperforms Pix2pix-Sep by a large margin, indicating that our model handles the task of MS-SBIG better than a collection of models trained on the single category. Besides, the images generated by SketchInverter can be better recognized by an image classifier than those of other baseline methods, which proves the superiority of our method. These results also demonstrate the benefits of using the pretrained image generator which learns a outstanding prior from large-scale image datasets.

4.4. Ablation Study

Effect of Conditional Encoder. To demonstrate the effectiveness of the proposed conditional encoder, we compare our conditional encoder with two variants: (i) using a simple convolution encoder which takes the sketch as the only input and maps it to a latent code; (ii) using two encoders, similar to [38], the input is the sketch only; while

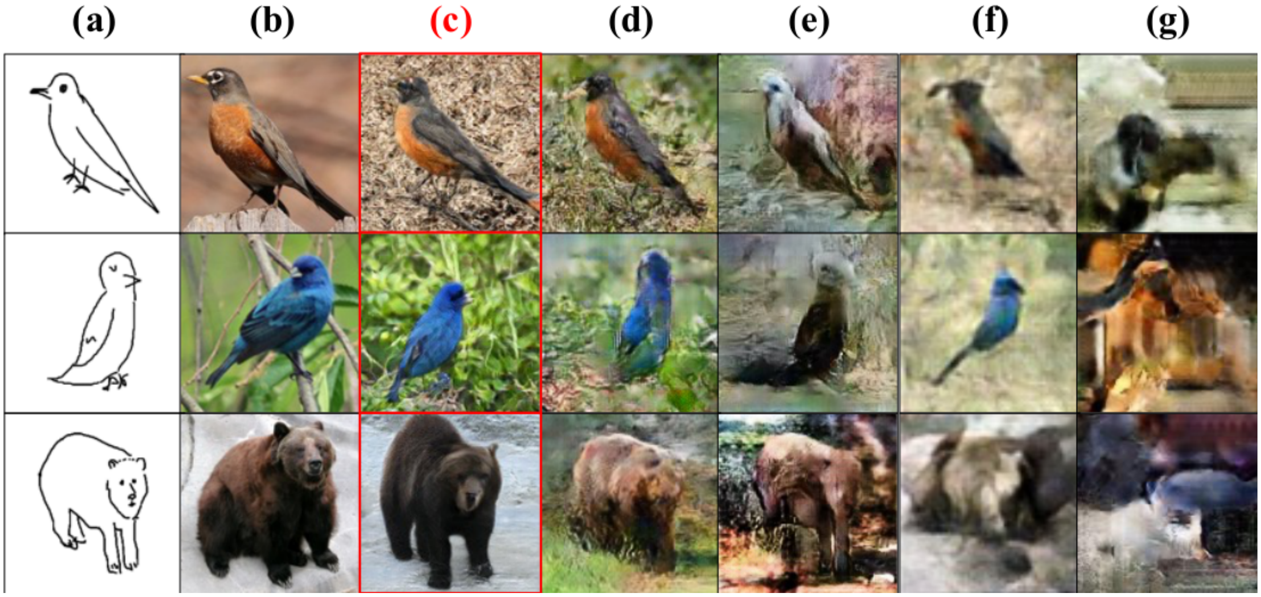

Figure 5. Visualization results tested on sketches from Sketchy database. (a) Sketch; (b) Ground-truth; (c) Our method; (d) Pix2pix-Sep: one model per class; (e) Pix2pix-Mix: a single model for all classes; (f) EdgeGAN-S; (g) AODA. (c)(d)(e)(f)(g) are all trained on our synthetic dataset and tested on the Sketchy database. More results are shown in supplementary.

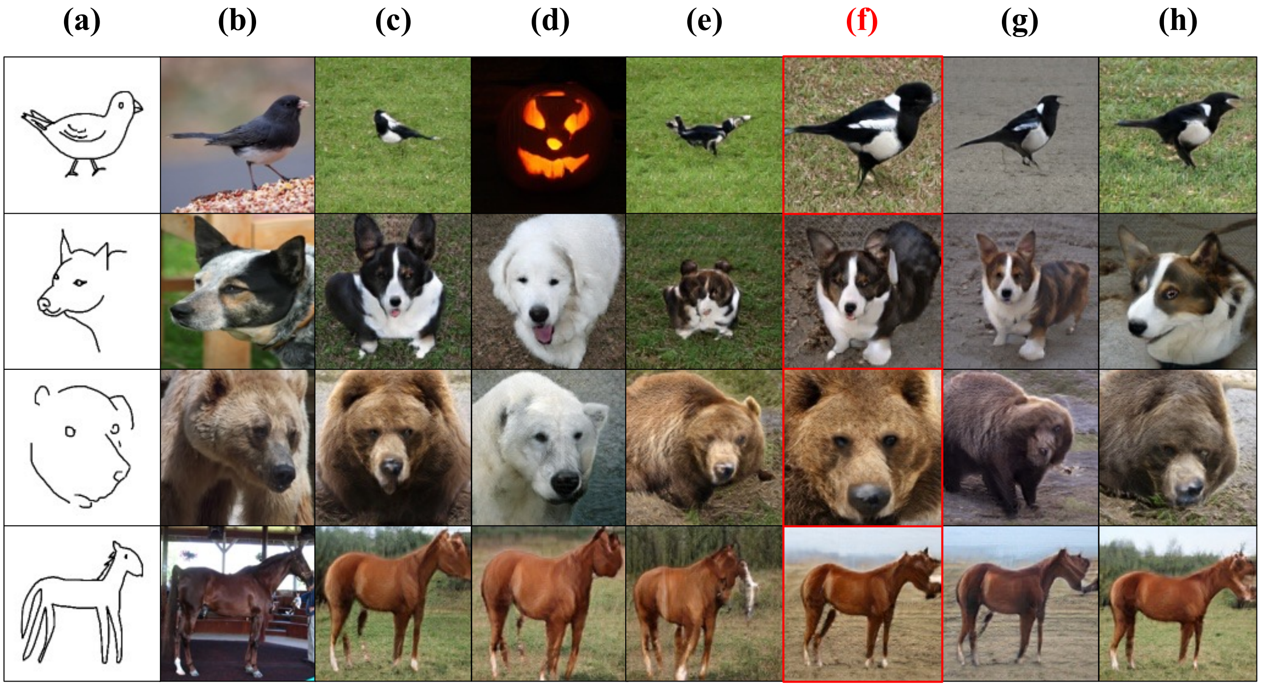

Figure 6. Visualization results of ablation study. (a) sketch; (b) ground-truth; (c) using one non-conditional encoder; (d) using two non-conditional encoder; (e) training without shape loss; (f) our full model; (g) our full model fine-tuned on the Sketchy database train set; (h) our full model trained on the mix of our synthetic dataset and the Sketchy database train set. Columns (c)(d)(e)(f) are the same setting that is training on our synthetic dataset and testing on the Sketchy database.

one encoder converts the sketch to a latent code and the other predicts a class label. Encoders used in (i) and (ii) have the same architecture. It is worthy to note that although the encoders used in these two variants are non-conditional, the generators used in these two variants and our proposed model are conditional, i.e., require class labels. Therefore,

Table 2. Comparison of different kinds of encoders. The conditional encoder we proposed generates more photo-realistic and faithfully results.

	FID↓	KID↓	IS↑	LPIPS↓	Acc↑
Single Encoder	61.86	0.013	8.74	0.6	**0.989**
Two Encoder	82.19	0.018	**12.21**	0.63	0.186
Ours	**56.71**	**0.012**	9.63	**0.55**	0.988

Table 3. Comparison of training with and without shape loss. The supervision of shape loss improves realism and diversity of results.

	FID↓	KID↓	IS↑	LPIPS↓
w/o shape loss	60.04	0.013	9.37	0.62
Ours	**56.71**	**0.012**	**9.63**	**0.55**

Table 4. Comparison of different training strategies.

	FID↓	KID↓	IS↑	LPIPS↓	Acc↑
Synthetic data only	**56.71**	0.012	9.63	0.55	0.988
Finetune on real data	58.58	0.014	8.95	**0.51**	0.982
Mixed	57.03	**0.011**	**9.65**	0.53	**0.991**

compared with these two variants, our proposed model does not use more supervision.

Table 2 shows that our proposed conditional encoder outperforms the two variants. Using a single non-conditional encoder has a higher classification accuracy. However, as shown in Fig. 6(c), this method fails to capture details indicated in a sketch (e.g., the orientation of the bird and the pose of the dog). Furthermore, in our experiments, we found that using single non-conditional encoder may suffer from the mode collapse problem that generated images have similar shapes and fall in several limited patterns. Thus this method achieves the lowest IS score (see Table 2).

The variant of using two encoders may predict an incorrect class label and thus produce wrong images. As shown in the first example of Fig. 6(d), the model predicts the bird as jack-o-lanterns so that the generated image is incorrect. This variant achieves the highest IS score and the lowest classification accuracy, indicating that the high IS score does not come from high quality but the unexpected abnormal diversity.

Compared with the non-conditional encoders, our proposed conditional encoder can learn a better mapping from the input sketch to the latent space, so the generated images are more faithful to the input sketches. Moreover, unlike previous GAN inversion works, our method allows the users to assign a specific class during generation, which is especially helpful when an input sketch is ambiguous.

Effect of Shape Loss. The shape loss is proposed to constrain the shape of the objects in generated images to be aligned with the input sketches. Figure 6(e) and (f) show the effectiveness of the shape loss. We can observe that the model trained with the shape loss can generate images being more faithful to the sketches in the shape and orientation. If without the shape loss, the model tends to generate objects with incorrect shapes, e.g., the head of the horse is missing. Quantitative results in Table 3 suggest that the shape loss can improve the model's performance in all aspects, including the realism, diversity, and faithfulness.

Comparison of Different Training Strategies. Our proposed model, SketchInverter, is trained on the synthetic dataset and tested on the real dataset. We wonder if including real data during the training process can further improve the model's performance. We compare three training strategies which have been introduced in Sec. 3.5. The Sketchy database is used as the real data for demonstration.

Table 4 compares the performance of different training strategies. Compared to training on synthetic data only, the strategy of finetuning on real data performs the best on LPIPS, indicating that the real data can enhance the faithfulness to the target images in the Sketchy database As shown in Fig. 6(g), the horse's color and the bird's background are closer to the ground truth (Fig. 6(b). However, due to the catastrophic forgetting phenomenon, finetuning on a smaller dataset leads to lower FID, KID, and IS scores. The strategy of training on mixed synthetic and real data achieves the best results on Acc, KID, and IS, indicating that this strategy can improve image quality and diversity. Figure 6(h) shows some results of this strategy. For example, the dog in the second row is more realistic and consistent to the ground-truth photo than others.

SketchInverter exhibits outstanding generalization ability to real data. Note that the strategy, i.e., training only on the synthetic dataset, performs comparatively well compared to the other two, which require real paired data during training. The experimental results suggest that training or finetuning on real data does not bring many benefits for image quality or diversity.

5. Conclusion

This paper has proposed the first GAN inversion-based framework for multi-class sketch-based image generation, which can generate images of high fidelity, realism, and diversity. This framework can significantly reduce the domain gap by using the prior knowledge of the pretrained image generator. A novel conditional encoder has been designed and developed to map the sketch to a latent space with a pre-assigned class label. We have also proposed a synthetic dataset and explored different training strategies to address the issue that paired sketch-photo data is limited.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62002012).

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

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

[3] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Inverting layers of a large generator. In *ICLR Workshop*, volume 2, page 4, 2019.

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

[5] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[7] Jiaxin Chen and Yi Fang. Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 605–620, 2018.

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

[9] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: internet image montage. In *ACM Transactions on Graphics (TOG)*, 2009.

[10] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *CVPR*, 2018.

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

[12] Guoxian Dai, Jin Xie, Fan Zhu, and Yi Fang. Deep correlated metric learning for sketch-based 3d shape retrieval. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[13] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. *Advances in Neural Information Processing Systems*, 30, 2017.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[15] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[16] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics*, 34(5):482–498, 2010.

[17] Mathias Eitz, Ronald Richter, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Photosketcher: interactive sketch-based image synthesis. *IEEE Computer Graphics and Applications*, 2011.

[18] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5174–5183, 2020.

[19] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *CVPR*, 2019.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[21] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *arXiv preprint arXiv:2007.01758*, 2020.

[22] Xiaoguang Han, Chang Gao, and Yizhou Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on graphics (TOG)*, 36(4):1–12, 2017.

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

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

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[29] Mengtian Li, Zhe Lin, Radomír Mˇ ech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *WACV*, 2019.

[30] Yuhang Li, Xuejin Chen, Binxin Yang, Zihan Chen, Zhihua Cheng, and Zheng-Jun Zha. Deepfacepencil: Creating face images from freehand sketches. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, New York, NY, USA, 2020. ACM.

[31] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017.

[32] Fang Liu, Changqing Zou, Xiaoming Deng, Ran Zuo, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. Scenesketcher: Fine-grained image retrieval with scene sketches. In *European Conference on Computer Vision*, pages 718–734. Springer, 2020.

[33] Runtao Liu, Qian Yu, and Stella X Yu. Unsupervised sketch to photo synthesis. In *European Conference on Computer Vision*, pages 36–52. Springer, 2020.

[34] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using contextual gan. In *ECCV*, 2018.

[35] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, pages 67–77. IEEE, 2017.

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

[37] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[38] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[39] Yonggang Qi, Jun Guo, Yi Li, Honggang Zhang, Tao Xiang, and Yi-Zhe Song. Sketching by perceptual grouping. In *2013 IEEE International Conference on Image Processing*, pages 270–274. IEEE, 2013.

[40] Yonggang Qi, Yi-Zhe Song, Tao Xiang, Honggang Zhang, Timothy Hospedales, Yi Li, and Jun Guo. Making better use of edges via perceptual grouping. In *CVPR*, 2015.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[42] Aditya Ramesh, Youngduck Choi, and Yann LeCun. A spectral regularizer for unsupervised disentanglement. *arXiv preprint arXiv:1812.01161*, 2018.

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[44] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

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

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[47] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. In *SIGGRAPH*, 2016.

[48] Yuefan Shen, Changgeng Zhang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepsketchhair: Deep sketch-based 3d hair modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3250–3263, 2020.

[49] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *ICCV*, 2017.

[50] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pages 9786–9796. PMLR, 2020.

[51] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14050–14060, 2021.

[52] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. N\" uwa: Visual synthesis pre-training for neural visual world creation. *arXiv preprint arXiv:2111.12417*, 2021.

[53] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *arXiv preprint arXiv:2101.05278*, 2021.

[54] Xiaoyu Xiang, Ding Liu, Xiao Yang, Yiheng Zhu, Xiaohui Shen, and Jan P Allebach. Adversarial open domain adaptation for sketch-to-photo synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[55] Jin Xie, Guoxian Dai, Fan Zhu, and Yi Fang. Learning barycentric representations of 3d shapes for sketch-based 3d shape retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5068–5076, 2017.

[56] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *CVPR*, 2016.

[57] Q. Yu, Y. Yang, Y. Song, T. Xiang, and T. Hospedales. Sketch-a-net that beats humans. In *BMVC*, 2015.

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[59] Fan Zhu, Jin Xie, and Yi Fang. Learning cross-domain neural networks for sketch-based 3d shape retrieval. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[60] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020.

[61] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[63] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017.