

STABLE DIFFUSION FEATURE EXTRACTION FOR SKETCHING WITH ONE EXAMPLE

Anonymous authors

Paper under double-blind review

ABSTRACT

Sketching is both a fundamental artistic expression and a crucial aspect of art. The significance of sketching has increased alongside the development of sketch-based generative and editing models. To enable individuals to use these sketch-based generative models effectively, personalizing sketch extraction is crucial. In response, we introduce DiffSketch, a novel method capable of generating various geometrically aligned sketches from text or images, using a single manual drawing for training the style. Our method exploits rich information available in features from a pretrained Stable Diffusion model to achieve effective domain adaptation. To further streamline the process of sketch extraction, we further refine our approach by distilling the knowledge from the trained generator into the image-to-sketch network, which is termed as DiffSketch_{distilled}. Through a series of comparisons, we verify that our method not only outperforms existing state-of-the-art sketch extraction methods but also surpasses diffusion-based stylization methods in the task of extracting sketches.

1 INTRODUCTION

Sketching, as an initial stage in artistic creation, serves as a foundational process for conceptualizing and conveying artistic intentions while visualizing the core structure and content of the final artwork. As sketches can exhibit distinct styles despite their basic form composed of simple lines, many studies in computer vision and graphics have attempted to train models for automatically extracting geometric sketches Winnemöller (2011); Illyasviel (2017); Ashtari et al. (2022); Chan et al. (2022); Seo et al. (2023). The majority of previous sketch extraction approaches utilize image-to-image translation techniques to produce high-quality results. These approaches typically require a large dataset when training an image translation model from scratch, making it difficult to personalize applications such as sketch auto-colorization, sketch-based editing, or conditional generation. Recently, advancements in abstract curve optimization have been made as an alternative that does not require training Mo et al. (2021); Vinker et al. (2022); Willett et al. (2023); Vinker et al. (2023). While these methods can effectively optimize curves based on a given text or image, they cannot follow the target style image, making it challenging to generate personalized sketches.

Meanwhile, recent research has explored the utilization of diffusion model Rombach et al. (2022); Saharia et al. (2022) features for downstream tasks Xu et al. (2023); Khani et al. (2023); Zhang et al. (2023a); Tumanyan et al. (2023). Features derived from pretrained diffusion models are known to contain rich semantics and spatial information Tumanyan et al. (2023); Xu et al. (2023), which can help train networks for various tasks using a small number of data. Previous studies have utilized these features extracted from a subset of layers Baranchuk et al. (2021), certain timesteps Zhang et al. (2023a); Xu et al. (2023), or every specific interval Luo et al. (2023). Unfortunately, these selected features often do not contain most of the information generated during the entire diffusion process.

To this end, we propose DiffSketch, a new method that can extract representative features from a pretrained Stable Diffusion (SD) Rombach et al. (2022) and train the sketch generator with one manual drawing. For feature extraction from the denoising process, we statistically analyze the features and select those that can represent the whole feature information from the denoising process. Our new generator aggregates the features from multiple timesteps, fuses them with Variational

The source code for both DiffSketch and DiffSketch_{distilled} will be released.

Autoencoder (VAE) Kingma & Welling (2013) features, and decodes these fused features into a sketch. In addition, we distill DiffSketch into a streamlined image-to-image translation network for improved inference speed and efficient memory usage, dubbed DiffSketch_{distilled}.

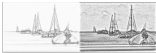



	DiffSketch _{distilled}	Reference based Sketch Extraction	Finetuning Diffusion Models	Curve Optimization
Stylize with example	 ✓	 ✓	 ✓	 ✗
Inference time	≈ 0.014s	<0.1s	>1s	>10m
No overfitting	✓	✓	✗	✓
Training Data	1	>1,000	<10	✗

Figure 1: The uniqueness of our method: DiffSketch_{distilled} is capable of extracting sketches of a given style after trained with only one example, without overfitting. DiffSketch_{distilled} is different from previous methods that require large datasets Seo et al. (2023), that are prone to overfitting Ruiz et al. (2023), and that cannot extract using a style example Vinker et al. (2023).

Our method is tailored specifically for sketch generation, utilizing a dedicated sketch generator trained on features from VAE and SD. This approach sets itself apart from traditional diffusion-based personalization or stylization techniques. While existing personalization methods rely on finetuning, re-prompting, or adding adaptation modules Ruiz et al. (2023); Gal et al. (2022); Zhang et al. (2023b); Hu et al. (2021), DiffSketch employs a Decoder which is trained with a domain adaptation technique, to address common issues such as mode collapse or color leakage. It processes diffusion features without modifying the original SD model and outputs fused features in a single channel sketch. In addition, unlike curve optimization methods Vinker et al. (2022; 2023); Xing et al. (2023), DiffSketch_{distilled} can control the style of sketches using a given style example. This is achieved in a few milliseconds. The differences of our method from previous methods are highlighted in Fig. 1

In addition to the newly proposed generator, we introduce a method for effective sampling performed during training. We found that training a network with data that share similar semantic information with that of the ground truth data is effective. However, relying solely on such data for training will hinder the full utilization of the capacity provided by the diffusion model. Therefore, we adopt a new sampling method to ensure training with diverse examples while enabling effective training. The resulting DiffSketch_{distilled} is the final network that is capable of performing a sketch extraction task.

2 RELATED WORK

2.1 SKETCH EXTRACTION

At its core, sketch extraction utilizes edge detection. Edge detection serves as the foundation not only for sketch extraction but also for tasks like object detection and segmentation Zhang et al. (2015); Arbelaez et al. (2010). Initial edge detection studies primarily focused on identifying edges based on abrupt variations in color or brightness Canny (1986); Winnemöller (2011). Although these techniques are direct and efficient without requiring extensive datasets to train on, they often produce outputs with artifacts, like scattered dots or lines.

To make extracted sketches authentic, learning-based strategies have been introduced. These strategies excel in identifying object borders or rendering lines in distinct styles Xiang et al. (2021); Xie & Tu (2015a); Illyasviel (2017); Li et al. (2019; 2017). Informative drawing Chan et al. (2022) took a step forward from prior techniques by incorporating the depth and semantic information of images to procure superior-quality sketches. In a more recent development, Ref2sketch Ashtari et al. (2022) permits to extract stylized sketches using reference sketches through paired training. Semi-Ref2sketch Seo et al. (2023) adopted contrastive learning for semi-supervised training. All of these

108 methods share the same limitation; they require a large amount of sketch data for training, which
109 is hard to gather. Due to data scarcity, training a sketch extraction model is generally challenging.
110 To address this challenge, our method is designed to train a sketch generator using just one manual
111 drawing.

112 2.2 DIFFUSION FEATURES FOR DOWNSTREAM TASK

113 Diffusion models Ho et al. (2020); Nichol & Dhariwal (2021) have shown cutting-edge results in
114 tasks related to generating images conditioned on text prompt Rombach et al. (2022); Saharia et al.
115 (2022); Ramesh et al. (2021). There have been attempts to analyze the features for utilization in
116 downstream tasks such as segmentation Baranchuk et al. (2021); Xu et al. (2023); Khani et al.
117 (2023), image editing Tumanyan et al. (2023), and finding dense semantic correspondence Luo
118 et al. (2023); Zhang et al. (2023a); Tang et al. (2023). Most earlier studies chose a specific subset
119 of features for their own downstream tasks. Recently, Diffusion Hyperfeature Luo et al. (2023)
120 proposed an aggregator that learns features from all layers and that uses equally sampled time steps.
121 We advance a step further by analyzing and selecting the features from multiple timesteps, which
122 represent the overall features. We also propose a two-stage aggregation network and feature-fusing
123 decoder utilizing additional information from VAE to generate finer details.

124 2.3 DEEP FEATURES FOR SKETCH EXTRACTION

125 Most of recent sketch extraction methods utilize the deep features of a pretrained model for sketch
126 extraction training Ashtari et al. (2022); Seo et al. (2023); Yi et al. (2019; 2020). These approaches
127 utilize deep features from a pretrained classifier Johnson et al. (2016); Zhang et al. (2018) or vision-
128 language models such as CLIP Radford et al. (2021) to measure semantic similarity Chan et al.
129 (2022); Vinker et al. (2022). They indirectly use the features by comparing them for the loss cal-
130 culation during the training process instead of using them to generate a sketch. DiffSketcher Xing
131 et al. (2023) utilizes a diffusion model to perform curve optimization from text. StyleSketch Yun
132 et al. (2024) utilizes GAN features to extract a facial sketch with a few data. These recent models
133 have successfully demonstrated that generative features can be used to create sketches. However,
134 neither method can extract a sketch from a single example because DiffSketcher cannot take a style
135 or content image as input, while StyleSketch requires 16 data for training in a single domain. To fa-
136 cilitate translating an image to a sketch in a provided style, we directly use the diffusion features that
137 contain rich information and generate geometric sketches using a network trained with one example
138 pair.

139 3 DIFFUSION FEATURES

140 During the backward diffusion process, UNet Ronneberger et al. (2015) produces several inter-
141 mediate features with different shapes while reducing noise. This collection of features contains
142 rich information about texture and semantics, which can be used to generate an image in various
143 domains. For instance, features from the lower to intermediate layers of the UNet reveal global
144 structures and semantic regions, while features from higher layers exhibit fine and high-frequency
145 information Tumanyan et al. (2023); Luo et al. (2023). Furthermore, features become more fine-
146 grained over time steps Hertz et al. (2022). As these features have different information depending
147 on their embedded layers and processed timesteps, it is important to select diverse features to fully
148 utilize the information they provide.

149 3.1 DIFFUSION FEATURES SELECTION

150 Here, we first present a method for selecting features by analysis. Our approach involves selecting
151 representative features from all the denoising timesteps and building our sketch generator, G_{sketch}
152 to extract a sketch from an image by learning from a single data. To perform analysis, we randomly
153 sampled images and collected all the features from multiple layers and timesteps during Denoising
154 Diffusion Implicit Model (DDIM) sampling, with a total of 50 steps Song et al. (2020). For an
155 experiment, features from a total of 50,000 data (50 UNet features with varying timesteps from
156 1,000 randomly generated images) were gathered.

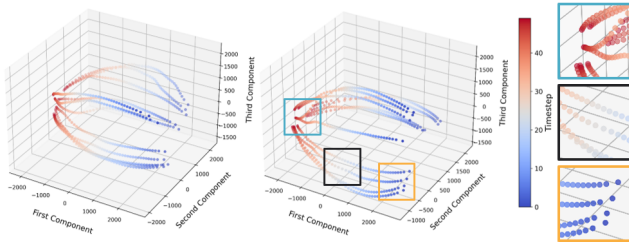


Figure 2: Analysis on sampled features. PCA is applied to DDIM sampled features, colored with denoising timesteps.

We conducted Principal component analysis (PCA) on these features from all timesteps to examine their distributions depending on their timesteps. The PCA results are visualized in Fig 2, in which smooth trajectories across the timesteps are shown. Therefore, selecting features from intervals can be more beneficial than using a single feature, as it provides richer information, as previously suggested Luo et al. (2023). Upon further examination, we can observe that the features tend to start at a similar point in their initial timesteps ($t \approx 50$) and diverge thereafter (cyan box). In addition, during the initial steps, nearby values do not show a large difference compared to those in the middle (black box), while the final features exhibit distinct values even though they are on the same trajectory (orange box).

These findings provide insights that can guide the selection of features. As we aim to capture the informative features across the timesteps instead of using all features, we first conducted a K-means clustering analysis (K-means) Hotelling (1933) using Within Clusters Sum of Squares distance (WCSS) to determine the number of feature clusters. From this process, we chose our K as 13 although this K value may vary with the number of diffusion sampling processes. We selected the features from the center of each cluster to use them as input to our sketch generation network. The detailed process of clustering and further experiments for different sampling processes and different models are presented in Sec. B of the Appendix.

3.2 DIFFUSION FEATURES AGGREGATION

Inspired by feature aggregation networks for downstream tasks Xu et al. (2023); Luo et al. (2023), we build our two-level aggregation network and feature fusing decoder (FFD), both of which constitute our new sketch generator G_{sketch} . The architectures of G_{sketch} and FFD are shown in Fig. 4 (b) and (d), respectively. The diffusion features $f_{l,t}$, generated on layer l and timestep t , are passed through the representative feature gate G^* . They are then upsampled to a certain resolution by U_m or U_{tp} , and passed through an aggregation network which consists of bottleneck layer (B_l^m or B_l^{tp}) and mixing layer with mixing weights w . The second aggregation network receives the first fused feature F_{fst} as an additional input feature.

$$F_{fst} = \sum_{t=0}^T \sum_{l=1}^{l_t-1} w_{l,t} \cdot B_l^m(U_m(G^*(f_{l,t}))), F_{fin} = \sum_{t=0}^T \sum_{l=l_t}^L w_{l,t} \cdot B_l^{tp}(U_{tp}(G^*(f_{l,t}))) + \sum_{l=l_t}^L w_l \cdot B_l^{tp}(U_{tp}(F_{fst})) \quad (1)$$

Here, L is the total number of UNet layers, while l_t indicates the middle layer, which are set to be 12 and 10, respectively. Bottleneck layers B_l^m and B_l^{tp} are shared across timesteps. T is the total number of timesteps. F_{fst} denotes the first level aggregated features and F_{fin} denotes the final aggregated features. These two levels of aggregation allow us to utilize the features in a memory efficient manner by mixing the features sequentially in a lower resolution first and then in a higher resolution.

3.3 VAE DECODER FEATURES

Unlike recent applications on utilizing diffusion features, where semantic correspondences are more important than high-frequency details, sketch generation utilizes both semantic information and high-frequency details such as texture. As shown in Fig. 3, VAE decoder features contain high-frequency details such as hair and wrinkles. From this observation, we designed our network to utilize VAE features following the aggregation of UNet features. Extended visualizations are provided in the Appendix.

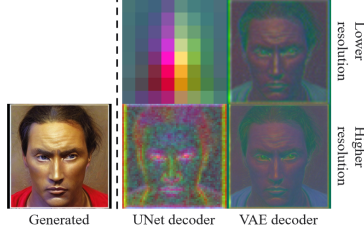
216
217
218
219
220
221
222
223
224

Figure 3: Visualization of features from UNet and VAE in lower and higher resolution layers. Lower resolution layers are the first layers while higher resolution layers are the 11th layer for UNet and the 9th layer for VAE.

225
226
227
228
229
230
231
232

We utilize all the VAE features from the residual blocks to build FFD. The aggregated features F_{fin} and VAE features are fused together to generate the output sketch. Specifically, in the fusing step i , VAE features with the same resolution are passed through the channel reduction layer followed by the convolution layer. These processed features are concatenated to the previously fused feature x_i and the result is passed through the fusion layer to output x_{i+1} . For the first step ($i = 0$), x_0 is F_{fin} . All features in the same step have the same resolution. We denote the number of total features at i as N without subscript for simplicity. This process is shown in Fig. 4 (d) and can be expressed as follows:

233
234
235

$$x_{i+1} = \text{FUSE}\left[\left\{\sum_{n=1}^N \text{Conv}(\text{CH}(v_{i,n}))\right\} + x_i\right], \hat{I}_{sketch} = \text{OUT}\left[\left\{\sum_{n=1}^N \text{Conv}(\text{CH}(v_{M,n}))\right\} + x_M + I_{source}\right] \quad (2)$$

236
237
238
239
240

where CH is the channel reduction layer, Conv is the convolution layers, FUSE is the fusion layer, OUT is the final convolution layer applied before outputting \hat{I}_{sketch} , \sum and $+$ represent concatenation in the channel dimension. Only at the last step ($i = M$), the source image, I_{source} is also concatenated to generate the output sketch.

241

242

4 DIFFSKETCH

243
244

DiffSketch learns to generate a pair of image and sketch through the process described below, which is also shown in Fig. 4.

245
246
247
248
249
250
251

1. First, the user generates an image using a prompt with Stable Diffusion (SD) Rombach et al. (2022) and draws a corresponding sketch while its diffusion features F are kept.
2. The diffusion features F , its corresponding image I_{source} , and drawn sketch I_{sketch} constitute a triplet data to train the sketch generator G_{sketch} with directional CLIP guidance.
3. With trained G_{sketch} , paired image and sketch can be generated with a condition. This becomes the input for the distilled network for fast sketch extraction.

252
253
254

In the following subsections, we will describe the structure of sketch generator G_{sketch} (Sec. 4.1), its loss functions (Sec. 4.2), and the distilled network (Sec. 4.4).

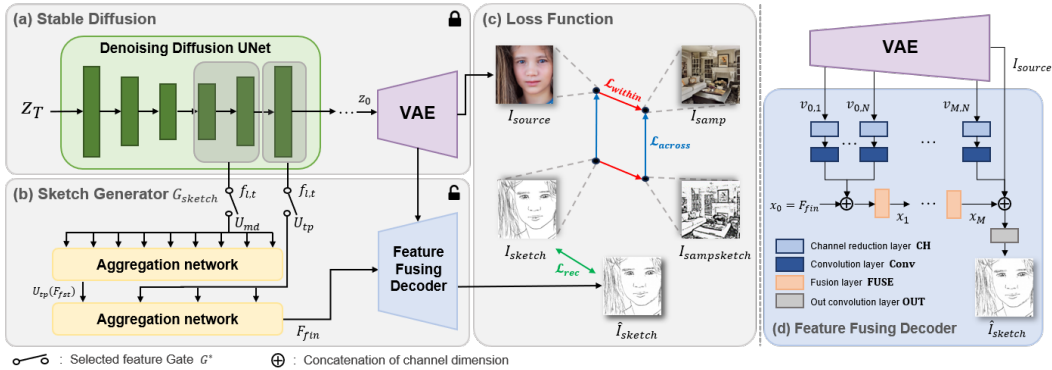
255
256
257
258
259
260
261
262
263
264
265
266
267268
269

Figure 4: Overview of DiffSketch. The UNet features generated during the denoising process are fed to the Aggregation networks to be fused with the VAE features to generate a sketch corresponding to the image that Stable Diffusion generates.

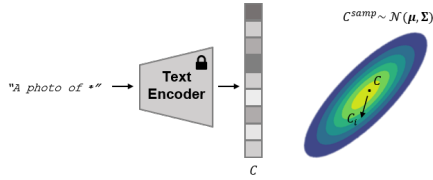


Figure 5: Illustration of CDST. Training starts with C , which is an encoded prompt and is diffused as the training iteration progresses to follow the distribution of SD.

4.1 SKETCH GENERATOR

Our sketch generator G_{sketch} is built to utilize the features from the denoising diffusion process performed by UNet and VAE as described in Secs. 3.2 and 3.3. G_{sketch} takes the selected features from UNet as input, and aggregate them and fuse them with the VAE decoder features $v_{i,n}$ to synthesizes the corresponding sketch \hat{I}_{sketch} . Unlike other image-to-image translation-based sketch extraction methods in which the network takes an image as input, our method accepts multiple deep features that have different spatial resolutions and channels.

4.2 OBJECTIVES

To train G_{sketch} , we utilize the following loss functions:

$$L = L_{rec} + \lambda_{within}L_{within} + \lambda_{across}L_{across} \quad (3)$$

where λ_{within} and λ_{across} are the balancing weights. L_{within} and L_{across} are directional CLIP losses for domain adaptation, proposed in Mind-the-gap (MTG) Zhu et al. (2022). L_{within} preserves the direction within the style (image-image and sketch-sketch), by enforcing the difference between synthetic image I_{samp} from SD and I_{source} to be similar to that between generated sketch $I_{samps sketch}$ from G_{sketch} and I_{sketch} in CLIP embedding space. Similarly, L_{across} enforces the difference between $I_{samps sketch}$ and I_{samp} to be similar to that between I_{source} and I_{sketch} . While MTG uses an MSE loss for the pixel-wise reconstruction, we use an L1 distance to avoid blurry sketch results, which is important in the generation of sketches Ashtari et al. (2022). Our L_{rec} can be expressed as follows:

$$L_{rec} = \lambda_{L1}L_{L1} + \lambda_{LPIPS}L_{LPIPS} + \lambda_{CLIPsim}L_{CLIPsim} \quad (4)$$

where λ_{L1} , λ_{LPIPS} , and $\lambda_{CLIPsim}$ are the balancing weights. L_{L1} calculates the pixel-wise reconstruction, L_{LPIPS} Zhang et al. (2018) captures the perceptual similarity, and $L_{CLIPsim}$ calculates the semantic similarity in the cosine distance. More details can be found in Sec. 5.1

4.3 SAMPLING SCHEME FOR TRAINING

Our method uses one source image and its corresponding sketch as the only ground truth when guiding the sketch style, using the direction of CLIP embeddings. Therefore, our losses rely on well-constructed CLIP manifold. We found that when the domains of two images I_{source} and I_{samp} differ largely, the confidence in the directional CLIP loss becomes lower (explanation and experiment are provided in Sec. 5.2). To fully utilize the capacity of the diffusion model and produce sketches in diverse domains, however, it is important to train the model on diverse examples.

To ensure learning from diverse examples without decreasing the confidence of directional CLIP losses, we propose a novel sampling scheme, condition diffusion sampling for training (CDST) in which the condition is diffused from a single point to whole sampling space of SD. We envision that this sampling can be useful when training a model with a conditional generator. CDST initially samples a data I_{samp} from one known condition encoded from prompt C and gradually changes the sampling distribution to the distribution of pretrained SD by using a diffusion algorithm when training the network (see Fig. 5).

Here, to estimate the distribution of SD, we randomly sampled 100k prompts from LAION-400M Schuhmann et al. (2021) and used them as a subset of the trained text-image pairs of the SD model. We then tokenized and embedded these prompts for preprocessing, following the process of the pretrained SD model. We then conducted Shapiro-Wilk test Shapiro & Wilk (1965), followed by Mardia test Mardia (1970; 1974) with a significance level of $\alpha = 5\%$ and found that the distribution of SD follows a multivariate normal distribution. The detailed process is stated in Sec. D of the Appendix. The condition on the iteration i ($0 \leq i \leq S$) can be described as follows:

$$\alpha_i = \sqrt{1 - \frac{i}{S}}, \quad \beta_i = \sqrt{\frac{i}{S}}, \quad C_i^{samp} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad C_i = \frac{\alpha_i}{\alpha_i + \beta_i}C + \frac{\beta_i}{\alpha_i + \beta_i}C_i^{samp} \quad (5)$$

where \mathcal{N} represents the multivariate normal distribution, approximating the distribution of the pre-trained SD. μ represents the mean vector and Σ represents the covariance matrix. S indicates the number of the total diffusion steps during training.

4.4 DISTILLATION

Once the sketch generator G_{sketch} is trained, DiffSketch can generate pairs of images and sketches in the trained style. This generation can be performed either randomly or with a specific condition. Due to the nature of the denoising diffusion model, in which the result is refined through the denoising process, long processing time and high memory usage are required. Moreover, when extracting sketches from images, the quality can be degraded because of the inversion process. Therefore, to perform image-to-sketch extraction efficiently while ensuring high-quality results, we train DiffSketch_{distilled} using Pix2PixHD Wang et al. (2018).

To train DiffSketch_{distilled}, we extract 30k pairs of image and sketch samples using our trained DiffSketch, adhering to CDST. Additionally, we employ regularization to ensure that the ground truth sketch I_{sketch} can be generated and discriminated effectively during the training of DiffSketch_{distilled}. With this trained model, images can be extracted in a given style much more quickly than with the original DiffSketch.

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

We implemented DiffSketch and trained generator G_{sketch} on an Nvidia V100 GPU for 1,200 iterations. When training G_{sketch} , we applied CDST with S in Eq. 5 to be 1,000. The model was trained with a fixed learning rate of $2e-4$. The balancing weights λ_{across} , λ_{within} , λ_{L1} , λ_{LPIPS} , and $\lambda_{CLIPsim}$ were fixed at 1, 1, 30, 15, and 30, respectively. DiffSketch_{distilled} was trained on two A6000 GPUs using the same architecture and parameters from its original paper except for the output channel, where ours was set to one. We also added regularization on every 16 iterations. DiffSketch_{distilled} was trained with 30,000 pairs that were sampled from DiffSketch with CDST ($S = 30,000$). LPIPS Zhang et al. (2018), SSIM Wang et al. (2004), and FID Heusel et al. (2017) were used for a comparison with baselines while only LPIPS and SSIM were used for an ablation study due to a limited number of test data (=100). LPIPS measured perceptual similarity, SSIM measured structural similarity, and FID measured distribution similarity.

5.2 CONFIDENCE SCORE TEST

An underlying assumption of CDST is that for a bi-directional CLIP loss which is used for domain adaptation Yoon et al. (2024); Zhu et al. (2022); Kim et al. (2022), two images with a similar domain (I_{source} and I_{samp}) leads to higher confidence compared to two images with a different domain. To examine this, we devised a new metric *confidence score*. As the first step, we measured similarity value Sim_{within} and Sim_{across} of the CLIP features of images from different domains in the same manner described as the Sec. 4.2. Specifically, the equation for similarity is as follows:

$$Sim(X, Y) = \frac{\cos(\overrightarrow{I_X I_Y} \cdot \overrightarrow{S_X S_Y}) + \cos(\overrightarrow{I_X S_X} \cdot \overrightarrow{I_Y S_Y})}{N} \quad (6)$$

where $\cos(a \cdot b)$ is the cosine similarity and N is the total number for averaging. I_X and I_Y corresponds to the CLIP embedding of images in each domain X and Y . Similarly S_X and S_Y corresponds to CLIP embedding of sketches in each domain X and Y . In detail, $\cos(\overrightarrow{I_X I_Y} \cdot \overrightarrow{S_X S_Y})$ corresponds to L_{within} and $\cos(\overrightarrow{I_X S_X} \cdot \overrightarrow{I_Y S_Y})$ corresponds to L_{across} described in Sec. 4.2. With these computed similarities, the confidence score in domain X and domain Y can be written as follows where $Sim_{(ALL, ALL)}$ denotes the average similarity of all images, for which higher is better:

$$confidence(A, B) = \frac{Sim(X, Y)}{Sim_{(ALL, ALL)}} \times 100 \quad (7)$$

We measured the *confidence score* using 4SKST Seo et al. (2023), which consists of four different sketch styles paired with color images. 4SKST is suitable for the *confidence score* test because it contains images from two distinct domains, photos and anime, presented in four different styles. We computed a *confidence score* to determine whether the directional CLIP loss is indeed reliable when

the images for comparison are from the same domain. We conducted the test with three settings using I_A (Photo) and I_B (Anime), along with their corresponding sketch embeddings, S_A and S_B . We then calculated the feature similarity within the photo domain, anime domain, and across the two domains. As shown in Table 1, for all four styles, confidence scores from the same domain were higher than those from different domains. Accordingly, we proposed a sampling scheme, CDST to train the generator in the same domain at the initial stage of the training, which leads to higher confidence while widening its capacity in the latter iterations of training.

Table 1: Confidence scores on 4SKST with four different styles.

Similarity	Style1	Style2	Style3	Style4	Average
$confidence(Anime,Anime)$	104.2608	102.8716	108.2026	101.3530	104.1720
$confidence(Photo,Photo)$	101.9346	98.8005	102.4516	100.5453	100.9330
$confidence(Photo,Anime)$	94.5036	94.0189	98.1867	92.3874	94.7742

5.3 DATASETS

For training, DiffSketch requires a sketch corresponding to an image generated from SD. To facilitate a numerical comparison, we established the ground truth for given images. Specifically, three distinct styles were employed for quantitative evaluation: 1) HED Xie & Tu (2015b) utilizes nested edge detection and is one of the most widely used edge detection methods. 2) XDoG Winnemöller et al. (2012) takes an algorithmic approach of using a difference of Gaussians to extract sketches. 3) Anim-informative Chan et al. (2022) employs informative learning, which is the state-of-the-art among single modal sketch extraction methods and is trained on the Anime Colorization dataset Kim (2018), which consists of 14,224 sketches. For perceptual study, we added hand-drawn sketches of two more styles. For testing, we employed the test set from the BSDS500 dataset Martin et al. (2001). As a result, our training set consisted of 3 styles and the test dataset consisted of 600 pairs (200 pairs for each style) of image-sketch for quantitative evaluation while 5 styles were used for the perceptual study. Two additional hand-drawn sketches were used only for perceptual study because there is no ground truth to compare with.

5.4 ABLATION STUDY

We conducted an ablation study on each component of our method compared to the baselines as shown in Table 2. Experiment were performed to verify the contribution of each component; feature selections, CDST, losses, and FFD. To perform the ablation study, we randomly sampled 100 images and extracted sketches with HED, XDog, and Anim-informative and paired them with all 100 images. All seeds were fixed to generate sketches from the same sample.

The ablation study was conducted as follows. For Random features, we randomly selected the features from denoising timesteps while keeping the number of timesteps equal to ours (13). We performed this random selection and analysis twice. For one timestep feature, we only used the features from the final timestep $t = 0$. To produce a result without CDST, we executed random text prompt guidance for the diffusion sampling process during training. For the alternative loss approach, we contrasted L1 Loss with L2 Loss for pixel-level reconstruction, as proposed in MTG. To evaluate the effect of the FFD, sketches were produced after removing the VAE features.

The evaluation results of the ablation study are shown in Table 2. Ours achieved the highest average scores for both metrics. Both Random features achieved overall low scores indicating that feature

Table 2: Quantitative results on ablation with LPIPS and SSIM. Best scores are denoted in bold.

Sketch Styles Methods	anim-informative		HED		XDoG		Average	
	LPIPS	SSIM	LPIPS	SSIM	LPIPS	SSIM	LPIPS	SSIM
Ours	0.2054	0.6835	0.2117	0.5420	0.1137	0.6924	0.1769	0.6393
Random features 1	0.2154	0.6718	0.2383	0.5137	0.1221	0.6777	0.1919	0.6211
Random features 2	0.2042	0.6869	0.2260	0.5281	0.1194	0.6783	0.1832	0.6311
One feature	0.2135	0.6791	0.2251	0.5347	0.1146	0.6962	0.1844	0.6367
W/O CDST	0.2000	0.6880	0.2156	0.5341	0.1250	0.6691	0.1802	0.6304
W/O L1	0.2993	0.3982	0.2223	0.5011	0.1203	0.6547	0.2140	0.5180
W/O FFD	0.2650	0.5044	0.2650	0.4061	0.2510	0.3795	0.2603	0.4300

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

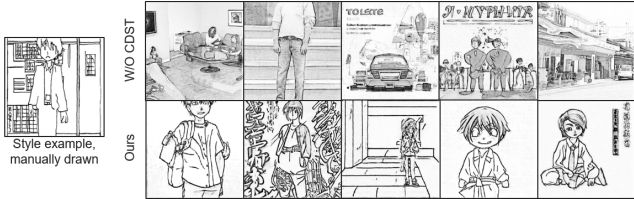


Figure 6: Comparison of results produced with and without using CDST. CDST was applied in both training and inference.

selection helps obtain rich information. Similarly, using one time step features achieved lower scores than ours on average, showing the importance of including diverse features. W/O CDST scored lower than ours on both HED and XDoG styles. W/O L1 and W/O FFD performed the worst due to lack of fine information from VAE.

Feature Selection We conducted an ablation study to examine if our selected features represent all features well during the diffusion process. For this, a comparison with two baselines was made, sampling at equal time intervals ($t=[i*4+1$ for i in the range of $(0,13)$]) similar to Luo et al. (2023) and randomly selecting 13 features. We calculated the minimum Euclidean distance from each feature and confirmed that our method resulted in the minimum distance across 1,000 randomly sampled images. As illustrated in Table 3, our selected features have the lowest distance in the feature space, while selecting equally similar to Luo et al. (2023) scored the second.

Table 3: Sum of the minimum distances from all features. Our selected features better represent overall denoising features compared to sampling equally and randomly.

Method	Ours	Equal time steps	Random sample
Euclidean Distance (10^3)	18.615	19.005	23.957

Condition Diffusion Sampling for Training While we tested on randomly generated images (without CDST), to maintain consistency in the test set, CDST should be applied during both the training of DiffSketch and the inference for training DiffSketch_{distilled}. Therefore, we conducted an additional ablation study on CDST, comparing Ours (trained and sampled with CDST), against W/O CDST (trained and sampled without CDST). The outline of the sketch was clearly reproduced, following the style, when CDST was used as shown in Fig. 6.

5.5 COMPARISON WITH BASELINES

We initially compared our method with 11 different alternatives, including state-of-the-art sketch extraction methods Ashtari et al. (2022); Seo et al. (2023), diffusion based stylization methods Kwon & Ye (2023); Yang et al. (2023); Zhang et al. (2023c); Chung et al. (2024b), and conventional style transfer Huang & Belongie (2017). However, four of the baselines Ruiz et al. (2023); Zhang et al. (2023b); Gal et al. (2022); Chung et al. (2024a) failed or had sever artifacts thus presented in Sec. E.1 of the Appendix. Among the remaining seven baselines, Ref2sketch Ashtari et al. (2022) and S-Ref2sketch Seo et al. (2023) are methods specifically designed to extract sketches in the style of a reference by training the network on large sketch data. DiffuseIT Kwon & Ye (2023), StyleID Chung et al. (2024b), and InST Zhang et al. (2023c) are designed for diffusion based image-to-image translation by disentangling style and content. AdaIN Huang & Belongie (2017) is conventional style transfer method, and ZeCon Yang et al. (2023) is text based stylization method.

Table 4 presents the result of quantitative evaluation. Overall, ours achieved the best scores. While S-Ref2sketch scored the second highest, it relied on a large sketch dataset to train unlike ours that required only one training data. Fig. 7 presents visual results produced by different methods. While S-Ref2sketch, Ref2sketch, StyleID, and AdaIn generated comparable quality in one or two sources, they did not faithfully follow the exact style in others. DiffuseIT sometimes failed to disentangle style and content, while InST and ZeCon failed to extract sketches following the target style. DiffSketch_{distilled} generated superior results compared to these baselines, effectively maintaining its styles and content.

5.6 PERCEPTUAL STUDY

We conducted a user study to evaluate different sketch extraction methods on human perception. We recruited 21 participants to complete a survey that used test images from five different styles, to extract sketches. Each participant was presented with a total of 20 sets of source image, target

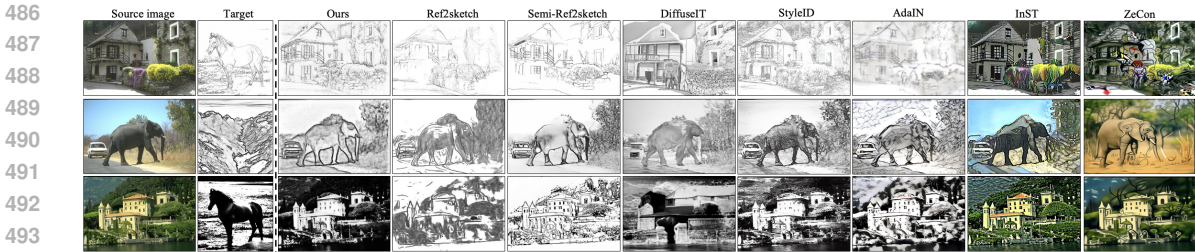


Figure 7: Qualitative comparison with baselines.

Table 4: Quantitative comparison of different methods on the BSDS500 datasets. Best scores are denoted in bold, and the second-best are underlined.

Methods	BSDS500 - anime			BSDS500 - HED			BSDS500 - XDoG			BSDS500 - average		
	LPIPS	SSIM	FID	LPIPS	SSIM	FID	LPIPS	SSIM	FID	LPIPS	SSIM	FID
Ours	0.218	<u>0.493</u>	<u>126.5</u>	0.227	0.593	110.6	0.143	0.649	62.8	0.196	0.578	100.0
Ref2sketch	0.336	0.469	155.2	0.420	0.315	168.6	0.571	0.131	274.5	0.442	0.305	199.4
S-Ref2sketch	<u>0.239</u>	0.510	99.1	0.397	<u>0.342</u>	<u>162.3</u>	0.505	0.309	192.6	0.380	<u>0.387</u>	<u>151.3</u>
DiffuseIT	0.484	0.298	215.2	0.492	0.191	214.2	0.573	0.110	215.3	0.516	0.200	214.9
StyleID	0.375	0.314	211.8	0.405	0.121	198.5	<u>0.241</u>	<u>0.459</u>	<u>135.4</u>	<u>0.340</u>	0.298	181.9
AdaIN	0.348	0.411	205.2	<u>0.392</u>	0.256	200.1	0.406	0.249	187.7	0.382	0.305	197.7
InST	0.677	0.180	245.9	0.592	0.129	187.8	0.477	0.294	244.3	0.582	0.201	226.0
ZeCon	0.702	0.243	254.6	0.619	0.160	253.3	0.494	0.341	262.5	0.605	0.248	256.8

sketch style, and resulting sketch. Participants were asked to choose one that best follows the given style while preserving the content of the source image. As shown in Table 5, our method received the highest scores among all competing methods. Ours outperformed the diffusion-based methods and even received a higher preference rating than the specialized sketch extraction method that was trained on a large sketch dataset.

Table 5: Results from the perceptual study performed given a style example and the source image. The percentages indicate the selection frequency. Ours was the most frequently chosen, with more than double the selection rate of the second-highest.

Ours	Ref2sketch	S-Ref2sketch	DiffuseIT	StyleID	AdaIn	InST	ZeCon
49.52%	1.90%	17.38%	1.19%	15.48%	8.10%	6.43%	0.0%

6 LIMITATION AND CONCLUSION

We proposed DiffSketch, a novel method to extract sketches in given styles by training a sketch generator using representative features. For the first time, we conducted the task of extracting sketches from the features of a diffusion model and demonstrated that our method outperforms previous state-of-the-art methods. The ability to extract sketches in input style, trained with one example, will have various use cases not only for artistic purposes but also for personalizing sketch-to-image retrieval and sketch-based image editing.

We built our generator network specialized for producing sketches by fusing aggregated features with the features from a VAE decoder. Consequently, our method works well with diverse sketches including dense sketches and outlines. However, because our method utilizes features during generation, it requires the user to draw a sketch, making it impossible to use existing sketch pairs. One possible future research direction could involve utilizing features from inversion. To help understand future research in this direction, we visualize the features from inversion to show that their characteristics are similar to the features from generation in Sec. B.2 of the Appendix.

Although we focused on sketch extraction, our analysis of selecting representative features and the proposed training scheme are not limited to the domain of sketches. Extracting representative features holds potential to improve applications leveraging diffusion features, including semantic segmentation, visual correspondence, and depth estimation. We believe that this research direction promises to broaden the impact and utility of diffusion feature-based applications.

REFERENCES

- 540
541
542 Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hi-
543 erarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*,
544 33(5):898–916, 2010.
- 545 Amirsaman Ashtari, Chang Wook Seo, Cholmin Kang, Sihun Cha, and Junyong Noh. Reference
546 based sketch extraction via attention mechanism. *ACM Transactions on Graphics (TOG)*, 41(6):
547 1–16, 2022.
- 548 Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khruikov, and Artem Babenko. Label-
549 efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.
- 550
551 John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis*
552 *and machine intelligence*, (6):679–698, 1986.
- 553 Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey
554 geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
555 *Pattern Recognition*, pp. 7915–7925, 2022.
- 556
557 Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis
558 for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*
559 *recognition*, pp. 8188–8197, 2020.
- 560 Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach
561 for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF*
562 *Conference on Computer Vision and Pattern Recognition*, pp. 8795–8805, 2024a.
- 563
564 Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach
565 for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF*
566 *Conference on Computer Vision and Pattern Recognition*, pp. 8795–8805, 2024b.
- 567 David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern*
568 *analysis and machine intelligence*, (2):224–227, 1979.
- 569 Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel
570 Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual
571 inversion. *arXiv preprint arXiv:2208.01618*, 2022.
- 572
573 Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or.
574 Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*,
575 2022.
- 576
577 Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter.
578 Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in*
579 *neural information processing systems*, 30, 2017.
- 580 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in*
581 *neural information processing systems*, 33:6840–6851, 2020.
- 582
583 Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal*
584 *of educational psychology*, 24(6):417, 1933.
- 585 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
586 and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint*
587 *arXiv:2106.09685*, 2021.
- 588
589 Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normal-
590 ization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510,
591 2017.
- 592 Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and
593 super-resolution. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The*
Netherlands, October 11–14, 2016, Proceedings, Part II 14, pp. 694–711. Springer, 2016.

- 594 Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative
595 adversarial networks. In *CVPR*, 2019.
- 596
- 597 Aliasghar Khani, Saeid Asgari Taghanaki, Aditya Sanghi, Ali Mahdavi Amiri, and Ghassan
598 Hamarneh. Slime: Segment like me. *arXiv preprint arXiv:2309.03179*, 2023.
- 599
- 600 Seongtae Kim, Kyoungkook Kang, Geonung Kim, Seung-Hwan Baek, and Sunghyun Cho. Dy-
601 nagan: Dynamic few-shot adaptation of gans to multiple domains. In *SIGGRAPH Asia 2022*
602 *Conference Papers*, pp. 1–8, 2022.
- 603 Taebum Kim. Anime sketch colorization pair. [https://www.kaggle.com/taebum/
604 anime-sketch-colorization-pair](https://www.kaggle.com/taebum/anime-sketch-colorization-pair), 2018.
- 605
- 606 Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*
607 *arXiv:1312.6114*, 2013.
- 608 Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and
609 content representation. In *The Eleventh International Conference on Learning Representations*,
610 2023. URL <https://openreview.net/forum?id=Nayau9fwXU>.
- 611
- 612 Elizaveta Levina and Peter Bickel. The earth mover’s distance is the mallows distance: Some in-
613 sights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision*.
614 *ICCV 2001*, volume 2, pp. 251–256. IEEE, 2001.
- 615 Chengze Li, Xueting Liu, and Tien-Tsin Wong. Deep extraction of manga structural lines. *ACM*
616 *Transactions on Graphics (SIGGRAPH 2017 issue)*, 36(4):117:1–117:12, July 2017.
- 617
- 618 Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring
619 contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer*
620 *Vision (WACV)*, pp. 1403–1412. IEEE, 2019.
- 621 llyasviel. sketchkeras. <https://github.com/llyasviel/sketchKeras>, 2017.
- 622
- 623 Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion
624 hyperfeatures: Searching through time and space for semantic correspondence. *arXiv preprint*
625 *arXiv:2305.14334*, 2023.
- 626
- 627 Kanti V Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57
628 (3):519–530, 1970.
- 629
- 630 Kanti V Mardia. Applications of some measures of multivariate skewness and kurtosis in testing
631 normality and robustness studies. *Sankhyā: The Indian Journal of Statistics, Series B*, pp. 115–
632 128, 1974.
- 633
- 634 David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented
635 natural images and its application to evaluating segmentation algorithms and measuring ecological
636 statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*,
637 volume 2, pp. 416–423. IEEE, 2001.
- 638
- 639 Haoran Mo, Edgar Simo-Serra, Chengying Gao, Changqing Zou, and Ruomei Wang. General virtual
640 sketching framework for vector line art. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.
- 641
- 642 Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models.
643 In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.
- 644
- 645 Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov,
646 Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning
647 robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- 648
- 649 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
650 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
651 models from natural language supervision. In *International conference on machine learning*, pp.
652 8748–8763. PMLR, 2021.

- 648 Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen,
649 and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine*
650 *Learning*, pp. 8821–8831. PMLR, 2021.
- 651
- 652 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
653 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF confer-*
654 *ence on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- 655 Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomed-
656 ical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–*
657 *MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceed-*
658 *ings, Part III 18*, pp. 234–241. Springer, 2015.
- 659 Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analy-
660 sis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- 661
- 662 Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman.
663 Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Pro-*
664 *ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–
665 22510, 2023.
- 666 Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar
667 Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic
668 text-to-image diffusion models with deep language understanding. *Advances in Neural Informa-*
669 *tion Processing Systems*, 35:36479–36494, 2022.
- 670
- 671 Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis,
672 Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of
673 clip-filtered 400 million image-text pairs, 2021.
- 674 Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi
675 Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An
676 open large-scale dataset for training next generation image-text models. *Advances in Neural*
677 *Information Processing Systems*, 35:25278–25294, 2022.
- 678
- 679 Chang Wook Seo, Amirsaman Ashtari, and Junyong Noh. Semi-supervised reference-based sketch
680 extraction using a contrastive learning framework. *ACM Transactions on Graphics (TOG)*, 42(4):
681 1–12, 2023.
- 682 Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete
683 samples). *Biometrika*, 52(3/4):591–611, 1965.
- 684
- 685 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv*
686 *preprint arXiv:2010.02502*, 2020.
- 687 Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent
688 correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023.
- 689
- 690 Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for
691 text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Com-*
692 *puter Vision and Pattern Recognition*, pp. 1921–1930, 2023.
- 693 Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano,
694 Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching.
695 *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.
- 696
- 697 Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with
698 different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Confer-*
699 *ence on Computer Vision*, pp. 4146–4156, 2023.
- 700 Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-
701 resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of*
the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

- 702 Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment:
703 from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–
704 612, 2004.
- 705 Nora S Willett, Fernando de Goes, Kurt Fleischer, Mark Meyer, and Chris Burrows. Stylizing
706 ribbons: Computing surface contours with temporally coherent orientations. *IEEE Transactions*
707 *on Visualization and Computer Graphics*, 2023.
- 708
- 709 Holger Winnemöller. Xdog: advanced image stylization with extended difference-of-gaussians. In
710 *Proceedings of the ACM SIGGRAPH/eurographics symposium on non-photorealistic animation*
711 *and rendering*, pp. 147–156, 2011.
- 712
- 713 Holger Winnemöller, Jan Eric Kyprianidis, and Sven C Olsen. Xdog: An extended difference-of-
714 gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):
715 740–753, 2012.
- 716 Xiaoyu Xiang, Ding Liu, Xiao Yang, Yiheng Zhu, and Xiaohui Shen. Anime2sketch: A
717 sketch extractor for anime arts with deep networks. [https://github.com/Mukosame/
718 Anime2Sketch](https://github.com/Mukosame/Anime2Sketch), 2021.
- 719 Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE*
720 *international conference on computer vision*, pp. 1395–1403, 2015a.
- 721
- 722 Saining Xie and Zhuowen Tu. Holistically-nested edge detection, 2015b.
- 723 XiMing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text
724 guided vector sketch synthesis through latent diffusion models. In *Thirty-seventh Conference on*
725 *Neural Information Processing Systems*, 2023. URL [https://openreview.net/forum?
726 id=CylxatvEQj](https://openreview.net/forum?id=CylxatvEQj).
- 727
- 728 Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-
729 vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the*
730 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2955–2966, 2023.
- 731 Serin Yang, Hyunmin Hwang, and Jong Chul Ye. Zero-shot contrastive loss for text-guided diffusion
732 image style transfer. *arXiv preprint arXiv:2303.08622*, 2023.
- 733
- 734 Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Apdrawinggan: Generating artistic portrait
735 drawings from face photos with hierarchical gans. In *Proceedings of the IEEE/CVF conference*
736 *on computer vision and pattern recognition*, pp. 10743–10752, 2019.
- 737 Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Unpaired portrait drawing generation via
738 asymmetric cycle mapping. In *Proceedings of the IEEE/CVF conference on computer vision and*
739 *pattern recognition*, pp. 8217–8225, 2020.
- 740 Soyeon Yoon, Kwan Yun, Kwanggyoon Seo, Sihun Cha, Jung Eun Yoo, and Junyong Noh. Lego:
741 Leveraging a surface deformation network for animatable stylized face generation with one ex-
742 ample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*
743 *(CVPR)*, pp. 4505–4514, June 2024.
- 744
- 745 Kwan Yun, Kwanggyoon Seo, Chang Wook Seo, Soyeon Yoon, Seongcheol Kim, Soohyun Ji, Amir-
746 saman Ashtari, and Junyong Noh. Stylized face sketch extraction via generative prior with limited
747 data. In *Computer Graphics Forum*, pp. e15045. Wiley Online Library, 2024.
- 748 Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun,
749 and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot
750 semantic correspondence. *arXiv preprint arXiv:2305.15347*, 2023a.
- 751 Kaihua Zhang, Lei Zhang, Kin-Man Lam, and David Zhang. A level set approach to image segmen-
752 tation with intensity inhomogeneity. *IEEE transactions on cybernetics*, 46(2):546–557, 2015.
- 753
- 754 Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image
755 diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
pp. 3836–3847, 2023b.

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

759
760 Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Chang-
761 sheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF*
762 *conference on computer vision and pattern recognition*, pp. 10146–10156, 2023c.

763 Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Mind the gap: Domain gap control for
764 single shot domain adaptation for generative adversarial networks. In *International Conference*
765 *on Learning Representations*, 2022.

766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

APPENDIX

This appendix consists of 5 Sections. Sec. A describes implementation details. Sec. B provides additional details and findings on diffusion features selection. Sec. C presents extended details of VAE decoder features. Sec. D contains the results of additional experiments on CDST. Sec. E presents additional comparison with baselines and additional qualitative results with various style sketches.

A. IMPLEMENTATION DETAILS

DiffSketch DiffSketch leverages Stable Diffusion v1.4 sampled with DDIM Song et al. (2020) pretrained with the LAION-5B Schuhmann et al. (2022) dataset, which produced images of resolution 512×512 . With the pretrained Stable Diffusion, we use a total of 50 time steps T for sampling. The training of DiffSketch was performed for 1200 iterations which required less than 3 hours on an Nvidia V100 GPU. For the training using HED Xie & Tu (2015b), we concatenated the first two layers with the first three layers to stylize sketch. In case of XDoG Winnemöller (2011), we used the Gary Grossi style.

DiffSketch_{distilled} DiffSketch_{distilled} was developed to conduct sketch extraction efficiently with the streamlined generator. The training of DiffSketch_{distilled} was performed for 10 epochs for 30,000 sketch-image pairs generated from DiffSketch, following CDST. The training of DiffSketch_{distilled} required approximately 5 hours on two Nvidia A6000 GPUs. The average inference time of both DiffSketch and DiffSketch_{distilled} was 4.74 seconds and 0.0139 seconds, respectively, when tested on an Nvidia A5000 GPU with 1,000 images with resolutions of 512×512 using automatic precision.

B. DIFFUSION FEATURES SELECTION

B.1 DETAILS OF DIFFUSION FEATURE SELECTION PROCESS AND ANALYSIS

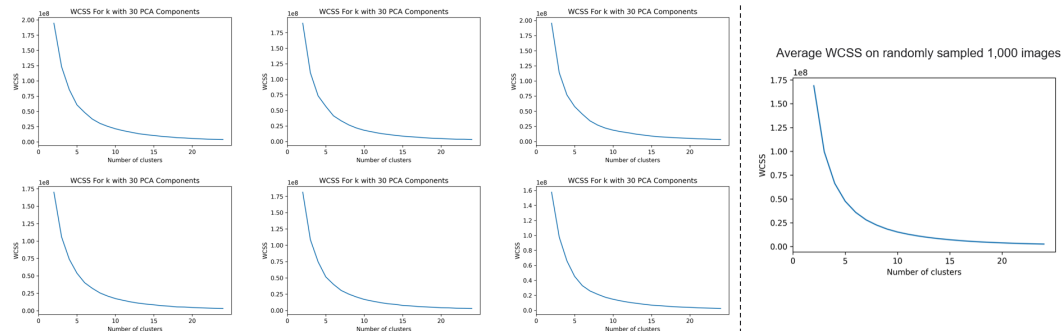


Figure 8: Visualization of WCSS values according to the number used for K-means clustering. The left plots are the WCSS values of the features from randomly sampled images while the right plot shows the average WCSS values of the features from all images.

To conduct K-means clustering for diffusion feature selection, we first employed the elbow method, visualizing the results. However, a distinct elbow was not visually apparent, as shown in Fig. 8. The left 6 images are WCSS values from randomly selected images. All 6 plots show similar patterns, making it hard to select a definitive elbow as stated in the main paper. The right image, which exhibits similar results, shows the average of WCSS on all 50,000 UNet features from 1,000 different images.

Therefore, we chose to use the Silhouette score Rousseeuw (1987) and Davies-Bouldin index Davies & Bouldin (1979), which are two of the most widely used numerical methods when choosing the optimal number of clusters. However, they are two different methods, whose results do not always match with each other. We first visualized and found the contradicting results of these two methods

as shown in Fig. 9. Therefore, we chose to use the one that first matches the i^{th} highest silhouette score and the i^{th} lowest Davies-Bouldin index simultaneously. This process of choosing the optimal number of clusters can be written as follows :

Algorithm 1 Finding the Optimal Number of Clusters

```

1:  $MAX\_clusters = Total\_time\_steps/2$ 
2:  $sil\_indicies \leftarrow sorted(range(MAX\_clusters), key = \lambda k : silhouette\_scores[k], reverse = True)$ 
3:  $db\_indicies \leftarrow sorted(range(MAX\_clusters), key = \lambda k : db\_scores[k], reverse = False)$ 
4: for  $i \leftarrow 0$  to  $MAX\_clusters$  do
5:   if  $sil\_indicies[i]$  in  $db\_indicies[i + 1]$  then
6:      $k\_optimal = sil\_indicies[i] + 1$ 
7:     break
8:   end if
9: end for

```

We conducted this process twice with two different numbers of PCA components (10 and 30), yielding the results shown in Fig. 10. The averages (13.26 and 13.34) and standard deviations (0.69 and 0.69) were calculated. As the mode value with both PCA components was 13, and the rounded average was also 13, we chose our optimal k to be 13. Using this number of clusters, we chose the representative feature as the one nearest to the center of each cluster. From this process, we ended up with the following t values: [0, 3, 8, 12, 16, 21, 25, 28, 32, 35, 39, 43, 47].

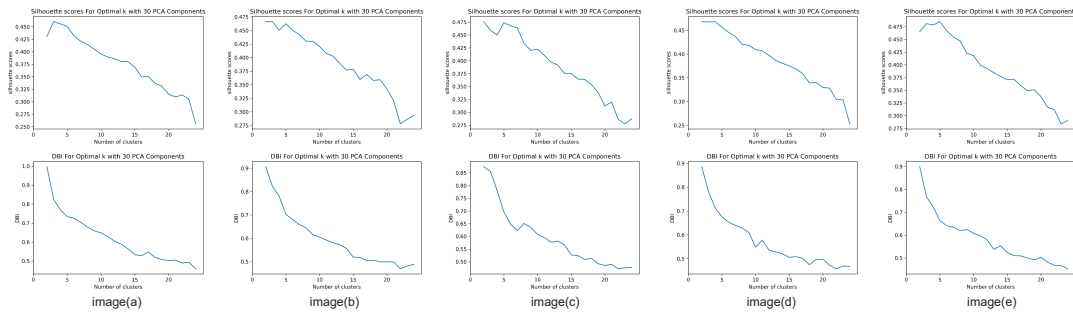


Figure 9: Visualization of contradicting results of Silhouette scores and Davis Bouldin indices on five different images.

In the main paper, we identified several key insights through the visualization of features. For future research and to provide additional insights, we manually classified images and visualized the trajectory of features from different classes as shown in Fig. 11. Here, we summarize extensively about our findings through feature analysis. First, semantically similar images lead to similar trajectories, although not identical. Second, features in the initial stage of the diffusion process (when t is approximately 50) retain similar information despite significant differences in the resulting images. Third, features in the middle stage of the diffusion process (when t is around 25) exhibit larger differences between adjacent features in their time steps. Lastly, the feature at the final time step (t=0) possesses distinctive information, varying significantly from previous values. This is also evident in the additional visualization presented in Fig. 11.

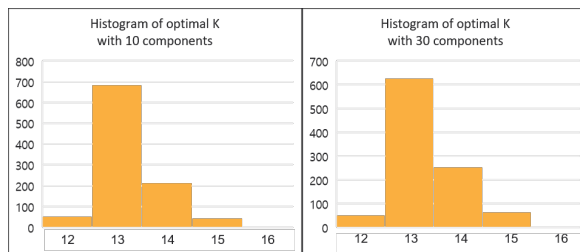


Figure 10: Visualization of histogram for the optimal k value with different numbers of PCA components.

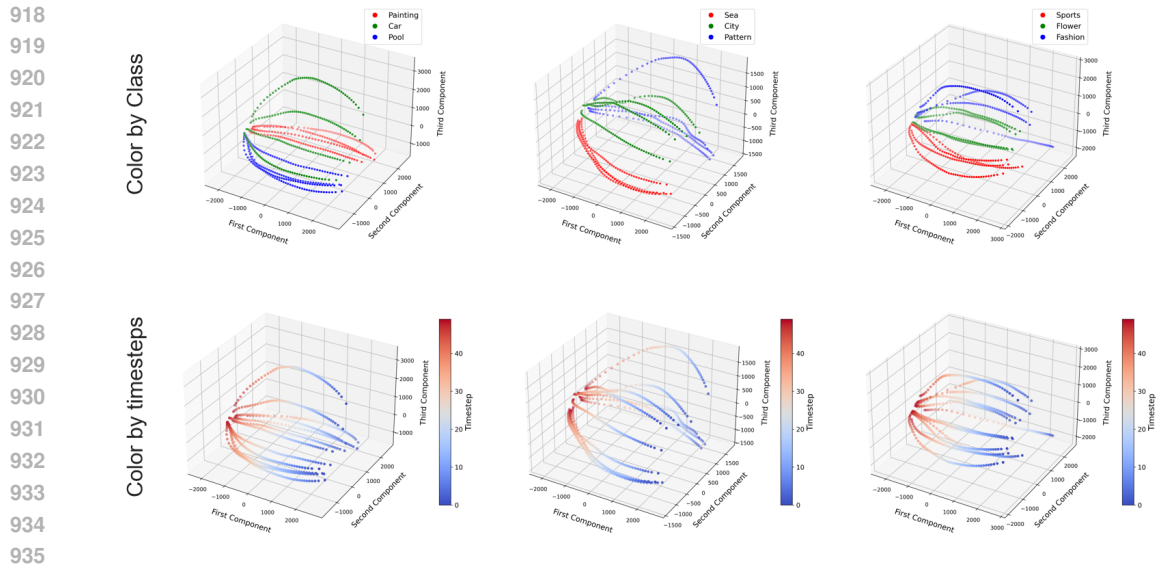


Figure 11: Additional analysis on sampled features. PCA is applied to DDIM sampled features from different classes. Up : features colored with human-labeled classes. Down : features colored with denoising timesteps

The automatically selected features indicate a prioritization of the final feature ($t=0$), and the selection was made more from middle steps than from initial steps ($t=[21,25,28]$ versus $t=[43,47]$). Our finding offers some guidance for manual feature selection to consider the time steps, especially when memory is constrained. The order of the preference is the features from the last step ($t=0$), from the middle (t is near 25), and from middle to final time steps while the features from initial steps are preferred less in general. For instance, when selecting five features from 50 time steps, a possible selection could be $t=[0, 10, 20, 30, 40]$ instead of using simple equal timesteps ($t = [9, 18, 27, 36, 45]$). However, for a task of semantic correspondence or segmentation, it is known that features from last 0 to 30% are more informative Baranchuk et al. (2021); Xu et al. (2023); Tang et al. (2023); Zhang et al. (2023a), therefore one possible choice can be $[0, 7, 14, 24, 34]$.

B.2 FEATURES FROM INVERSION, DIFFERENT STEPS, AND MODEL

While we focused on $T=50$ DDIM sampling, for generalization, we examined different intervals ($T=25, T=100$) and different models. For these experiments, we randomly sampled 100 images. While our previous experiments reported in Fig. 11 were conducted with manually classified images, we utilized DINOv2 Oquab et al. (2023), which was trained in a self-supervised manner and has learned visual semantics. With DINOv2, we separated the data into 15 different clusters and followed the process described in the main paper to plot the features. Here, we used 15 images from each cluster to calculate the PCA axis while we used 17 classes in the main experiments. The results, as shown in Fig. 13 and Fig. 14, indicate that even with different sampling methods, the same conclusions regarding the sampling method can be drawn. The last feature exhibits a distinct value, while the features from the initial time step have similar values.

In addition, we also tested on features extracted during the inversion process. We randomly selected 20 images from human face Karras et al. (2019) and cat photos Choi et al. (2020) to plot the features as shown in Fig. 12. Lastly, we tested on another model, Stable diffusion V2.1 which produces 768×768 images. Following the same process, we randomly sampled 100 images and clustered with DINOv2 and plot the results as shown in Fig. 15. This result also shows that even with different models with different resolutions, the same conclusions can be drawn, showing the scalability of our analysis.

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

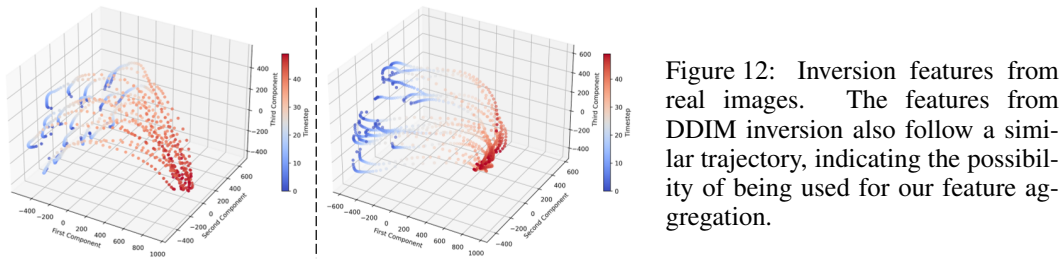


Figure 12: Inversion features from real images. The features from DDIM inversion also follow a similar trajectory, indicating the possibility of being used for our feature aggregation.

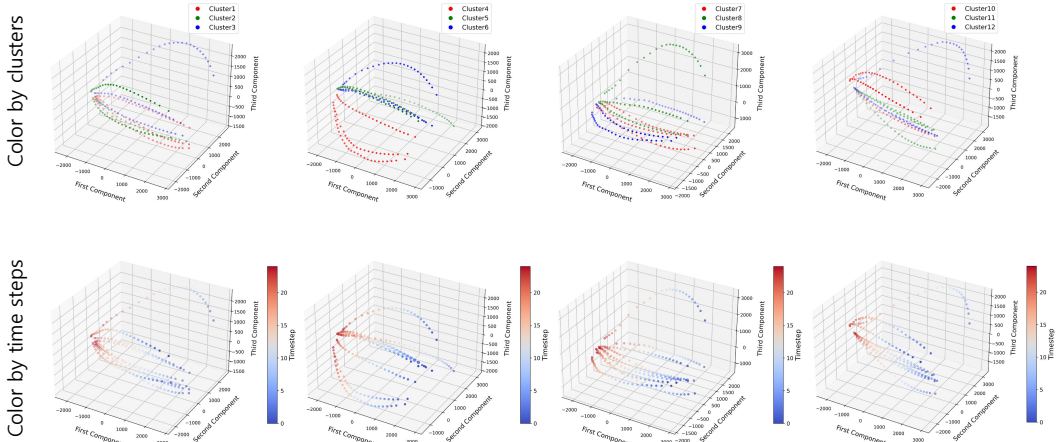


Figure 13: Additional analysis on sampled features. PCA is applied to 25 steps of DDIM sampled features with different clusters. Up : features colored with DINOv2 clusters. Down : features colored with denoising timesteps.

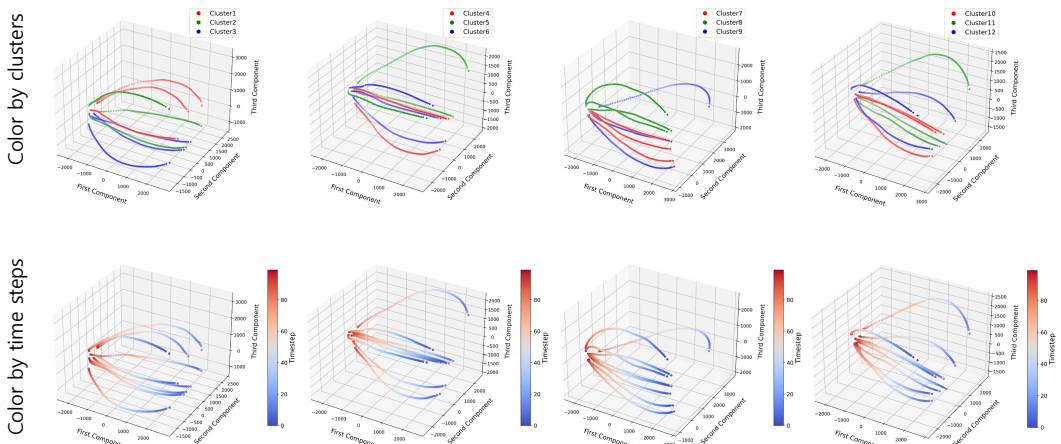


Figure 14: Additional analysis on sampled features. PCA is applied to 100 steps of DDIM sampled features with different clusters. Up : features colored with DINOv2 clusters. Down : features colored with denoising timesteps.

1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

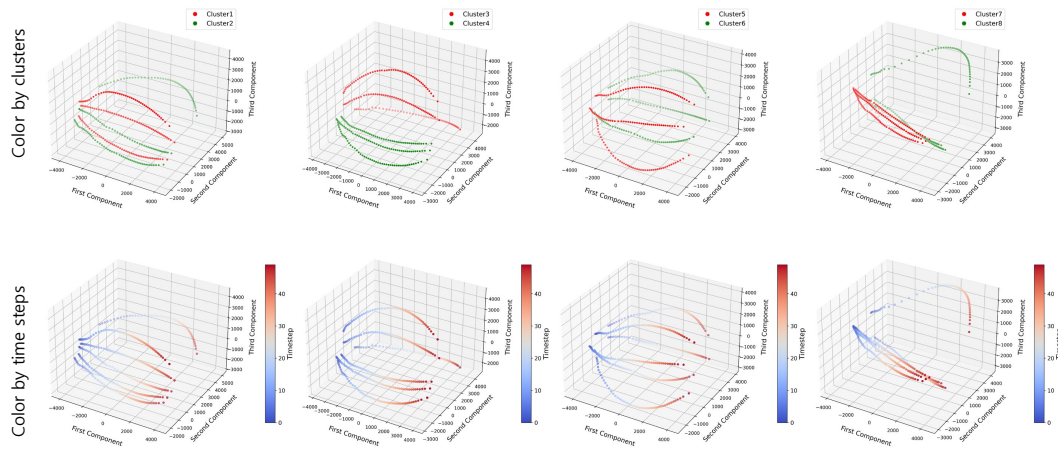


Figure 15: Additional analysis on Stable diffusion v2.1 sampled features. PCA is applied to 50 steps of DDIM sampled features with different clusters. Up : features colored with DINOv2 clusters. Down : features colored with denoising timesteps.

C. VAE DECODER FEATURES

The VAE features were fused with the Aggregation network features using FFD in the proposed model architecture to add fine details of the image. Fig. 16 shows a visualization of the VAE features. We used a set of 20 generated face images and extracted features from different decoder layers of the UNet and VAE decoders, at the last time step (t=0) similar to that of PNP Tumanyan et al. (2023). We observed that the use of VAE decoder resulted in higher-frequency details than the UNet decoder. While the features from the UNet decoder contain semantic information, the features from the VAE decoder produced finer details such as hair, wrinkles, and small letters.

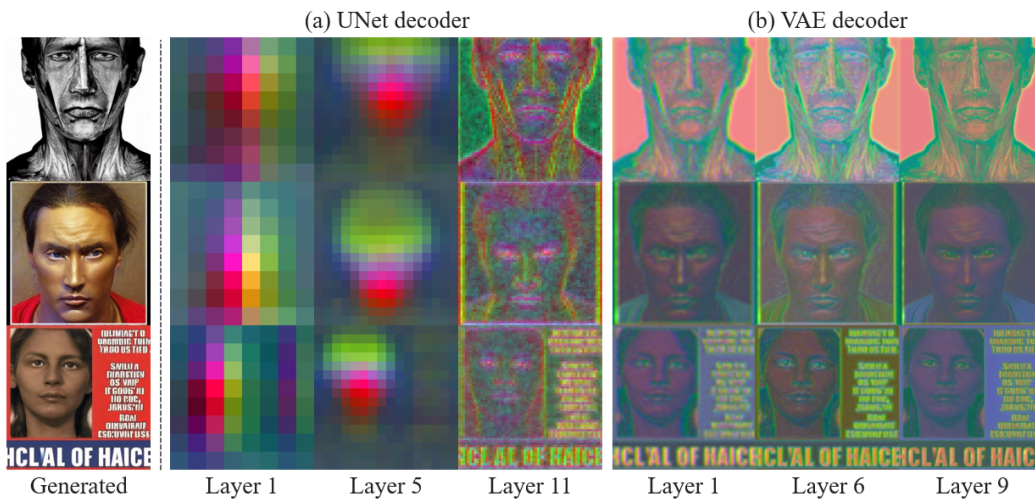


Figure 16: Extended visualization of features from UNet and VAE. (a) shows the UNet decoder features in lower resolution (layers 1), intermediate resolution (layers 5), and higher resolution (layers 11). (b) shows the VAE decoder features in lower resolution (layers 1), intermediate resolution (layers 6), and higher resolution (layers 9).

1080 D. CONDITION DIFFUSION SAMPLING FOR TRAINING

1081 D.1 ADDITIONAL DETAILS ON CDST

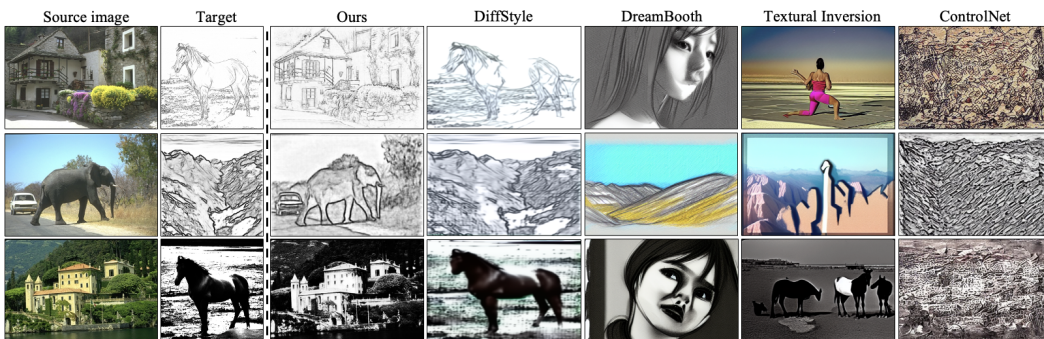
1082 As stated in the main paper, we randomly sampled 100k prompts to estimate the distribution of
 1083 SD. Specifically, we tokenized and embedded these 100k prompts in the space of the CLIP model.
 1084 With this embedding, we conducted PCA to extract 512 principal components. We then checked the
 1085 normality of the sampled embeddings with all 512 principal component axes using the Shapiro-Wilk
 1086 test Shapiro & Wilk (1965) with a significance level of $\alpha = 5\%$.

1087 As a result, 214 components rejected the null hypothesis of normality. This indicates that each of its
 1088 marginals cannot be assumed to be univariate normal. Next, we conducted the Mardia test Mardia
 1089 (1970; 1974) with the same 100k samples, taking into account skewness and kurtosis to check if
 1090 the distribution is multivariate. The results failed to reject the null hypothesis of normality with a
 1091 significance level of $\alpha = 5\%$. Therefore, we assumed D_{SD} as a multivariate normal distribution for
 1092 our sampling during training. In addition, we calculated the Earth Moving Distance (EMD) Levina
 1093 & Bickel (2001) with 100k samples from LAION- 400M, which were not used for our analysis.
 1094 For comparison, we used the normal distribution for each axis, and the uniform distribution to find
 1095 that our \mathcal{N} (244.22) is lower than the normal distribution for each axis (244.31) and the uniform
 1096 distribution (1480.57).
 1097
 1098

1099 E. ADDITIONAL EXPERIMENTS

1100 E.1 ADDITIONAL COMPARISON WITH BASELINES

1101 As stated in the main paper, we presented four additional methods that failed to extract sketches
 1102 following the desired style or exhibited severe artifacts. As shown in Figure 17, few-shot finetuning
 1103 methods Ruiz et al. (2023); Gal et al. (2022) were unable to extract sketches when trained with a
 1104 single example. The results of ControlNet Zhang et al. (2023b) showed severe artifacts because
 1105 the method was originally proposed to be trained with thousands of images. Diffstyle Chung et al.
 1106 (2024a), on the other hand, failed to preserve the content of source image. We also calculated LPIPS,
 1107 SSIM, and FID scores, as in our main experiments, and as noted in Tab. 6, our method achieved the
 1108 highest scores across all metrics.
 1109
 1110
 1111
 1112



1124 Figure 17: Experiment results on comparison with four additional baselines.
 1125
 1126
 1127

1128 E.2 EXAMPLES IN EXPERIMENTS

1129 We presented quantitative results and visual comparison with and without using CDST for the ablation
 1130 study described in the main paper. Here, we visualize additional results of the study in Fig. 18.
 1131 For a perceptual study, a total of 23 participants were asked to make 20 different comparisons and
 1132 determine which sketch style appeared most similar to the target sketch. Examples of our perceptual
 1133 study is provided in Fig. 19 and Fig. 20.

Table 6: Quantitative comparison of different methods on the BSDS500 datasets. Best scores are denoted in bold, and the second-best are underlined.

Methods	BSDS500 - anime			BSDS500 - HED			BSDS500 - XDoG			BSDS500 - average		
	LPIPS	SSIM	FID	LPIPS	SSIM	FID	LPIPS	SSIM	FID	LPIPS	SSIM	FID
Ours	0.218	0.493	126.5	0.227	0.593	110.6	0.143	0.649	62.8	0.196	0.578	100.0
DiffStyle	0.542	0.361	206.7	0.572	0.124	422.2	0.676	0.069	317.6	0.597	0.185	315.5
DreamBooth	0.806	0.302	233.5	0.746	0.185	277.8	0.723	0.195	276.1	0.758	0.227	262.5
TI	0.828	0.264	284.2	0.771	0.164	313.1	0.647	0.220	237.4	0.749	0.216	278.2
ControlNet	0.901	0.021	303.3	0.699	0.028	328.7	0.627	0.031	278.7	0.742	0.027	303.6

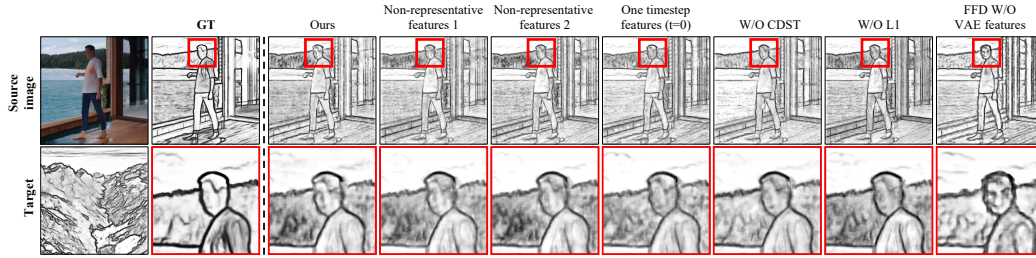


Figure 18: Visual examples of the ablation study. Ours generates higher quality results with details such as face, separated with hair region, compared to the alternatives.

E.3 QUALITATIVE RESULTS

We present additional results of our method extracted in diverse styles which share the source image, in Fig. 21 and those of the comparison with baselines in Fig. 22. The additional comparison results further confirm that DiffSketch_{distilled} extract superior results compared to the baseline methods.

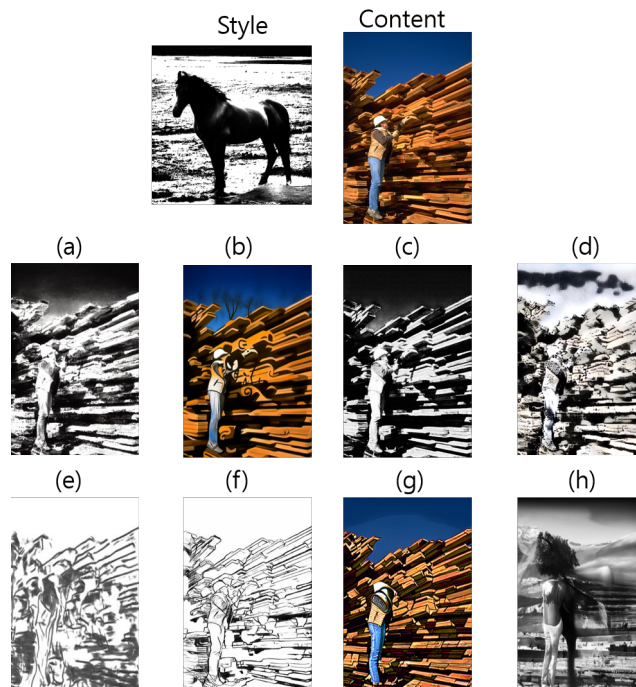


Figure 19: Example results of perceptual study. Participants were asked to choose one sketch image that has a style most similar to the style image while preserving the content of the content image faithfully. (c) is ours.

1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241

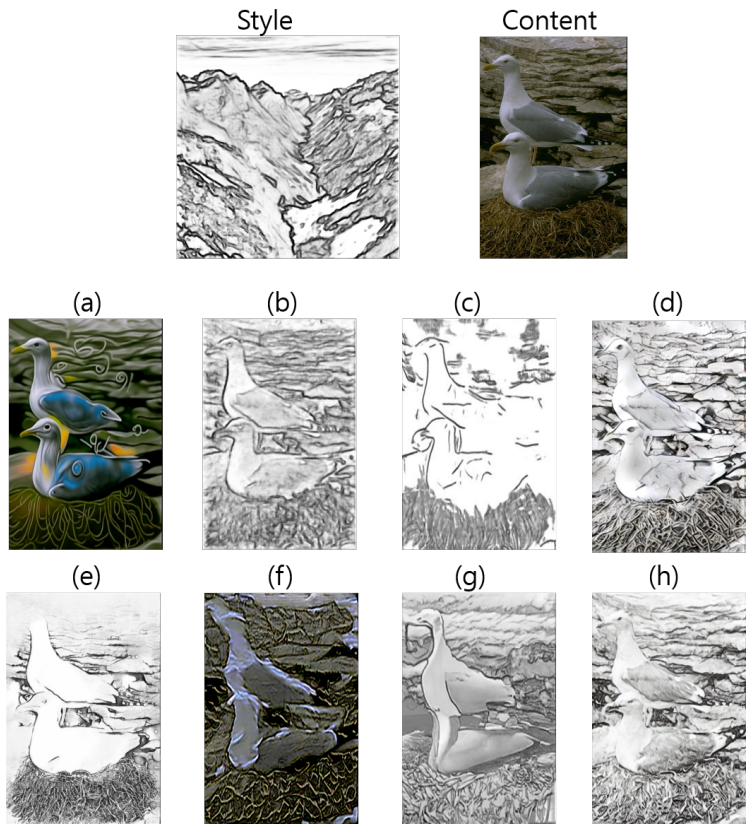


Figure 20: Example results of perceptual study. Participants were asked to choose one sketch image that has a style most similar to the style image while preserving the content of the content image faithfully. (b) is ours.

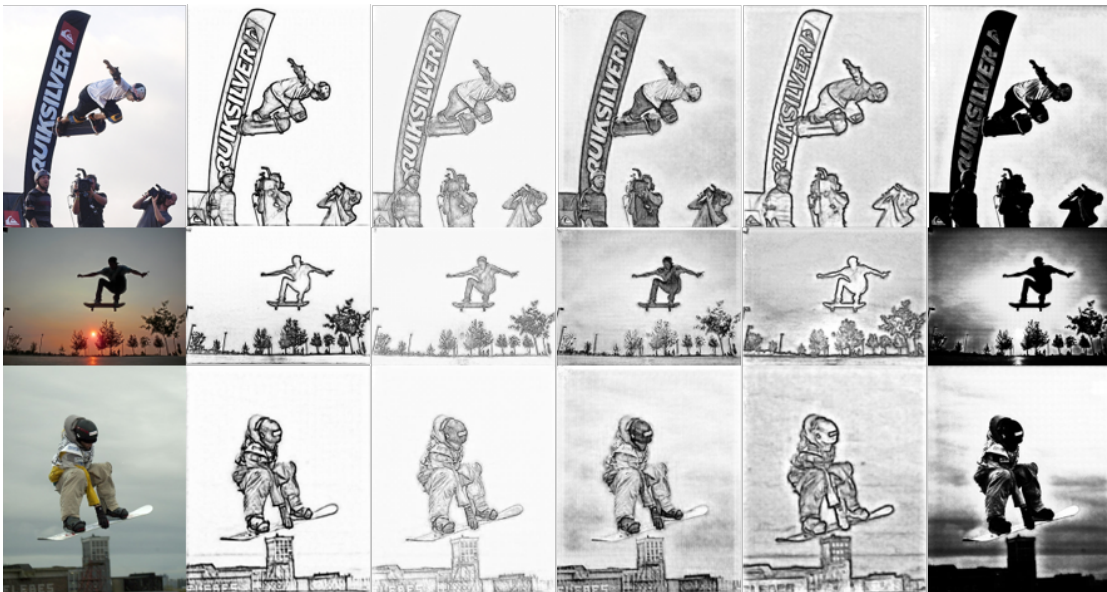


Figure 21: Additional results of DiffSketch_{distilled} from shared sources.

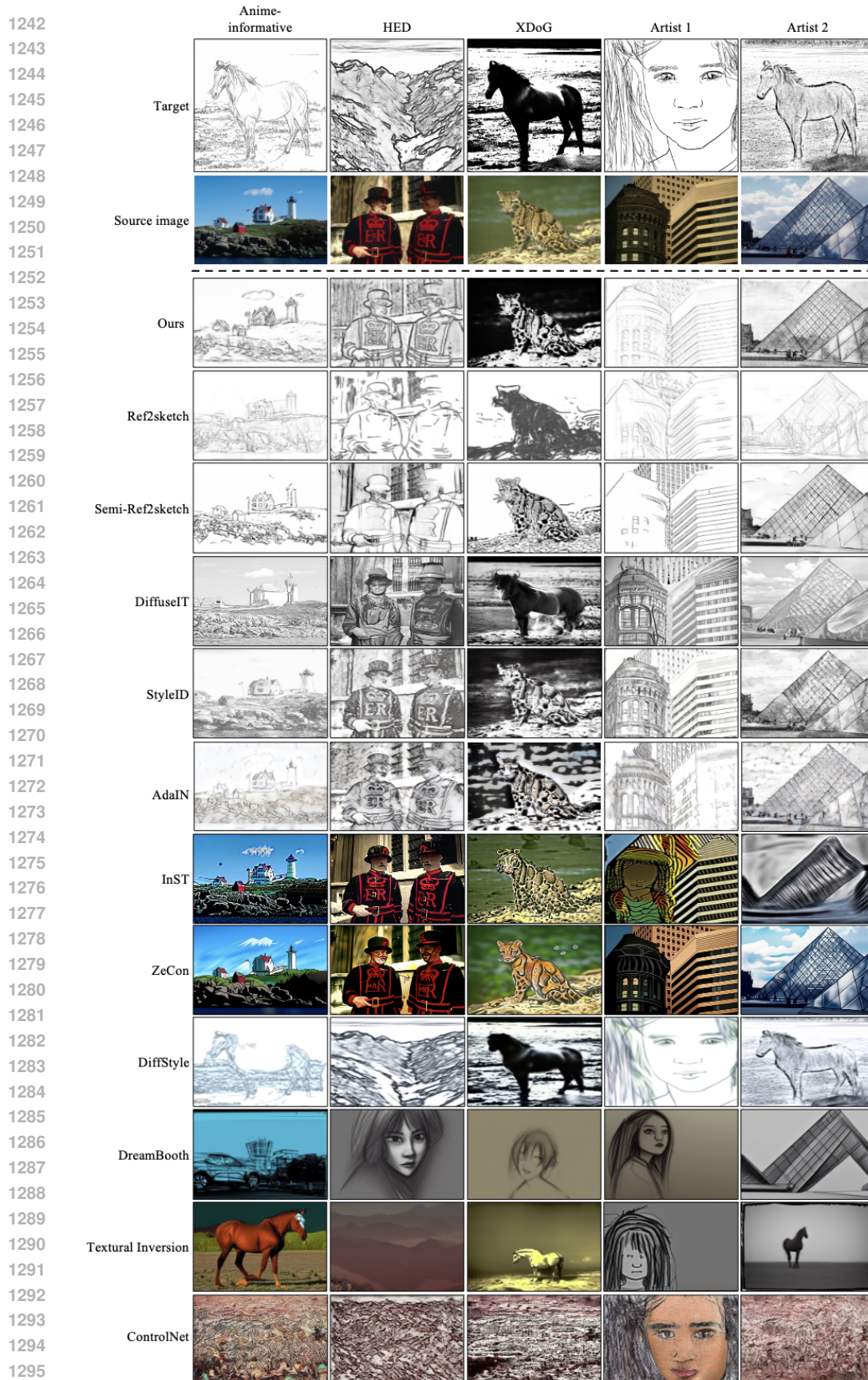


Figure 22: Qualitative comparison with alternative sketch extraction methods on the BSDS500 dataset.