ColorizeDiffusion: Adjustable Sketch Colorization with Reference Image and Text

Dingkun Yan
Tokyo Institute of Technology
Tokyo, Japan

Liang Yuan
Keio University
Tokyo, Japan

Yuma Nishioka
Tokyo Institute of Technology
Tokyo, Japan

Issei Fujishiro
Keio University
Tokyo, Japan

Suguru Saito
Tokyo Institute of Technology
Tokyo, Japan

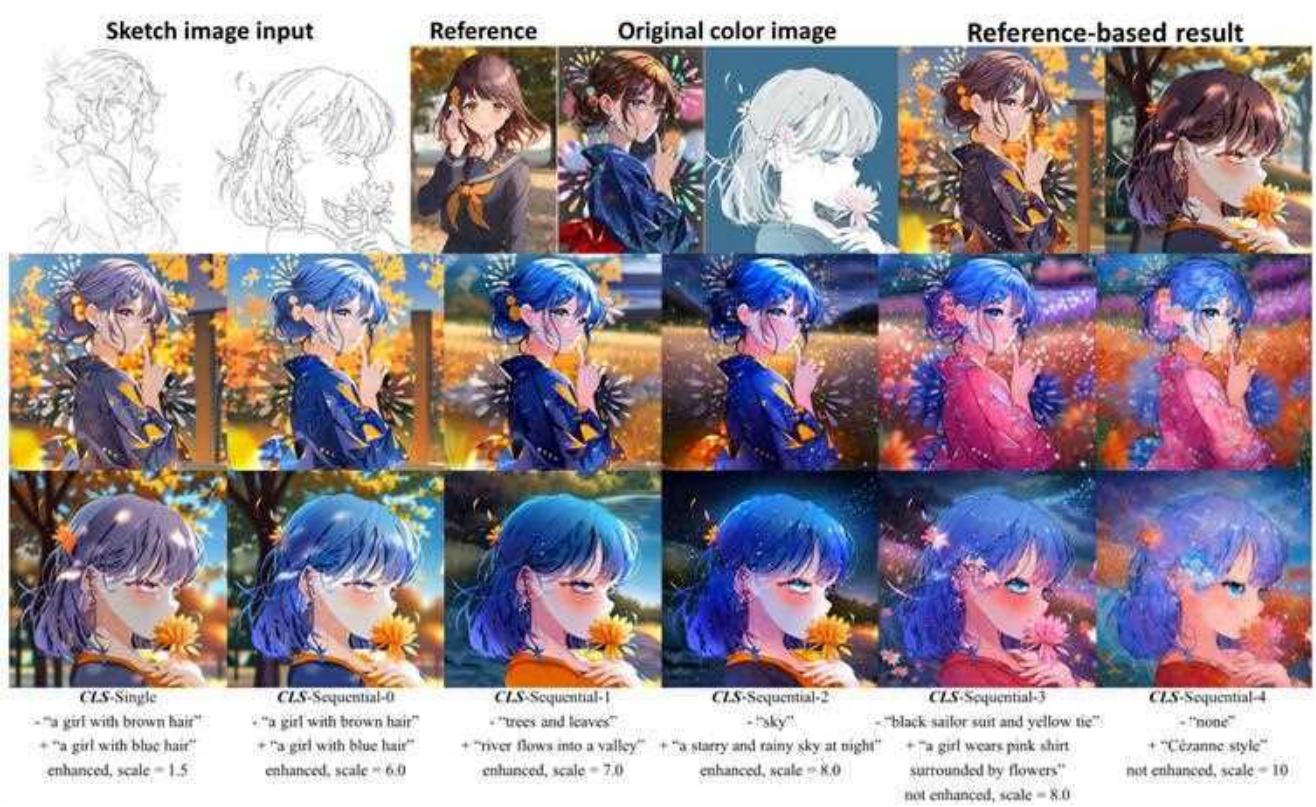

Figure 1. Our method colorizes sketch images based on a reference image and allows to edit the results sequentially using arbitrary text inputs with specified degrees. "+" and "-" denote the target text and anchor text for our text-based latent interpolation, respectively.

Abstract

Recently, diffusion models have demonstrated their effectiveness in generating extremely high-quality images and have found wide-ranging applications, including automatic sketch colorization. However, most existing models use text

to guide the conditional generation, with fewer attempts exploring the potential advantages of using image tokens as conditional inputs for networks. As such, this paper exhaustively investigates image-guided models, specifically targeting reference-based sketch colorization, which aims to colorize sketch images using reference color images. We investigate three critical aspects of reference-based diffusion models: the shortcomings compared to text-based counterparts, the training strategies, and the capability in zero-shot, sequential text-based manipulation. We introduce two variations of an image-guided latent diffusion model using different image tokens from the pre-trained CLIP image encoder, and we propose corresponding manipulation methods to adjust their results sequentially using weighted text inputs. We conduct comprehensive evaluations of our models through qualitative and quantitative experiments, as well as a user study. Code link: https://github.com/ydk-tellurion/colorizeDiffusion.

1. Introduction

Anime-style images have gained worldwide popularity over the past few decades thanks to their diverse color composition and captivating character design, but the process of colorizing sketch images remains laborious and time-consuming. Yet, swift advancements to diffusion models [17, 60] now enable large generative models to create remarkably high-quality images across a variety of domains, including anime style. While effective conditional diffusion models predominantly focus on text-based generation, few specialize in reference-based colorization, a complex generation task that utilizes both a reference and a sketch image. As such, this paper focuses on reference-based colorization by thoroughly analyzing a major challenge in the training of related models, examining the selection of reference conditions, exploring training strategies for relevant neural networks, and proposing two zero-shot text-based manipulation methods using tokens from pre-trained CLIP encoders.

A salient issue in multi-conditioned generation is the potential conflict between input conditions, which might not significantly impact methods using sketch and text conditions, as users are less likely to invoke contradictory conditions and can readily modify the text when such conflicts occur. Yet, this problem widely exists in reference-based colorization, because both sketch and reference images contain varied object identity information with potentially incompatible contents.

However, text-based models also exhibit several limitations when compared to image-guided methods, one of which is their inability to transfer features accurately from a reference image and to reflect the progressive changes in results for weighted text inputs [19,41,43], such as transitioning from "blue hair: 0.5" to "blue hair: 1.5." These problems arise due to the lack of image-based cross-attention modules, and the discrete representations of text embeddings. When trained using image-based cross-attention modules and image features that adapt in response to the confidence of corresponding attributes, such as hair color, image-guided models [11,26,31,38,40,57] could effectively mitigate this problem.

Given that anime-style images [6] are more sensitive to color variations and encapsulate ample visual attributes within each image, they are suitable to aid in analyzing the proposed reference-based generation and text-based manipulation methods. Our research demonstrates that reference-based models, leveraging image tokens from pre-trained CLIP encoders as conditions, are capable of progressively adapting their outputs in response to weighted text inputs.

Through rigorous experimentation with ablation models and baselines, we empirically prove the effectiveness of the proposed methods in reference-based colorization, style transfer, and text-based manipulation. We further conducted a user study to evaluate the proposed methods subjectively.

The contributions of this paper can be summarized as follows:

- We conduct a comprehensive investigation into the application of image-guided latent diffusion models to sketch colorization, especially of the distribution problem and training strategies.
- We design two zero-shot manipulation methods for the proposed reference-based models.

The following sections of this paper are organized as follows: Section 2 briefly reviews related works, including Latent Diffusion Models (LDMs), neural style transfer, and image colorization. Section 3 outlines the workflow of popular LDMs, describes a significant challenge called the "distribution problem," and introduces the proposed reference-based training and zero-shot manipulation methods. Section 4 presents several ablation studies and experimental comparisons with baselines, and Section 5 draws conclusions for this paper. Additional qualitative results related to the distribution problem, discussed in Sections 3 and 4, respectively, are included in the appendix for reference.

2. Related Work

Our work focuses on reference-based sketch colorization, an important subfield of image generation. We utilize the score-based generative model as our neural backbone, which is widely known as the diffusion model. Our training methods and overall pipeline are designed following previous style transfer and colorization methods, pursuing pixel-level correspondence and fidelity to the input sketch

image.

Latent Diffusion Models. Diffusion probabilistic Models (DMs) [17] are a class of latent variable models inspired by considerations from nonequilibrium thermodynamics [48]. Compared with Generative Adversarial Nets (GANs) [4, 5, 13, 24, 25], DMs excel at generating highly realistic images across various contexts. However, the autoregressive denoising process, typically computed using a deep U-Net network [42], incurs substantial computational costs for both training and inference, which limits further applications. To address this limitation, LDM [41], also known as StableDiffusion (SD), employ a two-stage synthesis and carry out the diffusion/denoising process within a highly compressed latent space to reduce computational costs significantly. Concurrently, many efficient samplers have been proposed to accelerate the denoising process [34, 35, 49, 50]. In this paper, we adopt a pre-trained text-based SD model as our neural backbone, utilize DPM++ [35] as the default sampler, and employ classifier-free guidance [7, 18] to strengthen the reference-based performance.

Neural Style Transfer. First proposed in [12], Neural Style Transfer (NST) has now become a widely adopted technique compatible with many effective generative models. Reference-based colorization, which aims to transfer colors and textures from reference images to sketch images, can be viewed as a subclass of multi-domain style transfer. However, compared to traditional network-based NST methods [4, 5, 20, 23, 66], which typically train networks using feature-level restrictions, reference-based colorization requires a higher level of color correspondence with the reference while maintaining fidelity to the sketch inputs. Consequently, our method is developed based on the principles of conditional image-to-image translation [22] to ensure pixel-level correspondence between the sketch and colorized results. We also demonstrate the efficiency of our approach to sketch-based style transfer.

Image Colorization. Developing automatic colorization algorithms has been a popular topic in the image generation field for years, and many effective methods have been developed for this purpose, all of which can be divided into traditional [9, 10, 37, 52] or Deep Learning (DL)-based methods [22, 63] according to the adoption of deep neural networks. Our work is highly related to DL-based methods, as they have been proven effective in generating high-quality images and controlling outputs using various conditional inputs. According to the conditions, existing DL-based methods can be categorized into three types: text-based [27, 60, 67], user-guided [61, 64], and reference-based [1, 30, 51, 56]. Text-based methods adopt text tags/prompts as hints to guide colorization, and they are the most popular subclass nowadays, owing to sufficient pre-trained Text-to-Image (T2I) models, as well as many effective plug-in modules and fine-tuning methods [19, 43, 60]. However, most text-based models cannot precisely adjust the scale of specific prompts or transfer features from references without training; meanwhile, user-guided methods require users to specify colors manually for each region using color spots or spray [61], assuming the user has a basic knowledge of line art. Yan et al. investigated the possibility of combining image and text tag conditions, but it was ineffective at generating backgrounds and at handling complex references, like many other GAN-based methods [5, 30]. To overcome the limitations of reference-based methods, we comprehensively investigate the application of image-guided LDMs and propose novel manipulation methods to enable text-based control.

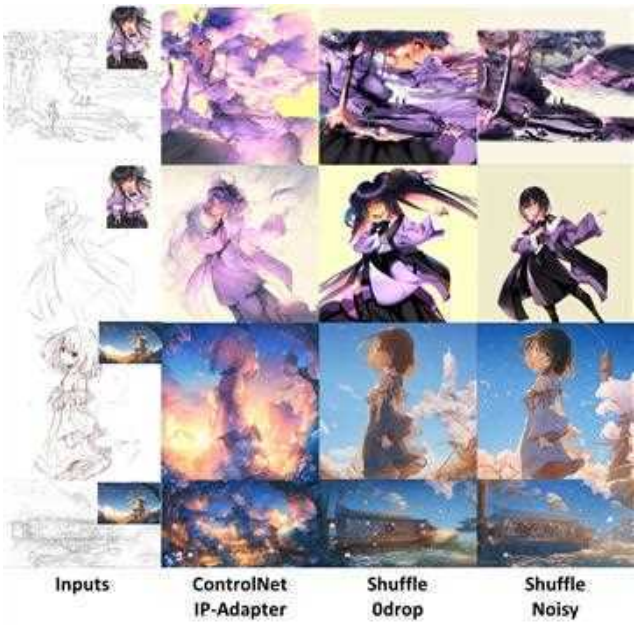

Figure 2. We generated different combinations of reference-based results using *ControlNet_lineart_anime + IP-Adapter v1.5* [57,62], our *shuffle-0drop* model, and *shuffle-noisy* model. We can observe that *ip-adapter* and *shuffle-0drop* generated incompatible texture.

3. Method

In this section, we briefly outline the workflow of LDMs in Section 3.1 and present the formulation of the so-called "distribution problem" that arises when applying LDMs to reference-based sketch colorization in Section 3.2. We propose various training strategies to tackle the data limitation and the distribution problem in Sections 3.3 and 3.4, and we design two zero-shot text-based manipulation methods

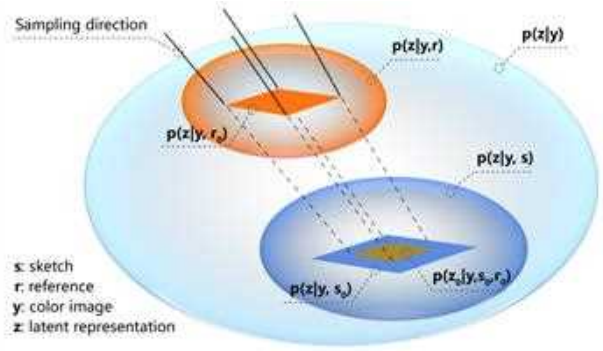

Figure 3. Ideal latent distributions and sampling process of reference-based colorization, where s_0 and r_0 denote specific instances of the sketch and reference, respectively.

for the proposed reference-based models in Sections 3.5 and 3.6, respectively.

3.1. Latent Diffusion and Denoising

1. Train a Variational AutoEncoder (VAE) [29] on the target image domain, comprising an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$ for perceptual compression and decompression, respectively.

2. The encoder $\mathcal{E}$ compresses an image y into latent representations $z_0 = \mathcal{E}(y)$ based on a scaling factor f, which is defined as $f = \frac{H}{h} = \frac{W}{w}$, where (H, W) and (h, w) denote the (height, width) of the input image and the latent representations, respectively. We set the scaling factor to 8 following popular SD models.

3. Autoregressively add noise $\epsilon \sim \mathcal{N}(0, 1)$ to z_0 through $z_t = \alpha_t z_0 + \beta_t \epsilon$, where t denotes the timestep, z_t the noisy representations, and α_t and β_t the hyper-parameters that control the noise schedule. This forward process is called diffusion and it is a fixed-length Markovian process with T steps in total, where T is set to $1,000$ in practice. The denoising U-Net θ learns to predict the noise ϵ at the t-step using the following function:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{E}(y),\epsilon,t,c}[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2], \qquad (1)$$

   where c denotes the guiding condition.

4. The denoising U-Net predicts ϵ_t to denoise z'_T to z'_0 autoregressively during the inference stage, where z'_T is usually a noise map sampled from a normal distribution.

5. Decompress the final latent representation to obtain the image output y' using the decoder $\mathcal{D}$, expressed as $y' = \mathcal{D}(z'_0)$.

Note that only steps 4 and 5 are undertaken during inference.

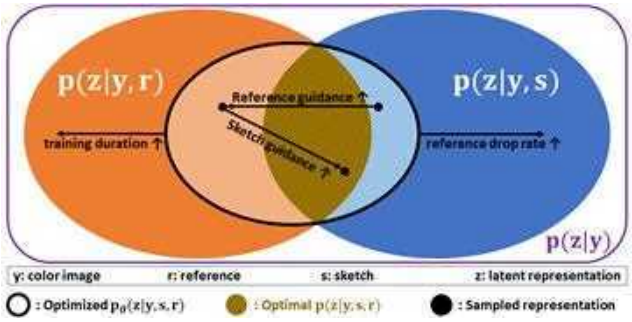

Figure 4. Illustration of the latent distribution and representations involved in the reference-based fine-tuning and sampling. The guidance scale is used for sampling with classifier-free guidance during inference. A higher guidance scale moves the denoised representations closer to the corresponding distribution, while longer training and a higher reference drop rate drag the optimized distribution $p_\theta(z|y, s, r)$ in the respective directions.

3.2. Distribution Problem

Because image-guided colorization primarily relies on reference images for color and texture generation, we introduce a significant challenge in this task, termed the "distribution problem." It can be defined as follows: given a target image domain $p(y)$, its corresponding latent distribution $p(z|y)$, and two conditions c_1, c_2, how does a generative model prioritize c_1 and c_2 when it is trained on the target distribution $y \sim p(z|y, c)$, where $c_1, c_2 \in c$? In the context of guided sketch colorization, where c_1 and c_2 represent the sketch image s and the guiding text/image r, respectively, it is preferable for the generated image to adhere more closely to the sketch rather than the reference image, especially for semantic regions. This problem is also reported in [60, Fig. 28] as a case of mistaken recognitions.

Unlike text- or user-guided colorization, where conflicting conditions are less likely to arise during inference, reference images often introduce elements absent in the sketch images. For example, using character-centric images to colorize landscape sketches may lead to pronounced discrepancies, such as erroneously adding eyes and hair to non-human sketches. As illustrated in Figure 2, image-guided networks are likely to generate visually unpleasant results. More examples of the distribution problem are included in the appendix.

We delve into this problem at the latent distribution level, and the ideal distributions and sampling are visualized in Figure 3. For a given reference r_0 and sketch s_0, the sampled latent representation z_0 is achieved by projecting the embedding plane $p(z|y, r_0)$ onto $p(z|y, s_0)$, ensuring $z_0 \in p(z|y, s_0)$ for generating reasonable results. However, to obtain substantial and semantically well-paired training data, sketch and reference images are produced from the original color images, sharing the same ground truth struc-

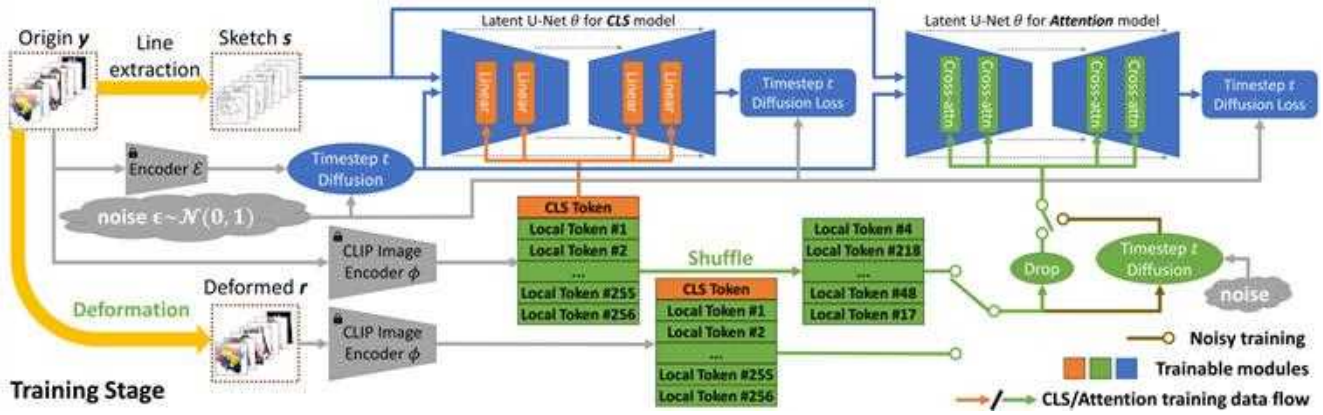

Figure 5. Our training pipelines. We propose two variations of reference-based models: *CLS* and *Attention*. For the *Attention* model, we introduce two training strategies, namely, deformation and shuffle training. Deformed images and sketch images are generated before training begins. Here, orange and green lines illustrate the respective data flows for the CLS model and the Attention model. While shuffle and deformation training are strategies specific to the *Attention* model, noisy training serves as a solution to the distribution problem, where the noisy training performs diffusion on the local tokens and is combined with either shuffle training or deformation training.

ture information at the latent level, so the ground truth color images can be easily reconstructed using either s or r, leading to an overlapped region $p(z|y,s,r)$ in the latent distributions $p(z|y,r)$ and $p(z|y,s)$. Thus, we can restate the distribution problem as follows: how can we train a dual-conditioned generative model to prioritize s over r using only one ground truth dataset, whose instances belong to $p(z|y,s)$ and $p(z|y,r)$, with $p(z|y,r) \subset p(z|y,s)$?

For simplicity, we collapse the visualization into 2D and illustrate the actual latent distributions in Figure 4. Because the information entropy of r is typically lower than that of s and noisy representations z_t during training, the denoised latent distribution $p_\theta(z|y,s,r)$ obtained during inference (marked by the black oval in Figure 4) usually tends to get closer to $p(z|y,r)$ rather than $p(z|y,s)$. This results in colorized outputs containing numerous identities specific to the reference image, leading to a suboptimal visual performance. In addition, the guidance scale of classifier-free guidance, which is widely used in denoising DM outputs, also has a strong influence on the distribution problem. To mitigate this issue, we employ dual classifier-free guidance, which adjusts the position of the sampled representation in the latent space.

3.3. Reference-based Training

Our reference-based models are fine-tuned from the Waifu Diffusion [14], and a pre-trained CLIP Vision Transformer (ViT) from OpenCLIP-H [3, 21, 39, 46] is used to extract image tokens from reference images and remains frozen during training. For a 224×224 image, the CLIP ViT outputs 257 tokens, comprising 256 local tokens and 1 CLS token. The CLS token encapsulates the semantic information of the reference image, whereas local tokens hold both structural and semantic content. We propose two reference-based models, *CLS* and *Attention*, differentiated by their token usage. The *CLS* model leverages only the CLS token, replacing all cross-attention modules in the latent U-Net with fully-connected layers, so it is a prompt-based model and less likely to suffer from the distribution problem; *Attention* models employ all local tokens for generation guidance, maintaining the architecture similar to SD v2.1 [41], the effectiveness of which in conditional generation has been demonstrated by various applications [43, 60].

Figure 5 illustrates our pipeline for reference-based fine-tuning, where we employ trainable convolutional layers in the denoising U-Net to downscale sketch inputs to the latent level. Following [60], these downscaled features are added to the forward features instead of being concatenated. Because the *CLS* token excludes structure information, the *CLS* model uses ground truth color images as reference inputs during training. However, the training of *Attention* models requires additional preprocessing for the reference inputs, so we accordingly propose two strategies to obtain the reference inputs and train the *Attention* model:

1. Deformation training: To tackle the data limitation, a common solution adopted by [30, 56, 61] is to generate reference images from ground truth color images using image deformation algorithms. In this paper, we utilize [45] to produce reference images before training.

2. Latent shuffle training: Generating reference images can be time-consuming and storage-intensive. Inspired by [8, 54], we propose latent shuffle training, which swaps the sequence of local tokens before inputting them into the U-Net, as shown in Figure 5. Note that the latent shuffle breaks the connection between neighbouring tokens, which strengthens the transfer ability of cross-attention modules

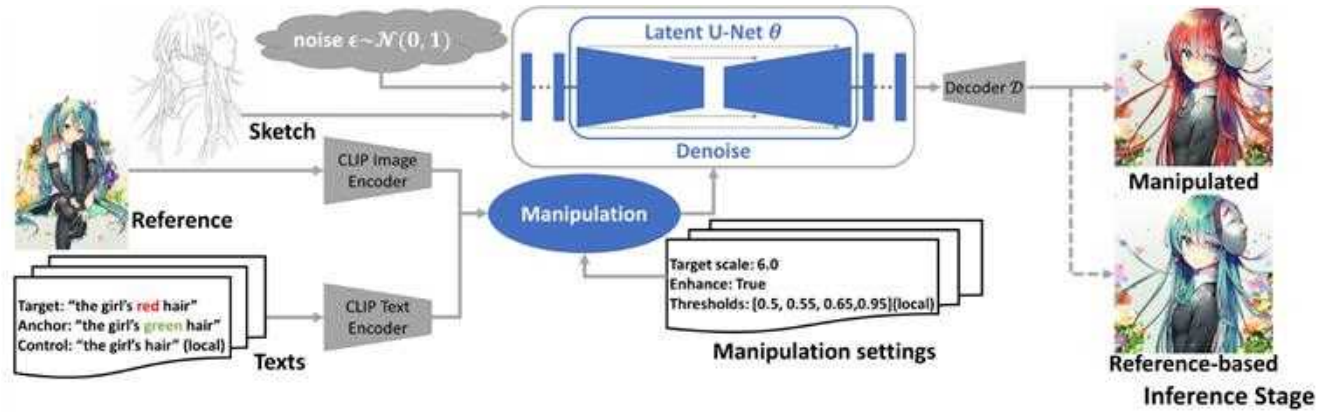

Figure 6. Our inference pipeline. The image tokens are edited before being input into the denoising U-Net. Illustrated results were generated by the *Attention* model at a resolution of 768×768.

but degrades the image composition of the generated results.

The diffusion loss for fine-tuning is defined as follows:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{E}(y),\epsilon,t,s,r}[\|\epsilon - \epsilon_\theta(z_t, t, s, \tau_\phi(r))\|_2^2], \quad (2)$$

where ϕ and τ_ϕ denote the CLIP ViT and extracted tokens, respectively. Compared to deformation-trained counterparts, shuffle-trained models can generate results with a more vivid texture, but they are more likely to suffer from the distribution problem. Therefore, most of our models were trained using latent shuffle to demonstrate the effectiveness of the proposed methods in mitigating the distribution problem, and we recommend adopting deformation-based training if possible.

3.4. Solutions to the Distribution Problem

To mitigate the distribution problem among *Attention* models, it is necessary to drag the denoised representations to $p(z|y,s)$, as explained in Section 3.2. To achieve this, we propose three solutions to designing the network aware of the distribution $p(z|y,s)$.

The first method, termed dropping training, randomly drops reference inputs during training with a drop rate much higher than 0.2, a suggested value in [18]. This slows down the optimization of cross-attention modules, enabling the network to generate $p_\theta(z_t|z_{t+1},s,t)$ (corresponding to $p(z|y,s)$ in Section 3.2). Default reference drop rates are empirically set to 0.75 for deformation training and 0.8 for shuffle training.

The second method is called noisy training, and it is identified by the brown switch in Figure 5. It performs diffusion on local tokens, dynamically increasing information entropy according to the timestep t. Therefore, its objective function is formulated as follows:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{E}(y),\epsilon,t,s,r}[\|\epsilon - \epsilon_\theta(z_t, t, s, \tau_{\phi,t}(r))\|_2^2], \quad (3)$$

where $\tau_{\phi,t}(r) = \alpha_t \tau_\phi(r) + \beta_t \epsilon_r$ and $\epsilon_r \sim \mathcal{N}(0,1)$. Compared with other solutions, this method eliminates the distribution problem but falls short of maintaining similarity with references, leading to a higher variance in the results.

Dropping reference conditions degrades the style transfer ability, becoming less effective after sufficient training. As illustrated in Figure 4, the $p_\theta(z|y,s,r)$ gradually shifts toward the $p(z|y,r)$ as the training duration increases. Therefore, we design dual-conditioned training to overcome this limitation.

The key goal of the dropping training is to enable the network to generate ϵ_t satisfying $z_t \in p_\theta(z_t|z_{t+1},s,t)$. In our proposed dual-conditioned diffusion training, we penalize the difference between the sketch-based results and the ground truth, rather than dropping references. The dual-conditioned loss is accordingly organized as follows:

$$\begin{aligned}\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{E}(y),\epsilon,\epsilon',t,s,r}[\|\epsilon - \epsilon_\theta(z_t, t, s, \tau_\phi(r))\|_2^2 + \\ \lambda\|\epsilon' - \epsilon'_\theta(z'_t, t, s)\|_2^2],\end{aligned} \quad (4)$$

where z_t and z'_t are diffused from z_0 using different noise ϵ and ϵ', respectively, and λ is set to 4 by default. Note that each batch travels the forward pass twice in the dual-conditioned training, so it takes approximately 1.6 times longer than other solutions. Models trained using the dropping, noisy, and dual-conditioned methods are referred to as the *Drop* model, *Noisy* model, and *Dual* model in the following sections, respectively. Qualitative results concerning the distribution problem are included in the appendix.

3.5. Global Text-Based Manipulation

Compared to T2I models, adjusting the semantic information of guiding conditions is more difficult for reference-based networks. We accordingly adopt a zero-shot and text-based manipulation for the proposed *CLS* model. DALL-E 2 [40] has demonstrated that an image-guided model using

Input: CLS token: $\vec{v}_{cls}$;
Normalized embeddings of target prompts: $\vec{e}[1..N]$;
Normalized embeddings of anchor prompts: $\vec{a}[1..N]$;
Target scales: $target\_scale[1..N]$;
Enhance flags: $enhance[1..N]$
for $i = 1, 2, .., N$ **do**
  if $\vec{a}[i]$ *is not null* **then**
    if $enhance[i]$ *is true* **then**
      $\vec{v}_{cls} \leftarrow \vec{v}_{cls} - (\vec{v}_{cls} \cdot \vec{a}[i]) * \vec{a}[i]$
      $\vec{v}_{cls} \leftarrow$
       $\vec{v}_{cls} + (target\_scale[i] - \vec{v}_{cls} \cdot \vec{e}[i]) * \vec{e}[i]$
    end
    else
      $\vec{v}_{cls} \leftarrow$
       $\vec{v}_{cls} + target\_scale[i] * (\vec{e}[i] - \vec{a}[i])$
    end
  end
  else
    if $enhance[i]$ *is true* **then**
      $\vec{v}_{cls} \leftarrow \vec{v}_{cls} + target\_scale[i] * \vec{e}[i]$
    end
    else
      $\vec{v}_{cls} \leftarrow$
       $\vec{v}_{cls} + (target\_scale[i] - \vec{v}_{cls} \cdot \vec{e}[i]) * \vec{e}[i]$
    end
  end
end
return $\vec{v}_{cls}$

Algorithm 1: Sequential global manipulation.

CLIP encoders can modify outputs based on text embedding. Therefore, we can adjust image embeddings to align with the target degree of visual attributes specified by texts before inputting them into the denoising U-Net θ. The inference pipeline is illustrated in Figure 6.

Our proposed method incorporates the normalized text embedding into the image embedding. We denote the extracted image tokens (previously represented as $\tau_\phi(r)$) and normalized text embeddings as vectors $\vec{v}$ and $\vec{e}$, respectively. The modified CLS token $\vec{v}^m_{cls}$ can be calculated as:

$$\vec{v}^m_{cls} = \begin{cases} \vec{v}_{cls} + target\_scale * \vec{e} & enhance \\ \vec{v}_{cls} + (target\_scale - \vec{v}_{cls} \cdot \vec{e}) * \vec{e} & not\ enhance \end{cases}, \tag{5}$$

where $target\_scale$ and $enhance$ are user-defined parameters. Similar to DALL-E-2, the manipulation can be improved through the normalized embedding of an anchor text, termed $\vec{a}$. The first method, where $enhance$ is false, calculates $\vec{v}^m_{cls}$ with the anchor text as:

$$\vec{v}^m_{cls} = \vec{v}_{cls} + target\_scale * (\vec{e} - \vec{a}). \tag{6}$$

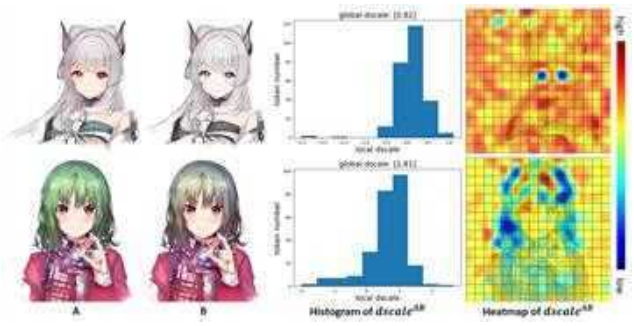

Figure 7. Visualization of $dscale^{AB}$ corresponding to the texts "the girl's red eyes" (upper) and "the girl's green hair" (lower), respectively. The first and second columns show the source images A and B used to generate image tokens, and the third column shows the histograms of $dscale^{AB}$, calculated between local tokens, with the $dscale^{AB}_{cls}$ (global $dscale$) shown at the top.

The global manipulation can be further enhanced by first eliminating the anchor attribute using $\vec{a}$ before adding $\vec{e}$. This step is taken when $enhance$ is activated and the CLS token $\vec{v}'_{cls}$ is then calculated as:

$$\begin{aligned} \vec{v}'^m_{cls} &= \vec{v}_{cls} - (\vec{v}_{cls} \cdot \vec{a}) * \vec{a}, \\ \vec{v}^m_{cls} &= \vec{v}'^m_{cls} + (target\_scale - \vec{v}'^m_{cls} \cdot \vec{e}) * \vec{e}. \end{aligned} \tag{7}$$

The sequential manipulation of $\vec{v}_{cls}$ is shown in Algorithm 1. Target scales ranging in $[4, 15]$ can generate reasonable results, and examples of how to perform the global zero-shot manipulation are included in Section 4.

3.6. Local Text-Based Manipulation

As *Attention* models employ local tokens as conditions, global manipulation becomes ineffective due to the absence of spatial information. Accordingly, we propose a semi-automatic algorithm for local tokens to accomplish manipulation. Importantly, to ensure the capability of accepting arbitrary text as input, the proposed local manipulation remains zero-shot.

In our observations, we noticed that the local tokens and the CLS token exhibit different directional changes when projected onto the text embedding. We define a scale called $dscale$, calculated as $dscale^{AB}_i = \vec{v}^A_i \cdot \vec{e} - \vec{v}^B_i \cdot \vec{e}$, where A and B represent the source images from which the image embeddings are extracted, and i denotes the index, with $i \in \{cls, 1, 2, .., n\}$ and n being the total number of local tokens. We find that for the given text, "a girl with green hair," as the hair becomes greener, the projection of the CLS token along the text embedding direction lengthens, which can be observed from the $dscale_{cls}$ value, which is illustrated in Figure 7 and labeled as $global\ dscale$ on top of the histograms. Conversely, the projections of the most relevant local tokens decrease, while those of irrelevant tokens increase. This can be observed from the heatmaps

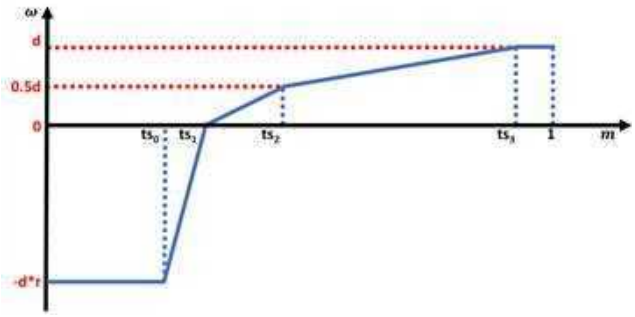

Figure 8. Plotting w_i as a function of m_i in Eq. 9. We divide the domain into five intervals to reduce the influence of the manipulation on unrelated attributes.

of $dscale^{AB}$, where the most related regions are marked in blue, indicating they have the lowest $dscale^{AB}$ values, which are negative according to the histograms. To locate the target region for manipulation, local manipulation additionally requires a control prompt whose embedding is denoted as $\vec{c}$. We calculate a Position Weight Matrix (PWM) $\boldsymbol{m}$ as:

$$\boldsymbol{m} = \mathcal{F}(\vec{\boldsymbol{v}} \cdot \vec{c}), \tag{8}$$

where $\mathcal{F}$ and $\vec{\boldsymbol{v}}$ indicate the min-max normalization and local tokens, respectively. By leveraging this PWM $\boldsymbol{m}$, we can simulate the change in the projection of a local token along the direction of the target text embedding. This is achieved using PWM $\boldsymbol{\omega}$, which is formulated as follows:

$$\omega_i = \begin{cases} -d * r, & m_i \leqslant ts_0 \\ -d + d * \frac{m_i - ts_0}{ts_1 - ts_0}, & ts_0 m_i \leqslant ts_1 \\ 0.5 * d * \frac{m_i - ts_1}{ts_2 - ts_1}, & ts_1 < m_i \leqslant ts_2 \\ 0.5 * d + 0.5 * d * \frac{m_i - ts_2}{ts_3 - ts_2}, & ts_2 < m_i \leqslant ts_3 \\ d, & m_i > ts_3 \end{cases} \tag{9}$$

where m_i and ω_i represent the i-th element of $\boldsymbol{m}$ and $\boldsymbol{\omega}$, respectively, with $i \in \{1, .., n\}$. We illustrate this function in Figure 8. In this equation, d is computed as:

$$d = \begin{cases} target\_scale - \vec{v}_{cls} \cdot \vec{a}, & enhance \\ target\_scale - \vec{v}_{cls} \cdot \vec{e}. & not\ enhance \end{cases}. \tag{10}$$

The hyperparameters r and ts_i in Eq. 9 denote the strength ratio for the most pertinent areas and the thresholds for differentiating all areas of the image, respectively. Here, $\boldsymbol{m} \leqslant ts_0$ selects the most relevant regions; $ts_0 < \boldsymbol{m} \leqslant ts_1$ identifies the positively correlated regions; $ts_1 < \boldsymbol{m} \leqslant ts_2$ covers negatively correlated regions; $ts_2 < \boldsymbol{m} \leqslant ts_3$ targets non-related regions; and $\boldsymbol{m} > ts_3$ randomly assigns the highest value to certain regions to emulate the *dscale* distribution. The default settings for the hyperparameters r

Input: Local tokens: $\vec{\boldsymbol{v}}$; CLS token: $\vec{v}_{cls}$;
Normalized embeddings of target prompts: $\vec{e}[1..N]$;
Normalized embeddings of anchor prompts: $\vec{a}[1..N]$;
Normalized embeddings of control prompts: $\vec{c}[1..N]$;
Target scales: $target\_scale[1..N]$;
Enhance flags: $enhance[1..N]$;
Thresholds list: $ts_{0,..,3}[1..N]$;
Strength factor: r;

for $i = 1, 2, .., N$ **do**
  if $\vec{a}[i]$ *is not null* **then**
    if $enhance[i]$ *is true* **then**
      $d \leftarrow target\_scale[i] - \vec{v}_{cls} \cdot \vec{a}[i]$
      $\boldsymbol{\beta} \leftarrow \mathbf{1}$
    end
    else
      $d \leftarrow target\_scale[i] - \vec{v}_{cls} \cdot \vec{e}[i]$
      $\boldsymbol{\beta} \leftarrow \mathbf{0}$
    end
    $\boldsymbol{m} \leftarrow \mathcal{F}(\vec{\boldsymbol{v}} \cdot \vec{c}[i])$
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}(\boldsymbol{m}, d, ts_{0,..3}[i], r)$ according to Eq 9
    $\vec{\boldsymbol{v}} \leftarrow \vec{\boldsymbol{v}} + (\boldsymbol{\omega} + \boldsymbol{\beta} * \vec{\boldsymbol{v}} \cdot \vec{a}) * (\vec{e}[i] - \vec{a}[i])$
  end
  else
    $d \leftarrow target\_scale[i]$
    $\boldsymbol{m} \leftarrow \mathcal{F}(\vec{\boldsymbol{v}} \cdot \vec{c}[i])$
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}(\boldsymbol{m}, d, ts_{0,..3}[i], r)$ according to Eq 9
    $\vec{\boldsymbol{v}} \leftarrow \vec{\boldsymbol{v}} + \boldsymbol{\omega} * \vec{e}[i]$
  end
end
return $\vec{\boldsymbol{v}}$

Algorithm 2: Sequential local manipulation, where $\mathcal{F}$ denotes min-max normalization.

and $[ts_0, ts_1, ts_2, ts_3]$ are 2 and $[0.5, 0.55, 0.65, 0.95]$, respectively. We set four thresholds to reduce the manipulation's influence on irrelevant visual attributes as much as possible. Experimentally, target visual attributes should be encompassed within the regions defined by $\boldsymbol{m} \leqslant ts_1$, while attributes intended for preservation should be within the $\boldsymbol{m} > ts_2$ region. Thereby, we can formulate the adjustment equation for the local tokens as:

$$\vec{\boldsymbol{v}}' = \vec{\boldsymbol{v}} + (\boldsymbol{\omega} + \boldsymbol{\beta} * \vec{\boldsymbol{v}} \cdot \vec{a}) * (\vec{e} - \vec{a}), \tag{11}$$

where $\boldsymbol{\beta}$ corresponds to the $enhance$ flag. If there is no anchor prompt, the equation is accordingly reorganized as:

$$\vec{\boldsymbol{v}}' = \vec{\boldsymbol{v}} + \boldsymbol{\omega} * \vec{e}. \tag{12}$$

Similarly, the calculation can be expanded to enable the sequential manipulation of multiple text pairs, as detailed in

Algorithm 2. Nevertheless, defining suitable thresholds for a control prompt can be challenging. To alleviate this difficulty, we have designed an interactive user interface that visually assists users in identifying the regions selected by each threshold.

4. Experiment

In this section, we first introduce the details of our implementation in Section 4.1, including the environment, data, and classifier-free guidance. We then experimentally compare the proposed models in ablation studies in Section 4.2 and compare them to baselines in Section 4.3. We present our text-based manipulation in Section 4.4, followed by the results of a corresponding user study in Section 4.5. Frechet Inception Distance (FID) [16, 47] estimates the distribution distance between generated images and real images, so it is used to evaluate the performance of generative models in this section.

4.1. Implementation Details

Training and Testing. We implemented our models using PyTorch and trained them on an NVIDIA DGX-Station A100 with 4x NVIDIA A100-SXM 40G. The *CLS* model was trained for seven epochs, and all the *Attention* models were trained for five epochs on the training set. The training of the *Shuffle-dual* model took 8 days, whereas the training of the other models took approximately 5 days using Distributed Data-Parallel Training (DDP) and AdamW optimizer [28, 33]. The training settings were as follows: learning_rate = 1e-5, batch_size_per_gpu = 10, betas = (0.9, 0.999), accumulative_batches = 2, weight_decay = 0.1. We adopted Stability-AI's official implementation of DPM++ solver, which is multi-step and second-order [34, 35], and our default sampling steps for testing were set to 20.

Dataset. We used Danbooru 2021 [6] as our original dataset to produce corresponding sketch and reference images. The sketch images were generated by jointly using SketchKeras [32] and Anime2Sketch [55], where as the total training set includes 4M+ triples of (sketch, reference, color) images, at a resolution of 512^2. All quantitative evaluations were taken on a subset of Danbooru 2021, including 48,000+ ground truth tags and (sketch, color) image pairs. Samples of the training data are included in the supplementary materials.

Dual Classifier-Free Guidance. Leveraging our dual-conditioned SD models, we can concurrently apply two forms of classifier-free guidance during inference. Both employ zero as the negative input. The guidance scales for reference-based and sketch-based guidance are denoted as GS and SGS, respectively, in subsequent sections.

Increasing the resolution for inference and applying Adaptive Instance Normalization (AdaIN) [20] as well as attention injection [15, 53, 65] could also enhance the quality of the generated images. Details can be found in the appendix.

4.2. Ablation study

Training Strategy and Architecture. We first evaluate two strategies introduced in Section 3.3. As shown in Table 1 and Figure 9, *Attention* models trained with different strategies achieved equivalent qualitative and quantitative results, demonstrating a better ability to transfer features than the *CLS* model. It can be observed that with higher guidance scales, the *Deform-0drop* model achieved lower FID scores compared to the *Shuffle-0drop* model, indicating that it performed better at avoiding the distribution problem. The *Dual* model achieved suboptimal FID scores compared to the other models, which we assume was due to the inappropriate λ value in Eq. 4. Though the *Noisy* model achieved the best score due to its effectiveness in eliminating the distribution problem, it outputs images with a higher variance, meaning more rounds are likely needed before obtaining results that correspond highly to the reference.

Guidance Scale and Drop Rate. We estimated the generation performance of ablation models under different GSs, as shown in Table 1. Most of our training followed the official implementation of SD and did not abandon conditions during training. The quantitative results indicate that adopting drop rates much higher than 0.2, which is suggested in [18], did not worsen the quality of generated images in reference-based colorization.

Training Epoch. The training duration also strongly influences the distribution problem, as indicated in Figure 4, where the distribution $p_\theta(z|y, s, r)$ gradually shifts to $p(z|y, r)$ as training continues. Qualitative evaluations regarding the training epoch are included in our appendix, which better reflect the distribution problem. Note that all attention-based models in our paper were trained for 5 epochs by default to balance style transfer and colorization.

4.3. Comparison to baseline

We compare our method to two baselines. The first baseline [56] effectively colorizes figure images but falls short when intricate backgrounds are involved, which is a common issue among GAN-based generative models. Our second baseline, *ControlNet* [36, 60], is an extension module introducing a conditional input for a pre-trained SD model. We adopted *Multi-ControlNet: lineart_lineart_anime + Reference* (simplified as *Multi-ControlNet*) for reference-based sketch colorization, and following the *ControlNet-lineart-*

Figure 9. Qualitative comparison to baselines and ablation models. Results (a)-(e) are real sketch images, with sketches (a),(b) by our human artist and (c)-(e) from [59]. The resolution of most results is 768^2, except for [56], which is 512^2.

anime guidelines, we employed *Anything v3*, a personalized model for anime style images [19, 58] as its SD backbone.

As *ControlNet* is based on T2I SD, we adopted ("masterpiece, best quality, ultra-detailed, illustration") as positive prompts and ("lowres, cropped, worst quality, low quality") as negative prompts in all its generations.

Qualitative Comparison. As Figure 9 illustrates, Yan et al.'s method struggles with colorization, especially when the reference includes diverse backgrounds. Although *Multi-ControlNet* demonstrates an impressive generation performance due to the well fine-tuned *Anything v3* [19], it falls short of maintaining color similarity with the references. For column (b) in Figure 9, due to the complex composition of the reference, we used a sketch guidance value of 1.5 and the attention injection to generate the results.

Because *Multi-ControlNet* incorporates three reference-based configurations, we generated *Multi-ControlNet*'s results using all three methods, showcasing the best ones in this paper. All results generated by other configurations, along with high-resolution images, are included in the supplementary materials.

Style Transfer. Both our models and *Multi-ControlNet* can facilitate reference-based style transfer when combined with a line extractor. As Figure 10 illustrates, our model transfers texture and color from the reference, while *Multi-ControlNet* manages to reconstruct the identities of the references in its generation domain. As our model is trained for high-fidelity sketch colorization, the quality of our results is decided by the sketch, occasionally yielding inferior segmentation compared to *Multi-ControlNet*. More examples and high-resolution images are provided in the supplementary materials.

Sketch Fidelity. Both our models and *Multi-ControlNet* can adjust the outputs' sketch fidelity according to respective hyperparameters, Sketch Guidance Scale (SGS) and control strength. We here qualitatively compare their differences in reference-based generation. As visualized in Figure 11, our guidance excels in maintaining color similarity with the original result (scale = 1) when increasing

Ablation study					
Model	GS-1	GS-2	GS-3	GS-5	GS-10
Deform-0	15.8590	10.8875	13.9459	20.7550	36.4256
Deform-0.75	17.4646	12.9854	11.7067	11.7067	15.5636
Shuffle-0	15.6971	10.3265	13.8398	22.1181	41.4941
Shuffle-0.8	15.2748	10.5986	9.1956	9.2383	12.0642
Noisy-0	15.5723	10.4629	9.0724	**8.9314**	11.5719
Dual-0	18.8059	13.6929	13.2995	14.7224	25.2262
CLS	13.5240	15.4600	19.9103	26.2609	41.8732
Baseline					
ControlNet, Text-based, GS-9					19.8511
†*ControlNet*, Text-based, GS-9, Shuffle					26.8437
‡*Multi-ControlNet-attn_AdaIN*, Reference-based, GS-9					22.2365
‡*Multi-ControlNet-attn_only*, Reference-based, GS-9					21.0125
‡*Multi-ControlNet-AdaIN_only*, Reference-based, GS-9					48.7509
[56]					26.1816

Table 1. FID scores achieved by various ablation models and baselines. Lower scores indicate better quality of generated images. {*Deform*,*Shuffle*, *Noisy*, *Dual*} and {0, 0.75, 0.8} indicate the corresponding training method and the reference drop rate applied to train the respective *Attention* models. Meanwhile, {GS-1, GS-2,GS-3,GS-5,GS-10} represent the respective guidance scales for each validation. The best score is highlighted in bold. †: Texts were randomly matched to unrelated sketch images during validation; ‡: Results were generated using the DPM++ 2M SDE Karras sampler rather than DPM++.

Figure 10. Comparison of style transfer outputs at 512^2. All results of *Multi-ControlNet* were generated using both attention injection and AdaIN, while ours were synthesized by the *Shuffle-0.8drop* model using either attention injection alone or both injection and AdaIN.

the scale.

Quantitative Comparison. Table 1 shows the FID scores of baselines. For reference-based baselines, color images were shuffled to colorize unrelated sketch images. As [56] is incapable of transferring complex features, it suffers less from the distribution problem and, thus, achieves a relatively lower FID. The gap between the two *ControlNet* results signifies the considerable effect of the distribution problem on text-based generation. Meanwhile, compared to the baselines, especially *Multi-ControlNet*, our models more effectively manage distribution issues while steadily generating superior colorized images.

Figure 11. Reference-based results generated using different sketch guidance weights and real sketch from our artist.

Figure 12. Global manipulation results using different target texts. Target scale and guidance scale were set to 10 and 3, respectively.

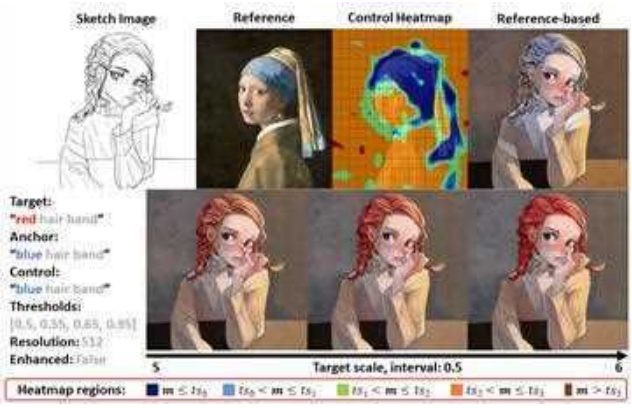

Figure 13. Local manipulation using the *Attention-Deform-0.75drop* model, with a guidance scale of 2, and a real sketch image. The stratified heatmap displays the regions selected by each threshold in Eq. 9.

4.4. Text-Based Manipulation

Global Manipulation. Two qualitative experiments were conducted to evaluate the controllability of the *CLS* model,

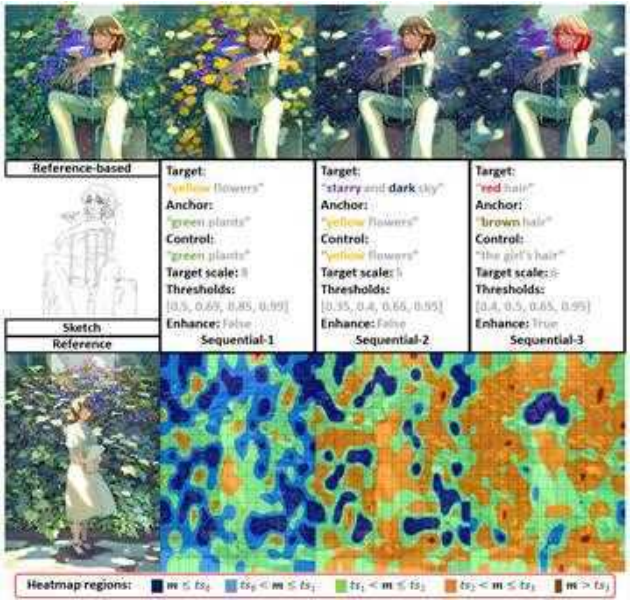

Figure 14. Illustration of the local manipulation. Stratified heatmaps corresponding to respective control texts in each step are listed under the input parameters. The results were generated at 768^2 using a real sketch, the *Attention-Deform-0drop* model, and guidance scale 2.

where Figure 1 shows the results of our sequential global manipulation, which also demonstrates the effectiveness of progressive change. We continue to show that global manipulation can also adjust highly abstract notions in Figure 12.

Local Manipulation. Unlike global manipulation, which relies solely on CLS token, local manipulation necessitates a PWM to adjust local tokens adaptively according to their association to the control text, leading to a more difficult manipulation. Figure 13 demonstrates that local manipulation can progressively adjust a specific visual attribute, while Figure 14 showcases sequential manipulation, altering backgrounds and hair color in sequential steps. Both figures adopt real sketch images.

Though our method effectively adjusts visual attributes, a significant challenge arises from the proposed local manipulation. Observing the heatmaps in Figure 14, which were generated from projections on the control text embedding, reveals substantial errors in segmentation, complicating the manipulation process.

4.5. User study

We implemented a user interface and invited 16 volunteers to experience our demo. Participants were required to test reference-based colorization and text-based manipulation for all proposed models. The average testing time for

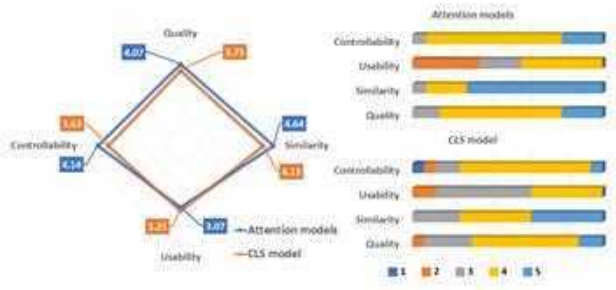

Figure 15. User study results. The radar charts show the average scores of four evaluations, and the bar chart showcases the distribution of user rating.

each individual exceeded 1 hour. After testing, we solicited participants' ratings across four dimensions:

Quality: Quality of generated images.

Similarity: Similarity with the reference image.

Usability: Ease of use.

Controllability: Correspondence between manipulated results and target texts.

The results, as shown in Figure 15, indicate overall satisfaction with image quality, control, and similarity. However, the relatively lower usability score demonstrates that the proposed manipulation requires further refinement to achieve simplicity. Materials used in our user study are included in the supplementary materials.

5. Conclusion

In this paper, we presented a thorough examination of the application of reference-based SD to sketch colorization, and we analyzed how the distribution problem leads to inferior outputs compared to text-based models. Leveraging a pre-trained CLIP, we proposed two strategies for training reference-based colorization SD and two kinds of zero-shot sequential manipulation methods. Our experimental results, including qualitative/quantitative evaluations and user studies, validate the effectiveness of our reference-based colorization and text-based manipulation methods. However, our work has four primary limitations:

1. Achieving precise segmentation based solely on the control text is challenging in the proposed local manipulation. In addition, adjustments without self-adaptive trainable modules struggle to replicate the real changes of tokens, especially for embeddings determined by multiple tokens, such as "daytime" and "night."

2. Because our manipulation is based on conditions rather than on forward representations, it is inevitable that some semantically unrelated visual attributes will be changed because they are colorized based on the manipulated regions in the reference. This can be observed in Figure 14, where the color of the right suitcase is changed.

3. It is challenging for the proposed models to achieve visually appealing results that are equivalent to well-personalized T2I models [19, 43].

4. OpenCLIP ViT takes images at 224×224 as inputs during training, a highly compressed resolution that hinders the learning of vivid strokes and detailed backgrounds, such as the lines and houses in *The Starry Night*.

Our future work will primarily focus on designing small-scale fine-tuning for T2I models [2, 44] to enable high-quality reference-based generation. In addition, we intend to identify a more suitable λ value for Eq. 4. The current value, which has not been thoroughly examined, resulted in suboptimal outcomes in ablation studies, as illustrated in Table 1. We also aim to enhance the controllability of local manipulation through three potential methods: 1) introducing a trainable module for adaptive PWM computation; 2) directly modifying features during the denoising process; and 3) designing advanced interactive systems to assist users in the selection of regions for local manipulation.

References

[1] Kenta Akita, Yuki Morimoto, and Reiji Tsuruno. Colorization of line drawings with empty pupils. *Comput. Graph. Forum*, 39(7):601–610, 2020. 3

[2] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *CVPR*, pages 18490–18500. IEEE/CVF, 2022. 13

[3] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 5

[4] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797. IEEE/CVF, 2018. 3

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8185–8194. IEEE/CVF, 2020. 3

[6] Danbooru community, Gwern Branwen, and Anonymous. Danbooru2021: A large-scale crowdsourced and tagged anime illustration dataset. https://gwern.net/danbooru2021, 2022. Accessed: DATE 2022-01-21. 2, 9

[7] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794, 2021. 3

[8] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883. IEEE/CVF, 2021. 5

[9] Sébastien Fourey, David Tschumperlé, and David Revoy. A fast and efficient semi-guided algorithm for flat coloring line-arts. In *Vision, Modeling and Visualization VMV*, pages 1–9. Eurographics Association, 2018. 3

[10] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri. Comicolorization: semi-automatic manga colorization. In *SIGGRAPH Asia*, pages 12:1–12:4. ACM, 2017. 3

[11] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Trans. Graph.*, 41(4):141:1–141:13, 2022. 2

[12] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423. IEEE/CVF, 2016. 3

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 3

[14] Reimu Hakurei. Hugging face/waifu-diffusion-v1-4. https://huggingface.co/hakurei/waifu-diffusion-v1-4, 2023. Accessed: DATE 2023-03-05. 5

[15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*. OpenReview.net, 2023. 9

[16] 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. In *NeurIPS*, pages 6626–6637, 2017. 9

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. 3, 6, 9

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022. 2, 3, 10, 13

[20] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519. IEEE/CVF, 2017. 3, 9

[21] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 5

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976. IEEE/CVF, 2017. 3

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, volume 9906, pages 694–711. Springer, 2016. 3

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410. IEEE/CVF, 2019. 3

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8107–8116. IEEE/CVF, 2020. 3

[26] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, pages 2416–2425. IEEE/CVF, 2022. 2

[27] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In *ICCV*, pages 9055–9064. IEEE/CVF, 2019. 3

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 9

[29] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 4

[30] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *CVPR*, pages 5800–5809. IEEE/CVF, 2020. 3, 5

[31] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *WACV*, pages 289–299. IEEE/CVF, 2023. 2

[32] lllyasviel. Sketchkeras. https://github.com/lllyasviel/sketchKeras, 2017. 9

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*. OpenReview.net, 2019. 9

[34] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022. 3, 9

[35] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *CoRR*, abs/2211.01095, 2022. 3, 9

[36] Lyumin Zhang Mikubill. sd-webui-controlnet. https://github.com/Mikubill/sd-webui-controlnet, 2023. Accessed: DATE 2023-07-01. 9, 16

[37] Amal Dev Parakkat, Pooran Memari, and Marie-Paule Cani. Delaunay painting: Perceptual image colouring from raster contours with gaps. *Computer Graphics Forum*, 41(6):166–181, 2022. 3

[38] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, pages 2065–2074. IEEE/CVF, 2021. 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763. PMLR, 2021. 5

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. 2, 6

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE/CVF, 2022. 2, 3, 5

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, volume 9351, pages 234–241. Springer, 2015. 3

[43] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, pages 22500–22510. IEEE/CVF, 2023. 2, 3, 5, 13

[44] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. *CoRR*, abs/2307.06949, 2023. 13

[45] Scott Schaefer, Travis McPhail, and Joe D. Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25(3):533–540, 2006. 5

[46] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 5

[47] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2023. Accessed: DATE 2023-05-17. 9

[48] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, volume 37, pages 2256–2265. JMLR.org, 2015. 3

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021. 3

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*. OpenReview.net, 2021. 3

[51] Tsai-Ho Sun, Chien-Hsun Lai, Sai-Keung Wong, and Yu-Shuen Wang. Adversarial colorization of icons based on contour and color conditions. In *ACM MM*, pages 683–691. ACM, 2019. 3

[52] Daniel Sýkora, John Dingliana, and Steven Collins. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Comput. Graph. Forum*, 28(2):599–608, 2009. 3

[53] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930. IEEE/CVF, 2023. 9

[54] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, pages 6306–6315, 2017. 5

[55] Xiaoyu Xiang, Ding Liu, Xiao Yang, Yiheng Zhu, Xiaohui Shen, and Jan P. Allebach. Adversarial open domain adaptation for sketch-to-photo synthesis. In *WACV*, pages 944–954. IEEE/CVF, 2022. 9, 17

[56] Dingkun Yan, Ryogo Ito, Ryo Moriai, and Suguru Saito. Two-step training: Adjustable sketch colourization via reference image and text tag. *Computer Graphics Forum*, 2023. 3, 5, 9, 10, 11

[57] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *CoRR*, abs/2308.06721, 2023. 2, 3

[58] Yuno779. https://civitai.com/models/9409, 2023. Accessed: DATE 2023-06-25. 10

[59] Lvmin Zhang. Style2paints v5, 2023. Accessed: DATE 2023-06-25. 10

[60] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *CoRR*, abs/2302.05543, 2023. 2, 3, 4, 5, 9

[61] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. Two-stage sketch colorization. *ACM Trans. Graph.*, 37(6):261, 2018. 3, 5

[62] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 3

[63] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, volume 9907, pages 649–666. Springer, 2016. 3

[64] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph.*, 36(4):119:1–119:11, 2017. 3

[65] Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. Real-world image variation by aligning diffusion inversion chain. *CoRR*, abs/2305.18729, 2023. 9

[66] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251. IEEE/CVF, 2017. 3

[67] Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu. Language-based colorization of scene sketches. *ACM Trans. Graph.*, 38(6), 2019. 3

A. Improvement on Generation

We introduce several important suggestions that can further improve the generation performance.

Resolution. Increasing the image resolution significantly improves reference-based sketch colorization. Sketch images in higher resolution provide detailed strokes and richer semantic information. Experimentally, optimal inference results often manifest at 1.5x the training resolution, e.g., training at 512^2 and inferring at 768^2. Real color images created by experienced artists contain numerous visual attributes that are difficult to transfer fully. However, reference-based models always manage to generate all these attributes in the sketch image, leading to overly saturated colors. Utilizing a larger resolution during inference can effectively moderate these reference features, yielding more appealing results.

As CLIP-ViT adopts inputs at a resolution of 224^2 and a fixed-length positional embedding, we must pre-process the inputs when generating images in a higher resolution. Given R_{tr} and R_{inf} as training and inference resolutions, respectively, and $f = \frac{R_{inf}}{R_{tr}}$, the reference image and positional embedding are interpolated to $(224f, 224f)$ and $256f^2 + 1$, respectively. Here, $R_{tr} = 512$ for all proposed models, as they were trained with 512^2 images.

Attention injection and AdaIN. Our implementation of attention injection and AdaIN is similar to *Multi-ControlNet* [36], and both techniques could be adopted to improve our generated results. We briefly introduce here how the attention injection is adapted to our reference-based colorization models. As illustrated in Figure 1, we utilize a sketch extracted from the reference image as the sketch input for the inversion $\boldsymbol{x}^R$ chain. Given the intermediate hidden states $\boldsymbol{h}^R$ obtained from the x^R chain, and $\boldsymbol{h}^G$ from the generation $\boldsymbol{x}^G$ chain, we concatenate them as $\boldsymbol{h}_c^G$ for computing K and V in self-attention modules, calculated as:

$$Q = W_q \cdot \boldsymbol{h}^G,\ K = W_k \cdot \boldsymbol{h}_c^G,\ V = W_v \cdot \boldsymbol{h}_c^G, where \\ \boldsymbol{h}_c^G = \boldsymbol{h}^R \oplus \boldsymbol{h}^G \tag{13}$$

where, W_q, W_k and W_v denote the weight matrix for Q, K and V, respectively.

B. Further discussion of Distribution problem

The training period has a strong influence on colorization and style transfer due to the distribution problem. We generated several results of colorization using different *shuffle*-based models, as shown in Figure 2 and Figure 3. It can be observed that the *Shuffle-0drop* model erroneously generates eyes in the right part of the results, starting from epoch 3, and the *drop* model synthesizes hair in the left part of the results starting at epoch 5; only the *Noisy* model can colorize the sketch reasonably all the time.

Table 1. Rates of the different training methods. The term "distribution" indicates the ability to mitigate the distribution problem. Note that deformation training must generate reference images, and dual-conditioned training takes 1.6 times longer than other solutions. As the variance in the *Noisy* model is higher than that of other methods, we rate it based on its best and worst results, respectively. †: Based on the best results. ‡: Based on the worst results.

	Distribution	Colorization	Style transfer
Training strategy			
Deformation	medium	good	medium
Shuffle	very bad	medium	good
Solution to the distribution problem			
Drop	good	good	medium
†*Noisy*	very good	very good	medium
‡*Noisy*	very good	medium	bad
Dual	medium	medium	good
0 drop	bad	bad	good

Two examples are shown in Figure 4 to demonstrate the differences between the four shuffle-based models. Compared to low-level fidelity sketch colorization, high-fidelity models struggle to recover missing parts reasonably if they suffer from the distribution problem. As visualized in the (a) part in Figure 4, both the *Drop* model and the *Noisy* model successfully recover the eyes with vivid strokes and texture without sketch-based guidance, while the *Dual* model and the *0drop* model fail. Corner cases are also presented in the same figure, where the SGS is set to 5. It can be observed that the *0drop* model still cannot generate clear edges of the eyes. A typical example of using sketch guidance to mitigate the distribution problem is provided in the (b) part, where the *0drop* model continues to fail at removing the mistakenly generated eyes.

In addition, several style transfer results are shown in Figure 5 to demonstrate the necessity of longer training. Combined with Figure 2 and Figure 3, we find that both *0drop* and *0.8drop* models are unable to generate vivid texture until the epoch at which they begin to encounter the distribution problem, which occured at epoch 3 and epoch 5, respectively. However, as the *0.8drop* model can mitigate the distribution problem with sketch guidance and resampling, it is more likely to generate better results than the *0drop* model.

Summary. As many different training methods have been introduced in this paper, we rate their overall performance based on applications, shown in Table 1. Here, the evaluation of colorization stresses color, while that of style trans-

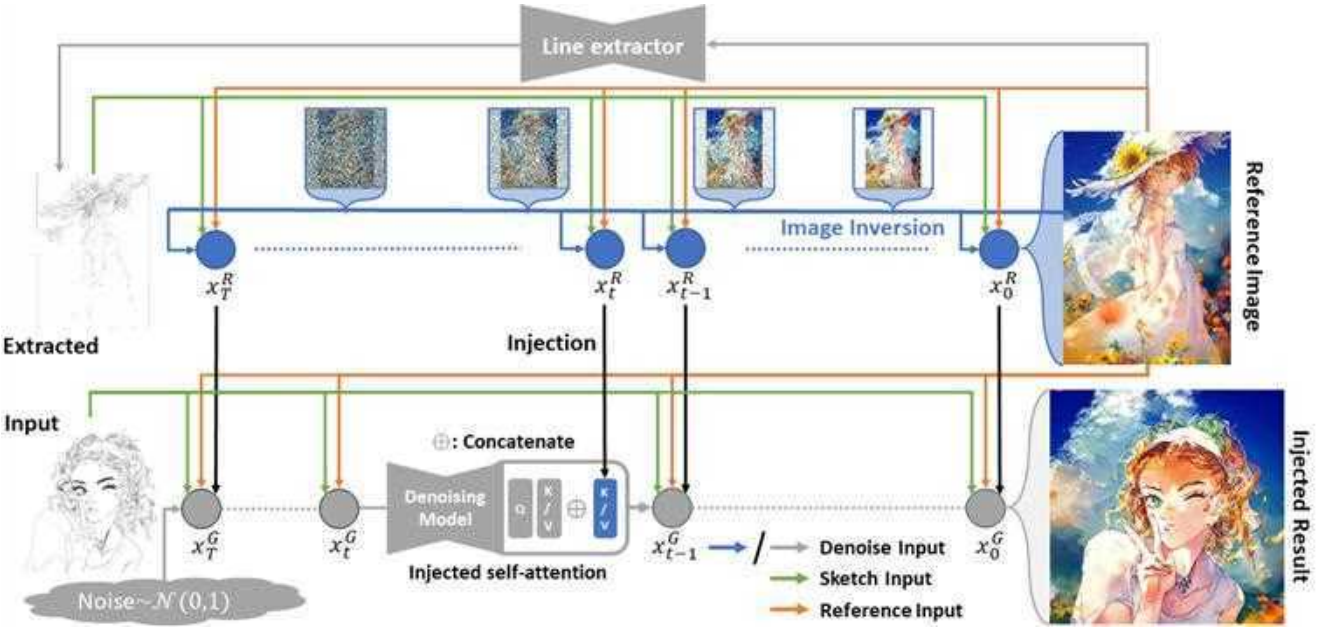

Figure 1. Illustration of our attention injection. The injected result is generated by the *shuffle-0.8drop* model. We adopt [55] as our default line extractor.

fer relies on texture and stroke. We suggest adopting noisy training if a higher variance in the generated results is acceptable.

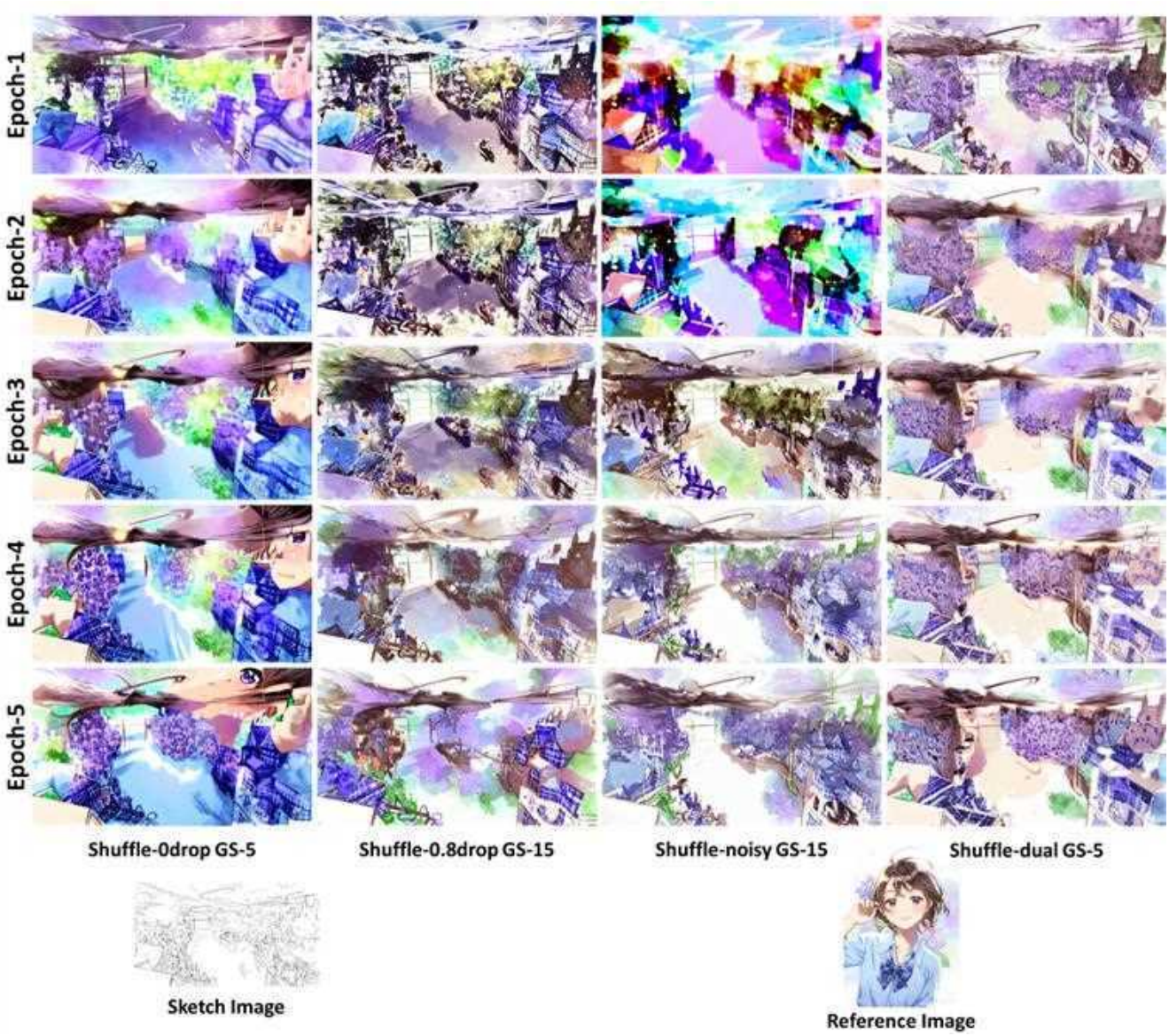

Figure 2. To better observe the distribution problem, we used extremely high reference guidance scales to generate the colorized results.

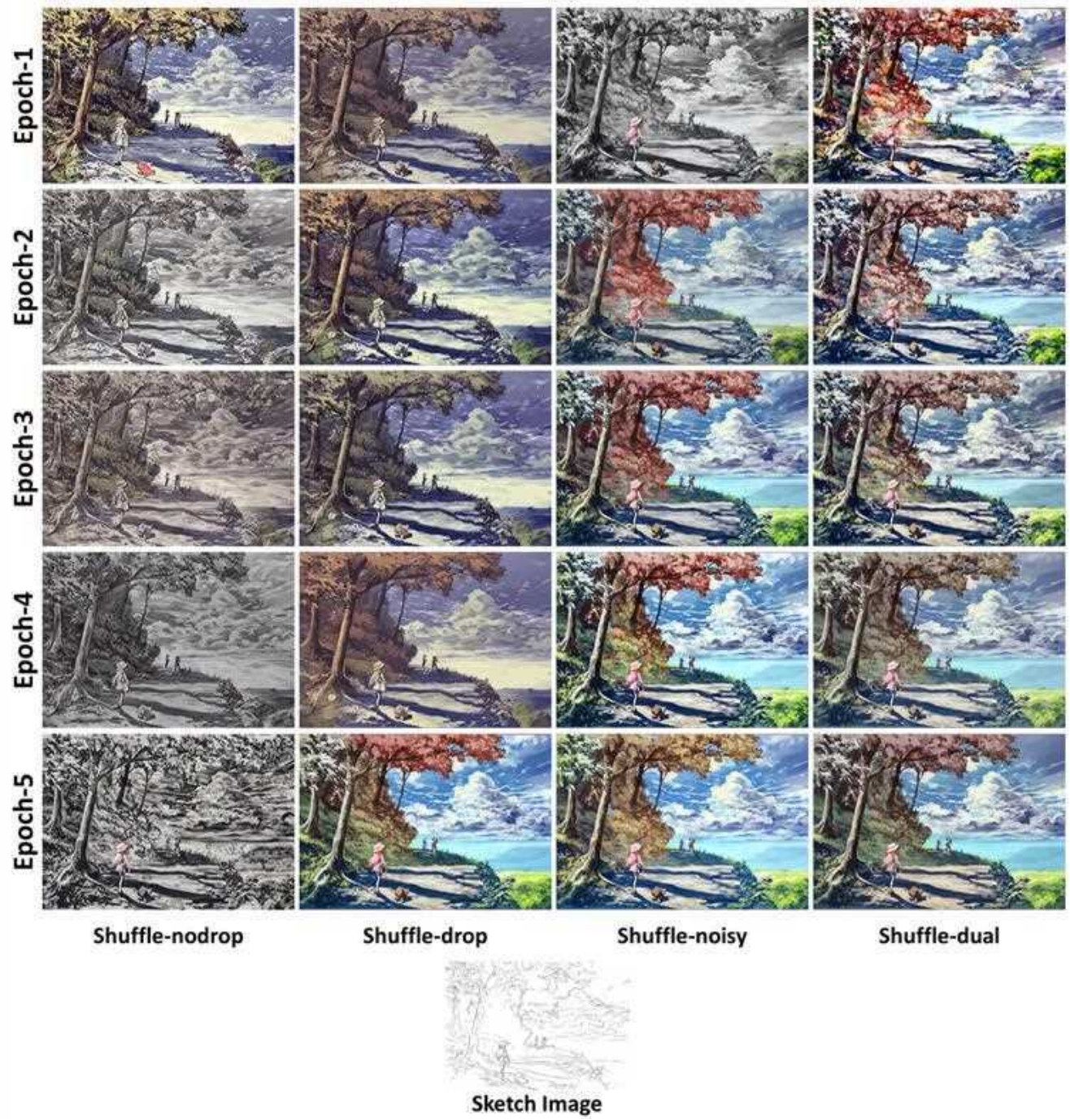

Figure 3. Colorization results without reference inputs, where we can find that the *0drop* model fails to generate texture that is faithful to the sketch as the training continues. We use a sketch guidance scale of 1.3 and the same seed for all models in this test.

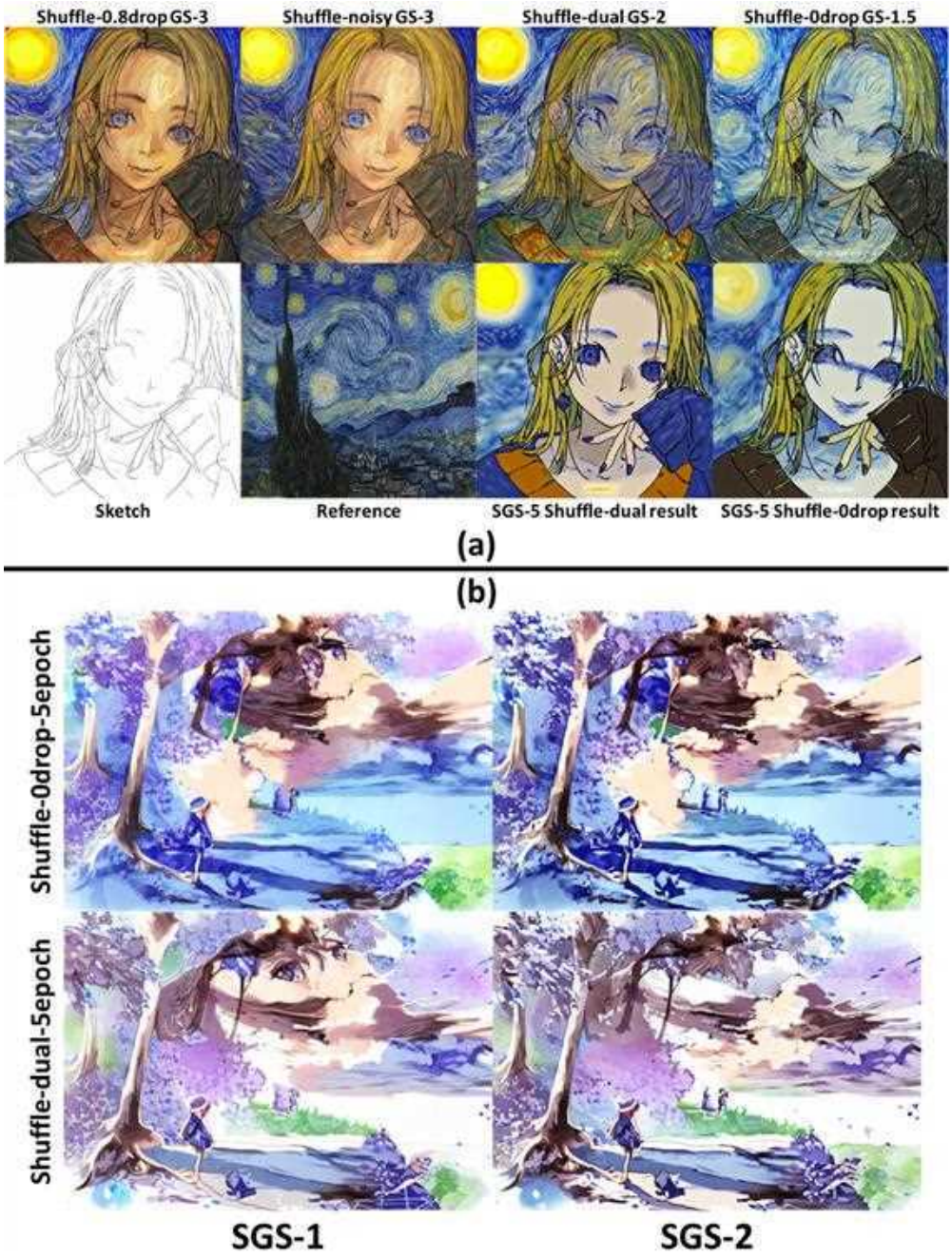

Figure 4. (a) Style transfer results for observing the distribution problem. High-fidelity models that suffer from the distribution problem struggle to reconstruct the missing parts of the sketch; (b) Using sketch guidance to mitigate the distribution problem. The *Dual* model removes the human eyes with sketch guidance while the *0drop* model fails to do so.

.

Figure 5. Sketch-based style transfer results. We generated the images with and without attention injection, respectively.