論文 / 著書情報
Article / Book Information

題目(和文)	参照画像とテキストを用いたスケッチの自動着色法
Title(English)	Automatic Sketch Colorization using Reference Image and Text
著者(和文)	Dingkun Yan
Author(English)	Dingkun Yan
出典(和文)	学位:博士(工学), 学位授与機関:東京科学大学, 報告番号:甲第477号, 授与年月日:2025年9月22日, 学位の種別:課程博士, 審査員:齋藤 豪,小池 英樹,篠田 浩一,下坂 正倫,井上 中順
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Institute of Science Tokyo, Report number:甲第477号, Conferred date:2025/9/22, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctor's Thesis

Automatic Sketch Colorization using Reference Image and Text

Dingkun Yan

Graduate Major in Artificial Intelligence
School of Computing
Tokyo Institute of Technology

Supervisor: Suguru Saito

June, 2025

Abstract

Anime-style imagery enjoys global acclaim for its vibrant palettes and expressive character designs, yet transforming monochrome sketches into fully colored frames remains a painstaking, time-intensive bottleneck in animation production. Early semi-automatic tools eased this workflow by propagating user-supplied color dabs, and subsequent machine-learning variants improved visual fidelity, but they still falter when sketches lack explicit texture cues. The emergence of deep generative networks has reshaped the landscape, enabling end-to-end systems that synthesize rich hues and fine-grained textures with minimal human intervention. Recent approaches fall into three classes based on how color hints are given to the deep networks: user-guided (color-spot input), text-conditioned, and reference-based. Although user-guided pipelines offer precise local control, they require artists to place strokes in anatomically consistent regions; text-driven models scale effortlessly but struggle to encode the nuanced style and color patterns typical of line art. In contrast, reference-based colorization balances automation and fidelity by drawing chromatic and textural guidance directly from exemplar images, making it the most practical choice for industrial production lines.

Unlike user-guided and text-conditioned pipelines, deep-learning reference-based colorizers face an acute form of over-fitting. They are typically trained on tightly aligned sketch–reference pairs, yet at inference, the system must transfer color between sketches and references that differ substantially in pose, composition, and semantic content. This distribution mismatch causes the network to memorize pair-specific correspondences, leading to washed-out hues, misplaced textures, or outright failure when conventional training recipes are applied. Production-quality performance, therefore, demands specialized architectures and optimization schedules that actively regularize against pair-wise memorization. This dissertation systematically dissects the phenomenon, tracing its empirical symptoms, uncovering why it departs from textbook over-fitting in vision tasks. A series of novel frameworks and strategies are proposed to suppress its adverse effects while retaining the automation and fidelity that make reference-based colorization attractive for industrial use.

Keywords: Deep learning, Sketch colorization, Style transfer

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Anime-style images have long stood as a powerful medium of artistic expression, captivating global audiences across generations with its unique ability to bring characters and stories to life. Despite advancements in digital tools, modern production of fine-grained illustration remains highly labor-intensive, requiring extensive human effort across multiple stages of the creative pipeline. Sketch colorization, the pivotal transformation of monochrome line art into richly rendered, emotionally resonant images, still consumes a disproportionate share of that effort. Because such illustrations serve as flagship visuals for anime-related works, including character designs, promotional posters, manga pages, and light-novel covers, an automatic colorization tool could dramatically accelerate production. This persistent bottleneck has therefore spurred growing interest in machine-learning techniques that automate and enhance colorization workflows. In particular, data-driven sketch-colorization models promise to reduce manual labor, boost efficiency, and, crucially, preserve the creative intent of artists.

Existing automatic sketch colorization methods still fall short of producing satisfactory results in real-world applications. Broadly speaking, current approaches can be categorized into two major groups based on their underlying techniques: traditional methods that rely on handcrafted image processing algorithms, and data-driven approaches powered by deep learning (DL). Traditional methods often require significant user interaction through step-by-step guidance and are typically constrained to narrow, case-specific scenarios, limiting their scalability and adaptability. Consequently, this thesis places its primary focus on deep learning-based methods, which hold greater promise for automation and generalization.

Based on the type of guidance provided for conveying color and semantic information, existing deep learning-based sketch colorization methods can be broadly categorized into three groups: user-guided, text-guided, and reference-based approaches. User-guided methods rely on manual inputs such as color dots or brush strokes applied to specific regions of the sketch, instructing the model where and how to apply color. While intuitive, these methods demand frequent user interaction and offer limited control over stylistic nuances such as stroke patterns or color consistency. Text-guided methods, on the other hand, utilize natural language prompts to guide the colorization process. Although these approaches offer a high-level, language-based interface, they struggle to accurately convey fine-grained visual details and often produce ambiguous or inconsistent results. In contrast, reference-based methods take a single color image as a style exemplar, allowing the

model to automatically extract color and stylistic information and apply it to the target sketch. This paradigm doesn't need users to give specific hints for each input, while ensuring greater consistency and fidelity in the output. Due to their automation potential and superior alignment with existing production workflows, reference-based methods are particularly well-suited for industrial applications.

Nevertheless, despite their recent progress, most DL-based reference-based methods show significant deterioration compared to their text-guided baselines and remain unsuitable for industrial deployment due to two key limitations: suboptimal visual quality of the generated outputs and limited generalization across diverse artistic styles of input pairs, character designs, or guiding conditions. Addressing these challenges is critical to bridging the gap between academic research and practical application in large-scale anime illustration production.

These limitations largely stem from a pervasive yet stubborn problem in image-guided learning—overfitting. Data-driven methods typically require the training data to be well aligned both semantically and spatially. This alignment requirement easily results in the reference-based sketch colorization methods overfitting their training data, and the extent of that overfitting depends heavily on how the training data are curated. In practice, sketches are automatically extracted from their ground-truth color images, while references are curated in one of two ways: (i) using the same ground-truth images or (ii) using near-duplicate frames of the same subject or scene (e.g., adjacent video frames or consecutive manga panels). This yields a training set of spatially aligned and semantically matched (sketch, reference, color target) triples. Networks trained on such data with pixel-level losses and jointly optimized sketch and reference encoders achieve excellent in-distribution performance; however, they are very likely to break down when the sketch and reference are even slightly misaligned. Unfortunately, in real-world applications, most sketch–reference pairs are highly mismatched in both semantic and spatial composition, causing the colorization networks to deteriorate severely due to overfitting.

To counter this, this thesis raises the level of correspondence from pixels (or latent tokens) to semantics by adopting a powerful pre-trained image encoder as the frozen reference branch, forcing the sketch pathway to match reference representations in a stable, high-level embedding space rather than memorize low-level spatial coincidences. Building on this principle, we develop a colorization framework that generalizes across diverse anime-style sketches and heterogeneous reference inputs, remaining robust under significant semantic and spatial mismatches.

The core contribution of this thesis is to address the critical overfitting issue inside reference-based sketch colorization, investigating the underlying causes of overfitting and introducing two series of colorization frameworks. Each series leverages distinct network architectures specifically designed to reduce overfitting and improve robustness across diverse input conditions. These approaches not only significantly enhance the generalization capability of the models, ensuring consistently high-quality colorization results, but also prioritize practical usability. Recognizing the frequent user requirement for additional refinement or customization of automatically generated colorizations, this thesis further proposes innovative text-based latent manipulation algorithms tailored explicitly for each reference-based framework. These algorithms empower users to perform intuitive, detailed adjustments through textual instructions, thereby expanding the functionality of

the proposed frameworks from strictly reference-guided sketch colorization to versatile, text-driven editing applications.

Chapter 2

Related work

2.1 Deep learning

2.1.1 Multilayer perceptrons

Multilayer Perceptrons (MLPs) are the earliest and most general-purpose neural-network family: a stack of affine projections followed by nonlinearities that, thanks to the universal approximation theorem, can model any Borel-measurable function to arbitrary precision given enough hidden units, illustrated in Figure 2.1. Although today's state-of-the-art models add convolutions, attention, or diffusion objectives, every one of those innovations ultimately reduces to MLP blocks under the hood, and their training dynamics, capacity, and failures can all be traced back to the properties of the classic dense layer plus activation described below. A standard MLP is composed of multiple layers of linear transformations—commonly referred to as fully connected or dense layers—interleaved with non-linear activation functions. These two components together enable the network to approximate complex, non-linear functions that cannot be captured by linear mappings alone. A canonical L-layer MLP can be formulated as

$$\begin{aligned} z_\ell &= \boldsymbol{W}_\ell \boldsymbol{h}_\ell + \boldsymbol{b}_\ell, \\ \boldsymbol{h}_\ell &= \phi(\boldsymbol{z}_\ell), \qquad \ell = 1, ..., L-1, \\ \boldsymbol{y} &= \boldsymbol{W}_L \boldsymbol{h}_{L-1} + \boldsymbol{b}_L, \end{aligned} \tag{2.1}$$

where ϕ is a nonlinear activation and $\boldsymbol{W}_\ell$,$\boldsymbol{b}_\ell$ are trainable parameters at ℓ-th layer. The learnability follows directly from back-propagation.

Normalization Layers. A basic MLP unit typically consists of a normalization layer (e.g., batch normalization or layer normalization) to stabilize and standardize the input distribution, followed by a linear projection layer to transform the feature space, and a non-linear activation function to introduce model expressiveness. This normalization layer is generally formulated as $\hat{\boldsymbol{z}} = \frac{\boldsymbol{z}_\ell - \boldsymbol{\mu}_\ell}{\boldsymbol{\sigma}_\ell}$ where $\boldsymbol{\mu}_\ell, \boldsymbol{\sigma}_\ell$ are batch/per-sample/per-instance statics in Batch Normalization [36]/Layer Normalization [4]/Instance Normalization [88], respectively. This modular structure forms the basis of more sophisticated computational blocks in modern neural networks, including transformer feed-forward layers, residual MLPs, and attention-based architectures.

Activation functions. The most widely used activation functions in MLPs include

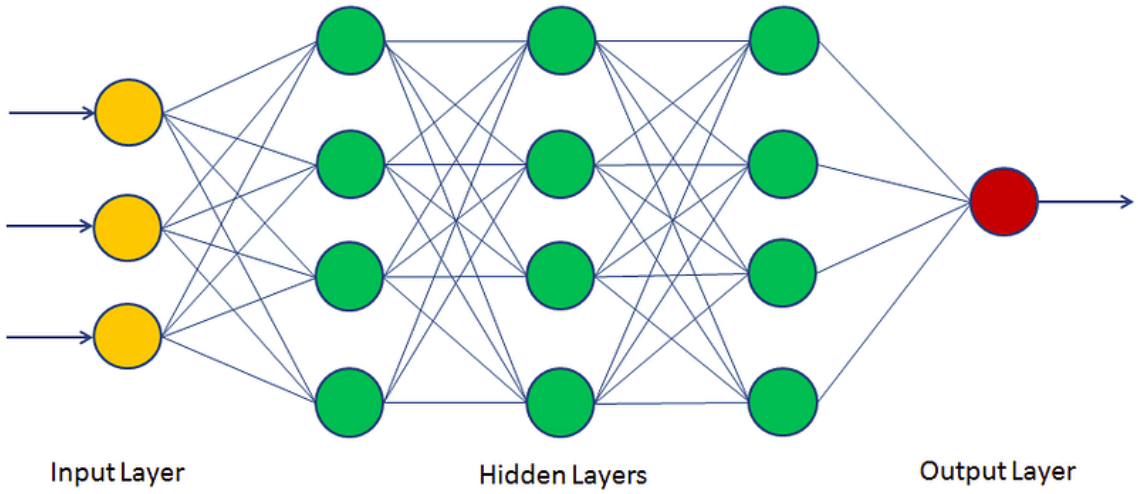


Figure 2.1: A basic visualization of Multilayer perceptron [61].

the ReLU (Rectified Linear Unit) family (e.g., ReLU [19], Leaky ReLU [96], ELU [10], GELU [26] in Figure 2.2 and in Table 2.1, as well as classical functions such as tanh and sigmoid in Figure. Each activation function introduces distinct nonlinear characteristics, affecting the model's convergence behavior, representational capacity, and gradient flow. For instance, ReLU and its variants are particularly effective in mitigating the vanishing gradient problem in deep architectures, while tanh and sigmoid remain useful in shallow networks or specific modules such as recurrent units.

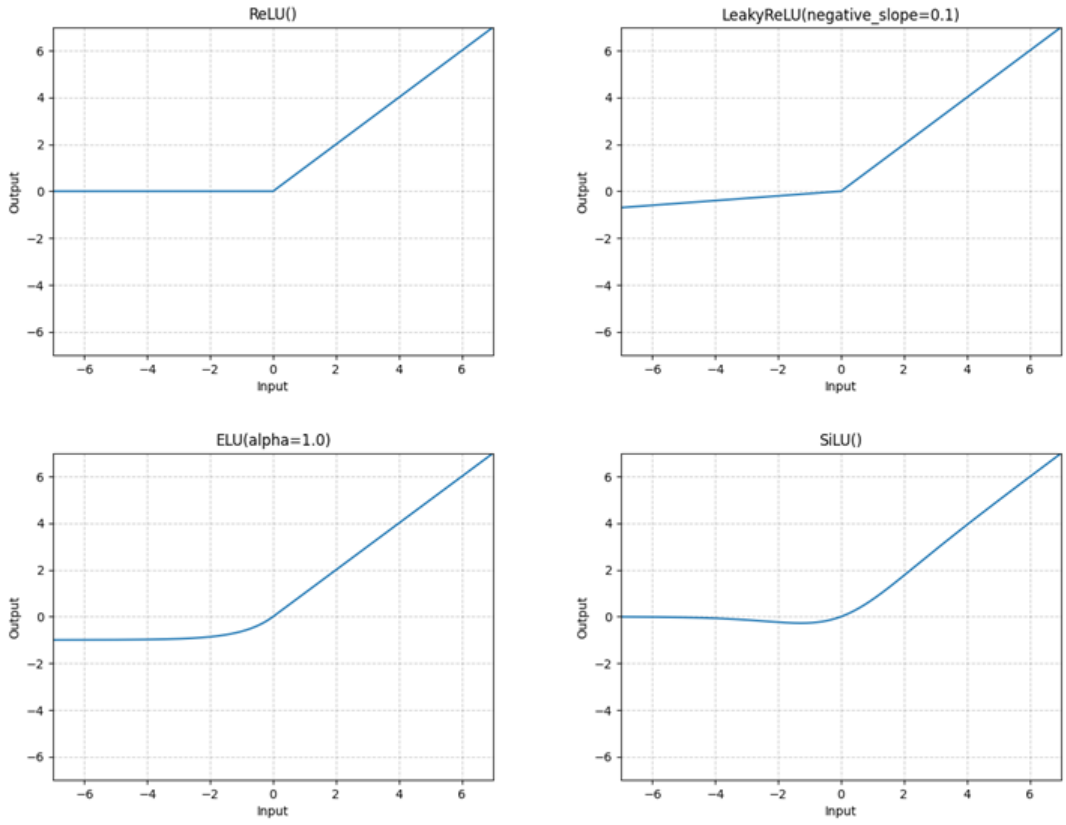


Figure 2.2: Most widely-used activation functions [19].

2.1.2 Backpropagation

Backpropagation—short for *backward propagation* of errors—is the learning algorithm that makes modern deep neural networks trainable. Originally formalized for MLP by Rumelhart, Hinton, and Williams [75] and derived independently by Werbos [91], backpropagation provides an efficient way to compute gradients of any scalar loss with respect to all model parameters by systematically applying the chain rule in reverse order. The algorithm's efficiency, together with the rise of GPU acceleration and automatic-differentiation software, transformed neural networks from academic curiosities into practical engines for computer vision, natural language processing, reinforcement learning, and, most recently, the diffusion and GAN architectures investigated later in this thesis. Historical background. Early adaptive-filter research introduced the delta rule for single-layer networks [92]. Control theory contributed the adjoint-state method [66] for computing sensitivities in dynamical systems. The breakthrough came when researchers extended these gradient ideas to multilayer, non-linear networks and showed that gradient descent remains tractable if intermediate activations are cached during a forward pass and reused during a single backward pass. LeCun's 1989 handwriting-recognition work [48] further demonstrated backpropagation's viability on real-world datasets and unevenly distributed pattern statistics. Algorithmic outline. Consider a network that comprises L differentiable layers:

$$y = f_L(\theta_\ell, f_{\ell-1}(\theta_{\ell-1}, ..., f_1(\theta_1, x))), \tag{2.2}$$

Table 2.1: Frequently-used activation functions.

Activation	**Formula**	**Key Properties**
Tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	Saturating; prone to vanishing gradients
Sigmoid	$\frac{1}{1+e^{-x}}$	Saturating; prone to vanishing gradients
ReLU	$\max(0, x)$	Alleviates vanishing gradients
Leaky ReLU	$\max(\alpha \cdot x, x)$	Retains gradient for $x < 0$
ELU	$\begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}$	Faster convergence than ReLU on some tasks
GELU	$x \cdot \Phi(x)$	Smooth stochastic gating
GELU (tanh)	$0.5 \cdot x \cdot \left(1 + \tanh\left(\sqrt{\frac{2}{\pi}} \cdot (x + 0.044715 \cdot x^3)\right)\right)$	Smooth stochastic gating

where θ_ℓ denotes the parameters of layer ℓ. For a scalar loss $\mathcal{L}(y, t)$ with target t, the goal is to compute the gradient $\nabla_{\theta_\ell}\mathcal{L}$ for every layer. Given the chain rule, which can be simplified as

$$\nabla_{\theta_\ell}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_\ell}\frac{\partial \boldsymbol{a}_\ell}{\partial \theta_\ell}, \tag{2.3}$$

where $\boldsymbol{a}_\ell$ is the pre-activation at layer ℓ. Backpropagation exploits the fact that $\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_\ell}$ can be expressed recursively through:

$$\delta_\ell \equiv \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_\ell} = (\boldsymbol{W}_{\ell+1}\delta_{\ell+1}) \odot \phi(\boldsymbol{a}_\ell) \tag{2.4}$$

where ϕ is the element-wise nonlinearity, *odot* denotes Hadamard multiplication, and $\boldsymbol{W}_{\ell+1}$ is the weight matrix of the next layer. A single forward a single backward pass re-uses these cached activations to compute all gradients, yielding an overall complexity O(cost-forward).

Key enablers of deep learning. (i) Automatic differentiation frameworks such as TensorFlow [2], PyTorch [1], and JAX [7] generalize backprop to arbitrary computation graphs, unrolling control flow and branching structures automatically. (ii) Hardware acceleration maps both the forward and backward tensor operations onto highly parallel GPU and TPU cores, exploiting fused convolution and matrix-multiply primitives. (iii) Improved regularization and normalization techniques—dropout, batch normalization, residual connections—address vanishing or exploding gradients, allowing networks hundreds of layers deep to train reliably. (iv) Scalable optimization methods such as Adam [46] and RMSProp [15] apply per-parameter adaptive learning rates, smoothing noisy gradients produced by mini-batch stochastic descent. (v) Memory-efficient training, such as gradient checkpointing, recomputes certain forward activations on-the-fly to trade computation for

reduced memory; reversible layers and swap-in/out schedules address GPU RAM limits encountered by billion-parameter models.

Beyond supervised learning. Backpropagation underlies almost every contemporary learning paradigm:

- In self-supervised representation learning, gradients flow through data-augmentation branches and contrastive objectives to sculpt invariant feature spaces.
- In reinforcement learning, policy-gradient and actor-critic algorithms apply backprop to networks that approximate value functions or directly parameterize action distributions.
- In generative modeling, score-based diffusion models, normalizing flows, and adversarial networks all rely on gradient signals—whether from denoising likelihoods, maximum-likelihood Jacobian penalties, or discriminator feedback—to steer high-dimensional density estimators.

The colorization framework proposed in later chapters employs end-to-end backpropagation to jointly optimize (i) a sketch encoder that captures line-art semantics, (ii) a generative backbone to synthesize colorized images. Fine-tuning modules via low-rank adaptation likewise depends on efficient gradient flow. Understanding the foundations and modern extensions of backpropagation, therefore, provides essential context for the methodological decisions and performance analyses presented throughout the remainder of this work.

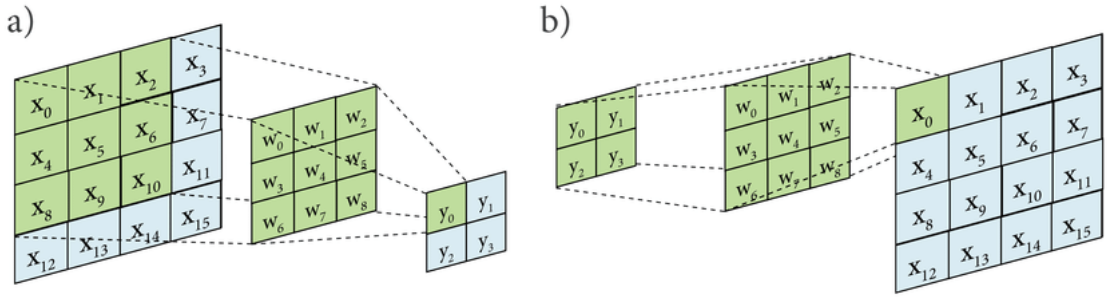


Figure 2.3: Convolutional layer and transpose convolutional layer [47, 59].

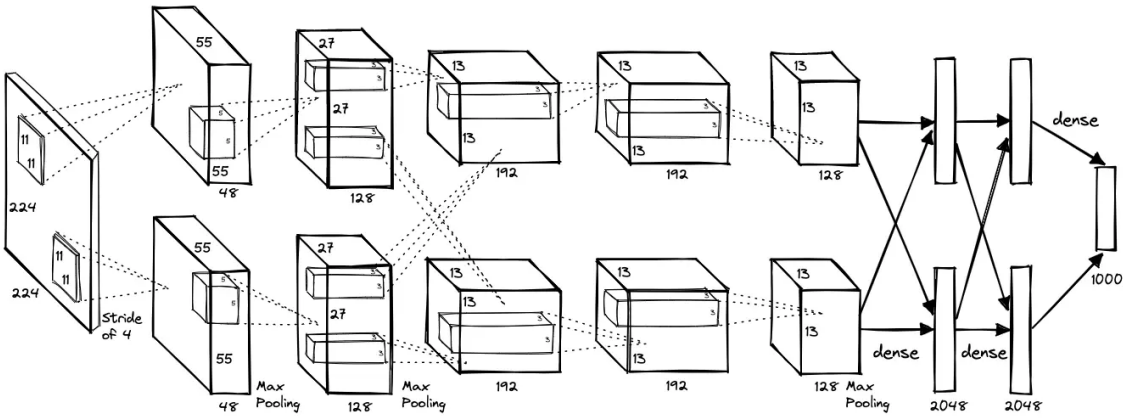


Figure 2.4: Illustration of Alexnet, a classification network [47].

2.1.3 Convolutional neural networks

Convolutional Neural Networks (CNNs) have become one of the most influential architectures in deep learning, particularly for tasks involving spatially structured data such as images, videos, and time series. Initially inspired by the hierarchical organization of the mammalian visual cortex [19, 47, 48], CNNs employ three fundamental principles—local receptive fields, parameter sharing, and spatial subsampling to exploit the spatial locality and compositional nature of visual data. These design choices significantly reduce the number of learnable parameters compared to fully connected architectures, facilitating the training of deep models while preserving spatial hierarchies in the input.

At the core of CNNs are convolutional layers, as shown in Figure 2.3, which apply a set of learnable filters (or kernels) that are spatially convolved over the input feature maps. Each filter is designed to detect specific local patterns—such as edges, textures, or object parts—by responding to spatially correlated features within a local neighborhood. As multiple layers are stacked, the network captures increasingly abstract and semantically rich representations, enabling effective hierarchical feature extraction. Non-linear activation functions (e.g., ReLU), batch normalization, and pooling operations (e.g., max pooling or average pooling) are often interleaved with convolutional layers to introduce non-linearity, stabilize training, and achieve spatial invariance.

The seminal success of AlexNet, as shown in Figure 4, on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 marked a turning point, demonstrating the viability of deep CNNs trained with large-scale data and GPU acceleration. This

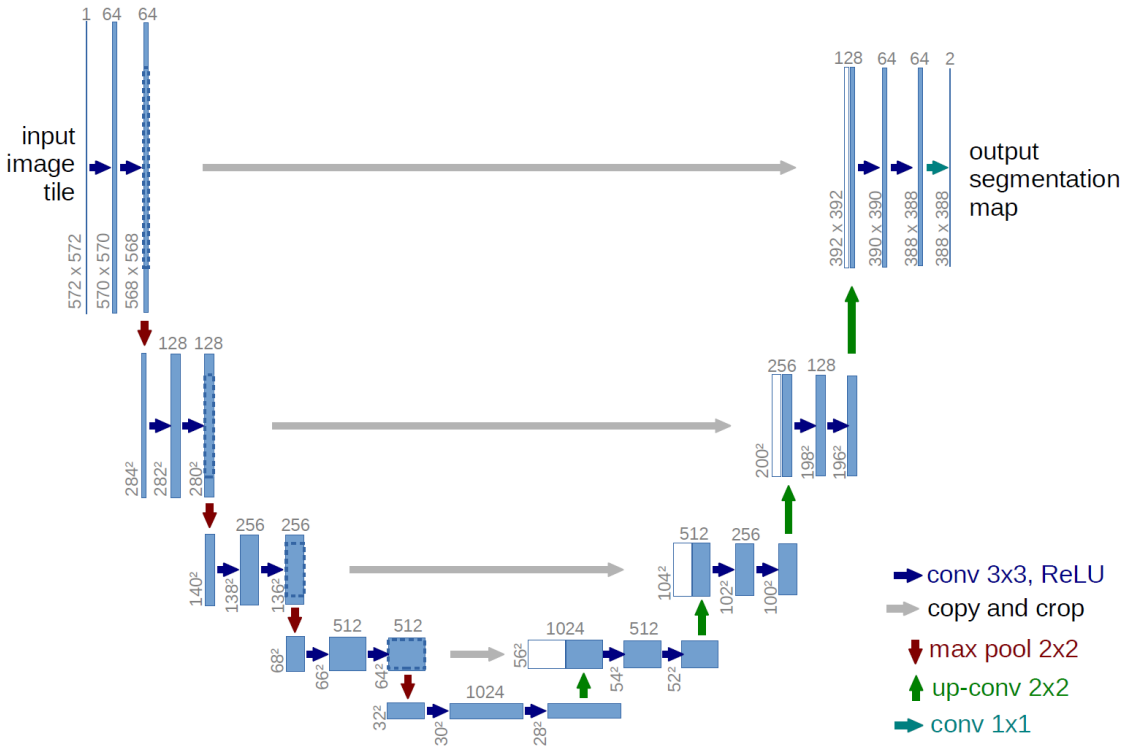


Figure 2.5: Illustration of U-net, an architecture widely used in various generative tasks [73]. A feature of U-Net is the skip connection between its encoder part and decoder part at each scale.

was followed by a series of architectural innovations that significantly extended the depth, efficiency, and expressive power of CNNs. VGGNet [79] demonstrated the effectiveness of deeper networks with small (3×3) filters; GoogLeNet [87] introduced the inception module to capture multi-scale features within a layer; ResNet [24] addressed the degradation problem in very deep networks using residual connections; and DenseNet [33] further enhanced information flow through dense layer-wise connections.

These developments not only advanced performance in core tasks such as classification, detection, and segmentation but also enabled CNNs to generalize to broader domains, including generative modeling, medical imaging, autonomous driving, and multimodal fusion, such as U-Net shown in Figure 2.5. In many contemporary systems, CNNs serve as versatile and modular components that can be integrated with other architectures such as recurrent neural networks (RNNs) for temporal modeling, transformers for capturing long-range dependencies, or graph neural networks (GNNs) for structured and relational data.

Despite the emergence of transformer-based architectures in vision tasks, CNNs remain widely used due to their inductive biases, computational efficiency, and robustness in data-scarce or resource-constrained settings. Furthermore, CNNs continue to evolve through hybrid models that incorporate attention mechanisms or learnable dynamic convolutions, reaffirming their central role in modern deep learning pipelines across both academic research and industrial deployment.

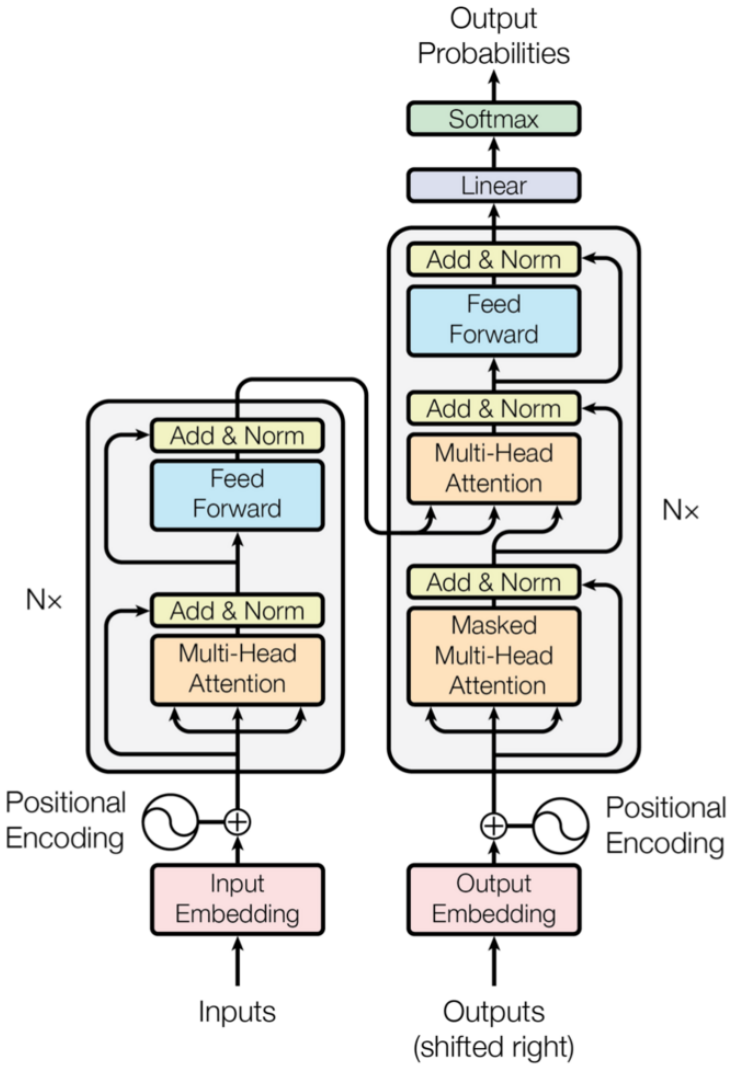


Figure 2.6: Illustration of vanilla transformer blocks [89].

2.1.4 Transformer

The Transformer architecture [14, 69, 89], as visualized in Figure 2.6. Illustration of vanilla trans-former blocks has emerged as the predominant paradigm in the field of natural language processing (NLP) over the past several years, largely due to its superior capacity for modeling long-range dependencies and its autoregressive nature.

Attention is the fundamental calculation inside transformers. It consists of two parts: scaled dot-product attention and a multi-head division. For the scaled dot-product, the attention layer linearly projects an input sequence $x \in \mathbb{R}^{n \times d_k}$ into query Q, key K, and values V, respectively, as follows:

$$Q = W_Q x, \qquad K = W_K x, \qquad V = W_V x, \qquad W_\cdot \in \mathbb{R}^{n \times d_k}. \tag{2.5}$$

A standard self-attention maps Q, K, V to an output Z via

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V \in \mathbb{R}^{n \times d_k}, \tag{2.6}$$

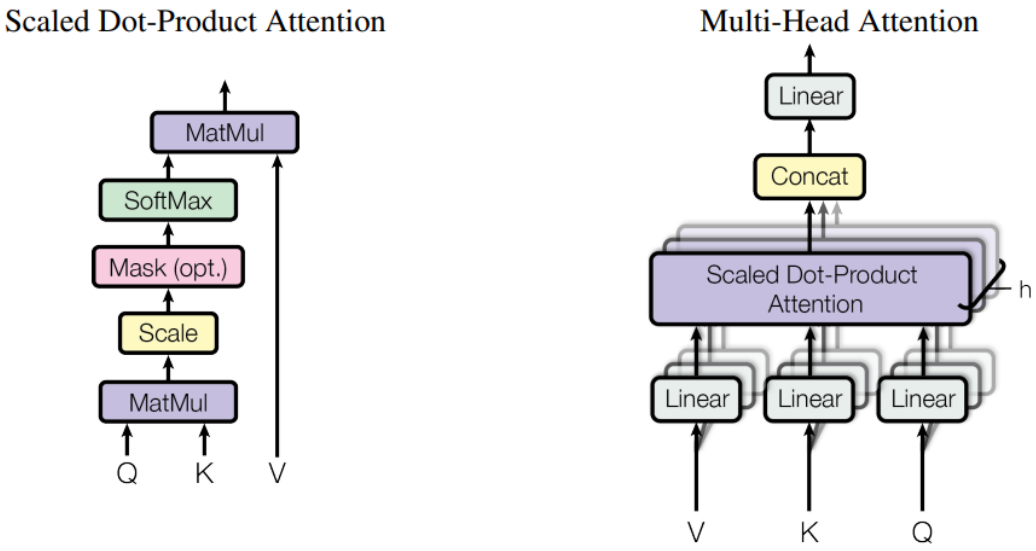


Figure 2.7: Visualization of multi-head attention.

where $\sqrt{d_k}$ is a scaling factor to stabilize the training by reducing the gradients. Instead of a single high-dimensional attention, multi-head attention learns h independent subspaces:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W_o, \\ \text{head}_i &= \text{Attention}(QW_Q^i, KW_K^i, VW_v^i), \end{aligned} \tag{2.7}$$

By leveraging attention mechanisms, visualized in Figure 2.7, and positional encoding, Transformers effectively capture contextual information across entire sequences, outperforming recurrent and convolutional models in both accuracy and scalability. Central to this framework are text tokenizers, which discretize raw text into sequences of tokens, and the encoder-decoder architecture, which facilitates diverse NLP tasks such as machine translation, dialogue systems, semantic understanding, and speech recognition. The introduction of large-scale pre-trained models, particularly those in the GPT (Generative Pre-trained Transformer) family [69], has further catalyzed progress in this domain. Notably, OpenAI's ChatGPT exemplifies the capabilities of such models, not only in achieving impressive performance across a wide range of language tasks but also in empirically validating the scaling laws of deep learning. These laws demonstrate that increasing model size, data volume, and compute resources can yield predictable improvements in performance, thereby underscoring the transformative potential of large generative models for both research and practical applications in NLP.

Following the success of Transformers in NLP, researchers in computer vision have been increasingly interested in leveraging their potential to address visual understanding problems [14,54]. Unlike text, which is inherently discrete and well-suited to tokenization, images consist of continuous-valued pixel arrays, presenting a fundamental mismatch with the discrete tokenization schemes typically used in language models. Despite this challenge, Transformer-based architectures have demonstrated remarkable performance in a wide range of vision tasks, such as image classification, object detection, and semantic seg-

mentation. This success is largely attributed to the self-attention mechanism's ability to model long-range dependencies and aggregate information globally, allowing the network to capture high-level semantic relationships across spatial regions. Compared to convolutional layers, which are inherently limited by their local receptive fields, attention layers enable a more holistic understanding of visual content. As a result, Transformer models have significantly advanced the state-of-the-art in vision by offering a powerful alternative to the inductive biases traditionally embedded in convolutional neural networks.

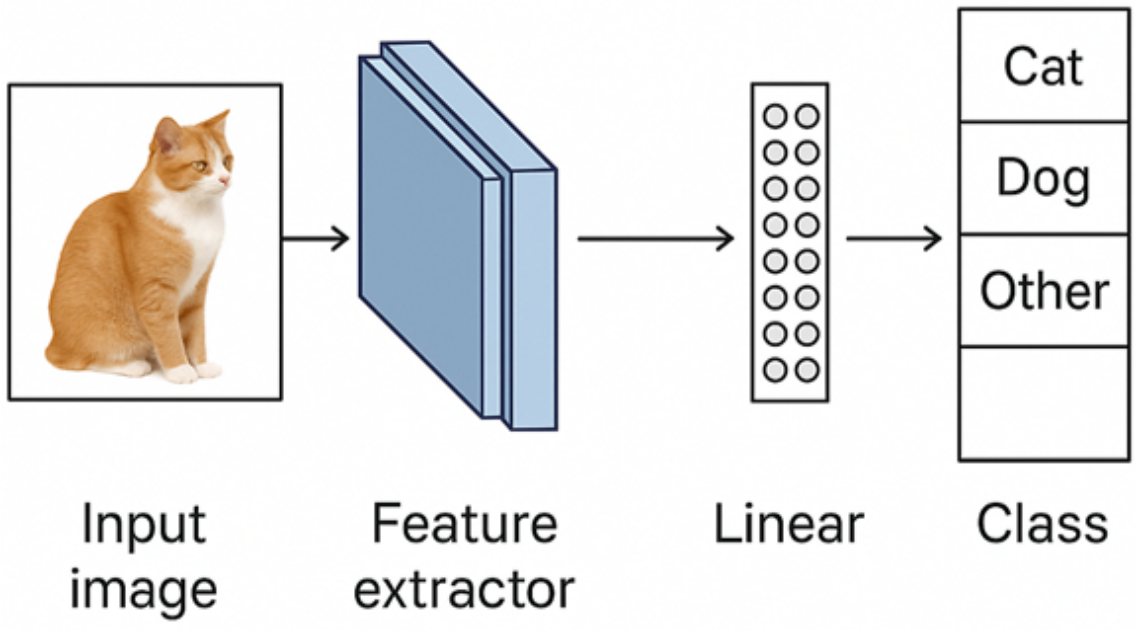


Figure 2.8: Classic image classification framework

2.1.5 Multi-modal models

Earlier multi-modal models were relatively simple in design and typically consisted of a single trainable neural backbone, most commonly based on CNNs [24,33,47,79,87]. These models primarily functioned as visual feature extractors, followed by a few projection or classification layers, as illustrated in Figure 2.8. Their primary applications were limited to foundational multi-modal tasks such as image classification and semantic segmentation. In such tasks, the model's objective was to map RGB visual inputs to predefined text labels or semantic masks [22], with limited contextual understanding or cross-modal reasoning. Due to their constrained architectures and task-specific training, these early models lacked the ability to generalize across more complex modalities or exhibit flexible conditioning capabilities required in generative or interactive systems. Nevertheless, they laid the groundwork for subsequent developments in multi-modal learning by demonstrating the feasibility of integrating visual representations with symbolic or textual supervision.

However, such a simple framework cannot solve the challenges emerged in recent years, specifically, zero-shot multi-modal understanding tasks that require networks to respectively embed inputs from different modalities into a shared embedding space, making these embeddings transferable to each other. To tackle this challenge, multi-modal frameworks have become much more complicated by involving multiple deep trainable DL modules, which are trained in different stages across various datasets.

One of the most influential multi-modal frameworks in recent years is CLIP (Contrastive Language–Image Pre-training) [68], which has significantly reshaped the field of cross-modal representation learning. CLIP is trained with a contrastive loss that aligns embeddings of paired images and texts in a shared latent space while simultaneously pushing apart non-matching pairs. Its large-scale pre-training pipeline, visualized in Figure 2.9, uses 400 million image–text pairs collected from the internet, allowing it to learn open-vocabulary associations without requiring manual annotation or task-specific supervision.

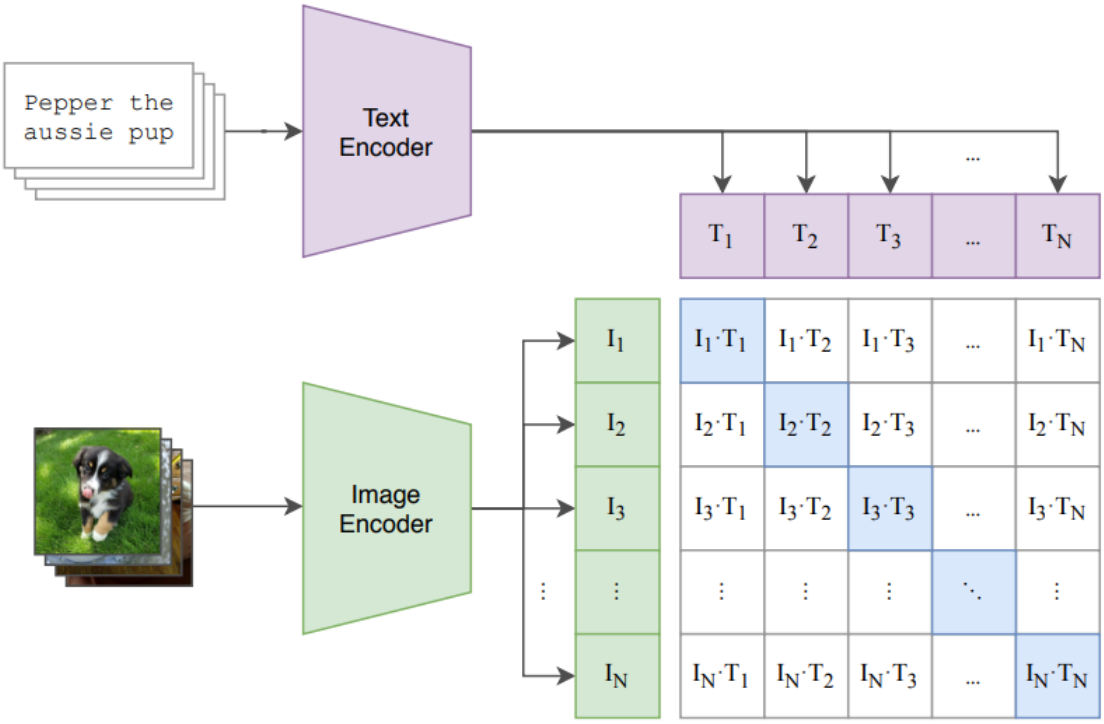


Figure 2.9: Contrastive pre-training in CLIP [68].

Architecturally, CLIP comprises two independent, modality-specific encoders:

1. Image encoder. Following CLIP's original design [68], the image encoder is instantiated as a ViT [14]. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the encoder first partitions it into

$$N = \frac{HW}{p^2} \tag{2.8}$$

non-overlapping $p \times p$ patches. Each patch is linearly projected to a d-dimensional *local patch token* $x_i \in \mathbb{R}^d$. A learnable [CLS] token x_{cls} is prepended, and fixed two-dimensional sine–cosine positional embeddings p_i are added to preserve spatial layout, yielding the input sequence

$$z_0 = \left[x_{\text{cls}} + p_0;\ x_1 + p_1;\ \ldots;\ x_N + p_N\right] \in \mathbb{R}^{(N+1)\times d}. \tag{2.9}$$

This sequence is processed by L transformer layers, each comprising multi-head self-attention and feed-forward sub-blocks, to produce the final hidden states z_L. The global image representation is taken as the last-layer [CLS] vector,

$$v_{\text{img}} = z_L^{(\text{cls})} \in \mathbb{R}^d, \tag{2.10}$$

which serves as a compact yet semantically rich descriptor for image–text alignment tasks. Meanwhile, the transformed patch tokens $\{z_L^{(i)}\}_{i=1}^N$ retain fine-grained spatial information and can be leveraged later for region-level conditioning or localization. These patch tokens are defined as [Local] tokens in the following sections.

2. Text encoder. The text encoder is a Transformer-based language model. Given an input text prompt T, it tokenizes the sentence and encodes it through self-attention layers, outputting a feature vector $v_{text} \in \mathbb{R}^d$, typically by pooling the final hidden state of the [EOS] or last token.

Both embeddings are then normalized (i.e., unit vectors) and mapped into a shared embedding space using learnable projection heads:

$$z_{\text{img}} = \frac{P_{\text{img}} \cdot v_{\text{img}}}{\|P_{\text{img}} v_{\text{img}}\|}, \qquad z_{\text{text}} = \frac{P_{\text{text}} \cdot v_{\text{text}}}{\|P_{\text{text}} v_{\text{text}}\|}, \tag{2.11}$$

where $P_{\text{img}}, P_{\text{text}} \in \mathbb{R}^{d \times d}$ are learnable linear projections. The full model is optimized using the InfoNCE contrastive objective, encouraging matched image–text pairs to have higher cosine similarity compared to unmatched ones while pushing apart mismatched pairs. This design facilitates semantically rich and transferable representations, enabling zero-shot and few-shot generalization across a wide array of downstream tasks.

Let a batch contain N image–text pairs $\{(I_i, T_i)\}_{i=1}^N$, the cosine similarity between any image i and text t is defined as:

$$s_{ij} = z_{\text{img}} \cdot z_{\text{text}} \tag{2.12}$$

The InfoNCE loss (a symmetric contrastive loss) is applied in both directions:

$$\mathcal{L}_{\text{img}} = -\frac{1}{N} \log\left(\frac{\exp(\frac{s_{ii}}{\tau})}{\sum_{j=1}^N \exp(\frac{s_i^j}{\tau})}\right), \qquad \mathcal{L}_{\text{text}} = -\frac{1}{N} \log\left(\frac{\exp(\frac{s_{ii}}{\tau})}{\sum_{j=1}^N \exp(\frac{s_j^i}{\tau})}\right) \tag{2.13}$$

where τ is a learnable temperature parameter. The total loss is:

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_{\text{img}} + \mathcal{L}_{\text{text}}) \tag{2.14}$$

This formulation ensures that each image embedding is maximally similar to its corresponding text embedding, and dissimilar to all others in the batch—and vice versa.

The success of CLIP has had a profound impact on the development of multimodal systems. Its embeddings are widely used as priors in generative models. It has also inspired a series of follow-up works, including ALIGN [38], which scales training with larger datasets and more powerful architectures, and BLIP [51], which integrates vision–language pre-training with retrieval and generation capabilities. Together, these works represent a paradigm shift from narrowly tailored models toward foundation models capable of generalizing across tasks, domains, and modalities.

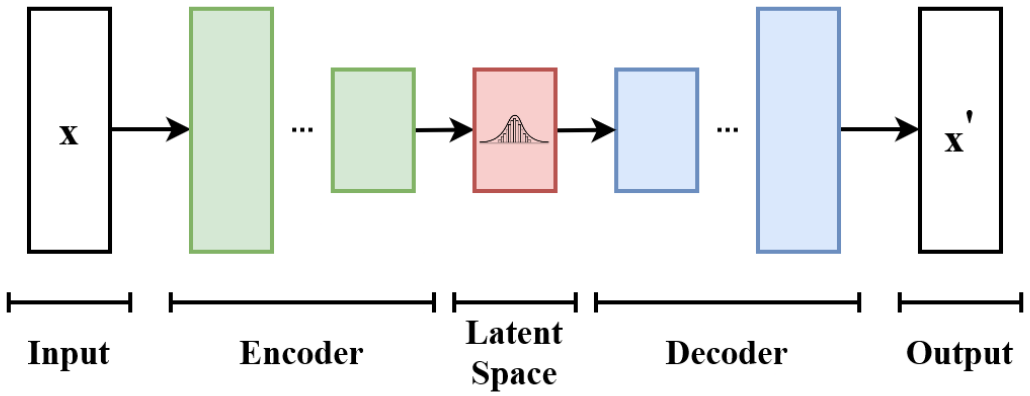


Figure 2.10: Illustration of variational autoencoder [45].

2.2 Generative Models in Vision

2.2.1 Auto-encoder and variational auto-encoder

In the context of computer vision powered by deep learning, image generation is broadly conceptualized as the process of mapping randomly sampled noise vectors—typically drawn from a standard normal distribution—into structured RGB images through a series of trainable deep neural modules. These noise vectors serve as abstract, high-level representations that reside in a semantic space where visual information is highly compressed. As such, their dimensionality is significantly lower than that of the output images. This compressed semantic representation space is commonly referred to as the latent space, and models that operate within this framework are known as Latent Variable Models (LVMs). These models learn to decode the latent codes into realistic visual outputs, effectively modeling the complex distribution of natural images. By operating in latent space, generative models such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Diffusion Models (DMs) can efficiently capture the underlying structure of image data while maintaining computational tractability. This latent-space-based generation paradigm enables both high-resolution synthesis and conditional generation when guided by external modalities such as text, pose, or segmentation maps.

One of the earliest and most fundamental forms of latent variable models (LVMs) in deep learning is the Autoencoder (AE) [28]. An AE consists of two primary components: an encoder, typically implemented as a convolutional neural network (CNN), which compresses an input image into a low-dimensional latent code; and a decoder, often a symmetric upsampling CNN, which reconstructs the image from this latent representation. Both components are trained jointly using a reconstruction loss, such as the L2 or mean squared error (MSE) loss, to ensure the output closely resembles the input. While AEs are effective at learning compact representations and reconstructing images, they suffer from a critical limitation: the latent space becomes fixed once training is complete, making it difficult to generate diverse or semantically meaningful variations of outputs from arbitrary latent vectors.

To address this limitation, the Variational Autoencoder (VAE) [45] was introduced

as a probabilistic extension of the standard AE framework. Instead of encoding each image into a single deterministic code, the VAE encodes it into a latent distribution, typically modeled as a multivariate Gaussian with a learned mean and variance. This process is visualized in Figure 2.10. During training, the decoder receives as input a sample drawn from this distribution, introducing stochasticity and enabling meaningful sampling in latent space. This design allows the model to generalize better and generate a diverse range of outputs. However, naively sampling from such distributions would break the backpropagation required for training. To overcome this, the VAE employs a reparameterization trick, which expresses the sampling operation as a differentiable function of the learned parameters and a source of noise. Additionally, the VAE's objective function incorporates a Kullback–Leibler (KL) divergence term to encourage the learned latent distributions to stay close to a standard normal prior, thereby regularizing the latent space and promoting smoothness and continuity in generation.

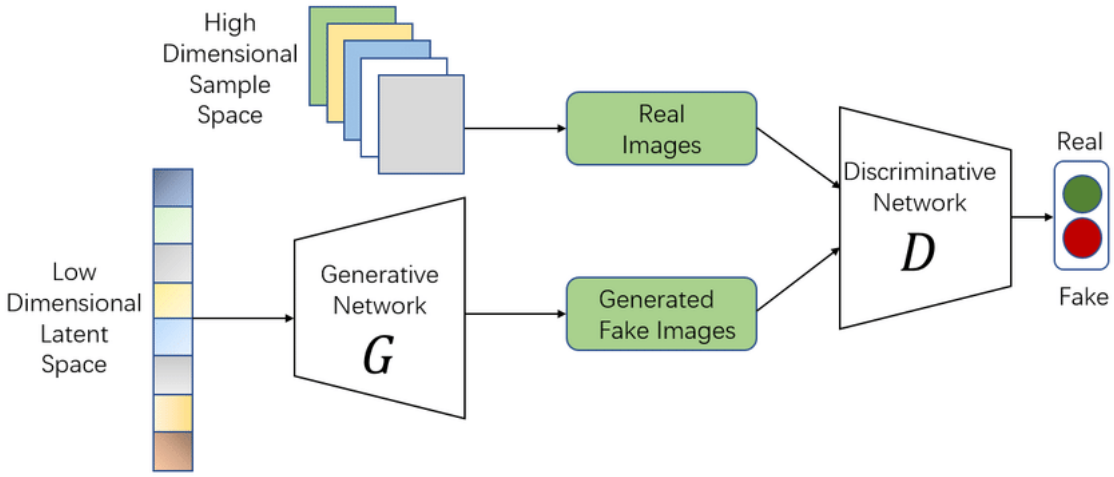


Figure 2.11: Training pipeline of GAN [5].

2.2.2 Generative adversarial networks

While the VAE successfully introduces stochasticity and diversity into the latent space through probabilistic modeling, it comes with a notable drawback: the incorporation of the KL divergence regularization often leads to blurry or low-contrast outputs, especially when trained on complex natural images. This degradation in perceptual quality is a direct consequence of the strong constraint that forces the learned latent distributions to closely match the prior, often at the expense of fine-grained visual fidelity.

To address this limitation, Generative Adversarial Networks (GANs) emerged as a powerful alternative during the same period. First proposed by Goodfellow et al. in 2014 [23], a GAN is composed of two neural networks—the generator and the discriminator—which are trained in an adversarial setting. Unlike AE or VAE architectures that require an encoder to map ground-truth images into latent codes during training, GANs begin directly with randomly sampled noise vectors drawn from a predefined latent distribution (e.g., Gaussian). The generator learns to decode this noise into synthetic RGB images.

These synthesized images are then evaluated by the discriminator, which is trained to distinguish between real images (i.e., ground-truth samples from the dataset) and fake images (i.e., outputs from the generator). The generator, in contrast, is optimized to fool the discriminator—that is, to produce images that are indistinguishable from real ones. The training process can be visualized as Figure 2.11. This min-max game between the two networks results in a powerful training dynamic that often leads to sharper, more realistic image synthesis than that achieved by VAEs. GANs have since become a cornerstone of generative modeling, inspiring a wide range of variants and extensions aimed at improving training stability, diversity, and controllability.

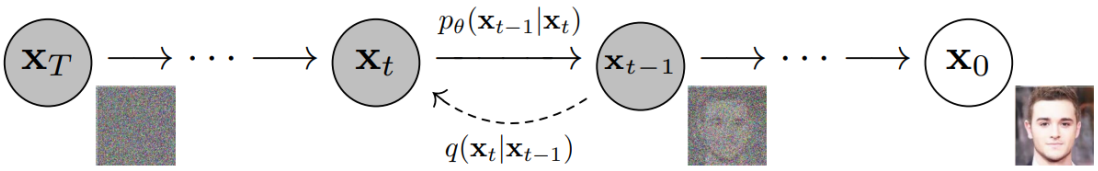


Figure 2.12: The diffusion and denoising processs of diffusion models [29].

2.2.3 Diffusion models and flow matching

Although GANs significantly outperformed VAEs in terms of perceptual quality and image realism, they suffer from several fundamental limitations that hinder their broader applicability. Most notably, GAN training is notoriously unstable, often requiring careful hyperparameter tuning, architecture balancing between generator and discriminator, and regularization techniques to avoid issues such as mode collapse and non-convergence. These challenges become especially pronounced when generating images with diverse content, intricate spatial composition, or complex visual styles.

In addition to training instability, GANs also face criticism regarding their lack of interpretability. Since GANs directly map latent vectors to RGB images without an explicit generative likelihood, it is difficult to understand or control how specific attributes in the latent space influence the final output. This opaqueness limits their integration into systems requiring controllable, transparent, or semantically grounded generation.

Due to these fundamental drawbacks, GANs were gradually supplanted by Diffusion Models (DMs)—a newer class of generative models that offer both training stability and high-fidelity synthesis. This paradigm shift was catalyzed by the work of Ho et al. and Song et al. [29,81–83], who formalized score-based generative modeling and established the theoretical foundations and practical viability of diffusion-based approaches. Since then, diffusion models have rapidly become the state-of-the-art across a wide range of generative tasks, including image, audio, and video synthesis.

DMs represent a fundamentally different approach to generative modeling compared to Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), both of which generate images in a single forward pass. In contrast, diffusion models decompose the generation process into a multi-step denoising trajectory, typically consisting of T discrete steps. The underlying idea is to learn how to reverse a gradual noising process that corrupts data over time, thereby generating clean images from pure noise through iterative refinement. This process is visualized in Figure 2.12, and the standard training objective for diffusion models is

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\right], \tag{2.15}$$

where $\mathbf{x}_0$ is the clean data sample, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is standard Gaussian noise, and t is uniformly sampled from $\{1, \ldots, T\}$. The noisy input $\mathbf{x}_t$ is constructed as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ is the cumulative product of predefined noise scaling factors $\alpha_s \in (0, 1)$. The network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is trained to predict the added noise $\boldsymbol{\epsilon}$.

Specifically, the diffusion process begins by progressively adding Gaussian noise to training images over T timesteps, eventually transforming them into pure noise. During

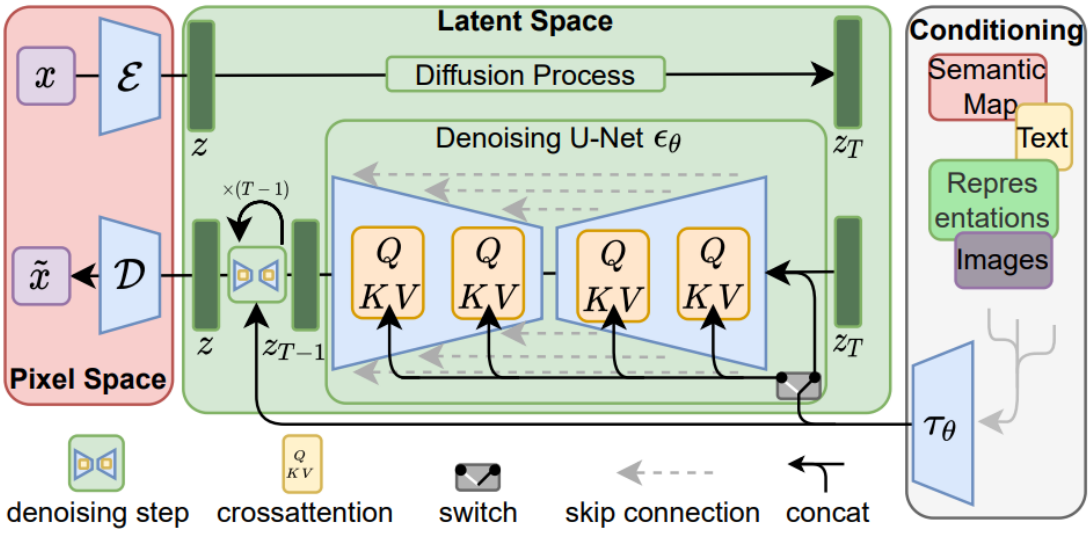


Figure 2.13: Framework of the LDM proposed by Rombach et al., which is also known as Stable Diffusion (SD) [72].

training, the model learns to predict the added noise at each timestep, effectively modeling the conditional distribution of the noise given the current noisy input and timestep. This is accomplished under a predefined noise schedule, which determines how much noise is added at each step.

At inference time, the model starts from a random noise sample and recursively denoises it step-by-step, reversing the diffusion process to synthesize high-quality images. By decomposing the generation task into a series of gradual refinements, diffusion models avoid common pitfalls seen in single-pass generation. They circumvent the mode collapse and sharpness–diversity trade-off observed in GANs, as well as the blurry reconstructions often produced by VAEs. Moreover, DMs benefit from stable training dynamics and well-grounded probabilistic formulations, which further enhance their appeal in both research and practical applications.

The numerical stability of score-based diffusion models revolutionized image-synthesis research, empowering the community to scale architectures to billions of parameters and tackle cross-modal tasks such as text-to-image (T2I) generation. Pioneering work began with GLIDE [62], which introduced cascaded diffusion decoding and classifier-free guidance, producing photorealistic yet diverse samples. This momentum continued with DALL·E 2 [71], whose CLIP-conditioned prior and diffusion decoder translated complex prompts into high-resolution imagery, igniting mainstream fascination with AI art. Almost concurrently, Google released Imagen [76], pairing a large T5 language backbone [70] with a cascaded diffusion pipeline to achieve state-of-the-art FID scores and underscoring the value of powerful text encoders in T2I systems.

Despite these successes, early diffusion frameworks faced a significant computational bottleneck: hundreds to thousands of iterative denoising steps were needed for a single sample. Latent Diffusion Models (LDMs)—first formalized by Rombach et al. [72]-addressed this drawback by performing the diffusion process in a perceptually compressed latent space learned by a pretrained VAE, cutting both training and inference cost by

roughly an order of magnitude without sacrificing semantic fidelity (see Figure 2.13). The open-source release of Stable Diffusion, built directly on the LDM blueprint, democratized high-quality image synthesis via permissive licensing and consumer-GPU support, which in turn sparked a thriving ecosystem of community fine-tuning and downstream applications. Additional commercial efforts—such as Google's Parti autoregressive-diffusion hybrid [99] and the artist-centric Midjourney [58] platform—expanded creative possibilities by providing distinctive style priors and intuitive user interfaces.

Collectively, these early frameworks established three core algorithmic pillars—cascaded generation, latent-space diffusion, and prompt-based guidance—that continue to underpin modern multimodal generative systems.

Flow Matching (FM) [53, 82] reformulates generative modeling as solving an ordinary differential equation (ODE) that transports samples from a base distribution to the data distribution via a learned time-dependent velocity field. This continuous formulation enables the use of efficient ODE solvers to accelerate sampling. A variety of solvers have been introduced or adapted for this purpose, including the Euler method [40] and DPM-Solver [56]. These solvers significantly reduce the number of sampling steps—often from hundreds to fewer than twenty—while maintaining high sample quality. Their compatibility with deterministic trajectories makes them particularly effective for continuous-time models such as those based on Flow Matching or ODE-formulated diffusion.

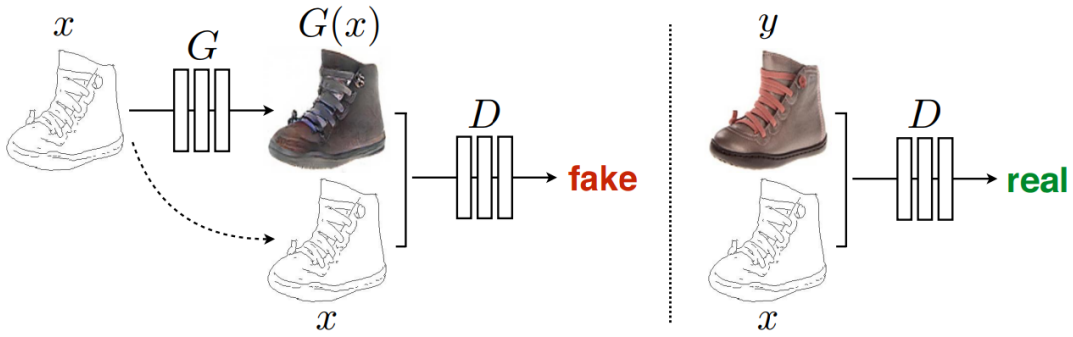


Figure 2.14: Different from vanilla GAN, cGANs condition generation on input image.

2.3 Controllable generation

2.3.1 Conditional GANs

Compared to research-oriented investigations that typically focus on enhancing the performance and theoretical properties of Latent Variable Models (LVMs), industrial applications place significantly greater emphasis on practical aspects, particularly the controllability and visual fidelity of the synthesized content. Industrial scenarios often demand precise and intuitive manipulation of generated outputs to align with user intentions or specific business requirements. Such controllability, alongside achieving superior visual realism and consistency, is generally accomplished by selectively sampling representations from carefully designed conditional distributions embedded within LVM frameworks. Motivated by these stringent industrial requirements, numerous conditional generative networks have been proposed. These networks employ various conditional cues—ranging from explicit user instructions and textual descriptions to specific visual contexts—to effectively guide the generative process, thereby delivering synthesized outputs that are not only visually compelling but also readily customizable and controllable according to application-specific demands.

Researchers have developed a wide range of conditional generative networks leveraging GANs, as GANs were the predominant generative backbone during the early years of generative modeling research. Notably, influential frameworks such as pix2pix emerged [37], pioneering supervised image-to-image translation tasks. These frameworks significantly advanced the field by enabling effective manipulation of visual attributes and providing robust control over generative processes, thus establishing foundational standards for subsequent conditional generation methods. Different from vanilla GANs, which generate images from randomly sampled noise, the generator networks adopted in conditional GANs (cGANs) are typically U-Net and take conditional RGB images as inputs, as visualized in Figure 2.14.

The other conditional generation framework of GANs focuses on directly manipulating latent codes within the latent space, a representative approach exemplified by StyleGAN [41, 42]. Proposed by Karras et al., StyleGAN introduces an innovative hierarchical architecture that separates the latent space into distinct layers controlling different levels of visual detail, such as coarse structure, intermediate features, and fine-grained texture. By

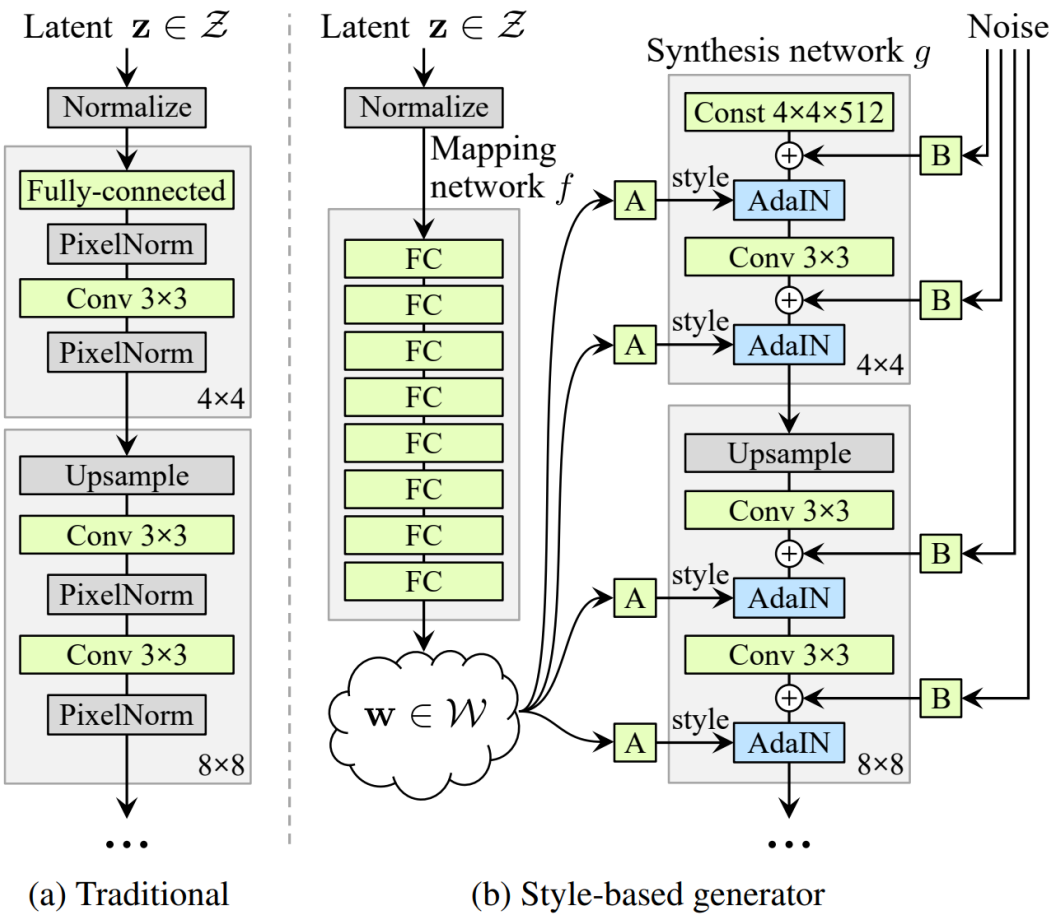


Figure 2.15: Comparison between vanilla GANs and StyleGAN from [41].

leveraging a carefully designed mapping network that projects random latent vectors into an intermediate latent space and modulates adaptive instance normalization (AdaIN) parameters across generator layers, StyleGAN enables highly disentangled and interpretable latent representations. A comparison regarding the difference of network architectures is given in Figure 2.15. Consequently, this structured latent code manipulation allows precise control over image attributes, facilitating tasks like semantic editing, style transfer, and diverse image synthesis with superior visual quality and fidelity.

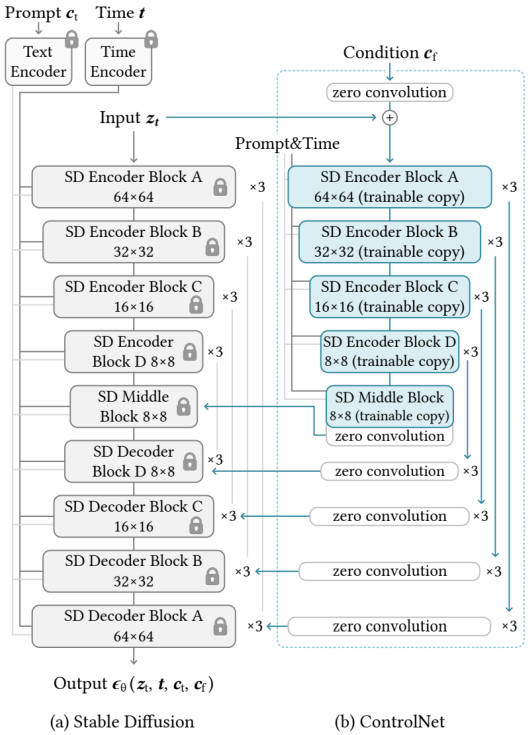


Figure 2.16: Stable Diffusion's U-net architecture with a connected ControlNet [101].

2.3.2 Classifier-free Guidance and Diffusion Adapter

Different from GANs, most diffusion models (DMs) can be effectively trained for text-to-image generation, showcasing their superiority in accurately capturing and translating textual descriptions into high-quality visual outputs. This characteristic indicates that DMs consistently outperform other generative models when guided by text prompts, primarily due to their explicit modeling of conditional probability distributions through iterative denoising steps. However, despite their remarkable generation fidelity and enhanced controllability, DMs incur significantly higher computational costs and require substantially larger parameter counts compared to other generative architectures, thereby escalating the training expenses dramatically.

Consequently, to mitigate the computational burden associated with directly fine-tuning entire diffusion models for specialized conditional generation tasks, researchers propose classifier-free guidance [30] for strengthening the guidance of prompt conditions and frequently adopt modular adaptation strategies to introduce additional control conditions.

Classifier-Free Guidance (CFG). CFG is a low-overhead alternative to the original *classifier guidance* technique for conditional diffusion. Rather than training and repeatedly querying an external image classifier at sampling time, CFG reuses the same diffusion network to produce *two* score estimates at each timestep t:

- *Conditional score* $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t \mid \mathbf{c})$, obtained by running the network with the desired conditioning signal $\mathbf{c}$ (text, sketch, class label, *etc.*).
- *Unconditional score* $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t)$, obtained from the same network with its conditioning input replaced by a null token.

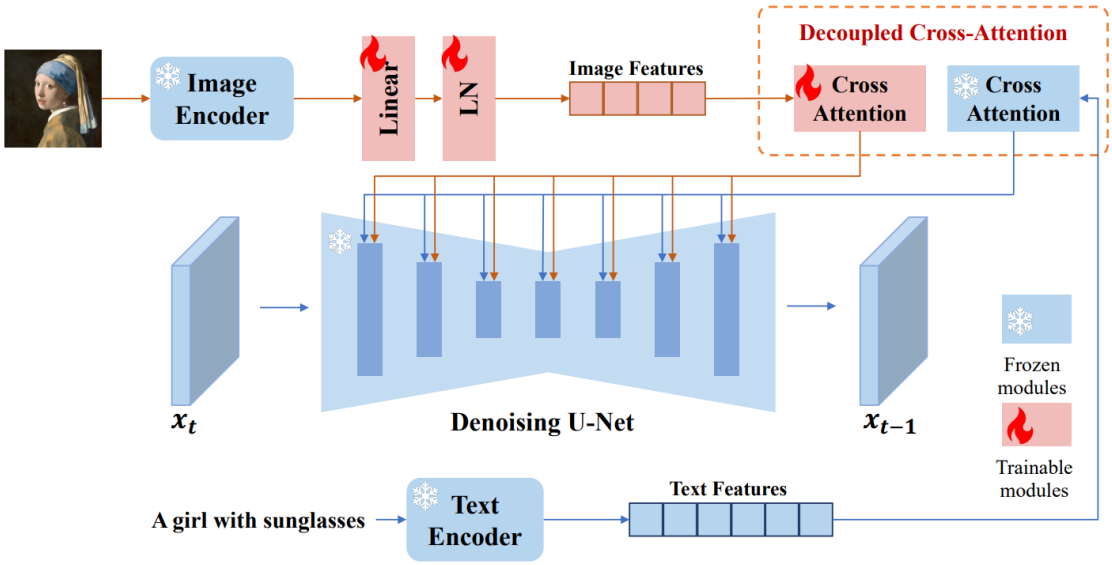


Figure 2.17: IP-Adapter injects image-prompt cues, enabling text-to-image models to follow reference visuals [98].

At inference, these scores are linearly combined

$$\tilde{\nabla}_{\mathbf{x}} = \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t \mid \mathbf{c}) + \gamma \Big(\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t \mid \mathbf{c}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t) \Big), \tag{2.16}$$

where the *guidance scale* $\gamma \geq 0$ balances fidelity and diversity: $\gamma = 0$ recovers the n-conditional model, moderate values ($1 \leq \gamma \lesssim 2$) provide a good trade-off, and large values (($\gamma > 5$) enforce the prompt strongly at the expense of diversity and may introduce oversaturation artefacts. Because both passes share all weights, CFG incurs no additional training cost and only a negligible extra forward call per denoising step; it is therefore the de facto standard in modern systems such as GLIDE, Stable Diffusion [72], DiT [64], and related models [30].

Adapter-based modular fine-tuning. Adapter modules—compact, trainable components are introduced to the existing diffusion architecture. These adapters can be efficiently optimized in isolation, while the core denoising backbone remains frozen. This approach not only dramatically reduces computational requirements and accelerates training convergence but also preserves the generalization capabilities of the pre-trained backbone, enabling swift and cost-effective adaptation to diverse downstream tasks.

Three representative adapters that tackle these limitations are ControlNet [101], IP-Adapter [98], and T2I-Adapter [60]. A persistent weakness of conventional text-to-image diffusion models is their poor control over spatial composition: because the text prompt provides only high-level semantics, the generator often ignores precise layout, perspective, or object boundaries. To inject explicit spatial structure, ControlNet originally proposed for Stable Diffusion (SD) [72] adds a parallel, fully trainable copy of the SD U-Net encoder–decoder. This copy is initialized with the frozen backbone weights, but each convolution is preceded by a zero-initialized "control" convolution. During fine-tuning, these zero convolutions gradually learn residual features from external condition maps (e.g., Canny edges, depth, human pose, and semantic masks) while the original backbone remains untouched. The two branches are merged at every resolution level, allowing the

model to respect geometric constraints without sacrificing the pre-trained generative prior. Because only the zero convolutions are updated, ControlNet adds approximately 20% extra parameters yet enables pixel-level layout fidelity and one-shot domain adaptation, as illustrated in Figure 2.16.

While ControlNet excels at geometry, it does not transmit rich style or reference-image cues. IP-Adapter addresses this gap by learning a lightweight image-prompt adapter that converts a reference image into "style tokens." Built on a frozen CLIP image encoder, the adapter produces a small set of learnable vectors fed into the cross-attention layers of SD in place of (or alongside) text tokens, as visualized in Figure 2.17. Training the adapter alone, typically 2% of the backbone size, enables efficient personalization: users can steer color palette, texture, lighting, or artistic style simply by supplying one or a few reference frames, all without touching the gigaparameter diffusion core.

Together, these adapters demonstrate a pragmatic direction for conditional generation research: instead of expensive end-to-end fine-tuning, task-specific, plug-and-play modules can be optimized to inject geometry (ControlNet) or appearance (IP-Adapter) cues, achieving fine-grained controllability with minimal compute and memory overhead.

Figure 2.18: In grayscale image colorization, geometry semantics can be extracted from structural inputs, which are grayscale images in such tasks. Illustration from [25].

Figure 2.19: Style transfer aims to transfer specific visual features from a reference image to an original input [21].

2.4 Sketch colorization

2.4.1 Image colorization and style transfer

Sketch colorization belongs to the broader family of image-colorization problems, yet it poses a markedly different set of challenges from classical color recovery of grayscale photographs. In a grayscale image, the luminance channel still preserves all low- and mid-level visual cues—edges, shading, material texture, even subtle stroke patterns, so modern frameworks such as Colorful Image Colorization and Deep Exemplar-based Colorization [25,105] can concentrate on learning a mapping from luminance to chrominance. The network's task is essentially to "repaint" an already complete geometric canvas, injecting plausible hues while relying on the input itself for structure and detail (see Figure 2.18. The colorization framework in such tasks only needs to transfer **chromatic colors** without considering **achromatic style** information, such as **textures/strokes**.

A hand-drawn sketch, by contrast, is an extremely sparse representation: outlines are often one-pixel wide, interior regions lack shading, and no material or texture cues are present. Consequently, a successful sketch colorizer must accomplish two coupled

objectives:

Semantic hallucination. It must infer region boundaries, object identities, and depth ordering purely from abstract strokes—information that a grayscale photo already provides.

Style-aware propagation. It must transfer not only global color palettes but also fine-grained stylistic attributes—cell-shaded tones, cross-hatching, painterly textures, or manga-specific screen tones—from its guiding conditions (text, user scribbles, or reference images) onto the outline while respecting stroke fidelity.

These requirements render sketch colorization much closer to texture synthesis and neural style transfer [21, 34, 39] — which transfer primarily texture information from a reference image to an original image via features drawn from specific neural layers—than to conventional color recovery. An illustration of neural style transfer is shown in Figure 2.19. Naïvely applying grayscale-oriented models produces color bleeding across line boundaries, washed-out regions, and stylistic mismatches, underscoring the need for edge-aware attention, stroke-conditioned diffusion, or region-aware correspondence modules. In short, whereas grayscale colorization mainly predicts what colors to add, sketch colorization must also decide where and how to place them, simultaneously reconstructing missing geometry and applying stylistically coherent texture.

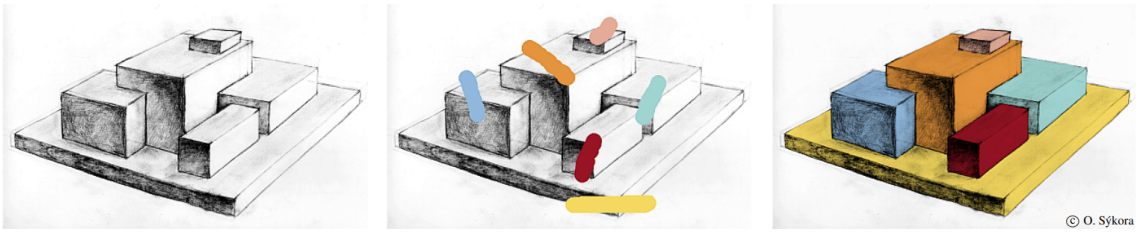


Figure 2.20: Lazybrush [86] implements an accelerated colorization with user-given color spray.

2.4.2 Traditional sketch colorization methods

Colorizing line art—especially at production scale for animation or manga—is still a labor-intensive bottleneck. To ease this burden, researchers have introduced a series of semi-automatic tools that propagate color from a handful of user strokes. LazyBrush [86], for example, formulates the task as a graph-cut labeling problem: the artist places a few coarse scribbles, and the algorithm flood-fills entire regions while respecting line boundaries. Building on this principle, Delaunay Painting [63] improves edge adherence by using Delaunay triangulation to guide color diffusion across broken contours; Sato et al.'s method [77] employs quadratic programming over a graph to maintain global palette consistency in manga pages; and Fourey et al.'s fast flat-coloring algorithm [18] accelerates the process with lightweight, region-growing heuristics. Although these techniques dramatically reduce manual effort, they are fundamentally hint-driven: color propagation succeeds only when the artist provides explicit scribbles or region selections. Such reliance on direct user input makes them ill-suited for reference-based colorization, where the goal is to transfer palettes and shading cues automatically from a sample image without extensive interactive guidance.

Figure 2.21: User-guided methods require users to give color spots at desired regions [104].

2.4.3 Deep learning sketch colorization methods

Deep learning has become the de-facto framework for automatic sketch colorization, consistently producing high-quality, high-resolution outputs that classical optimization pipelines struggle to match. Current approaches are typically grouped by their guiding modality: user-given hints, natural language prompts, and reference color images.

User-guided methods [20,104,106] leverage explicit, localized inputs—such as scribbles, point strokes, region-based fills, or spray patterns—to offer artists fine-grained control over both hue and shading. These approaches are particularly valued in professional workflows where precise artistic intent must be preserved. Although they enable high-fidelity color placement and stylistic consistency, their dependence on manual input severely limits scalability and automation, making them unsuitable for large-scale deployment or real-time applications. A typical pipeline of such user-guided frameworks is illustrated in Figure 2.21. Despite the stark difference in guiding modality compared to reference-based sketch colorization, their underlying neural architectures share notable similarities with image-guided generative models. In these frameworks, the user-provided hints—often formatted as RGB maps—are processed through a dedicated encoder in an independent forward pass. The extracted condition features are then fused with the main generative stream, enabling the network to modulate the colorization of sketch features based on localized guidance. This modular structure not only facilitates flexible conditioning but also underscores the architectural generalizability of encoder-decoder-based colorization systems across different guiding modalities.

Text-prompted methods [44,101,111] capitalize on the recent breakthroughs in text-to-image (T2I) diffusion models and vision–language alignment, enabling natural-language descriptions to be translated into plausible colorization outputs. These approaches eliminate the need for paired visual references by instead leveraging rich, human-readable prompts to guide the generative process. Following the general framework of T2I generation, text-guided colorization methods typically employ pre-trained text encoders—such as CLIP [68] or T5 [70]—to convert input prompts into high-dimensional embeddings, which are then injected into the diffusion process as conditioning signals. This paradigm offers an intuitive interface for users and facilitates semantic-level control over style and color tone. However, in practice, such methods often struggle with enforcing precise chromatic relationships, maintaining local textural fidelity, or preserving structural consistency, especially when compared to reference- or user-guided alternatives. As a result,

Figure 2.22: Text-guided colorization methods require users to input text prompts for color guidance [60, 101].

users frequently resort to iterative prompt tuning or elaborate descriptions to steer the output toward a desired result. A representative implementation of this pipeline combines Stable Diffusion (SD) [72] with ControlNet [101] to anchor the generation process to sketch inputs while responding to textual cues, as illustrated in Figure 2.22.

Reference-based methods transfer chromatic and stylistic cues from a designated exemplar—such as an illustration, comic panel, or photograph—to the target sketch, enabling fine-grained and context-aware colorization. These methods typically employ explicit structural alignment techniques, including convolutional neural networks (CNNs) or transformer-based correspondence modules, to establish spatial and semantic mappings between the reference and the sketch. By leveraging these cross-domain alignments, reference-based models can achieve highly accurate reproduction of textures, color palettes, and stroke styles, offering significantly greater control than both user-guided and text-prompted approaches, particularly when high-quality and structurally similar references are available. This precision makes them especially well-suited for artistic workflows demanding consistency across frames or panels, such as animation and manga production.

However, their effectiveness heavily depends on the quality and relevance of the provided reference. Discrepancies in pose, composition, or domain (e.g., differences in art style or medium) may lead to visual artifacts, including color bleeding, unnatural shading, or incomplete transfer of stylistic elements. A comprehensive discussion of reference-based colorization frameworks, including architectural design and training strategies, is presented in the following section.

Despite notable progress, particularly with diffusion-driven correspondence networks, reference-based colorization still faces a fundamental challenge: reconciling the sparse, abstract geometry of sketches with the dense spatial semantics of arbitrary references. Most prior work, therefore, presupposes tightly matched sketch–reference pairs; otherwise, color bleeding, misalignment, or loss of fine detail often occurs.

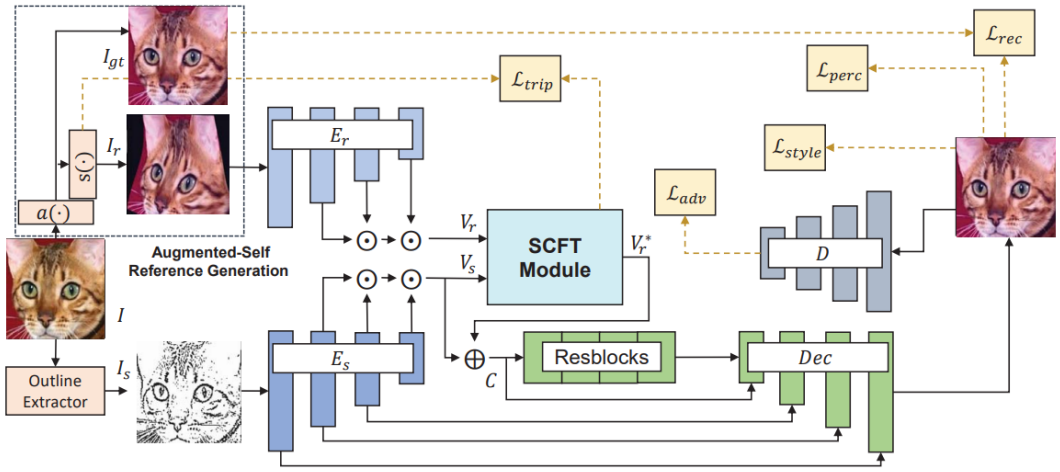


Figure 2.23: SCFT proposed in [49] The training pipeline warps the reference image and logs the control points into a spatial-matching loss, driving the network to align low-level features precisely.

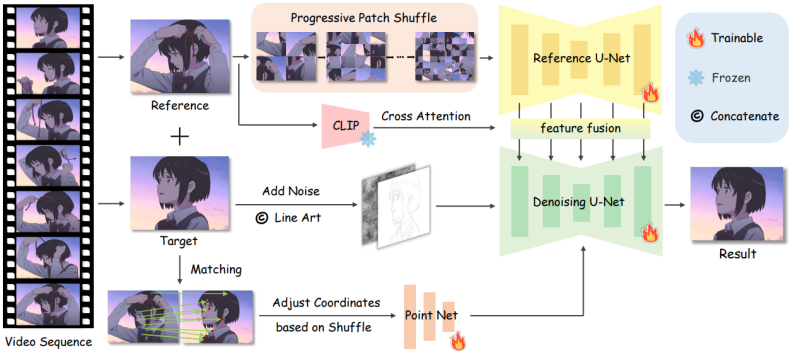


Figure 2.24: Like SCFT, MangaNinja [55] jointly trains its reference encoder, implemented as a U-Net within the overall architecture.

2.4.4 Reference-based Sketch Colorization

Reference-based sketch colorization is the fundamental utility of the systems proposed in this thesis. To better contextualize and highlight the advantages of our approach, this subsection reviews the latest baselines in detail and discusses their inherent limitations.

Since reference-based sketch colorization must transfer chromatic and textural cues from a reference image onto a line drawing, the neural backbone must first extract rich color representations from the reference and then inject them, in a structure-aware manner, into the sparse sketch domain. To achieve this, existing frameworks almost always employ two separate encoders: one for the sketch and one for the reference. Depending on whether the reference encoder's parameters are updated during training, current baselines fall into two broad categories:

- **Jointly trained reference encoder.** The reference encoder is optimized end-to-end together with the sketch encoder and the generative backbone. Most reference-based sketch colorization systems [49, 55, 57, 95, 110] adopt the second strategy. Two

Figure 2.25: Baseline methods that jointly train the reference encoder easily overfit to their training dataset and are unable to colorize unaligned input pairs. Results synthesized by [110].

canonical examples are the architectures in [49,55], illustrated in Figure 2.23 and Figure 2.24. When the inference pair (sketch, reference) is semantically and structurally close to the training distribution, these jointly trained models achieve impressive perceptual fidelity.

- **Frozen reference encoder.** The other training paradigm utilizes a pre-trained image embedder that is typically trained on large-scale vision–language objectives, such as multi-label tagging [24] or zero-shot classification with CLIP [51, 68]. Then, the reference encoder is frozen during sketch-colorization training.

All baseline reference-based methods rely on the first training paradigm and, as a result, exhibit three recurring shortcomings: reduced perceptual fidelity, weaker generalization, and limited controllability. These issues stem from overfitting inherent to reference-based sketch colorization, particularly the use of a trainable reference encoder, which co-adapts to the training pairs and degrades sharply at inference time when the sketch and reference diverge in semantics or pose. Typical failure modes include color bleeding across line boundaries, large-area desaturation, and stylistic inconsistencies, as illustrated in Figure 2.25.

In contrast to these baselines, the frameworks proposed in this thesis first demonstrate the effectiveness of the second training strategy in reference-based sketch colorization: the reference branch is a large, pre-trained image encoder kept frozen, while only the sketch encoder and generative backbone are optimized. This choice—rooted in early neural style transfer [21] and the pix2pix paradigm [37], and validated for sketch colorization by [103] though not previously applied for reference-based colorization, elevates representation transfer to higher-level representations. As a result, this strategy substantially mitigates overfitting and improves both perceptual quality and generalization. Building

on this foundation, the thesis further introduces text-based manipulation methods that enhance the controllability of the colorized outputs.

Chapter 3

Dataset and evaluation

3.1 Dataset curation

Deep-learning colorization pipelines are inherently data-hungry: both generative adversarial networks (GANs) and diffusion models (DMs) deliver convincing results only when trained on large, diverse image sets. Accordingly, this thesis explores two successive synthesis engines:

GAN-based framework (early stage). Before DMs became mainstream, GANs were the workhorse for conditional generation thanks to their fine-grained, attribute-level control. However, their notoriously unstable training dynamics—mode collapse, vanishing gradients, and delicate loss balancing—hamper scale-up. In sketch colorization, these constraints restrict GANs to figure-only tasks, so the training and validation corpus consists mainly of single-character drawings with plain backgrounds. Therefore, this framework only utilized the figure subset of Danbooru2021, which comprises 0.7 million figure-focused images as the ground truths.

DM-based framework (later stage). Diffusion models replace adversarial training with likelihood-based denoising, eliminating mode collapse and enabling near-linear scaling in both network depth and dataset complexity. Modern variants—latent diffusion, ControlNet, T2I-Adapter, and others—inject reference features via cross-attention or adapter layers at every denoising step, capturing long-range dependencies, intricate textures, and subtle color harmonies that GANs often miss. This architectural stability lets us enlarge the dataset to multi-character scenes, dynamic poses, cluttered props, and varied lighting while still achieving better generation performance.

At inference time, image-guided sketch colorization consumes a pair (sketch, reference); therefore, the network must be trained on triples (sketch, reference, ground truth). Pixel-wise losses such as MSE are meaningful only when the reference and ground-truth images are tightly aligned, which means that they should have the same identities, similar color scheme, palette, and composition. Compared to text-guided generation tasks, the requirements for semantic alignment as well as geometry misalignment between reference and ground truth make the data collection much harder to curate at scale. In practice, there are two collection strategies adopted by researchers to collect raw data:

Ground-truth–as-reference paradigm. In the first training paradigm, each ground-truth color image doubles as its own reference. During optimization, the network receives

Figure 3.1: An example of processed Danbooru data triple, comprising an extracted sketch image, accompanying classification tags, and the corresponding ground-truth color image. This thesis directly utilizes ground truths as training references.

Figure 3.2: Two representative paradigms of collecting reference-based training data. Images from the animation film *Princess Mononoke*.

a triplet (sketch, reference = ground truth, target = ground truth); the reference encoder is frozen and has been pre-trained on large vision–language corpora (see Training Strategy 1 in Section 2.4.4). Because the encoder parameters never update, the generator must learn to reproduce the reference style using only the information already embedded in the fixed feature space. To increase stylistic variety without collecting extra data, simple image-space augmentations—rotation, horizontal flip, thin-plate-spline (TPS) warping, and color jitter—are applied to create perturbed reference images from the same ground truth. This paradigm is attractive because it: (i) eliminates the need for manual reference collection, (ii) scales to millions of samples with negligible overhead, and (iii) delivers strong cross-domain generalization, as demonstrated later in the thesis.

Neighboring-frame (or panel) paradigm. The second approach harvests references from adjacent frames in an animation or consecutive panels in a manga where the same character appears (Training Strategy 2 in Section 2.4.4). Because the reference is genuinely distinct from the target frame, the network can jointly optimize the reference encoder and the generative backbone, more closely simulate some of the real-world applications. However, compiling such datasets requires access to raw animation or comic sequences—a far narrower source pool than in the first paradigm, especially for style variance—and the resulting training corpora are usually smaller and less diverse. Net-

works trained exclusively on neighboring frames tend to generalize poorly when faced with novel characters, complex textures, or non-anime sketches. Figure 3.2 contrasts these two paradigms. Throughout this thesis, I adopt the ground-truth–as-reference strategy, which streamlines data acquisition and, as later chapters show, achieves superior generalization across a wide spectrum of test conditions. Concretely, all training samples are drawn from Danbooru [12], a large-scale, multi-label dataset containing more than six million artist-created anime images. From each raw image, I derive (i) a text caption, (ii) a synthetic sketch, and (iii) the original color artwork, yielding the tri-modal training tuples required by the proposed frameworks. A processed example is illustrated in Figure 3.1.

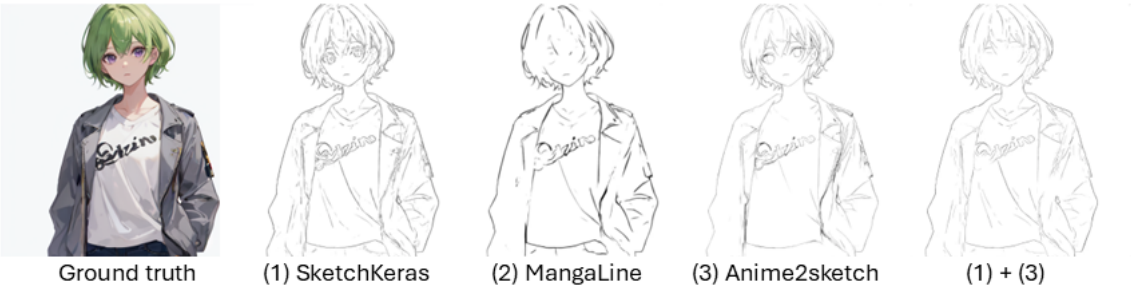


Figure 3.3: Examples of extracted sketches.

Extractor	Tool & Rationale	Key Design Notes	Typical Failure Modes
(1) [102]	Lightweight U-Net trained with perceptual + GAN losses; excels at retaining long, coherent outlines	32-channel encoder, multi-scale discriminators; trained on paired line art/photo data for 60 epochs	May miss faint interior strokes in densely shaded manga panels
(2) [50]	Specialized network that separates structure from *screentone*	Residual + skip architecture; explicit *screentone* suppression branch	Susceptible to aliasing when input resolution $<$ 512 px
(3) [94]	Open-domain adaptation GAN transfers edge detectors from photos to anime; yields noise-free, high-contrast lines	Cycle-consistent loss with domain classifier; pretrained weights available	Occasionally exaggerates line thickness on pastel backgrounds

Table 3.1: Comparison of sketch extraction tools used to produce training sketch data in the thesis.

3.2 Preprocessing techniques

Sketch extraction. Building a reliable, style-diverse line-art corpus is the foundation of our colourisation pipeline. Starting from the 4.9-million-image Danbooru2021 archive [12], I first discard low-resolution or heavily water-marked items, then run a three-stage extractor ensemble: SketchKeras [102], the Manga Structural Line CNN [50], and Anime2Sketch [94], whose complementary inductive biases collectively recover clean outer contours and delicate interior hatching. A light post-processing step merges over-segmented strokes and equalizes line widths, after which 30% of the sketches are further simplified by re-routing the outputs of SketchKeras through Anime2Sketch to enlarge style variance and reduce extractor artifacts. The resulting sketch set spans everything from minimal animation cels to cross-hatched illustrations.

To broaden the stylistic coverage of our corpus and avoid a model that only excels on densely inked drawings, I augment the dataset by re-synthesizing roughly 30 percent of the sketches with a two-stage pipeline that feeds the output of SketchKeras directly into Anime2Sketch. SketchKeras' perceptual-plus-adversarial training flattens mid-frequency texture and harmonizes stroke width, acting as an implicit low-pass filter on the source image. Anime2Sketch's open-domain adaptation GAN then eliminates residual high-frequency noise, boosts edge contrast, and produces clean contours that mimic professional inks. This simplification track not only diversifies the line-art styles—spanning

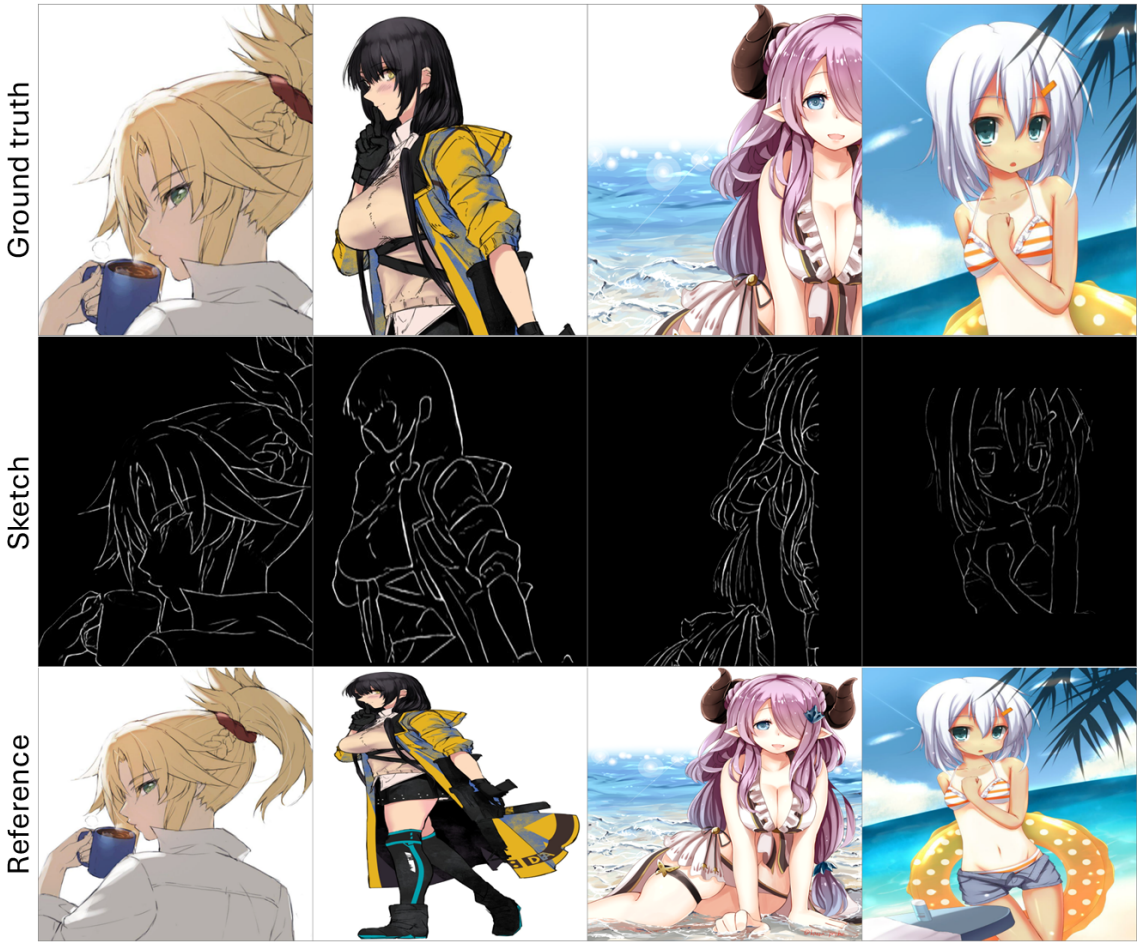


Figure 3.4: Samples of training data after on-the-fly preprocessing.

minimalist cel animation to heavy cross-hatching—but also makes the colorization network more resilient when it is asked to fill large, stroke-free regions at inference time. Examples of extracted line drawings are illustrated in Figure 3.3.

On-the-fly preprocessing. At every training step, the three inputs, reference image R, sketch S, and ground-truth color image C, are augmented independently to break trivial correspondences while still preserving the semantic fidelity of the reference:

- Spatial decoupling for (S, C). Both the sketch S and its ground-truth color target C are first isotropically rescaled by a random factor uniformly sampled from [1.0, 1.3], and are then cropped at a random spatial offset so that each retains the fixed training window chosen afresh for every mini-batch. This stochastic resize–crop routine intentionally de-registers low-level pixel layouts between S and C, compelling the network to discover semantic correspondences (shape, contour, tone) rather than relying on pixel-aligned shortcuts.

- Semantics-preserving resize for R. In contrast, the reference image is only resized (no cropping, flipping, or rotation), as such transformations are observed to degrade the similarity between colorized results and references during inference. By keeping the full composition and context intact, I guarantee that R continues to provide an unambiguous stylistic exemplar, allowing the network to focus on transferring global

palette and texture cues instead of recovering lost geometry. Since the reference embedder is pre-trained for image understanding, such deformation would not influence the extracted embeddings.

- Random cropping for S. To endow the generator with genuine outpainting capability, i.e., the skill to hallucinate plausible content beyond the strokes that are actually provided, I apply an asymmetric border-crop augmentation to every sketch during training. Concretely, for each iteration I draw four independent offsets $\Delta_{t,b,l,r} \sim \mathcal{U}(0, 0.15)$ and remove that proportion from the top t, bottom b, left l, and right r margins, respectively. Only the sketch is cropped; the paired color ground truth remains intact. The resulting input therefore contains an inner region with normal guidance and an outer "sketch-free halo." After cropping, the sketch is re-padded to the original canvas size with zeros, ensuring dimensional consistency. This simple operation serves three intertwined purposes: (i) it exposes the network to partial-guidance conditions that mimic the inference-time outpainting scenario; (ii) it discourages boundary-locked convolutions, thereby reducing ringing artifacts along image edges; and (iii) it enlarges the effective receptive field without increasing model depth, because the generator must propagate semantic cues from the central strokes outward to the unguided margins. A corresponding ablation study will be introduced in Section 5.6.

- **Line–polarity inversion for S.** Building on the empirical observation of [101] that line–art encoders achieve stronger activations when stroke energy resides in the *high-intensity* channel, I invert every input sketch from the conventional *black-on-white* (0-line, 1-background) representation to a *white-on-black* format. This polarity swap accentuates edge contrast, dampens low-level convolutional noise, and consistently accelerates training convergence. In pilot experiments, however, I found that mapping line pixels to $+1$ and background pixels to -1 (rather than the standard $\{0, 1\}$ range) further improves both outpainting fidelity and downstream segmentation accuracy. Accordingly, all sketches are rescaled from the original domain [-1 (lines), 1 (backgrounds)] to the inverted domain [-1 (backgrounds), 1(lines)], ensuring maximal edge saliency for subsequent encoder stages.

The resulting asymmetric augmentation pipeline delivers millions of stylistically diverse yet semantically faithful (Reference, Sketch, Color) triplets—an essential ingredient for training the high-capacity diffusion and GAN backbones analyzed in subsequent chapters. Representative augmented samples are illustrated in Figure 3.4.

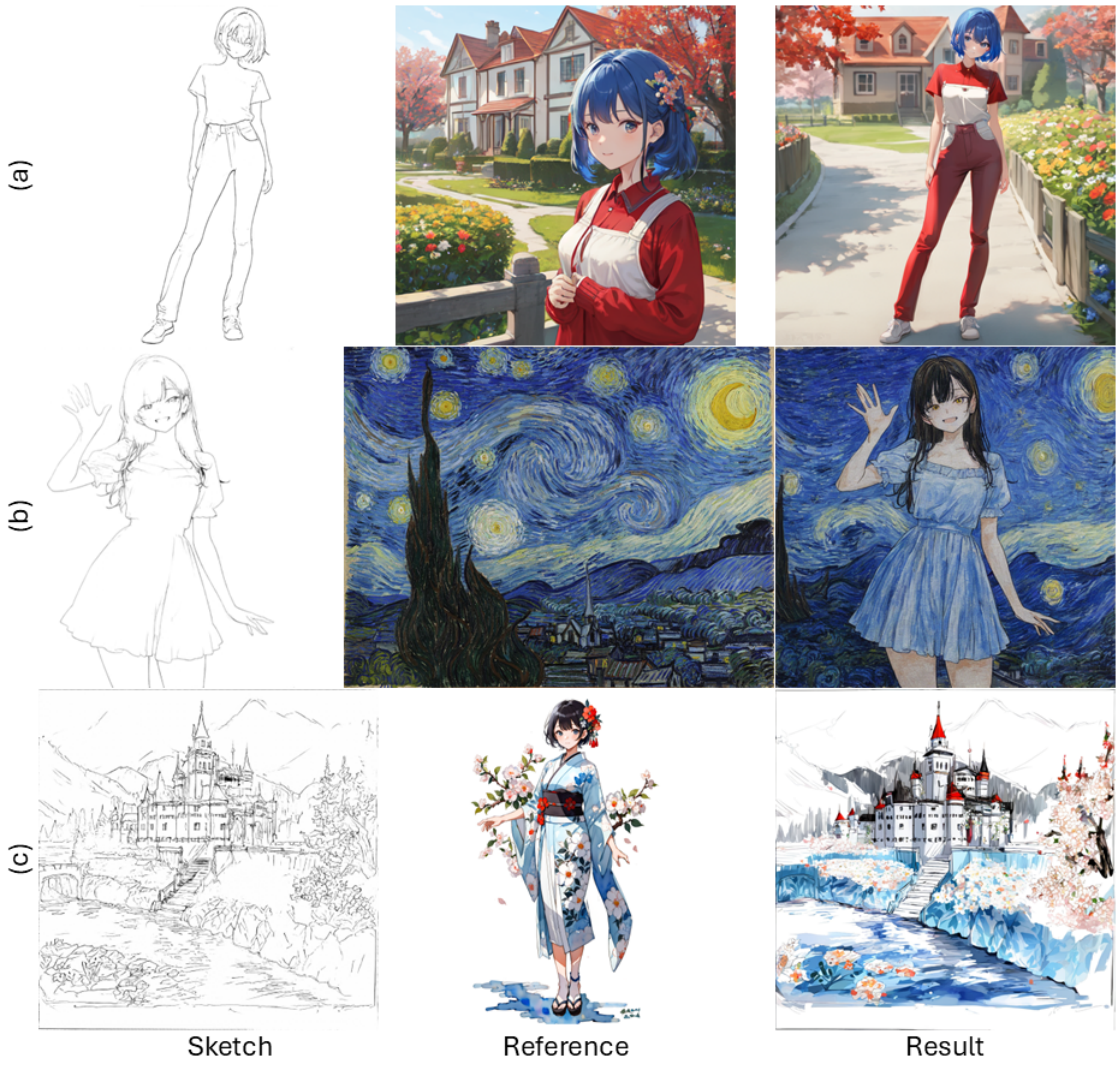


Figure 3.5: Reference-based colorized results should follow the sketch segmentation while effectively transferring textures/strokes/colors from the reference.

3.3 Evaluation protocol

Compared with other conditional generation tasks, reference-based sketch colorization still lacks an objective metric that can reliably measure similarity when the validation pair—*input sketch* and *color reference*—is semantically unrelated. As a consequence, evaluation currently depends heavily on human judgment, supported by a few domain-aware quantitative scores. In this thesis, the assessment is organized around three complementary aspects.

3.3.1 Qualitative evaluation (primary)

Reference images provide complete information about hue, texture, and stroke style, whereas text- or user-guided methods offer only coarse hints. Because mismatches become obvious at a glance, visual inspection remains the most important yardstick. Three dimensions are checked:

1. Visual performance: Are the colorized results pleasant to the eye and free of artifacts?

Typical failures in image-guided colorization appear as color bleeding/spray/additional body parts outside the line art or incomplete color synthesis when the sketch/reference gap is large.

2. Similarity with the reference: Do the generated colors, textures, and strokes closely follow the given reference image? This dimension requires the colorization system not only transfers color information, but also able to transfer style details, which significantly in-creases the generalization ability of the proposed framework, making it able to colorize sketches with various styles, even when the networks are only trained with anime-style images.

3. Semantic fidelity to the sketch: Does the network respect the geometry implied by the sparse line art? Maintaining this fidelity is particularly challenging for models trained with "ground truth as reference" shortcuts and a frozen, pre-trained embedder. Different from the training paradigm 2 introduced in Section 3.1.

To facilitate and visualize the criteria for successful reference-based sketch colorization, Figure 30 presents three illustrative examples. Each row highlights key factors that define whether a generated result effectively reflects the intended reference while preserving the structure of the input sketch.

In row (a), both the sketch and the reference depict a single female character. The sketch omits background information, while the reference contains a complete outdoor scene. A successful colorization in this case requires the system to faithfully transfer key visual attributes—such as the character's blue hair, blue eyes, and red-and-white outfit—while also generating a plausible background that stylistically aligns with the reference image. This demonstrates the model's ability to extend beyond local color transfer and infer missing contextual information in a visually coherent way.

Row (b) emphasizes the importance of style-aware but structure-preserving transfer. Here, the sketch depicts a girl, while the reference is Vincent van Gogh's *The Starry Night*, featuring vivid colors and heavily textured brushstrokes. An ideal result would retain the sketch's structure—particularly facial features and anatomy—while adapting the reference's stylistic elements (e.g., swirling strokes, blue–yellow palette) in a restrained, contextually appropriate manner. However, in this case, the result suffers from severe over-transfer of texture, especially in the face and hair, leading to noticeable artifacts. This highlights a common failure mode where excessive stylistic transfer overrides structural fidelity, violating the sketch's intended form.

In row (c), the sketch and reference depict semantically different subjects: a castle landscape and a traditionally dressed female character, respectively. Despite this mismatch, the successful result reflects key stylistic elements from the reference—such as the ink-wash textures, floral motifs, and red-blue-black color palette—while preserving the spatial structure and identity of the original sketch. This demonstrates a key strength of reference-based colorization: the ability to abstract and recontextualize visual style, even when the source and target domains are not directly aligned.

Together, these examples outline the standards for high-quality reference-based sketch colorization: faithful structure preservation, semantically consistent style transfer, robustness to missing information, and restraint in applying textures or patterns to avoid artifacts.

3.3.2 Quantitative evaluation (supporting)

To complement qualitative evaluations, this thesis reports several automatic scores. Their roles and limitations are summarized below so that subsequent numerical results can be interpreted correctly.

1. Fréchet Inception Distance - primary metric

Fréchet Inception Distance (FID) has become the de facto standard for benchmarking the perceptual quality of images synthesized by generative models. Introduced by Heusel et al. [27], FID quantifies how closely the distribution of synthesized images X_g matches the distribution of real (ground-truth) images X_r. FID embeds both the generated images and a set of real images into the feature space of the pre-trained Inception-v3 network [87], specifically, the output features of the layer *2048-d pool*$_3$. Producing feature sets $F_g = \{\mathrm{f}_g^i\}_{i=1}^{N_g}$ and $F_r = \{\mathrm{f}_r^i\}_{i=1}^{N_r}$ for generated images and real images, respectively. These features are then empirically modeled as multivariate Gaussians

$$\begin{gathered} \mathcal{N}(\mu_g, \Sigma_g) \quad \text{and} \quad \mathcal{N}(\mu_r, \Sigma_r) \quad \text{with} \\ \mu_* = \frac{1}{N_*}\Sigma \mathrm{f}_*^i, \quad \Sigma_* = \frac{1}{N_* - 1}\Sigma(\mathrm{f}_*^i - \mu_*)(\mathrm{f}_*^i - \mu_*)^\top, \end{gathered} \tag{3.1}$$

and then computes the Fréchet (2-Wasserstein) distance between these Gaussians as

$$\mathrm{FID}(X_g, X_r) = \|\mu_g - \mu_r\|_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}} \tag{3.2}$$

Where $\mathrm{Tr}(\cdot)$ is the matrix trace and $(\Sigma_r\Sigma_g)^{\frac{1}{2}}$ is the principle square root. Lower values therefore indicate that the synthetic distribution is statistically closer to the real one in terms of high-level semantics captured by the network.

Because FID summarizes an entire distribution, it cannot assess one-to-one correspondence between a generated image and a specific reference; however, it is sensitive to both mode dropping and excessive artifacts, making it a reliable proxy for overall fidelity and diversity. Competing metrics such as the Inception Score (IS) [6] and Perceptual Path Length [41] use class-posterior statistics that were tuned on photographic ImageNet data. When applied to anime or line-art domains, these networks misinterpret stylistic cues as "noise," inflating or deflating scores unpredictably. FID's reliance on feature covariances rather than classification logits reduces—but does not eliminate—this domain bias. In practice, FID remains preferred in the sketch-to-colorization literature precisely because it balances robustness with computational efficiency while avoiding the unreliable rankings produced by IS in non-photographic domains.

2. Peak Signal-to-Noise Ratio (PSNR) - secondary metric

PSNR is a classical, full-reference fidelity metric that quantifies pixel-wise reconstruction accuracy between a synthesized image I_g and its ground truth counterpart I_r. It is derived directly from the mean-squared error (MSE):

$$\mathrm{MSE}(I_g, I_r) = \frac{1}{HWC}\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C}(I_g(h,w,c) - I_r(h,w,c))^2, \tag{3.3}$$

where H, W, C denote the height, width, and channel count of images. Because the logarithm rescales the ratio, higher PSNR values indicate lower distortion, i.e., synthetic images whose pixel intensities more closely match the reference ground truth.

However, PSNR isn't reliable as FID for reference-based sketch colorization due to:

- Necessity of an aligned ground truth. Image-guided sketch colorization often draws colors from arbitrary reference images rather than a single canonical target. Because PSNR can be computed only when each generated image has an exact, pixel-registered ground truth, it breaks down whenever references are chosen freely or when multiple "correct" colorizations are possible.

- Penalizing stylistic detail. Fine-grained strokes, hatching, and textured fills—hallmarks of professional line-art—introduce high-frequency variations that PSNR interprets as noise. Models that faithfully reproduce these artistic details may paradoxically receive lower PSNR than blurrier outputs, despite being preferred by human evaluators.

- Weak correlation with perceptual quality. Small spatial misalignments (e.g., a one-pixel shift) can drastically reduce PSNR even when images look identical to the eye, while over-smoothed images can earn high PSNR. This mismatch is especially pronounced in stylized or non-photographic domains where color harmony, brushwork, and stroke integrity outweigh raw pixel coincidence.

- Incompatibility with diverse-reference evaluation protocols. Modern studies often report distribution-level or perceptual metrics (e.g., FID, IS) that thrive without paired data. PSNR cannot participate in such protocols, limiting cross-paper comparability and making it an unreliable benchmark for ablation studies in reference-based pipelines.

In short, although PSNR offers a mathematically simple gauge of pixel accuracy, its reliance on a single aligned ground truth and its tendency to down-score richly detailed outputs make it an inferior choice for evaluating the creative, reference-driven results typical of sketch colorization systems.

3. Structural similarity index measure (SSIM) - secondary metric

SSIM was introduced by Wang et al. [90] to quantify perceptual similarity between a synthesized image I_g and a ground-truth reference image I_r. Instead of comparing raw pixel errors, SSIM decomposes quality into luminance l, contrast c, and structure s components, each computed over small, sliding windows (typically 11×11 Gaussian-weighted patches):

$$\begin{aligned} l(I_g, I_r) &= \frac{2\mu_g\mu_r + C_1}{\mu_g^2 + \mu_r^2 + C_1}, \\ c(I_g, I_r) &= \frac{2\sigma_g\sigma_r + C_2}{\sigma_g^2 + \sigma_r^2 + C_2}, \\ s(I_g, I_r) &= \frac{\sigma_{gr} + C_2/2}{\sigma_g + \sigma_r + C_2/2}, \end{aligned} \tag{3.4}$$

where μ_* and σ_* are local means and standard deviations, σ_{gr} is the cross-covariance,

and C_1, C_2 are small stabilizing constants proportional to I_{max}^2. The per-patch SSIM is

$$\text{SSIM} = l^\alpha c^\beta s^\gamma, \tag{3.5}$$

with exponents usually set to $\alpha = \beta = \gamma = 1$. A final image-level score is obtained by averaging SSIM over all patches; values range from -1 to 1, with 1 denoting a perfect structural match. Similarly, SSIM is a poor fit for reference-based sketch colorization because of the following points:

- Requires an aligned ground truth. Like PSNR, SSIM can only be computed when every generated image has a pixel-registered target. In image-guided colorization, references are often freely chosen and may enable multiple equally valid colorizations, leaving no single "correct" counterpart for SSIM to compare against.
- Penalizes stylistic detail. Fine cross-hatching, line-weight variation, and stippling increase local variance, triggering lower contrast/structure scores even though these details are artistically desirable. Blurred outputs that suppress stroke detail can actually raise SSIM, reversing intuitive quality rankings.

4. CLIP cosine similarity (CLIP Score) - secondary metric

CLIP score is a reference-based perceptual metric that evaluates how well a synthesized image semantically aligns with a given text or visual condition. It leverages the Contrastive Language–Image Pretraining (CLIP) model, formerly introduced in Section 2.1.5, a large-scale vision-language embedding system trained to align images and texts in a shared latent space. In the context of image-guided sketch colorization, CLIP score is often computed between the generated image and its reference image, both encoded via the CLIP image encoder.

Formally, let I_g be the generated image and I_r the reference ground truth image. Their CLIP embeddings are denoted:

$$\mathrm{v}_g = \text{CLIP}_{\text{img}}(I_g) \qquad \mathrm{v}_r = \text{CLIP}_{\text{img}}(I_r), \tag{3.6}$$

which are normalized to unit vectors. The CLIP score is then defined as the cosine similarity:

$$\text{CLIP} - \text{Score}(I_g, I_r) = \frac{\mathrm{v}_g \mathrm{v}_r}{\|\mathrm{v}_g\| \|\mathrm{v}_r\|} \in [-1, 1] \tag{3.7}$$

A higher score indicates that the synthesized image captures similar semantic content and style to the reference, as understood by the CLIP model.

Though CLIP score offers the advantage of not requiring a paired ground-truth image, it remains an imperfect metric for evaluating reference-based sketch colorization, primarily due to its limited sensitivity to modality-specific visual factors. CLIP embeddings are trained to capture high-level semantic alignment—such as object categories, scene types, or general style—but are relatively agnostic to fine-grained visual attributes like local geometry, color accuracy, texture fidelity, or stroke integrity. As a result, the CLIP score cannot reliably disentangle structural semantics (e.g., pose or layout) from color, texture, or artistic stylization.

This becomes problematic in the context of sketch colorization, where the goal is to faithfully transfer color and style from a reference image to a structurally different sketch. If the model overfits to the reference and copies large regions of the reference image—including its shapes, contours, or lighting—into the output, it may receive a higher CLIP score due to improved global similarity, even though the result violates the structural constraints of the input sketch and introduces visible artifacts (e.g., background leakage, mismatched edges, or texture bleeding). In other words, a higher CLIP score can perversely reward outputs that are less faithful to the input sketch and more similar to the reference, undermining the intended role of sketch guidance.

Consequently, using CLIP loss or score as a training signal or evaluation metric in sketch colorization can introduce a misalignment between optimization objectives and visual quality. It encourages reference mimicry over structural fidelity, leading to visually inconsistent or semantically implausible results, especially in domains like anime illustration, where local consistency, clean edges, and detailed line preservation are essential.

Chapter 4

Generative adversarial networks framework

4.1 Overview

Before introducing the diffusion model (DM)-based framework, this thesis begins with an earlier stage of research that explores a GAN-based approach to reference-based sketch colorization. This foundational work lays the groundwork for multi-modal control in the colorization process, enabling both visual fidelity and semantic flexibility.

The system is designed to take a line-art sketch and a reference image as input, where the reference serves as a source of stylistic cues—such as color palette, texture, and tone. The generative backbone, built upon a conditional GAN architecture, learns to synthesize a fully colorized output that reflects the visual semantics of the reference while preserving the structural layout of the sketch. To further enhance controllability and user interaction, the framework integrates a multi-label classification branch, which associates each generated image with a high-dimensional semantic tag vector derived from a curated set of 6,000 attributes (e.g., "blonde hair," "yellow eyes," "blue sky," "red skirt"). These tags are automatically predicted and can be edited by users during inference, allowing fine-grained semantic adjustment of the generated results without requiring paired ground truth or manual recoloring.

This design supports a two-stage workflow for the proposed generative backbone: first, users generate a base colorization result using any reference image; second, they can refine or steer the output by modifying the associated textual tags, effectively guiding the model toward semantically richer or contextually appropriate renderings. This pipeline demonstrates the feasibility of combining visual guidance and text-based semantic control within a unified generative framework. Yet, before the optimization of the generative backbone, a pre-training for the reference encoder, a ResNet-34, is necessary to align its feature space with that of anime-style images.

A qualitative example from the proposed GAN-based colorization system is presented in Figure 4.1. It showcases the model's ability to harmonize structural fidelity with stylistic transfer, as well as the potential for interactive user refinement via tag manipulation. These early results underscore the importance of multi-modal conditioning mechanisms, which later evolve into the more expressive and controllable diffusion-based architecture

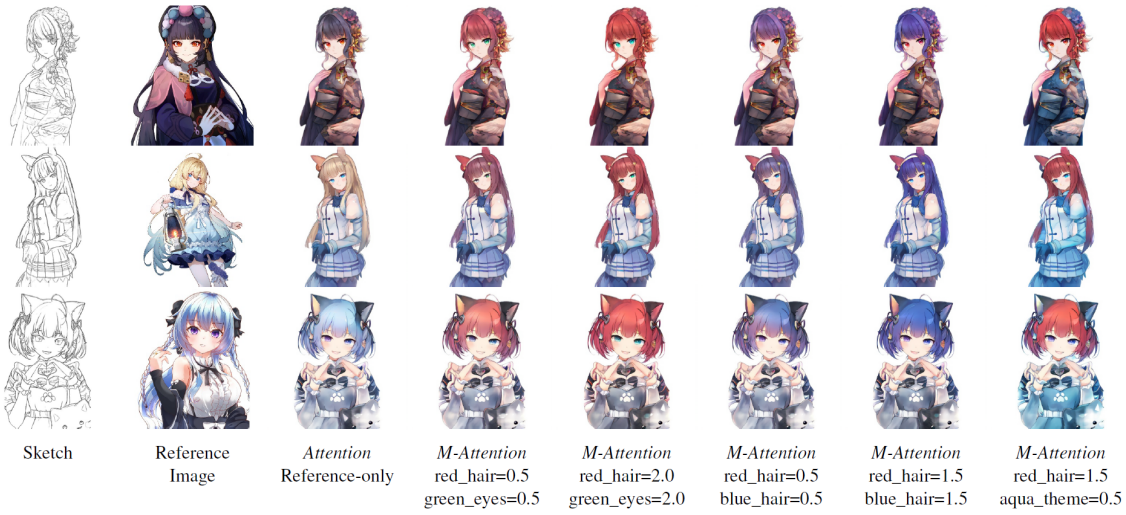


Figure 4.1: The proposed framework is capable of colorizing sketch images using reference images. Then, users can further edit the colorized results using text tags with an interpolation scale.

described in the subsequent chapters.

This chapter begins by introducing reference-based training, detailing the associated architectures and loss functions. It then presents the underlying principle of tag-based manipulation and explains how it is implemented within the proposed framework.

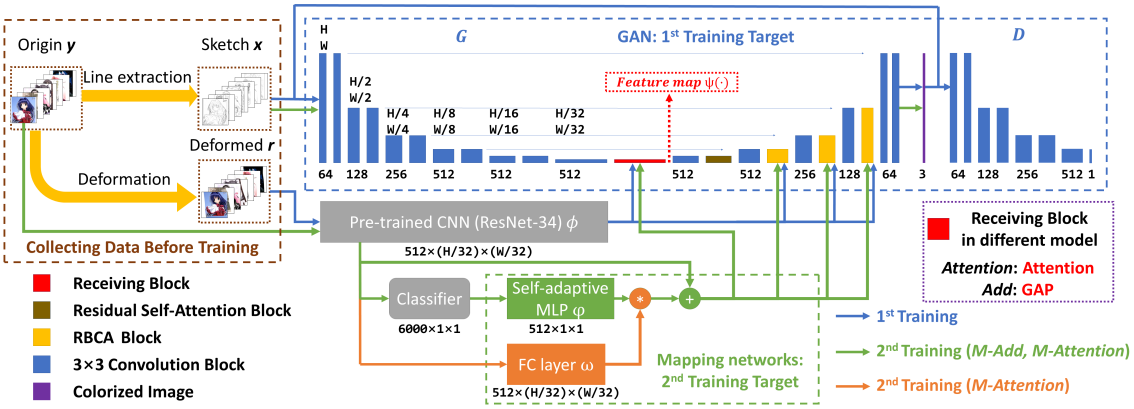


Figure 4.2: Pipeline of the proposed GAN-based framework, which comprises two training stages. In the GAN-based framework, the reference inputs are randomly deformed to create spatial misalignment.

4.2 Reference-based sketch colorization

This framework involves three distinct optimization stages. The first stage is the pre-training of the reference encoder, which is initially trained for multi-label classification on anime-style imagery and then frozen throughout the generative training process to serve as a stable, domain-adapted feature extractor. The subsequent two stages pertain to the generative training pipeline, which is divided into two phases to facilitate progressive learning.

The first-stage generative training focuses on **reference-based sketch colorization**. Its primary goal is to establish a semantically meaningful and stable latent space, while simultaneously enabling the generator to perform effective color transfer from the reference image to the sketch input. During this phase, **a conditional U-Net generator** is trained alongside **a CNN-based discriminator** under an adversarial learning framework.

The second-stage generative training focuses on **tag-based manipulation**, enabling semantic editing of the generated images by interpolating along interpretable directions in the latent spaces. This is achieved through **latent interpolation** in two aligned domains: (1) the **reference feature space**, extracted by the pre-trained reference encoder, and (2) the **latent representation** space of the GAN generator. By performing controlled interpolations guided by target tag embeddings, this stage empowers the model to adjust specific visual attributes—such as hair color, clothing style, or global color scheme—based on user-specified tag modifications.

This section will introduce these training stages in sequence with a detailed explanation for each design, with the full training pipeline visualized in Figure 4.2.

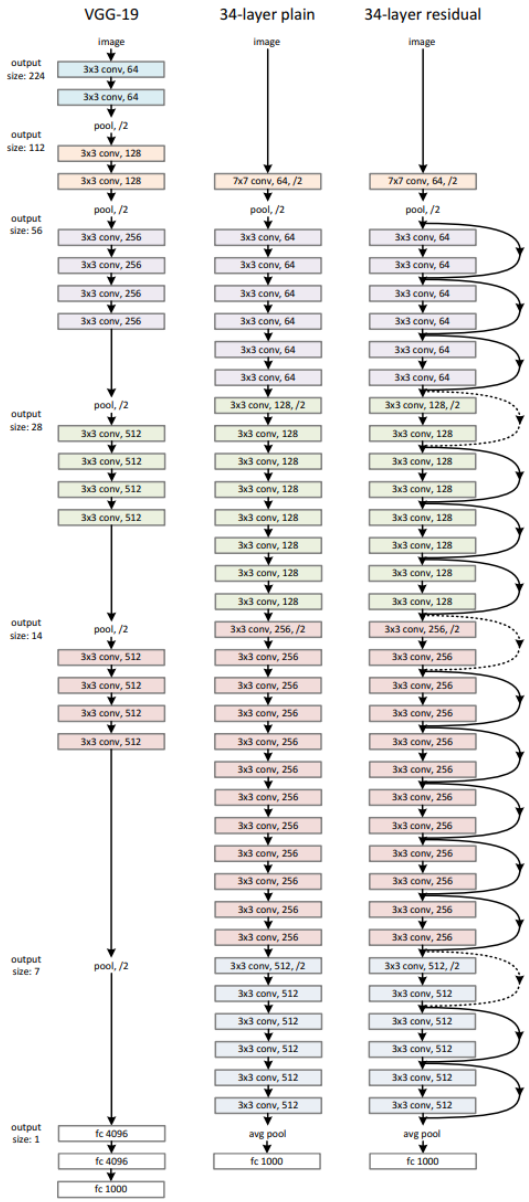


Figure 4.3: The adopted reference encoder, ResNet-34 [24]. During fine-tuning, the last *fc* layer is adapted for 6000 classes prediction.

4.2.1 Reference embedder selection and pre-training

As discussed in Section 2.4.4 and Section 3.1, this thesis focuses on a training paradigm that leverages ground-truth color images as references, along with a pre-trained and frozen reference encoder, to construct a robust colorization system.

Within this framework, I adopt ResNet-34 [24] as the reference encoder. Originally pre-trained on the ImageNet dataset for single-label classification tasks, ResNet-34 is selected for its favorable balance between model complexity and feature representation capability. As illustrated in Figure 4.3, the model is repurposed to extract multi-label semantic features corresponding to style- and content-related tags from anime-style imagery. Preliminary experiments indicate that ResNet-34 demonstrates superior performance in capturing col-or-relevant visual semantics when compared to deeper networks like ResNet-

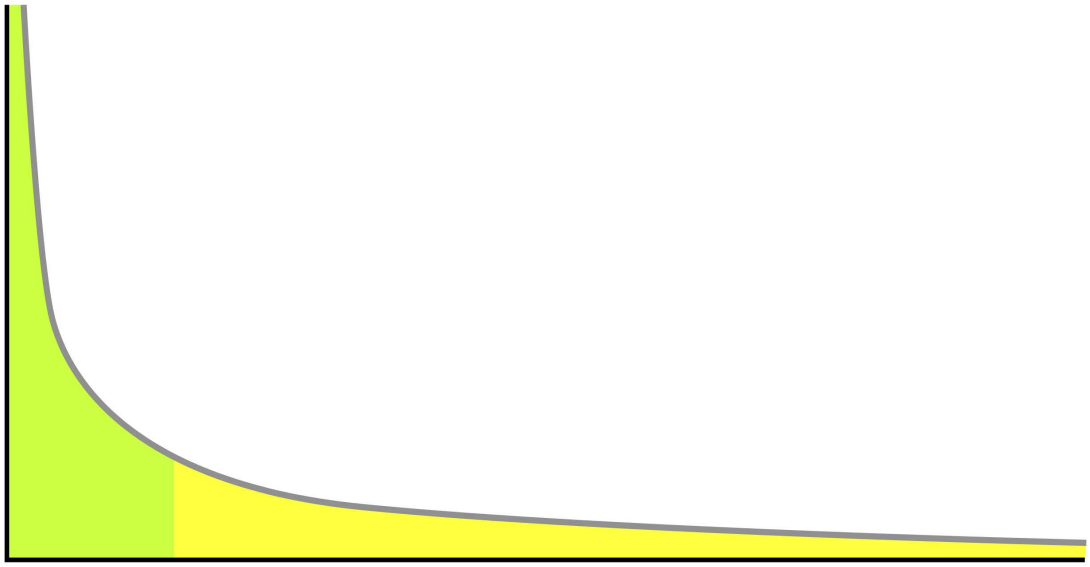


Figure 4.4: Long-tailed distribution, which indicates that the frequencies of head tags are much higher than those of tail tags

50/152 or channel-attention based alternatives such as Squeeze-and-Excitation Networks (Se-Net) [32].

Given that the reference encoder is frozen during the training of the generative backbone, it is crucial to mitigate the domain shift between ImageNet features and anime-style imagery. To this end, I conduct a domain-specific fine-tuning phase, transforming the encoder into a multi-label classifier trained on a curated tag dataset from anime illustrations. However, a key challenge arises from the long-tailed tag distribution, in which a small number of frequent tags dominate the dataset while the vast majority of tail tags are underrepresented (Figure 34). This imbalance results in the model's bias toward head classes and significantly degrades its ability to extract discriminative features for rare tags.

To address this, I reformulate the learning objective using both the standard Binary Cross Entropy (BCE) loss and the Focal Loss [52] to balance the gradients during optimization. For a sample with ground-truth tag vector $\boldsymbol{y} \in \{0,1\}^C$ and predicted probabilities $\hat{\boldsymbol{y}} \in [0,1]^C$ where C is the total number of tags, the multi-label BCE loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\sum_{i=1}^{C}[y_i \log(\hat{y_i}) + (1-y_i)\log(1-\hat{y_i})] \tag{4.1}$$

To mitigate the dominance of head tags, I augment the loss with Focal Loss, which down-weights the easy examples and focuses learning on hard, misclassified samples. The focal loss for each tag is defined as:

$$\mathcal{L}_{\text{focal}} = -\sum_{i=1}^{C}[\alpha_i(1-\hat{y_i})^{\gamma} y_i \log(\hat{y_i}) + (1-\alpha_i)\hat{y_i}^{\gamma}(1-y_i)\log(1-\hat{y_i})] \tag{4.2}$$

where $\alpha_i \in [0,1]$ is the weighting factor for class i, and γ is the focusing parameter that controls the degree of down-weighting on well-classified examples. In this thesis, I empirically set $\gamma = 2$ and α_i inversely proportional to class frequency to emphasize tail

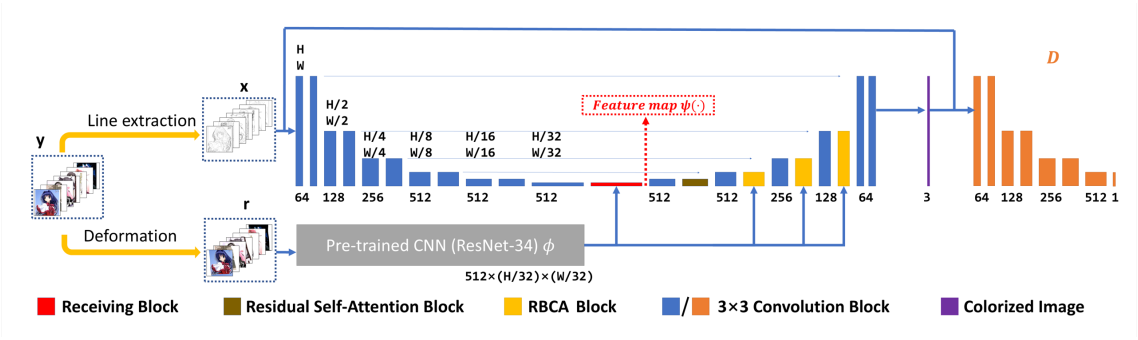


Figure 4.5: The generative backbone involved in the first-stage GAN training.

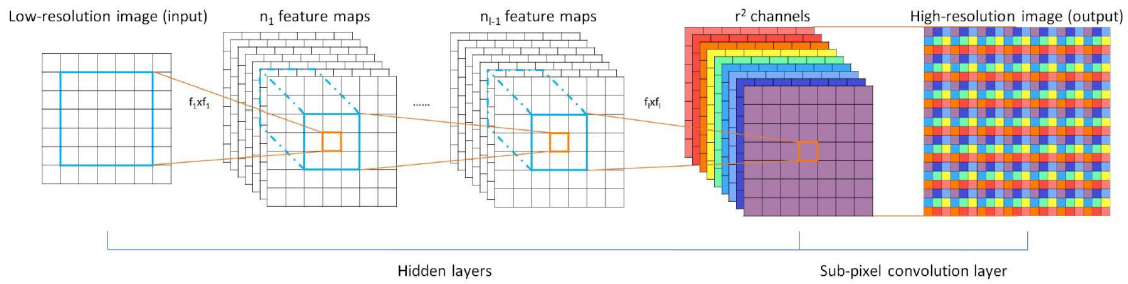


Figure 4.6: Visualization of sub-pixel, an upsampling layer used to replace vanilla transpose convolutional layers to interpolate features from low resolution to higher resolutions [78].

classes.

By combining both BCE and focal losses during fine-tuning, the ResNet-34 is able to be trained for more balanced and semantically rich representations across the full spectrum of tags, significantly enhancing the downstream performance of the colorization network, especially when dealing with rare or stylistically unique references.

The fine-tuning is conducted on a curated subset of the Danbooru dataset, comprising approximately 855,876 anime-style figure images annotated with corresponding multi-label tags, divided into 766,454 triples for training and 89,422 for validation, respectively. This large-scale tag supervision provides rich semantic guidance for adapting the reference encoder to the visual characteristics of anime artwork.

4.2.2 Architecture and loss

We start from introducing the reference-based sketch colorization training and related modules, which are visualized in Figure 4.5. At this step, the U-Net generator and the CNN-based discriminator are optimized for image-conditioned colorization. Following the experience from [37], the discriminator CNN outputs patch-wise feature maps instead of a single scalar.

To improve the generation performance and overcome the limitations of existing methods available at the time, several key architectural innovations are introduced in the design of the proposed framework:

1. **Pixel-shuffle upsampling.** Instead of using simple transpose convolutional layers for upsampling, which are known to produce artifacts such as checkerboard patterns,

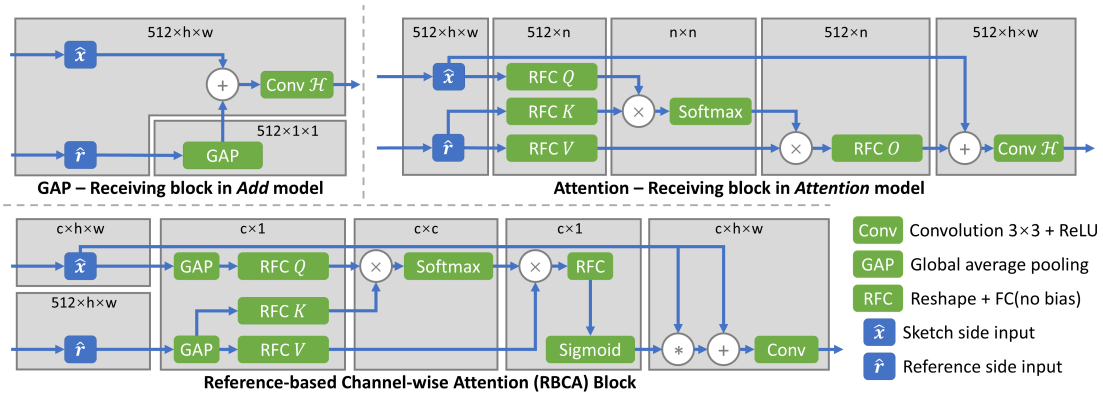


Figure 4.7: Illustration of the adopted spatial attention and proposed reference-based channel-wise attention (RBCA). To demonstrate the effectiveness of the adopted spatial attention, as well as the proposed RBCA blocks, baseline models without these modules were trained for an ablation study in the following subsection.

the generator adopts Pixel-Shuffle operations [78]. Originally introduced for image super-resolution tasks, pixel-shuffle rearranges feature maps along the channel and spatial dimensions to achieve smoother and more precise upsampling. This modification significantly improves the spatial fidelity and continuity of the generated images. The redesigned upsampling strategy is visualized in Figure 4.6.

2. **Spatial Attention Mechanism and Reference-Based Channel-Wise Attention (RBCA).** To strengthen the alignment between the sketch input and the reference image, the generator incorporates a spatial attention mechanism that serves as a receiving block and is highlighted as a red block in Figure 4.5, which selectively emphasizes visually relevant regions during the fusion of reference features into the generation stream. Additionally, a novel Reference-Based Channel-Wise Attention (RBCA) block is proposed. Pipelines of both blocks are visualized in Figure 4.7. Unlike conventional attention mechanisms that operate solely within the generative stream, RBCA dynamically reweights the generator's intermediate feature channels using global statistics derived from the reference features. This enables the network to enhance the influence of semantically important channels that carry color, texture, and style information from the reference, thereby improving visual coherence and style transfer fidelity. Though utilizing spatial attention to transfer cross-modalities embeddings has become a common practice in recent architectures and more likely to be called "cross-attention," it's still novel when the framework was proposed.

3. **Carefully Fine-Tuned ResNet-34 as Reference Encoder.** As introduced in Section 4.2.1. The reference encoder plays a crucial role in extracting semantic features from the reference image. In contrast to previous works that utilize encoders pre-trained on ImageNet or jointly trained in an end-to-end fashion with the generator, this framework adopts a carefully fine-tuned ResNet-34 encoder, specifically adapted for anime-style multi-label classification. As detailed in Section 3.1, the training paradigm 1 requires the reference embedder to be frozen during the reference-based colorization training. Therefore, this model is first fine-tuned on a large-scale anime illustration dataset to align its distribution with anime-style im-

ages and then frozen during generative training to ensure stability and semantic consistency in reference feature extraction. This design choice results in stronger feature representations for style and color, contributing significantly to the quality of the final colorization outputs.

To maximize both realism and faithfulness in the first-stage sketch-to-color training, we jointly optimize a composite objective that blends adversarial learning with pixel-wise supervision and a smoothness prior. Concretely, the total loss is formulated as

$$\mathcal{L}_{\text{total}} = \lambda_{\text{cGAN}}\mathcal{L}_{\text{cGAN}} + \lambda_{\text{L1}}\mathcal{L}_{\text{L1}} + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}} \tag{4.3}$$

where each term plays a complementary role:

1) **Conditional GAN (cGAN) loss** $\mathcal{L}_{\text{cGAN}}$
Following the cGAN paradigm [37], the discriminator D receives the input sketch x and judges whether the accompanying color image is real y or generated $G(x, \phi(r)))$. The generator G is encouraged to fool D while exploiting the latent reference code $\phi(r)$ to introduce inter-example color variance. Formally,

$$\mathcal{L}_{\text{cGAN}} = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,r}[\log(1 - D(x, G(x, \phi(r))))]. \tag{4.4}$$

This term shapes the global color and texture distribution so that generated images are indistinguishable from artist-drawn references.

2) **Pixel-level reconstruction loss** $\mathcal{L}_{\text{L}_1}$
A per-pixel ℓ_1 distance directly penalizes deviations between the generated image and its ground truth,

$$\mathcal{L}_1(G) = \mathbb{E}_{x,y,r}[\|y - G(x, \phi(r))\|_1] \tag{4.5}$$

This term anchors the colorization to the precise line structure of the sketch, stabilizes adversarial training, and preserves fine details that are easily lost when relying on perceptual losses alone.

3) **Total variation loss** $\mathcal{L}_{\text{tv}}$
To suppress high-resolution artifacts such as checkerboard patterns, we adopt the isotropic total-variation regularizer used in earlier colorization works [39, 100]:

$$\mathcal{L}_{tv}(G) = \sum_{i,j}(\|G_{i+1,j} - G_{i,j}\|^2 + \|G_{i,j+1} - G_{i,j}\|^2)^{\frac{\eta}{2}} \tag{4.6}$$

where $G_{i,j}$ indicates the (i, j)-th element of the output RGB image $G(x, \phi(r))$, $\eta = 1$ empirically. This smoothness term encourages locally coherent color transitions while preserving edges, resulting in clean and artifact-free colorization.

Since the network design is extremely important for stabilizing the GAN training. A full parameter setting of the proposed generative model is given in Table 4.1.

4.2.3 Discussion and experiments on reference embedder

In the GAN-based framework, two widely used losses for GAN-based generation in style transfer tasks, perceptual loss [39] and cycle consistency loss [109], were discarded due to

Table 4.1: Layer specifications of the proposed architecture.

Layers	Size^2	Stride	Input dim	Output dims
Encoder				
Conv + Conv	3	1 + 2	3	64 + 64
Conv + Conv	3	1 + 2	64	128 + 128
Conv + Conv	3	1 + 2	128	256 + 256
Conv + Conv	3	1 + 2	512	512 + 512
Conv + Conv	3	1 + 2	512	512 + 512
Intermediate layers				
Conv	3	1	512	512
Attention + Conv	3	1	sketch: 512 reference: 512	512
Decoder				
Conv + Upsample + Self-Attention	3	1 + 2 + 1	512	2048 + 512 + 512
Conv + Upsample + RBCA	3	1 + 2 + 1	1024	2048 + 512 + 512
Conv + Upsample + RBCA	3	1 + 2 + 1	1024	1024 + 256 + 256
Conv + Upsample + RBCA	3	1 + 2 + 1	512	512 + 128 + 128
Conv + Upsample + Conv + Tanh	3	1 + 2 + 1	256	256 + 64 + 3

their ineffectiveness in generating natural colors in sketch colorization. Since pixel-level restriction is necessary when training a sketch colorization network, pixel-level correspondence between (sketch, color) pairs and semantic similarity between (reference, color) pairs are required to establish the training. As mentioned at Section 3.1, the training reference images r were originally sourced from the ground truths y by applying deformation/transformation $\mathcal{H}$, so we can re-write Eq. 4.5 as:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\|y - G(\boldsymbol{x}, \phi(\mathcal{H}(y)))\|_1] \tag{4.7}$$

If G and ϕ are jointly trained, they can be viewed as a united generator G', and the optimization becomes the following process:

$$\arg\min_{G'} \mathcal{L}_{L1}(G') = \mathbb{E}_{x,y}[\|\boldsymbol{y} - G'(x, \mathcal{H}(y))\|_1] \tag{4.8}$$

We can determine the optimal G' for the re-organized loss to be $\mathcal{H}^{-1}$, the inverse transformation of $\mathcal{F}$, and ignore the sketch input x. This leads to a substantial deterioration in inference.

Let F and E denote the decoder and trainable encoder(s), respectively. When jointly training the encoders, the reference-based colorization can be expressed as $y = F(z) \sim P(y|x, r)$, where $z = E(x, r)$, and the latent distribution of z is, therefore, $P(z|y, x, r)$. Adopting a pre-trained reference encoder stabilizes this process by dividing it into two steps. First, the sketch encoder generates z' by encoding the sketch image, expressed as $z' = E(x)$, and the latent distribution of z' is $P(z'|y, x)$. Then, the receiving block, highlighted as red block in Figure 4.5, obtains the embeddings z by conditioning on the reference information, such that $z = \psi(\hat{x}, \hat{r})$, where $\hat{x} = z' \sim P(z'|y, x)$. As the reference

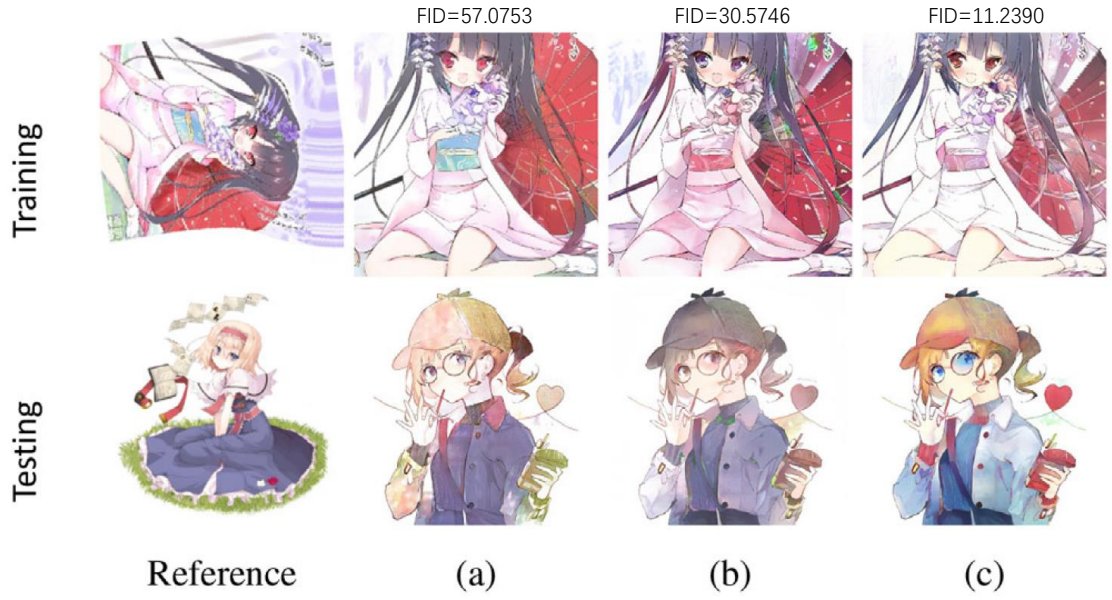


Figure 4.8: Qualitative comparison of results generated by models using different reference encoders. The reference encoder was (a) jointly trained with GAN, (b) fixed and pre-trained on ImageNet and (c) fixed and pre-trained on ImageNet and Danbooru. GAN, generative adversarial network.

encoder is fixed, the optimization target shifts from the latent distribution $P(z|y, x, r)$ to the distribution $P(z'|y, x)$, where the latent distribution $P(z'|y, x)$ is irrelevant to the reference r and decides the image quality. Therefore, this change significantly improves the generated results, particularly compared to the cases when jointly trained encoders fail to match semantically corresponding regions between x and r.

Ablation studies were conducted to demonstrate this deterioration, with two baseline models using reference encoders that were either 1) jointly trained with the generative backbone or 2) pre-trained solely on ImageNet [13]. Figure 4.8 provides a qualitative comparison of colorization results under different reference encoder training strategies, highlighting the impact of encoder optimization on generalization. During training (top row), both group (a) and group (c) effectively colorize sketches when the reference shares visual style with the training distribution. In contrast, group (b) yields inferior color fidelity due to the misalignment between the latent representations of the ImageNet-pretrained encoder and the anime-style domain. However, under the testing setting (bottom row), where the reference is out-of-distribution, the model using the jointly trained encoder (a) demonstrates clear overfitting to training textures, leading to distorted and semantically inaccurate results. In comparison, the models with ImageNet-pretrained (b) and ImageNet+Danbooru-pretrained (c) encoders generalize more effectively. Notably, group (c) achieves the most visually coherent and stylistically faithful outputs, benefiting from domain-specific knowledge acquired through Danbooru pretraining. This is further supported by the Fréchet Inception Distance (FID) scores: group (a) records the highest FID of 57.08, indicating poor generalization; group (b) achieves a moderate FID of 30.57; while group (c) yields the best performance with a substantially lower FID of 11.24. These results confirm that although end-to-end training enhances reference fidelity in-distribution,

Table 4.2: FID score evaluation for the ablation study and comparison with baseline methods. A lower FID score indicates better quality of the generated image. "Fix" and "Train" indicate that the reference encoder is fixed or trained in the colorization training, respectively, and "D" and "I" indicate that the reference encoder is pre-trained on Danbooru [12] + ImageNet [13] or ImageNet only, respectively. The second training scores will be introduced in Section 4.3.

Full setting		w/o RBCA		Ablation reference encoder			
Attention	Add	Attention	Add	D + Train	I + Fix	I + Train	Train
11.2390	11.8531	15.6212	13.6563	23.1408	30.5746	51.3302	57.0753
		2nd training score		Baseline methods			
		M-Attention	M-Add	[9]	[85]	[49]	[35]
		12.3845	11.9800	34.8503	59.6767	62.1494	121.5455

it also introduces a strong inductive bias toward the training domain, ultimately compromising generalization to unseen references. Full quantitative evaluation is given in the next subsection, together with the baseline comparison in Table 4.2.

Similar degradation is commonly observed in reference-based colorization baseline methods that adopt **training paradigm 2** (Section 3.1), as most high-quality image generation frameworks are optimized using reconstruction-based losses. Even when these baseline methods incorporate real reference images, such as those from neighboring frames or spatial grids, this form of deterioration remains inherent to their framework. A more detailed discussion of this issue will be provided in Section **??**.

4.2.4 Ablation Study and Comparison with Baseline Methods

FID calculation. As introduced in Section 3.3, the Fréchet Inception Distance (FID) is regarded as the most reliable metric for evaluating both the generation quality and generalization ability of image synthesis frameworks. To rigorously assess generalization, validation references were randomly selected from the validation set—completely disjoint from the training set—resulting in 7,000 triplets of (sketch, ground-truth color image, reference). This setting ensures that the reference styles used for evaluation are out-of-distribution with respect to the training data, thus revealing how well the model adapts to novel inputs.

Ablation study on network architecture. One core component evaluated in this ablation study is the reference feature injection mechanism. While global feature pooling has traditionally been used to inject reference information [103, 104], this approach lacks spatial precision and often fails to preserve fine-grained stylistic details. In this study, a spatial attention mechanism was first adopted to better transfer localized reference cues to the generative backbone. However, the inherent instability of GAN-based training rendered the attention maps unreliable, leading to performance degradation. To overcome this limitation, I introduced the Reference-Based Channel-wise Attention (RBCA) block, which enhances colorization performance by focusing on semantically relevant channels and mitigating GAN-induced noise. Quantitative results, measured using FID, confirm that the RBCA-enhanced architecture yields improved stability and fidelity compared to both spatial attention and global pooling baselines. This is evident when comparing the

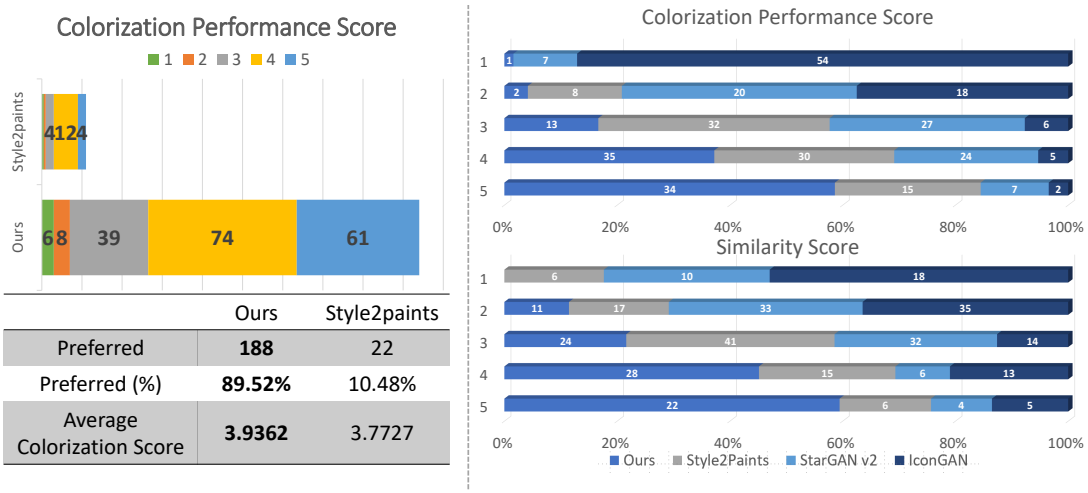
FID score of the *Attention* model, the proposed framework, with that of the *Add* model, an ablation variant in which reference representations are globally averaged and directly added to the query features. The inferior performance of the *Add* model highlights the limitations of simple global aggregation and underscores the importance of spatially-aware, learnable reference integration. Further ablation studies show that removing RBCA blocks results in a consistent drop in performance, reinforcing their critical role in aligning reference features and guiding generation. These findings collectively demonstrate that RBCA not only enhances feature conditioning but also contributes to more stable and faithful image synthesis, as evidenced by lower FID scores across variants.

Brief overview of baseline methods. Several notable reference-based sketch colorization methods were selected as baselines for evaluating the proposed GAN-based framework. Based on their relevance to the task definition and practical applicability, the selected baselines include: Style2Paints [104], StarGAN v2 [9], IconGAN [85], and MUNIT [35]. Among these, StarGAN v2 and MUNIT are representative style-transfer frameworks that encode the input image into a latent representation, which is then modulated using a style code extracted from the reference image. IconGAN employs dual discriminators to independently enforce structural and chromatic consistency during generation. SCFT [49], on the other hand, introduces a spatial correspondence module by deforming the reference images and recording the deformation, enabling the network to learn correspondences in a supervised manner.

These baselines typically follow training paradigm 2 (as introduced in Section 3.1) to jointly optimize multiple encoders specialized for different input styles, except for Style2paints [104], which also utilizes paradigm 1. Besides, Style2Paints adopts a two-stage training pipeline and leverages a pre-trained InceptionNet [87] for feature extraction. However, Style2Paints is designed as a semi-automatic application requiring user interaction, such as manual stroke hints and post-processing operations. Consequently, generating a large-scale dataset for FID-based evaluation is infeasible. To fairly assess its performance, two separate user studies were conducted to investigate user preferences and subjective evaluation of the results.

Comparison with baseline methods. To further assess the effectiveness of the proposed framework, comparisons were made against the aforementioned baselines, each of which employs distinct architectural assumptions and reference utilization strategies. While these methods demonstrate promising results under constrained conditions, their dependence on reconstruction-based objectives and tightly coupled reference encoders limits their ability to generalize beyond the training distribution. This limitation is particularly evident when reference images deviate significantly from those seen during training, leading to inconsistent or semantically inaccurate outputs.

By contrast, the proposed method introduces two key innovations: the RBCA block, which allows for fine-grained reference feature integration, and domain-specific encoder pretraining on both ImageNet and Danbooru. These components collectively enhance generalization performance and produce outputs that remain visually consistent and stylistically faithful under diverse and challenging conditions. As demonstrated by the FID evaluation in Table 4.2, this advantage is quantitatively supported by consistently lower



	Ours	Style2paints
Preferred	**188**	22
Preferred (%)	**89.52%**	10.48%
Average Colorization Score	**3.9362**	3.7727

Figure 4.9: User study results. Left: The first user study conducted between the proposed framework and Sytle2Paints. Participants are invited to rate the quality of their preferred result from 1 to 5, with 5 as the best. The average colorization score is calculated as $\frac{\sum score}{\sum pt}$, where pt denotes the preferred time. Right: Rating score distribution in the second user study. A higher score indicates better performance.

Table 4.3: Average scores in the second user study. The participants needed to rate colorization performance and similarity for each group, which contains a sketch image, a reference image, and a corresponding colorized result.

	Colorization Performance	Similarity
Attention	**4.165**	**3.718**
[104]	3.612	2.976
[9]	3.047	2.541
[85]	1.624	2.435

FID scores, confirming the proposed method's superior robustness and effectiveness across a wide range of test cases.

User studies with baseline methods. To complement the quantitative evaluation and provide a more comprehensive assessment of the proposed GAN-based framework, two user studies were conducted. The first user study focused on a direct comparison between our method and Style2Paints [104], as these two approaches were the only ones capable of consistently producing satisfactory results. This study comprised 10 image groups, each consisting of a sketch, a reference image, and two corresponding colorized outputs—one from our method and one from Style2Paints. A total of 21 participants were invited to evaluate each group by selecting the preferred image and assigning a rating for colorization quality on a five-point Likert scale. The results, summarized in the left part of Figure 4.9, demonstrate a clear preference for our method, which was chosen 188 times (89.52% of votes) and achieved a higher average colorization score of 3.9362 compared to 3.7727 for Style2Paints.

The second user study was designed to benchmark our method against a broader set of baselines, including Style2Paints, StarGAN v2, and IconGAN. Four questionnaires were prepared, each containing 20 image triplets consisting of a sketch, a reference, and a gen-

erated image. In each questionnaire, the image triplets were divided into four groups (e.g., groups [1–5], [6–10], [11–15], and [16–20]), with each group corresponding to one of the four methods under comparison. A total of 17 participants were invited to rate each colorized result based on two criteria: overall colorization performance and similarity to the reference image. As illustrated in Figure 4.9 and detailed in Table 4.3, our method outperformed all other baselines across both evaluation dimensions, consistently receiving the highest average ratings. These results support the effectiveness of the proposed framework in delivering visually appealing and reference-faithful outputs. Samples of both user studies are provided in the supplementary materials for reference.

4.3 Tag-based manipulation

After optimizing the generative backbone for reference-based sketch colorization, this study establishes a stable latent space mediated through the receiving block. Inspired by previous works on latent space manipulation and attribute control in generative models [40, 41, 93, 97], this section explores the potential of utilizing the reference representations, acquired via the spatial attention mechanism within the receiving block, not only for guiding colorization but also for editing specific visual attributes. Crucially, these reference representations are directly fed into a linear projection layer to produce a probability vector for multi-label classification, where each label corresponds to a distinct visual attribute. This direct mapping between representation and attributes suggests the possibility of inverting the projection process—enabling the projection of desired attribute probabilities back into the reference representation space. In doing so, the framework can potentially support controllable edits of visual attributes by modifying the reference representation itself.

This section begins by formulating the manipulation process through a detailed analysis of how to establish an invertible mapping between visual attribute probabilities and reference representations. Building on this foundation, I introduce the architecture of the trainable modules designed to approximate the inverse of the non-linear projection, along with the corresponding loss functions tailored to optimize this inversion. Finally, qualitative experiments are presented to demonstrate the effectiveness of the proposed approach in enabling text-based manipulation of visual attributes within the latent space.

4.3.1 Latent interpolation objective

Previous studies have shown that the class probabilities output by a pre-trained CNN encode rich latent semantic information that can be leveraged for tasks such as sketch colorization. Building upon this insight, our second training stage is designed to bridge these class probability vectors with the visual features employed in the first training stage. To this end, I introduce a mapping network φ, which is trained to satisfy the following approximation:

$$\mathrm{GAP}(\phi(r_t)) - \mathrm{GAP}(\phi(r_a)) \approx \varphi(\mathrm{cls}_t) - \varphi(\mathrm{cls}_a). \tag{4.9}$$

Here, $\phi(r_t)$ and $\phi(r_a)$ denote the visual representations extracted from the target reference image r_t and the anchor reference image r_a, respectively, while cls_t and cls_a represent their corresponding class probability vectors produced by the pre-trained CNN classifier—typically the final linear layer followed by a Sigmoid activation. By learning a neural network φ to map from class probability space to the latent visual space, we enable a more linear and interpretable manipulation of features. This mapping not only increases robustness but also permits extrapolation, allowing input probability values beyond the typical [0, 1] range.

Models trained with this second-stage optimization, in conjunction with the visual encoder ϕ, are referred to as *M-Attention*. An ablated variant, *M-Add*, replaces the attention mechanism with a simple additive operation for feature fusion. Notably, M-Attention incorporates an additional fully connected layer ω, which learns a spatial modulation factor

to support the proposed spatially-aware manipulation scheme.

We now proceed to define the optimization objectives used to train the extended framework for effective latent interpolation and semantic transformation:

$$\arg\min_{\varphi,\omega} \mathcal{L}(\varphi,\omega) = \mathcal{L}_{\text{hybrid}}(\varphi,\omega) + \mathcal{L}_{\text{inv}}(\varphi,\omega) \tag{4.10}$$

In this formulation, $\mathcal{L}_{\text{inv}}$ and the spatial modulation layer ω are specifically introduced to compensate for the absence of spatial information in $\phi(\text{cls})$, the output of the mapping network applied to class probabilities. These components are excluded when training the *M-Add* model, which does not utilize spatial information. Notably, this simplification does not significantly degrade colorization performance, since the second training stage focuses exclusively on aligning intermediate representations and does not involve the generator G in the optimization process.

The core idea of the second training stage is to modify the latent reference codes using the mapped class probability vectors. To achieve this, I revisit a decomposition of the reference representation $\phi(r)$:

$$\phi(r) = \frac{\phi(r)}{\text{GAP}(\phi(r))} * \text{GAP}(\phi(r)) \tag{4.11}$$

This identity allows us to separate $\phi(r)$ into two interpretable components: a spatial factor matrix $w(r) = \frac{\phi(r)}{\text{GAP}(\phi(r))}$, which retains localized structural information, and a semantic vector $\text{GAP}(\phi(r))$, obtained via global average pooling. Through this decomposition, the representation can be viewed as a composition of $(\frac{H}{\text{patch\_num}}, \frac{W}{\text{patch\_num}})$ latent codes, where H and W denote the input image's height and width. Following the standard configuration of the adopted ResNet-34 backbone, I set $\text{patch\_num} = 32$.

Using this formulation, we can express the difference between two reference representations r_t (target) and r_a (anchor) as:

$$\phi(r_t) - \phi(r_a) = w(r_t) * \text{GAP}(\phi(r_t)) - w(r_a) * \text{GAP}(\phi(r_a)) \tag{4.12}$$

Given that our framework is designed for reference-based editing in sketch colorization, I make a reasonable assumption: the spatial factor matrices $w(r_t)$ and $w(r_a)$ remain approximately equivalent when the two reference images differ primarily in semantics—such as hair color—while maintaining similar structural layouts. For instance, consider r_t as an image with red hair and r_a with blue hair; in this case, both $w(r_t)$ and $w(r_a)$ exert a similar influence on spatial attention due to the shared underlying spatial structure.

Under this assumption, Eq. 4.12 can be approximated as:

$$w(r_t) * \text{GAP}(\phi(r_t)) - w(r_a) * \text{GAP}(\phi(r_a)) \sim w(r_a) * (\text{GAP}(\phi(r_t)) - \text{GAP}(\phi(r_a))) \tag{4.13}$$

This leads to the simplified approximation:

$$\phi(r_t) \approx \phi(r_a) + w(r_a) * (\text{GAP}(\phi(r_t)) - \text{GAP}(\phi(r_a))) \tag{4.14}$$

This final expression encapsulates the core idea of our proposed manipulation strategy: semantic editing is achieved by modifying the global semantic vector, while preserving the

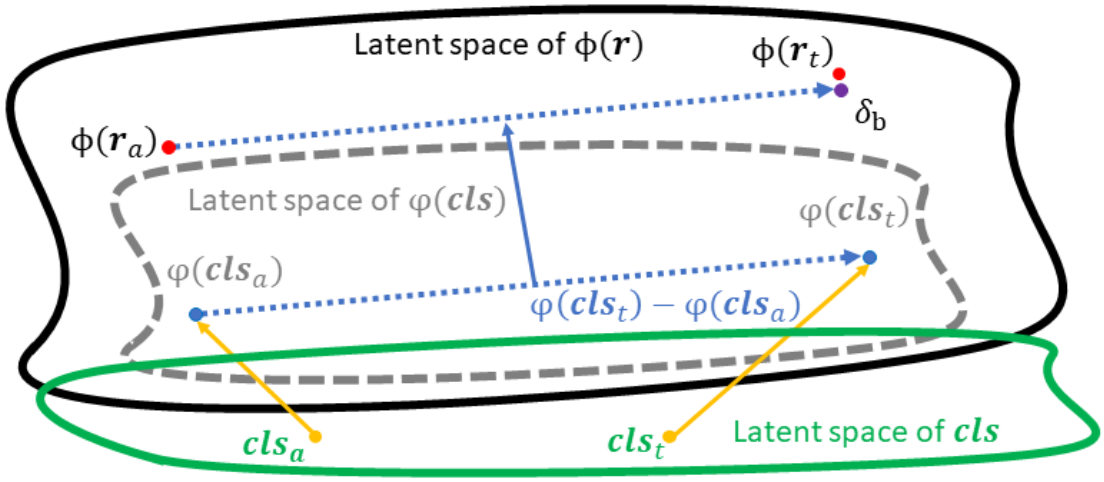


Figure 4.10: Illustration of how to approximate $\phi(r_t)$ using δ_b, defined in Eq. 4.15. Converting $\phi(r_a)$ to $\phi(r_t)$ on the basis of the vector distance is better than directly mapping $\phi(cls_t)$ to $\phi(r_t)$ as it ignores the difference of latent space.

spatial structure of the reference representation. The interpolation in the latent space can be formulated as:

$$\delta_b = \phi(r_a) + \mathcal{F}(r_a) * (\varphi(\text{cls}_t) - \varphi(\text{cls}_a)) \,, \tag{4.15}$$

where $\varphi(\text{cls}_*)$ denotes the mapped class probability vectors for the target and anchor images, broadcasted to match the spatial dimensions of $\phi(r_a)$ by replicating the channel values. The transformation function $\mathcal{F}$ serves as a learned approximation of the spatial factor w, and is defined as:

$$\mathcal{F}(r_a) = \omega \left(\frac{\phi(r_a)}{\text{GAP}(\phi(r_a))} \right) , \tag{4.16}$$

with ω implemented as a single linear (fully connected) layer. Given that the ResNet-34 encoder utilizes a ReLU activation in its final layer, we have $\phi(r_a) \geq 0$. To address potential division by zero when computing the normalized representation, I define the ratio $\frac{\phi(r_a)^c}{\text{GAP}(\phi(r_a))^c} = 0$ whenever the denominator is zero for any channel c.

This formulation enables us to interpolate between latent representations by adjusting only the semantic component, guided by class probabilities, while maintaining the spatial configuration inherited from the reference image. The visual effect of this interpolation is demonstrated in Figure 4.10, where controlled semantic alterations—such as changes in color attributes—are realized without disturbing the structural integrity of the output. This provides a robust and interpretable mechanism for reference-based sketch colorization.

4.3.2 Architecture and training

Hybrid reconstruction loss $\mathcal{L}_{\text{hybrid}}$

We now delve into the training objective for the second-stage optimization of the mapping network φ and the spatial factor simulator ω to enable tag-based manipulation. As described in Eq. 4.15, the mapping network φ is designed to transform the class probability vectors such that the resulting representation δ_b approximates the target latent code $\phi(r_t)$. To guide this transformation, I propose a hybrid L1 loss that enforces consistency at both the pixel and feature levels:

$$\begin{aligned}\mathcal{L}_{\text{hybrid}}(\varphi,\omega) = &\underbrace{\mathbb{E}x, r_t, \delta_b \left[\|G(x,\phi(r_t)) - G(x,\delta_b)\|_1\right]}_{\text{Pixel-level constraint}} + \\ &\underbrace{\mathbb{E}_{x,r_t,\delta_b} \left[\|\psi(x,\phi(r_t)) - \psi(x,\delta_b)\|_1\right]}_{\text{Feature-level constraint}},\end{aligned} \tag{4.17}$$

where G denotes the generator, x is the input sketch, and ψ refers to the intermediate feature maps extracted from the generator during forward propagation (as visualized in Figure 4.2).

The feature-level L1 loss serves as the core supervision signal, ensuring that the synthesized latent representation δ_b captures the fine-grained semantics of the target reference representation $\phi(r_t)$. This loss component directly influences the ability of the network to perform accurate and controllable semantic manipulation in the latent space.

The pixel-level L1 loss, on the other hand, enforces global appearance consistency between the generated images conditioned on $\phi(r_t)$ and δ_b, respectively. It primarily constrains broader visual attributes—such as background color, lighting, or scene theme—which are strongly influenced by the spatial attention mechanisms introduced through RBCA blocks.

Together, these two components form a balanced training objective that encourages φ and ω to produce semantically meaningful and spatially coherent latent representations, enabling precise and interpretable sketch colorization through class-guided manipulation.

Inversion loss $\mathcal{L}_{\text{inv}}$

While the feature-level L1 loss plays a crucial role in guiding the training of the mapping network φ, it is insufficient on its own in the context of the *M-Attention* model. This limitation arises from the dot product operation used within the spatial attention mechanism of the receiving block, which restricts effective gradient flow back to the parameters of φ. To address this issue and facilitate more efficient convergence of the mapping network, I introduce an additional supervision signal in the form of an inversion loss, formulated as:

$$\mathcal{L}_{\text{inv}}(\varphi,\omega) = \mathbb{E}r_t, r_a \left[\|(\text{GAP}(\phi(r_t)) - \text{GAP}(\phi(r_a))) - (\varphi(\text{cls}_t) - \varphi(\text{cls}_a))\|_1\right] \tag{4.18}$$

This loss explicitly encourages the mapping network φ to satisfy the latent space alignment condition described in Eq. 4.9, thereby reinforcing the semantic consistency between the mapped probability vectors and their corresponding visual features. In combination with the hybrid L1 loss, the inversion loss also enables the spatial factor simulator ω to

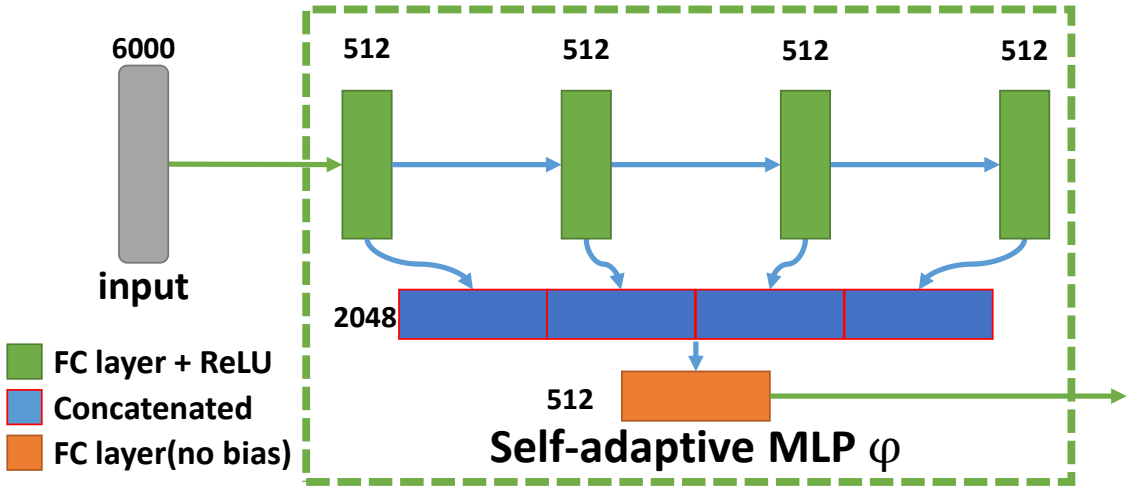


Figure 4.11: Self-adaptive MLP used to generate latent codes from classification probabilities. It takes the classification probabilities as input. MLP, multi-layer perceptron

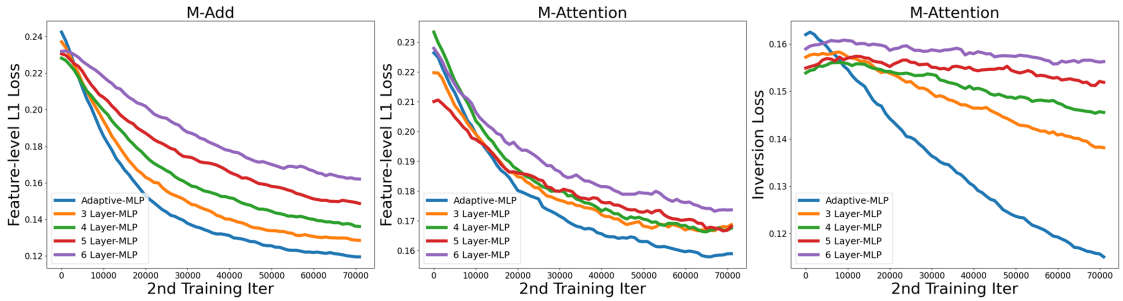


Figure 4.12: Comparison of feature-level L1 and inversion losses during training. The losses are smoothed by exponential moving average with the smoothness weight set to 0.9.

better modulate $\frac{\phi(r_a)}{\text{GAP}(\phi(r_a))}$ by learning its underlying channel-wise relationships. This synergy between loss functions ensures that both semantic and spatial aspects of the reference representation are effectively aligned for robust and interpretable manipulation.

Mapping network design

Preliminary works [40, 41, 93, 97] and experiments indicate that different visual attributes are controlled by latent codes emerging from distinct layers within the generator's latent space. For instance, attributes such as "hair" and "eyes" are primarily influenced by the earlier layers (e.g., the first and second fully connected layers), whereas more global features like "sky" and overall "theme" are governed by deeper layers. However, as the network progresses, earlier semantic cues, such as those controlling "hair" and "eyes," often become entangled within the deeper layers. This leads to a loss of control over fine-grained attributes and results in undesirable entanglement in the generated outputs.

To address this issue, I propose a novel architecture termed the self-adaptive MLP, designed to mitigate such entanglement. Instead of relying solely on the final output of a deep multi-layer perceptron, our self-adaptive MLP concatenates the outputs from multiple intermediate layers. These concatenated features are then adaptively weighted to preserve and balance both local and global semantic information. This design allows the

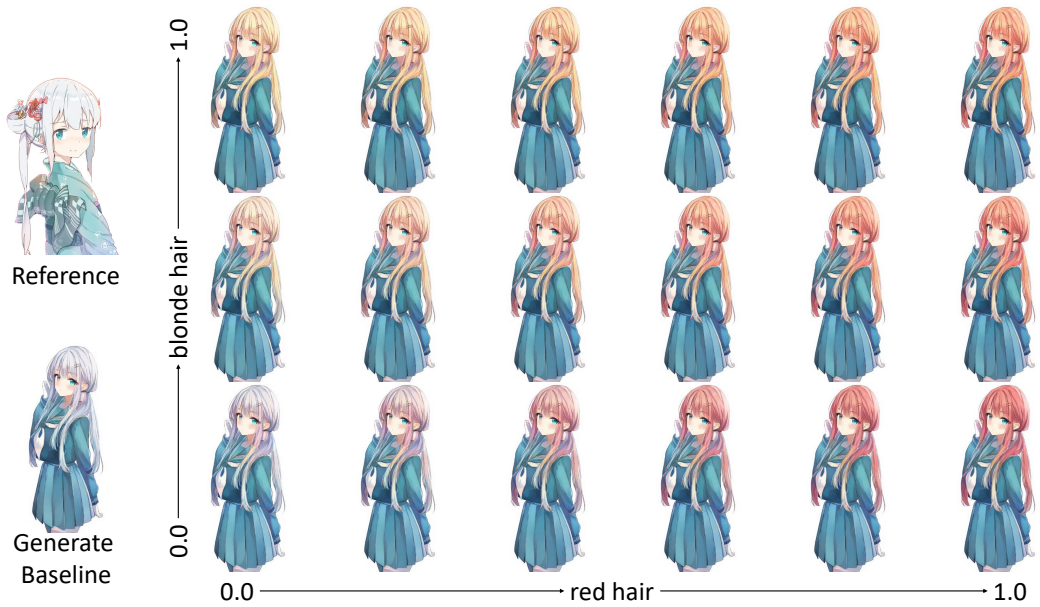


Figure 4.13: Multi-attribute results rendered by the *M-Attention* model. From left to right, the respective "red_hair" values are $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, and from bottom to top, the respective "blonde_hair" values are $\{0, 0.5, 1.0\}$.

network to maintain fine-grained control over early-layer attributes while incorporating the broader contextual understanding captured by deeper layers. The detailed architecture is illustrated in Figure 4.11.

Figure 4.12 corroborates the effectiveness of the self-adaptive MLP in disentangling local and global semantics. By explicitly fusing and re-weighting intermediate representations, the model retains fine-grained cues from shallow layers (e.g., "hair" and "eyes") while simultaneously leveraging the holistic context learned in deeper layers. As a result, it not only converges more rapidly than heavier baselines but also attains a substantially lower terminal loss, indicating a cleaner optimization landscape with reduced attribute entanglement. This synergy of faster convergence, improved semantic control, and a lightweight parameter budget makes the self-adaptive MLP an attractive choice for both resource-constrained environments and large-scale generative applications.

4.3.3 Experimental validation

To evaluate the controllability of the proposed models, I conducted multi-attribute manipulation experiments by varying the values of hair-related tags. As illustrated in Figure 4.13, the tag values were progressively increased along the axes, resulting in smooth and continuous changes in hair color across the generated samples. This visual progression highlights the model's ability to respond consistently to semantic input changes.

To further assess disentanglement and control over global attributes, I performed targeted manipulations on the "sky" labels, with the corresponding results shown in Figure 4.14. These results demonstrate that the global hue and thematic color of the image can be altered based on changes in the "sky" attribute, without inadvertently affecting other localized features such as hair or eyes. This supports the effectiveness of our model in maintaining semantic isolation across attributes.

Figure 4.14: Multi-attribute results generated by respective models. All results are generated with "red_hair"=2.0 and "yellow_eyes"=2.0. The sky labels can control the global hue of the generated image. Background and theme labels have a similar effect, such as "simple_background," "red_background" and "green_theme," "red_theme," respectively.

Figure 4.15: Multi-attribute results generated by the proposed *M-Attention* and *M-Add* models, where the baseline columns show the respective reference-based results. The manipulated tags are "blue_shirt," "green_hair" and "yellow_eyes."

I also investigated the linearity of the manipulation process by comparing results from the proposed *M-Attention* and ablation *M-Add* models, as shown in Figure 4.15. Both models exhibit consistent semantic transitions when the input tag values are extended beyond the typical [0, 1] range. Notably, our method maintains controllability and coherence even with values up to approximately 5, confirming the robustness of the proposed framework in extrapolative settings.

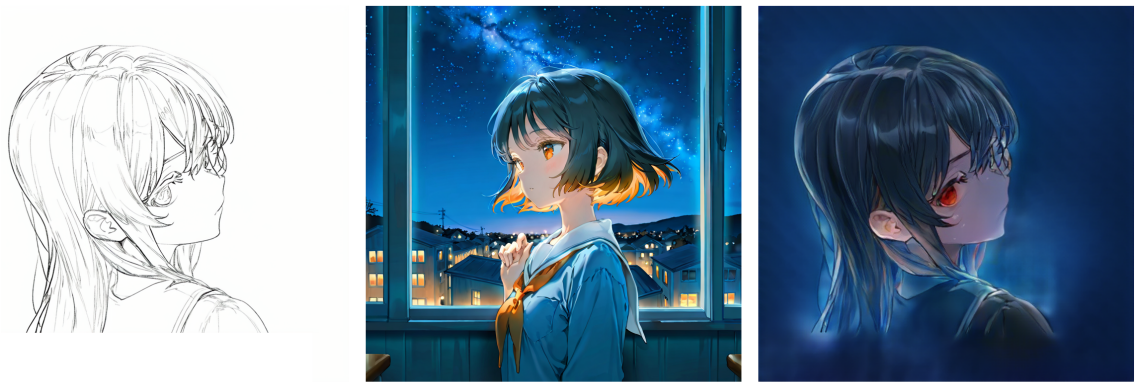


Figure 4.16: The limited generation ability and unstable training of GANs make it ineffective in synthesizing images with complicated compositions and content.

4.4 Limitation and conclusion

In this chapter, the proposed GAN-based framework introduces a reference-based sketch colorization system that yields visually pleasing, easily adjustable results by employing a pre-trained CNN as a frozen reference encoder, a design that, to the best of my knowledge, is the first to use a frozen image embedder to extract reference representations. This choice elevates transfer to higher-level semantics and, compared with baselines, delivers substantial gains in perceptual quality and generalization. To further strengthen transfer without compromising GAN stability, the framework integrates spatial attention and a newly proposed channel-wise cross-attention. An ablation study attributes clear improvements to both the RBCA block and a self-adaptive MLP, while comprehensive qualitative and quantitative evaluations—including a user study—consistently favor the proposed model over competing methods. Additional experiments on multi-label manipulation demonstrate fine-grained controllability and illuminate the model's spatial-latent behavior. Overall, the results show that freezing a pre-trained image encoder as the reference branch markedly improves performance and greatly enhances generalization in reference-based sketch colorization.

However, there are still a number of limitations. First, the performance of colorization deteriorates as the line density of the input sketch images decreases, resulting in a loss of texture information. While most generative models rely on noise inputs to compensate for this missing information when applied to single-condition generation, our model is designed for dual-condition generation (sketch + reference image or sketch + text tags) and eschews this approach due to its negative impact on the stability of training and image quality. Second, our ResNet-34 is inefficient in multi-label classification, as I found it performs much worse than DeepDanbooru [43] or other open-sourced classification networks, which is too heavy for our research, and this drawback decreases the GAN's segmentation ability. Finally, the proposed model cannot directly manipulate the latent code in the GAN, which may degrade the controllability of the results. The most important limitation comes from the inferior generation ability of GAN. As shown in Figure 4.16, the GAN-based framework is ineffective in transferring backgrounds and vivid textures.

Most of the foregoing constraints are rooted in the inherent training instability of

GANs. Because GAN optimization is formulated as a two-player minimax game between a generator and a discriminator, it demands a delicate and continuous equilibrium between their competing objectives. In practice, even minor mismatches in learning dynamics can precipitate mode collapse, oscillatory behavior, or outright divergence, trapping the model in sub-optimal local minima. These stability concerns force practitioners to adopt conservative architectural choices and limit the total number of trainable parameters, thereby constraining GANs' capacity to synthesize imagery with both stylistic diversity and structural fidelity. As a result, traditional adversarial frameworks remain ill-suited for the fine-grained, condition-aware control required in high-quality reference-based colorization.

To surmount these obstacles, the next chapter introduces a DM–based colorization pipeline. By replacing adversarial training with a likelihood-based objective and a progressive denoising process, DMs circumvent the generator–discriminator tug-of-war, offering markedly improved stability, scalability, and semantic controllability.

Chapter 5

Diffusion model framework

5.1 Overview

From GANs to Diffusion Models

Early work in reference-based sketch colorization relied on GANs, but their limited generative capacity made it difficult to faithfully transfer colors, textures, and even coarse background structure from the reference image to the target sketch. Recent DMs have largely removed these capacity constraints; their iterative denoising process can recover fine details and complex global layouts far beyond the reach of GANs. Unfortunately, DMs introduce a more challenging failure mode: strong prompt guidance often amplifies mismatches between reference and sketch during inference, producing conspicuous artifacts, segmentation errors, and unexpected geometry changes to sketches. Eliminating these DM-specific artifacts is therefore the central challenge addressed in this chapter. The proposed framework is called *ColorizeDiffusion*, labeled by their corresponding versions *v1* and *v1.5*, respectively.

The Overfitting Dilemma

As detailed in Section 3.1, reference-based colorization must be trained on triplets (sketch, reference, ground truth) that are *semantically related*. In practice, this means the reference and ground truth share not only high-level identity cues (character, clothing, pose) but also low-level spatial compositions (hairstyle, clothes type, and so on). While such alignment establishes the convergence of image-guided optimization, it also encourages the model to memorize spatial correspondences between sketch and reference inputs instead of learning how to *disentangle* spatial control from semantic transfer.

Proposed Remedies

To retain the benefits of semantic alignment between training data, which enhances the generalization (reasons of why this is considered "enhancement" will be discussed in Section), while suppressing its side effects, this chapter introduces three complementary techniques in sequence:

1. **Noisy Training with fine-tuning.** I inject controlled Gaussian noise into reference embeddings during a long-term optimization, followed by a shorter-period clean fine-

tuning. The noise perturbs low-frequency semantic cues (e.g., shapes and coarse regions) more than high-frequency style cues (e.g., brushstroke texture), forcing the network to rely on robust style features while preventing it from blindly copying spatial layouts. Empirically, noisy training widens the model's domain of acceptable inputs and reduces segmentation collapse.

2. **Split Cross-Attention.** Conventional cross-attention allows every reference token to influence every sketch token, so background features in the reference often pollute foreground regions in the prediction. I therefore partition the attention map into foreground and background sub-spaces and block cross-talk between them. Foreground tokens attend only to foreground reference embeddings; background tokens attend only to background embeddings. This quarantining strategy drastically reduces halo artifacts and texture leakage.

3. **Representation-Separated Framework.** Further analysis revealed that artifacts are modulated not just by *where* attention flows but also by the *semantic level* of the reference representation. I generalize split attention into a hierarchical separation scheme that propagates low-level color/style cues and high-level semantic cues through separate pathways. This refinement improves both color fidelity and stylistic consistency on challenging benchmarks.

5.1.1 Text-Driven, Zero-Shot Manipulation

Beyond faithfully transferring reference style, practical colorization systems should be able to support user intent. I therefore introduce a zero-shot, text-based manipulation module that performs spatial latent interpolation guided by textual descriptions (e.g., "make the coat pastel blue," "add autumn leaves in the background"). Because it operates in the same latent space as the diffusion process, the module adds no extra training cost and maintains full compatibility with the techniques above.

Most of the difficulties tackled here originate from *inevitable* overfitting: by definition, the reference and ground-truth images always share semantics. Thus, the training paradigm, outlined in Section 2.4.4 and revisited in Section 3.1, is just as critical as model architecture. To overcome the distribution shift issue that stems from the definition of reference-based sketch colorization, this thesis involves a series of works developed using DMs. These works will be introduced in sequence as: ults. Finally, an extensive discussion of the training paradigm is given in Section **??**.

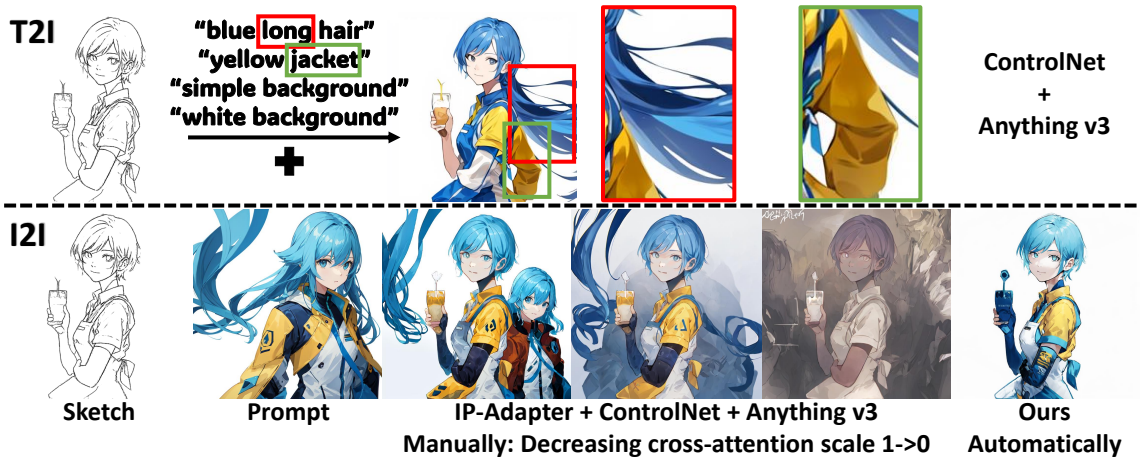


Figure 5.1: Illustration of spatial entanglement, a typical type of **artifacts** caused by the distribution shift. The T2I model prioritizes prompt semantics and thus generates results with long hair and a jacket outside the sketch. Similar conflicts widely exist in I2I colorization but result in much worse artifacts, such as the extra person in column (c) and the messy background in column (d). Column (f) illustrates our result as correct colorization.

5.2 Distribution shift and spatial entanglement

In the context of reference-based sketch colorization, overfitting is a pervasive yet often overlooked issue. Unlike conventional overfitting, where a model memorizes specific training outputs or over-adapts to training samples. Here, the overfitting manifests in the *mapping patterns* between input pairs of sketches and conditional references. Importantly, this overfitting does not necessarily degrade the visual quality of generated images in an easily recognizable way, making it harder to detect and diagnose. Instead, what becomes overfitted is the statistical relationship, which is the sub-distribution jointly decided by the sketch and its associated condition, rather than the ground truth distribution of plausible outputs. In other words, the model becomes overly specialized in transforming sketches into colorized outputs under the assumption that prompt conditions are always perfectly aligned both spatially and semantically. This phenomenon is particularly troublesome in the realm of conditional generation and is not unique to sketch colorization; it is widely observed across tasks such as text-to-image (T2I) synthesis, video prediction, and conditional inpainting. A representative example of such an issue in T2I is illustrated in the top row of Figure 5.1.

In contrast to text-guided or user-guided colorization, where user inputs typically consist of abstract prompts that lack fine spatial resolution and are designed to semantically match the sketch both during training and inference, inference scenarios for image-guided methods often involve references that are only loosely correlated with the target content. During training, however, these models rely on meticulously curated (sketch, reference) pairs that are fully aligned in terms of both structure and semantics. This discrepancy between training and inference conditions introduces a critical generalization gap. As a result, when confronted with imperfect or mismatched references at inference time, the model may generate visually implausible or semantically inconsistent content, such as incorrect textures, misplaced colors, or disorganized compositions.

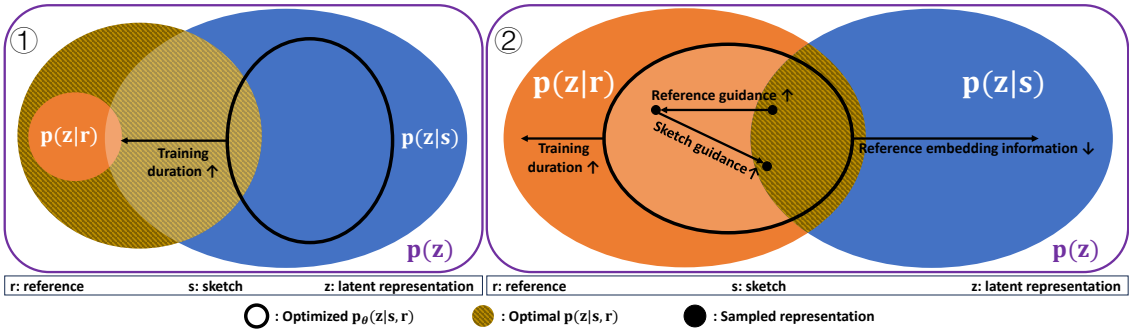


Figure 5.2: Illustration of the distribution shift. The optimized distribution gradually deviates from the optimal distribution, resulting in artifacts when reference images are semantically unaligned with sketches during inference. A solution is to reduce the information of reference embeddings during training.

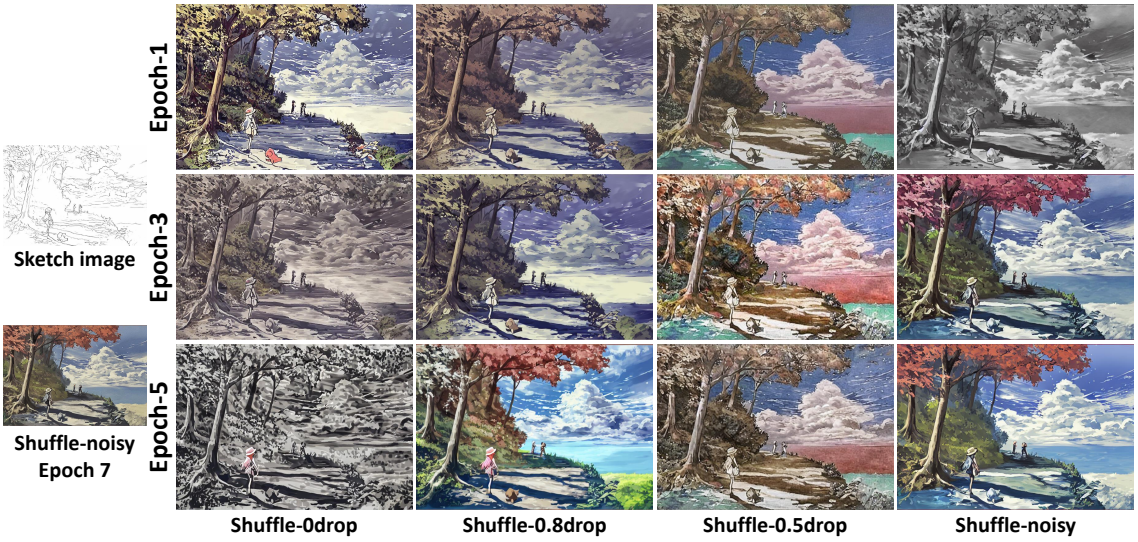


Figure 5.3: Qualitative colorization results synthesized without inputting reference images. Adding noise/increasing reference drop rate can reduce the reference information involved in the training and drag the optimized distribution $p_\theta(z|s,r)$ slightly back to $p(z|s)$.

Let the sketch be denoted as s, the ground truth image as y, the reference image as r, and the latent representation of the ground truth as z. Define $p(z|y)$ as the distribution of latent representations conditioned on the ground truth, and $p(z|s)$ and $p(z|r)$ as the ideal conditional distributions based on the sketch and reference, respectively. As discussed in Section 3.1, the perceptual semantics of the reference images are often closely aligned with those of the ground truths, resulting in $p(z|r) \approx p(z|y)$ during training. Since sketches typically encode only coarse structural and compositional cues, the gradient contributions from the reference side are significantly stronger than those from the sketch side. This imbalance causes the optimized generative model, parameterized by θ, to learn a conditional distribution $p_\theta(z|s,r)$ that increasingly resembles $p(z|r)$ rather than $p(z|s)$. This phenomenon is called "**distribution shift**." At inference time, however, the distribution $p(z|r)$ often lies outside the true support of $p(z|s)$, leading the model to sample implausible features from $p_\theta(z|s,r)$. This mismatch results in visually unacceptable artifacts, and in this thesis, all such background artifacts caused by the distribution shift are termed

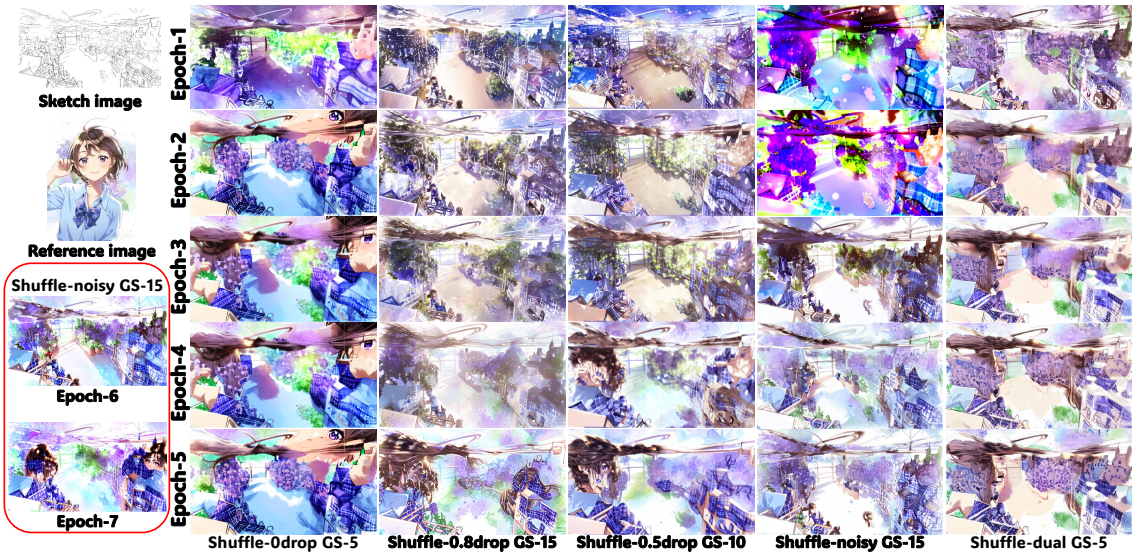


Figure 5.4: As training progresses, the optimized latent distribution inevitably shifts toward $p(z|r)$, manifesting in the synthesis of reference semantics that conflict with the sketch-guided regions.

"**spatial entanglement**," illustrated in Figure 5.2. Qualitative results are given in Figure 5.3 and Figure 5.4 to demonstrate the influential factors of the distribution shift.

Furthermore, image embeddings implicitly encode information related to object size and layout [65]. If these embeddings are directly transferred to structurally incompatible sketches, it can significantly degrade the perceptual quality. Unfortunately, standard reference-based training exacerbates this issue over time by reinforcing the transfer of such embeddings. This degradation is empirically validated through ablation studies comparing model outputs at various training stages.

One possible mitigation strategy is to adjust sampling hyperparameters, such as the scale of cross-attention output, to reduce the influence of the reference during generation. As shown in Figure 5.1, scaling down the reference input or modifying attention weights can lead to modest improvements in reducing visual artifacts. However, such adjustments generally prove ineffective in addressing the deeper issue of spatial entanglement and often lead to a noticeable drop in style fidelity or semantic richness. Furthermore, identifying optimal hyperparameter settings on a per-sample basis is computationally infeasible and impractical for real-world applications.

Although distribution shift is an intrinsic and therefore unavoidable aspect of reference-based sketch colorization, the objectionable artifacts it produces can still be firmly controlled. The most conspicuous of these artifacts is spatial entanglement—the bleeding or misalignment of reference colors across semantic boundaries—clearly illustrated in Figure 5.4. This thesis concentrates on eliminating such spatial entanglement by introducing dedicated techniques that realign color information and disentangle erroneous spatial correlations, thereby restoring visual fidelity even in the continued presence of distribution shift.

Table 5.1: Detailed U-Net architecture of Stable Diffusion v2.1 latent denoising network.

Stage	Resolution	Channels	Module
Input	64×64	4	Latent input (from VAE encoder)
Down Block 1	$64 \rightarrow 32$	320	Conv + ResBlock $\times 2$
Down Block 2	$32 \rightarrow 16$	640	Conv + (ResBlock + Cross-Attn) x2
Down Block 3	$16 \rightarrow 8$	1280	Conv + (ResBlock + Cross-Attn) x2
Down Block 4	$8 \rightarrow 8$	1280	(ResBlock + Cross-Attn) x2
Middle Block	8×8	1280	ResBlock + Cross-Attn + ResBlock
Up Block 1	$8 \rightarrow 8$	1280	(ResBlock + Cross-Attn) x3
Up Block 2	$8 \rightarrow 16$	$1280 \rightarrow 640$	(ResBlock + Cross-Attn) x3 + Upsample
Up Block 3	$16 \rightarrow 32$	$640 \rightarrow 320$	(ResBlock + Cross-Attn) x3 + Upsample
Up Block 4	$32 \rightarrow 64$	$320 \rightarrow 320$	ResBlock $\times 3$ + Upsample
Output	64×64	$320 \rightarrow 4$	Final LayerNorm + SiLU + Conv2D

5.3 Architecture

5.3.1 Denoising backbone

To overcome the limitations associated with the inferior generation quality of GAN-based methods, I begin by constructing sketch colorization systems based on diffusion models. In particular, Stable Diffusion (SD) v2.1 [72] is selected as the foundational architecture due to two key advantages that make it especially well-suited for this task.

1. **Extensive ecosystem of pre-trained weights and an active open-source community.** Among publicly available diffusion models, Stable Diffusion is by far the most widely adopted and actively maintained framework, primarily due to its open-source design. Community-driven platforms such as Stable-Diffusion-WebUI [3] and ComfyUI [11] have significantly lowered the barrier for experimentation, allowing independent developers to fine-tune and deploy countless custom models. Many of these community-developed models demonstrate state-of-the-art visual quality across a wide range of content and artistic styles, often surpassing outputs seen in the academic literature. This vibrant ecosystem has also motivated researchers to propose novel techniques and user-friendly tools [31, 60, 74, 98, 101] aimed at empowering individual creators and extending the applicability of diffusion models to diverse industrial and artistic domains. A fundamental prerequisite for building the proposed sketch colorization system is the availability of T2I models pre-trained specifically for anime-style generation. Fortunately, the SD community provides a rich repository of such fine-tuned weights.

2. **Integration of CLIP-based encoders.** Unlike many other T2I diffusion models developed in the same era, Stable Diffusion is built entirely on CLIP text encoders. As discussed in Section 2.1.5, CLIP encoders project both text and images into a shared semantic embedding space. Although this projection does not provide a continuous interpolation from discrete tokens to dense representations, the alignment between image and text embeddings enables Stable Diffusion to be readily extended for image-guided generation by changing guiding conditions from CLIP text embeddings to image embeddings. This makes the model inherently more flexible and

Table 5.2: Layer configuration of the downsampling convolutional layers inside the sketch encoder, where K/S/P represent kernel size, stride, and padding size, respectively. ZeroModule indicates all weights of the layer are zero-initialized [101].

Layer	K/S/P	Input dims	Output dims
Conv2d + SiLU	3 × 3 / 1 / 1	in_dims	16
Conv2d + SiLU	3 × 3 / 1 / 1	16	16
Conv2d + SiLU	3 × 3 / 2 / 1	16	32
Conv2d + SiLU	3 × 3 / 1 / 1	32	32
Conv2d + SiLU	3 × 3 / 2 / 1	32	96
Conv2d + SiLU	3 × 3 / 1 / 1	96	96
Conv2d + SiLU	3 × 3 / 2 / 1	96	256
ZeroModule(Conv2d)	3 × 3 / 1 / 1	256	out_dims

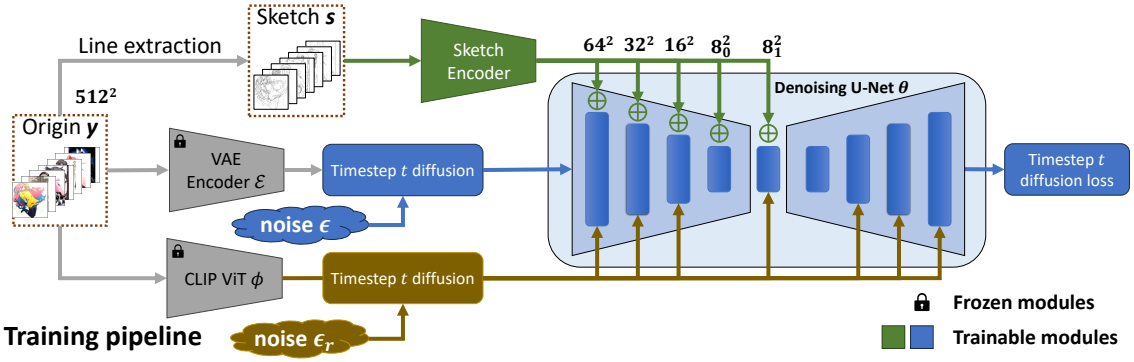


Figure 5.5: The pipeline of proposed noisy training. Timestep-dependent noises are added to reduce the effective information of reference embeddings and thereby achieve the

modular, allowing it to accommodate both textual and visual conditions within the same architectural backbone. In the DM-based frameworks proposed in this thesis, I utilized *local* tokens to provide reference representations for the colorization backbone.

Since the original SD2.1 is designed for T2I generation, the training target of the proposed framework is to alter it for reference-based generation that utilizes image prompts. A detailed architecture table is given in Table 5.1 for reference.

5.3.2 Sketch encoder

Different from the GAN-based framework introduced in Section 4.2, which directly encodes sketch images as an input for the generative backbone, the DM-based framework utilizes an independent sketch encoder to introduce the sketch conditions. The majority of the sketch encoder is similar to that of ControlNet [101], composed of several convolutional layers and in-between SiLU [16] layers. The configuration of the downsampling layers is given in Table 5.2.

A major difference between the adopted sketch encoder and that of ControlNet is the multi-scale injection. Similar to T2I-Adapter [60], the sketch encoder utilized in the proposed framework injects encoded sketch features at different levels through element-wise additions

$$z_i' = z_i + s_i, \quad i \in \{64, 32, 16, 8_0, 8_1\} \tag{5.1}$$

where z_i, s_i indicate the forward features and sketch features at corresponding scale i. A visualization of the multi-scale feature injection is given in Figure 5.5, together with the first-stage noisy training.

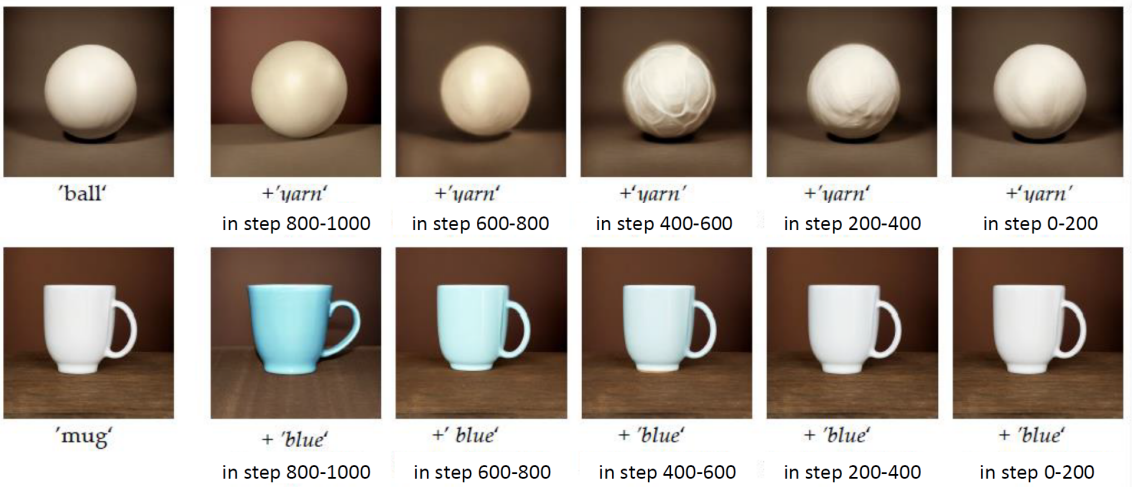


Figure 5.6: Diffusion models synthesize different visual attributes at distinct denoising steps. Empirically, global properties such as color schemes and identity-related semantics are generated in the early stages, while high-frequency details—such as textures and fine strokes—are progressively refined in the later steps. The visualization results from Prospect [107] clearly illustrate this timestep-dependent generation behavior.

5.4 Two-stage training for major backbone

To mitigate the negative effects caused by the distribution shift discussed above, the first crucial step is to reduce its negative influence on the sketch-side segmentation. Specifically, the model tends to overfit the spatial correspondences between sketches and reference images when trained on perfectly aligned pairs, making it tends to produce results with less semantic fidelity and spatial entanglement artifacts. To address this issue, I propose a two-stage training strategy that aims to decouple and balance the optimization of style transfer and spatial semantic alignment.

The key idea is to first stabilize the generative backbone by focusing on learning spatial structures and segmentation conditioned primarily on the sketch input. Once the model has learned to reconstruct accurate spatial layouts from sketches alone, reference-based conditioning is gradually introduced to enhance stylistic fidelity without compromising the spatial integrity established in the first stage. This prevents the early dominance of reference information and helps the model generalize better to a wider range of reference inputs during inference.

This training approach is further supported by recent studies in diffusion-based generative models. Prior works, such as [17, 60, 107], have shown that different semantic components of an image emerge at different stages of the denoising process. As visualized in Figure 5.6, coarse spatial structures are typically synthesized in the early timesteps, while fine details and stylistic elements are added in the later stages. These insights suggest that separating spatial and style learning phases aligns well with the inherent generative process of diffusion models, improving both robustness and controllability in reference-based image synthesis.

Based on this principle, I propose a noisy training strategy followed by a shorter-period fine-tuning phase to carefully optimize a reference-based sketch colorization framework. Specifically, the model is first initialized from well-trained text-to-image (T2I) diffusion

weights to retain rich prior knowledge regarding visual semantics, compositional structures, and generalization capability. This initialization provides a robust foundation for downstream tasks and accelerates convergence.

During the noisy training stage, the model is exposed to imperfect and partially mismatched references to simulate realistic inference conditions. This helps reduce overfitting to idealized training pairs and encourages the model to become more robust to semantic and spatial variance in the reference inputs. Following this stage, a fine-tuning phase is applied with more reliable data and stabilized conditions to refine the generation quality while preserving the robustness acquired during noisy training. AdamW optimizer [46] and a learning rate 1×10^{-5} are adopted for training the networks.

In addition to the two-phase training strategy, a critical training technique is employed to further improve synthesis quality. These include adaptive attention reweighting, reference dropout, gradient clipping, and normalization techniques that help stabilize training dynamics and prevent over-reliance on either input modality. Together, these components form a cohesive training pipeline designed to maximize both controllability and perceptual fidelity in reference-based sketch colorization.

5.4.1 Stage I - Noisy training

An inferable conclusion from Section 5.2 is that prolonged training tends to exacerbate the distribution shift problem—an observation consistent with general overfitting behavior seen in many machine learning domains. However, the overfitting encountered in reference-based sketch colorization deviates from standard patterns. Notably, the model learns high-level semantics such as composition and identity significantly faster than fine-grained visual features like textures and strokes. According to the definition of reference-based sketch colorization introduced in Section 3.3, a complete and well-functioning colorization system should ensure that both structural and stylistic details are faithfully transferred from the reference image to the sketch. Therefore, such an imbalance in the learning dynamics is undesirable and undermines the integrity of the generated results.

Moreover, visually unacceptable artifacts, such as extraneous body parts appearing in background regions or unintended changes to the sketch's original layout, are closely linked to early-stage identity and color semantics. These components are primarily determined during the initial steps of the denoising process. From this perspective, the early-stage optimization, which governs semantic layout and identity transfer, should proceed more cautiously than the later stages, which are responsible for refining textures and fine visual details.

To decelerate the optimization of spatial transfer and rebalance the optimization process, two classic strategies are typically considered: (1) performing uneven timestep sampling during training [60, 65], which allocates more emphasis on late denoising steps, and (2) applying a higher dropout rate to early-stage noise levels [101]. However, preliminary experiments indicate that these techniques are insufficient when applied to reference-based sketch colorization. Due to the richness of semantic information embedded in the reference images—which far exceeds that contained in sketches—such methods merely reduce the overall optimization rate without effectively controlling the dominance of early-stage semantics. As a result, they fail to address the core imbalance and have a limited impact

Figure 5.7: The color synthesized by the noisy-trained model is visually flat. Therefore, a refinement stage fine-tuning is adopted to fine-tune the transfer of color details.

on mitigating the observed artifacts.

Instead, I directly apply the same diffusion process to the reference embeddings, extracted from the CLIP image encoder. The training pipeline is illustrated in Figure 5.5. Given the pre-trained VAE encoder $\mathcal{E}$, the combination of denoising U-Net and the sketch encoder θ, CLIP image encoder (ViT) ϕ, sketch inputs x, reference inputs r, ground truths y and its encoded latents z, the noisy training is formulated as

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{E}(y),\epsilon,t,s,r}[\|\epsilon - \epsilon_\theta(z_t, t, s, \tau_{\phi,t}(r))\|_2^2], \tag{5.2}$$

where $\tau_{\phi,t}(r) = \alpha_t \tau_\phi(r) + \beta_t \epsilon_r$ and $\epsilon_r \sim \mathcal{N}(0, 1)$. As the reference images used during training are ground truth, r can be replaced by y in this equation. The training data, as discussed in Section 3.1, is the full set of Danbooru2021. I trained the network for 5 epochs (approximately 100k gradient steps per GPU) and dropped 10% reference inputs for classifier-free guidance (CFG) [30].

5.4.2 Stage II - Short-period fine-tuning

After Stage I, the diffusion backbone has been retargeted from pure text-to-image synthesis to reference-based sketch colorization that utilizes image prompt guidance, without introducing the large-scale spatial entanglement that often plagues direct fine-tuning. Nevertheless, two shortcomings remain: the generated hues look washed-out, and fine structures—highlights on hair, jewelry accents, and similar details—do not always adopt the reference palette faithfully. To restore vivid chroma while protecting the newly learned structure awareness, I adopted a short "refinement burst" lasting roughly 1-2 epochs (approximately 40k gradient steps) on a single Nvidia H100 GPU.

The fine-tuning processes as vanilla diffusion training but with a much higher reference

Figure 5.8: A comparison of inpainting. The upper result is generated by an ablation model trained without center cropping.

Table 5.3: Quantitative comparison of FIDs with ablation models at the resolution of 512^2. I use the uniform noise scheduler [84] for validation. Tested CFG scales are represented by GS-3 and GS-5, where optimal results are usually achieved. †: Tested at epoch 5. ‡: Tested at epoch 7.

Fréchet inception distance (50K-FID) ↓		
Model	GS-3	GS-5
CLS token, Proj-0.1	10.5273	10.3981
CLS token, CLS-0.1	17.6103	24.2609
†*Drop-0.5*	7.9077	8.2407
‡*Drop-0.5*	8.1842	9.1032
†*Noisy trained*	9.4761	10.9010
‡*Proposed model*	7.3676	**6.8551**

drop rate 50% to constrain the distribution shift. The loss function is the MSE error between ground-truth noise and the model's noise prediction, formulated as

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{E}(y),\epsilon,t,s,r}[\|\epsilon - \epsilon_\theta(z_t, t, s, \tau_\phi(r))\|_2^2], \tag{5.3}$$

A qualitative comparison is given in Figure 5.7 to show the advancement achieved by the second-stage fine-tuning.

5.4.3 Center cropping

Image-guided networks trained using both conditions show an inability to inpainting, caused by their sensitivity to sketch edges and view-related embeddings, which are implicitly expressed by image prompts. An example is shown in Figure 5.8. The upper result is semantically correct but visually unsatisfying due to its narrow composition. Consequently, I applied center cropping only to sketch inputs, while other pre-processing was simultaneously applied on both sketch and ground truth during training. Thus, the network learned to generate perceptually pleasant content in the margins. The effectiveness of center cropping degrades without the proposed noisy training.

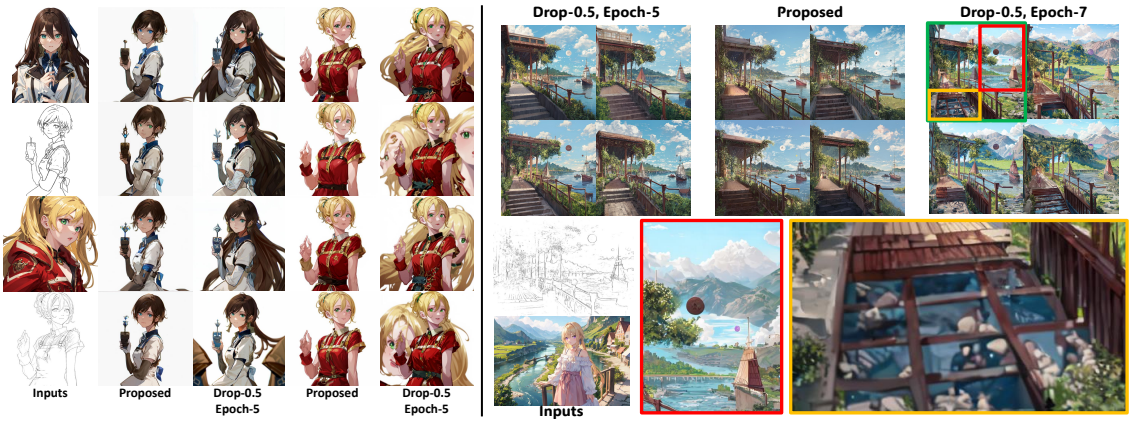


Figure 5.9: Results generated in one batch by respective models. As seen in the left comparison, the five-epoch *Drop-0.5* model shows a much higher probability of generating spatial entanglement compared to the proposed model. This tendency increases as training continues, highlighted in the right comparison, where compositions of results generated by the seven-epoch Drop-0.5 model are visually chaotic.

5.4.4 Experimental validation

Architecture. As the first trial on using DMs for reference-based sketch colorization, I tested different training strategies and architecture designs. All ablation models were trained for 7 epochs. Since the reference drop rate is a critical factor in the image-guided generation, the reference drop rates used in the ablation training are labeled with the model name. For example, {*model_name*}*-0.1* indicates the reference drop rate is 10%.

1. Dropping model: This ablation model utilizes the same architecture as the proposed one but was trained without the noisy training to demonstrate the deterioration caused by the distribution shift. Following [101], I dropped 50% prompt inputs during training, which are reference images in our task. This model is labeled as *Drop-0.5*.

An alternative solution to reduce spatial entanglement is adopting the CLS token as prompt input instead of local tokens. As the CLS token is globally compressed, it contains much less spatial information. The following ablation models utilize the CLS token in two distinct ways:

2. Projection model: CLS token is decomposed into 256 heads through two linear layers with an in-between activation. This decomposition occurs before the token is input into the denoising U-Net. This model is labeled as *Proj-0.1*.

3. CLS model: Since the CLS token is a 1024-dimensional vector, I replace cross-attention modules with linear layers to reduce computational cost. This model is labeled as *CLS-0.1*.

Discussion. A quantitative evaluation measured by 50K-FID is shown in Table 5.3. Notably, the inferior scores of *Proj-0.1* and *CLS-0.1* models suggest that the CLS token is less effective than local tokens for training reference-based models. Besides, the spatial entanglement still appears in these models as the CLS token also contains enough spatial information to reconstruct images, inferable from IP-Adapter [98]. Therefore, I consider local tokens a better choice as reference embeddings.

For the *Drop-0.5* model, I calculated its FIDs at two different epochs to demonstrate the deterioration in image quality caused by the distribution shift, which intensifies as

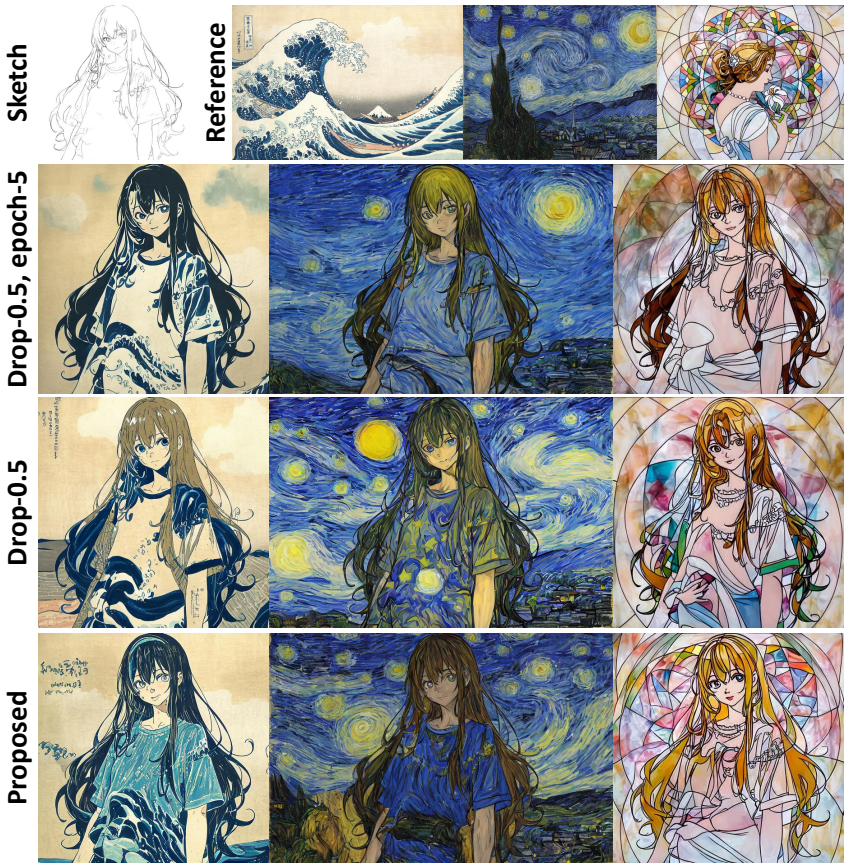


Figure 5.10: Comparison with ablation models to demonstrate the influence of training duration on style transfer. The proposed noisy training effectively slows down the optimization of identity/color semantic transfer, allowing the framework to be more optimized for high-frequency style details.

training progresses, as discussed in Section 3.2.

The FID results of *Drop-0.5* model at the 5th epoch are closer to those of the proposed model and better than those calculated at the 7th epoch. Therefore, I compare the model at 5th epoch for spatial entanglement, as illustrated in Figure 5.9, where the results of the *Drop-0.5* model are still inferior to the proposed model trained for seven epochs. As is demonstrated in Figure 5.10, the results of two models trained for seven epochs have more fine-grained textures and stories, indicating that longer training is necessary for improving style transfer performance.

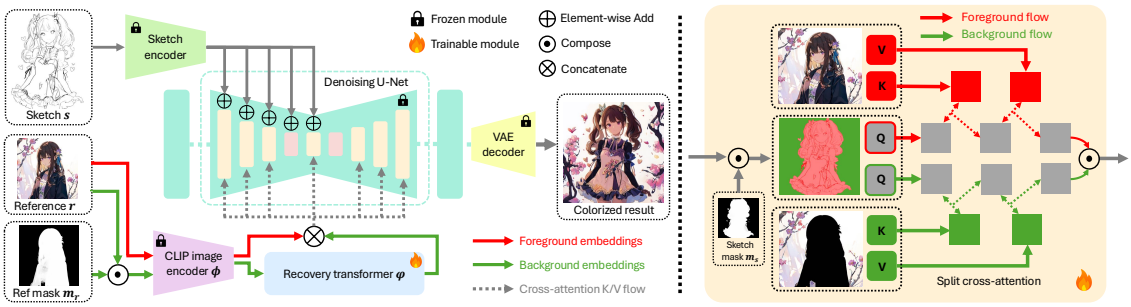


Figure 5.11: Illustration of the proposed framework *ColorizeDiffusion*-v1.5 after introducing the split cross-attention and a recovery transformer for the background embeddings. In this study, I use reference masks to separate reference images into foreground and backgroundand regions, and the CLIP image encoder ϕ to extract both regions into embeddings. The background embeddings first go through the recovery transformer φ to recover detailed information, then are concatenated with foreground embeddings as final K and V inputs for split cross-attention. The equation of split cross-attention is given in Eq. 5.4.

5.5 Low-Rank Fine-Tuning for Foreground–Background Separation

Although the proposed two-stage training strategy markedly enhances colorization fidelity and substantially mitigates spatial entanglement, residual artifacts can still arise when reference features intended for the character leak into background regions. To further suppress this leakage, we introduce a *split cross-attention* (SCA) module that replaces the vanilla cross-attention blocks inside the frozen, two-stage-optimized denoising backbone. The updated framework is illustrated in Figure 5.11.

5.5.1 Motivation

Empirically, I find that spatial entanglement originates from uncontrolled propagation of reference embeddings: semantics tied to the character (e.g., hair or body structure) are sometimes copied into non-sketch pixels. This occurs because the original cross-attention has no explicit notion of *where* the reference embeddings should be transferred to. My remedy is to route information flow along two disjoint channels, foreground and background, so that each region of the sketch attends only to semantically compatible regions of the reference.

5.5.2 Split Cross-Attention (SCA)

Let the forward (noisy) features at timestep t be $\mathbf{z}_f$ (foreground) and $\mathbf{z}_b$ (background). Let $\mathbf{e}$ be the full reference embedding extracted by the frozen encoder $\phi(\cdot)$, and $\mathbf{e}_b = \varphi\big(\phi(\mathbf{r}_b)\big)$ be the background-only embedding processed by our trainable *recovery transformer* $\varphi(\cdot)$. Here, $\mathbf{r}_b$ is the background crop of the reference, obtained via a semantic matting network. The split cross-attention is then

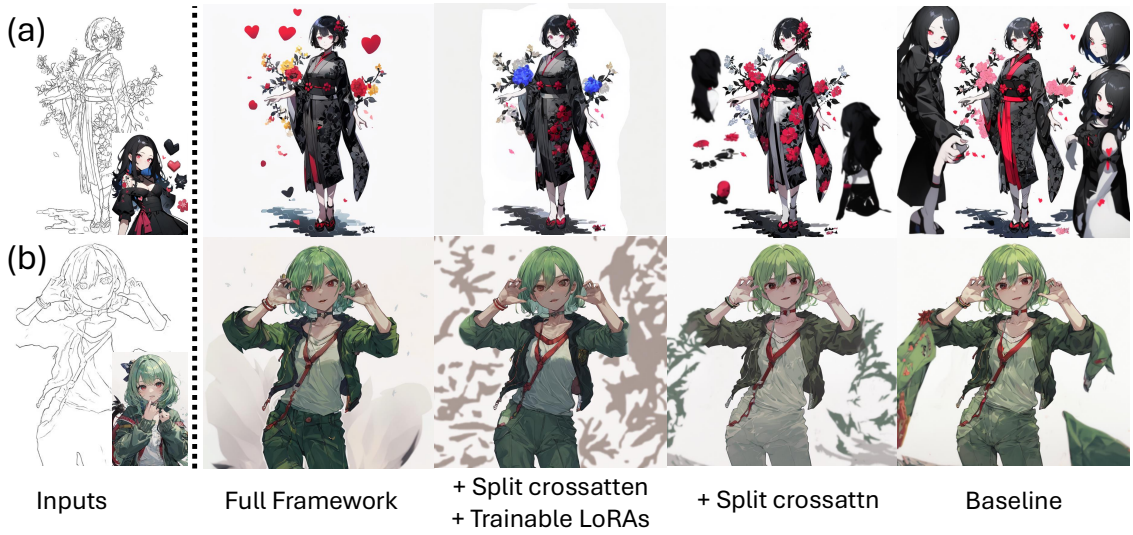


Figure 5.12: Results of the ablation study. The baseline model demonstrates significant spatial entanglement; incorporating split cross-attention reduces artifacts, the trainable LoRAs improve color saturation and details, and the proposed complete pipeline produces high-quality results free of artifacts.

$$\mathbf{y} = \begin{cases} \text{Softmax}\big(\frac{(\hat{W}_f^q \mathbf{z}_f)(\hat{W}_f^k \mathbf{e})^\top}{\sqrt{d}}\big)(\hat{W}_f^v \mathbf{e}) & \text{if } m_s > \tau_s, \\ \text{Softmax}\big(\frac{(\hat{W}_b^q \mathbf{z}_b)(\hat{W}_b^k \mathbf{e}_b)^\top}{\sqrt{d}}\big)(\hat{W}_b^v \mathbf{e}_b) & \text{otherwise,} \end{cases} \tag{5.4}$$

where d is the key dimensionality, m_s is the sketch foreground mask, and τ_s is a user-adjustable threshold. We write the fine-tuned weights as $\hat{W}^{\{q,k,v\}}_{\{f,b\}} = W^{\{q,k,v\}} + \Delta W^{\{q,k,v\}}_{\{f,b\}}$, where $W^{\{q,k,v\}}$ are the frozen backbone weights and ΔW are the low-rank updates.

Low-Rank Adaptation. To keep training lightweight and preserve the carefully tuned dynamics of the backbone, I adopt *LoRA* [31]. Separate LoRA adapters of rank 16 (foreground) and rank 4 (background) are sufficient: the character channel benefits from higher capacity to model diverse appearance, while the more homogeneous background needs only coarse adjustment.

5.5.3 Mask Generation and User Control

Foreground masks for the sketch (m_s) and reference (m_r) are extracted with lightweight segmentation networks [67,80,108]. Because automatic segmentation may misclassify thin line art or complex scenery, users can override the defaults via thresholds τ_s and τ_r, enabling interactive refinement without retraining. Masks values $m_s > \tau_s$ and $m_r > \tau_r$ are considered foreground.

5.5.4 Recovery Transformer φ

Directly injecting raw background embeddings into the SCA was found to degrade structural fidelity and stylistic coherence. The recovery transformer φ (a 4-layer Transformer encoder with 64-head_dim self-attention) re-encodes background embeddings, harmonizing their statistics with those of the foreground pathway.

5.5.5 Experimental validation

The proposed method aims to address spatial entanglement by simulating the animation workflow. To demonstrate the effectiveness of this workflow, I set up three frameworks: 1) a baseline model without split cross attention, trainable LoRAs, and recovery transformer, 2) a baseline model with split cross attention but no trainable LoRAs and recovery transformer, and 3) the proposed full framework. We show the qualitative comparison in Figure 5.12 to validate the effectiveness of the proposed modules. The baseline model causes severe spatial entanglement in generating additional figures in (a) and undesired clothes in (b). The application of split cross-attention mitigates the spatial entanglement but still causes artifacts and degrades the color saturation and details of the results. Collaborating split cross attention with trainable LoRAs improves the quality of results and further improves the background, but still suffers from artifacts. The proposed full framework, enhanced by recovery transformers, effectively eliminates spatial entanglement and synthesizes colorization results that have clear boundaries and rich details and textures, and loyally preserves the color distribution of reference images.

Figure 5.13: The distribution shift artifacts increase when more detailed information is injected in the common training scheme.

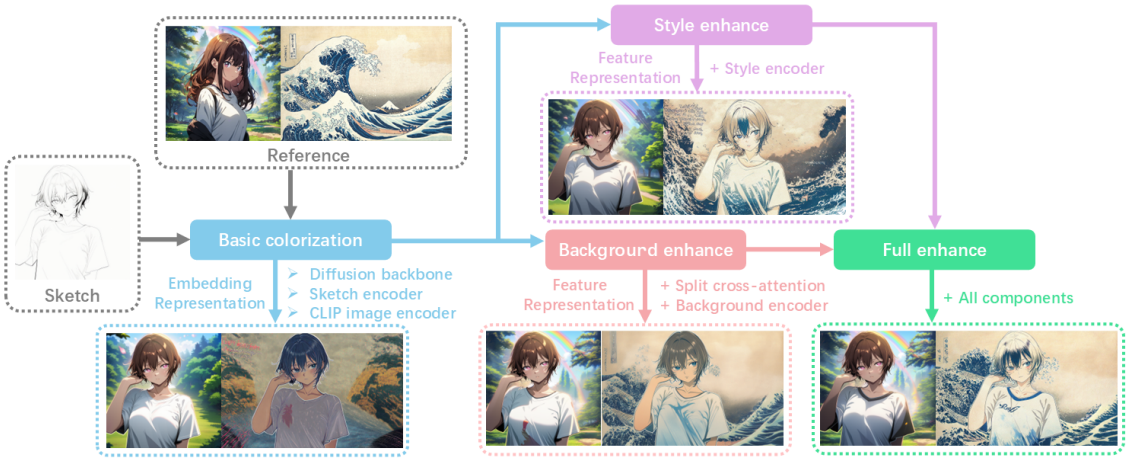


Figure 5.14: Illustration of the further-improved reference-based colorization pipeline. In this framework, reference representations are separated based on their semantic levels and are respectively transferred into the denoising backbone through different neural modules.

5.6 Separation of reference representation

Previous frameworks rely on a frozen embedder, pre-trained for generic multimodal tasks, to extract reference information. Although this approach suffices for transferring chromatic and parts of achromatic cues, its high-level nature inevitably discards many low-level stylistic details, especially for texture, shading, and stroke.

A straightforward remedy is to jointly optimize the reference encoder together with the generative backbone, where the MSE loss used in the diffusion training would drive the reference encoder to extract the reference representations as low as possible to best reconstruct ground truths. Yet, as discussed in Section 4.8, joint optimization introduces a stronger distribution shift between the reference and sketch domains. This misalignment manifests as conspicuous artifacts that disregard sketch semantics and spatial composition, ultimately degrading visual fidelity.

However, by observation, I notice that since the artifacts caused by different level

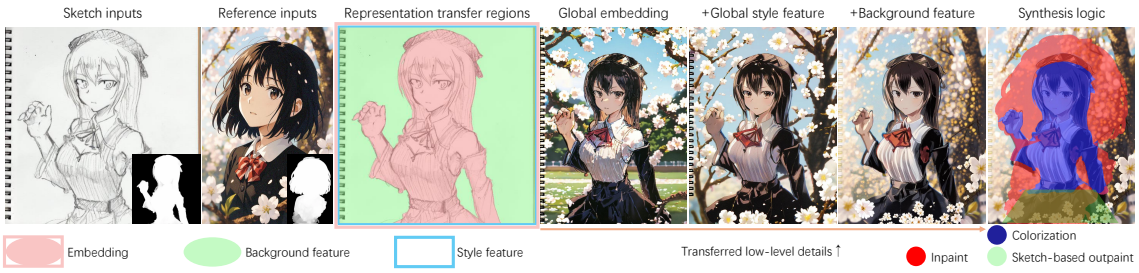


Figure 5.15: Illustration of the proposed reference-based sketch colorization workflow. To eliminate artifacts and enhance colorization quality, we separate colorization into three parts, leveraging foreground masks extracted from the reference and sketch inputs: embedding guidance for sketch-covered regions, style modification for global details, and low-level transfer for non-sketch regions. Moreover, the network should be able to properly inpaint the missing regions based on neighboring features in the sketch and reference images. As highlighted by red, the proposed network inpaints the skirt based on prior knowledge from the sketch and the flowers based on neighboring features from the reference.

representations show different properties, as visualized in Figure 5.13, we can effectively separate them for the guidance of different regions. Leveraging the foreground-background separation proposed in Section 5.5, we can further separate the reference representations used for different regions.

This separation of representations is visualized in Figure 5.14, and Figure 5.15 illustrates how different synthesis logits and reference representations are combined for colorization.

5.6.1 Architecture and multi-stage training strategy

To achieve the colorization logits established in Figure 5.14, and Figure 5.15, I propose a multi-stage training strategy to introduce respective trainable modules step by step. The full training pipeline is given in Figure 5.16. A significant difference between this framework with the one introduced in Section 5.5 is the replacement of the recovery transformer with a background encoder. The recovery transformer receives background embeddings as input and output recovered background embeddings, while in this design, the background encoder receives background latents, a low-level visual representations, instead of embeddings.

To inject reference representations at different levels, this multi-stage training strategy optimizes the diffusion backbone, the background encoder, and the style encoder separately: 1. Colorization pre-training stage: this training follows the strategy introduced in Section 5.4 to avoid severe deterioration in the segmentation and perceptual quality of results; 2. Foreground-background separated training stage: I add the split cross-attention module and the background encoder into the major framework and optimize them with other parameters frozen. This stage helps eliminate spatial entanglement caused by the reference embeddings and enhances the synthesis of backgrounds and; 3. Hybrid training stage of style encoder: the style encoder is optimized with the background encoder and split cross-attention not trained but randomly activated at a rate of 50% to generate extra conditions for the denoising backbone and other parameters frozen. In stage 2 and stage

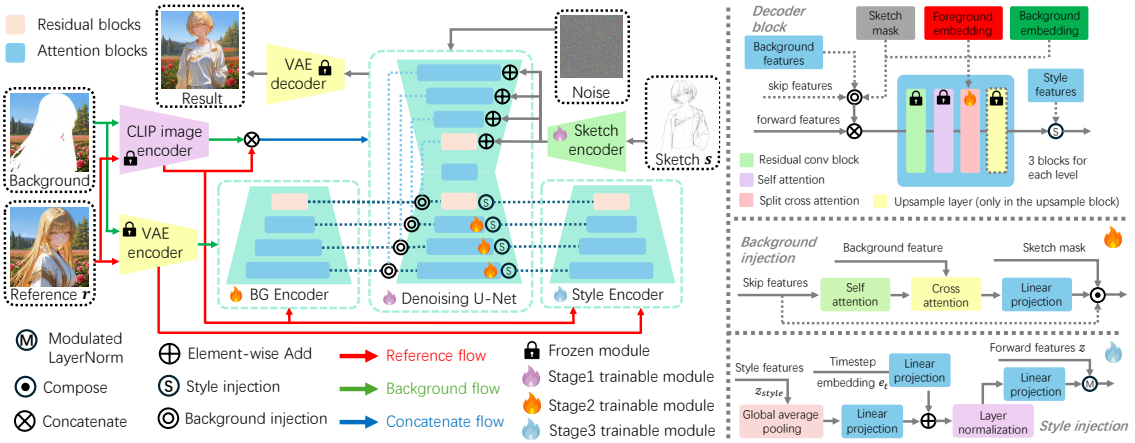


Figure 5.16: Illustration of the proposed framework. The Left shows the whole framework. The CLIP image encoder and the VAE encoder are fixed during training. The extracted image embeddings and latent images are injected into the corresponding modules in the same way as standard LDM. The denoising U-Net, the background encoder, and the style encoder are trained separately in 3 stages. The Right shows the detailed architectures of a decoder block, a style injection block, and a background injection block.

3, the reference embeddings for the denoising U-Net are dropped at a fixed rate of 50%. Given noise ϵ, sketch $\boldsymbol{s}$, ground truth $\boldsymbol{y}$, encoded latent representations (forward features) $\boldsymbol{z}_t$ at timestep t, VAE encoder $\mathcal{E}$, denoising U-Net and sketch encoder θ, background encoder with background injection φ_{bg}, style encoder with style injection φ_{style}, and CLIP image encoder ϕ, the training objective for all training stages can be defined as

$$\mathcal{L}(\vartheta) = \mathbb{E}_{\mathcal{E}(\boldsymbol{y}),\epsilon,t,\boldsymbol{s},\boldsymbol{c}}[\|\epsilon - \epsilon_\theta(\boldsymbol{z}_t, t, \boldsymbol{s}, \boldsymbol{c})\|_2^2], \tag{5.5}$$

where ϑ and $\boldsymbol{c}$ represent the optimization targets and conditional inputs, and a detailed explanation for each stage is as follows. **Stage 1**: ϑ represents the denoising U-Net and the sketch encoder, and $\boldsymbol{c}$ represents image embeddings $\boldsymbol{e} = \phi(\boldsymbol{r})$; **Stage 2**: ϑ represents the background encoder and injection modules, as well as LoRAs inside split cross-attention layers, and $\boldsymbol{c}$ represents background embeddings $\boldsymbol{e}_{bg}$, sketch mask $\boldsymbol{m}_s$, and background features $\boldsymbol{z}_{bg} = \varphi_{bg}\big(\mathcal{E}(\boldsymbol{r}_{bg}), \boldsymbol{e}\big)$; **Stage 3**: ϑ represents the style encoder and injection modules, and $\boldsymbol{c}$ represents style features $z_{style} = \varphi_{style}\big(\mathcal{E}(\boldsymbol{r}), \boldsymbol{e}\big)$. As ground truths are directly used as references, $\boldsymbol{r}$ can be replaced with $\boldsymbol{y}$ throughout all equations.

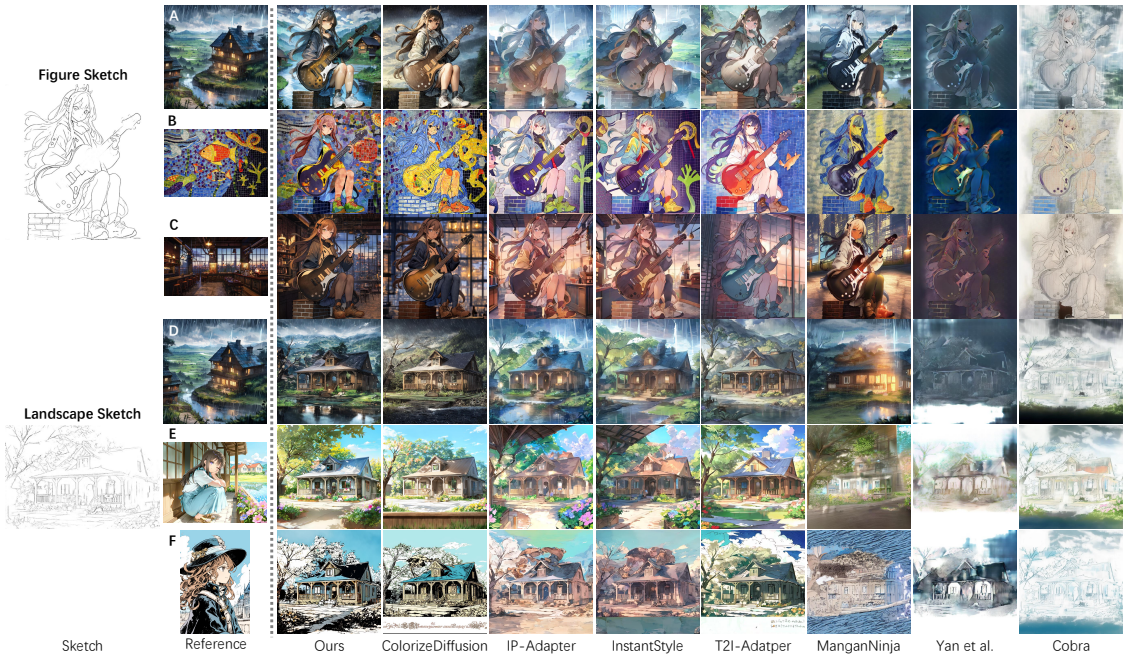


Figure 5.17: A comparison between the proposed models with baseline methods, where both "ours" and *ColorizeDiffusion* are the proposed frameworks of this thesis.

5.7 Comparison with baseline methods

In this section, I made a comprehensive comparison with the state-of-the-art baseline methods that focus on reference-based sketch colorization across qualitative evaluation, quantitative evaluation, and user studies, to demonstrate the significant superiority of the proposed framework. Step-by-step, the proposed framework establishes an effective reference-based denoising backbone, eliminates spatial entanglement backgrounds, and finally further improves the transfer of style/background details. These improvements make the proposed framework able to produce visually pleasant results for various inputs and transfer various styles of reference representations. Comparisons given in this section selected all state-of-the-art baseline methods as baselines to demonstrate the superiority of the proposed framework. Specifically, there are two groups involved in these comparisons developed based on this thesis, and the two groups are labeled as "**ours**" and "**ColorizeDiffusion**" in the figures, respectively.

5.7.1 Qualitative comparison

I present two qualitative comparisons in Figure 5.17 and Figure A.1. Among the baselines, MangaNinja [55] and Cobra [110] are specifically designed for character colorization. These methods mitigate spatial entanglement artifacts by using training references composed of different images featuring the same identities. However, this strategy often leads to overfitting, resulting in notable performance degradation when the input sketch and reference image are semantically or geometrically misaligned. This limitation reduces the generalization ability of their models. Consequently, their models exhibit limited generalization. This issue is evident in Figure A.1, where results in rows (a)–(g) show significantly lower

Table 5.4: A full quantitative evaluation on 768^2 resolution between the proposed framework and baseline methods. †: These evaluations randomly selected color images as references, making them close to real-application scenarios. ‡: Ground truth color images were deformed to obtain semantically paired and spatially similar references for evaluations. §: Tested at 512^2 resolution.

Method	†FID ↓	‡PSNR↑	‡MS-SSIM↑	‡CLIP similarity↑
Ours	**5.6330**	29.3626	**0.7081**	**0.9056**
ColorizeDiffusion	9.6423	28.7215	0.5899	0.8753
IP-Adapter	38.9232	28.5124	0.5464	0.8632
InstantStyle	40.2134	28.0921	0.4467	0.8039
T2I-Adapter	41.1569	28.1321	0.3194	0.7134
§*MangaNinja*	42.9741	**29.5741**	0.6715	0.7304
§Yan et al.	27.0032	29.1293	0.5239	0.8894

visual quality compared to those in rows (h)–(j). The GAN-based method, introduced in Chapter 4, struggles to synthesize accurate colors and backgrounds for complex inputs due to limitations in its neural backbone. Compared to these baselines, the proposed multi-representation framework consistently generates high-quality colorized images with strong similarity to the reference images while effectively avoiding artifacts across a wide range of content.

Furthermore, I exhibit cross-content colorization of figure and landscape sketches in Figure 5.17. This scenario lacks clear correspondence between input identities, so subjective evaluations usually prioritize the similarity of style, color scheme, and texture/stroke details, all of which require a reasonable transfer of low-level visual features. The proposed framework significantly outperforms baseline methods in this regard while closely adhering to the sketch semantics to achieve clearer visual segmentation outcomes.

5.7.2 Quantitative comparison

Quantitative evaluation is critical for objectively evaluating the performance of reference-based sketch colorization. I conducted FID evaluation, introduced in Section 3.3, on the entire validation set for the DM-based framework, which contains 52k+ (sketch, reference) image pairs. Reference images were randomly selected during the validation. I tested the multi-scale structural similarity index measure (MS-SSIM), peak signal-to-noise ratio (PSNR), and CLIP score for assessing the similarity between generated images and the given ground truth. As these metrics require the reference to be aligned with ground truth, I selected 5000 color images as ground truth to generate extracted sketches and deformed references, where references were deformed using thin plate spline (TPS) transformation. I show the results of quantitative evaluation in Table 5.4, where the proposed method significantly outperforms in all evaluations owing to the removal of artifacts, higher fidelity to the sketch composition, and stronger style transfer ability.

5.7.3 User study

To further reveal the subjective evaluation of the proposed method and existing methods by real persons, I demonstrate a user study with 30 participants from Anime lovers

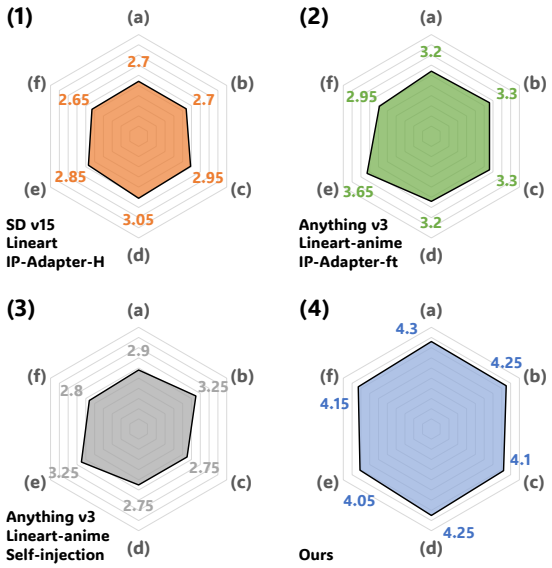


Figure 5.18: Results of user study. The proposed method is preferred across all shown methods in overall quality and geometric preservation.

Proposed	Controlnet-based	Cobra [110]	MangaNinja [55]	GAN-based
1387ms	2617ms	1754ms	3147ms	57ms

Table 5.5: Inference time for different architectures to generate a 1024^2 image. GAN-based frameworks only need one forward pass, so they are much faster than DM-based frameworks.

communities invited to select the best results with two criteria: the overall colorization quality and the preservation of the geometric structure of the sketches. 35 image sets are prepared, and each participant is shown 15 image sets for evaluation. I present to participants the colorization results of the proposed method and those generated by six existing methods for each image set. I present the results of the user study in Figure 5.18, with the results showing that our proposed method has received the most numbers of preferences across all the methods illustrated. For further validation of the comparison, the Kruskal-Wallis test is employed as a statistical method. The results demonstrate that our proposed method outperforms all previous methods significantly in terms of user preference with a significance level of $p < 0.05$. All the images shown in the user study are included in the supplementary materials.

5.7.4 Inference speed

Although diffusion-based (DM) frameworks inevitably incur longer inference times than their GAN-based counterparts, the proposed DM framework remains faster than most existing DM baselines, detailed in Table 5.5. Competing approaches often add a full extra U-Net pass—either for a ControlNet branch [60, 98, 101] or for a dedicated reference network [55], whereas our design avoids these additional forward steps. Consequently, it delivers superior speed without sacrificing quality.

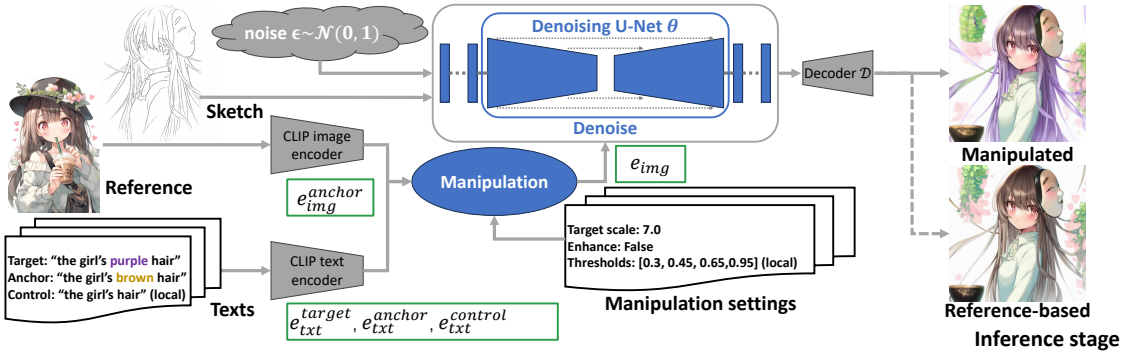


Figure 5.19: The inference pipeline for the proposed local text-based manipulation. The local tokens are edited before being input to the denoising U-Net.

5.8 Text-based Embedding Manipulation

Finally, I introduce a *zero-shot text–driven manipulation* mechanism that endows the proposed framework with prompt-level controllability. Editing a reference image is considerably harder than rewriting a text prompt in conventional (T2I) systems, because the prompt itself is a high-dimensional visual embedding. To overcome this limitation, I introduce a zero-shot semantic-interpolation scheme that operates directly in CLIP's latent space. As described in Section 5.4, I evaluate an ablation variant, denoted *CLS* model, that discards all local tokens and conditions the denoising U-Net solely on the CLIP [CLS] token extracted from the reference image.

This design is inspired by DALL-E 2 [71], which showed that the [CLS] token furnishes a compact yet semantically rich summary for image-conditioned generation while supporting smooth attribute morphing without additional fine-tuning. By linearly interpolating the [CLS] vector toward text-derived attribute directions (e.g., "blue hair," "red eyes," or "blue clothes"), we can continuously modulate the target appearance before feeding the embedding into the U-Net θ. Figure 5.19 depicts the resulting text-based manipulation pipeline: the denoised latent is fused with sketch features to introduce spatial control, then manipulation text inputs are encoded by the CLIP text encoder into the same embedding space as image embeddings and edit the image embeddings on specific visual attributes, finally the modulated image embeddings are input to the denoising U-Net to perform the "reference-based" sketch colorization as normal.

5.8.1 Global Text-Based Manipulation

The CLIP score is widely used to evaluate the correlation between a generated image and a given caption, as explained in Section 2.4.3 and Section 3.3. It is calculated as the projection of the image CLS token onto the text CLS token. While using image tokens as prompt inputs, we can directly modify the generated results using this projection-based correlation. To simplify the expression, we denote the extracted image tokens (previously represented as $\tau_\phi(r)$) and the normalized text CLS token as vectors $\vec{\boldsymbol{v}}$ and $\vec{e}$, respectively. Specifically, the CLS token is denoted as $\vec{v}_{cls}$, and we can calculate the modified CLS token

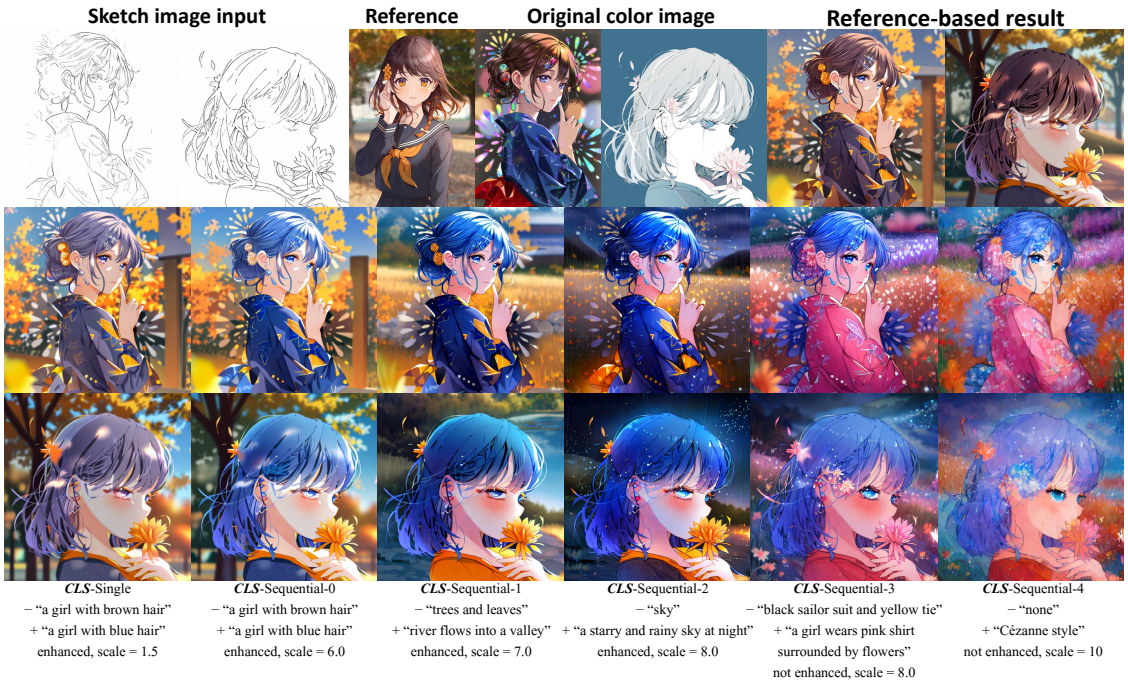


Figure 5.20: The proposed manipulation method allows sequential editing on reference-based results with specified degrees. Symbols "+" and "–" respectively denote the target text and anchor text for our text-based manipulation.

$\vec{v}_{cls}^{m}$ as

$$\vec{v}_{cls}^{m} = \begin{cases} \vec{v}_{cls} + target\_scale * \vec{e} & enhance \\ \vec{v}_{cls} + (target\_scale - \vec{v}_{cls} \cdot \vec{e}) * \vec{e} & not\ enhance \end{cases}, \tag{5.6}$$

where *target_scale* and *enhance* are user-defined parameters. They indicate the target scale of the interpolation and whether the manipulation should be enhanced to achieve a more obvious change, respectively. Similar to DALL-E-2 [71], the manipulation can be improved through the normalized embedding of an anchor text, termed $\vec{a}$. The first method, where *enhance* is set to false, calculates $\vec{v}_{cls}^{m}$ with the anchor text as

$$\vec{v}_{cls}^{m} = \vec{v}_{cls} + target\_scale * (\vec{e} - \vec{a}). \tag{5.7}$$

The global manipulation can be further enhanced by first eliminating the anchor attribute with $\vec{a}$ before adding $\vec{e}$. The modified CLS token $\vec{v}'_{cls}$ is then calculated as

$$\begin{aligned} \vec{v}'^{m}_{cls} &= \vec{v}_{cls} - (\vec{v}_{cls} \cdot \vec{a}) * \vec{a}, \\ \vec{v}^{m}_{cls} &= \vec{v}'^{m}_{cls} + (target\_scale - \vec{v}'^{m}_{cls} \cdot \vec{e}) * \vec{e}. \end{aligned} \tag{5.8}$$

However, enhancing the manipulation with an anchor text would make unrelated attributes more likely to be jointly changed. The sequential manipulation of $\vec{v}_{cls}$ is shown in Algorithm 1. The target scales proposed in [4, 15] can generate reasonable results. Sequential editing results are visualized in Figure 5.20

Algorithm 1: Sequential global manipulation.

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 $\vec{v}_{cls}$

5.8.2 Local Text-Based Manipulation

As noted in Section 5.3, the proposed framework conditions the diffusion backbone on *local* CLIP tokens rather than the global [CLS] token. The global manipulation is therefore ineffective because it ignores the spatial specificity encoded in *local* tokens. To overcome this limitation, I devise a *zero-shot, semi-automatic* manipulation algorithm that operates directly on local tokens while still accepting *arbitrary* text prompts.

For better clarification, I first define three terms used in the proposed local manipulation: *dscale*, Position Weight Vector (PWV) $\boldsymbol{m}$, and PWV $\boldsymbol{\omega}$. We already know that the correlation between an image and a caption can be evaluated through the CLIP projection, formulated as $corr = \vec{v}_{cls} \cdot \vec{e}$. We have observed that the local tokens also demonstrate the ability of zero-shot segmentation, which suggests that such correlation is also computable using local tokens. Therefore, we extend the calculation of the correlation vector as $corr_i = \vec{v}_i \cdot \vec{e}$, with $i \in \{cls, 1, 2, .., n\}$ and n being the total number of local tokens, which is 256 for the adopted OpenCLIP-H, and define $dscale_i^{AB} = \vec{v}_i^A \cdot \vec{e} - \vec{v}_i^B \cdot \vec{e}$. Our aim is to use $dscale_{cls}$ and PMVs $\boldsymbol{m}, \boldsymbol{\omega}$ to simulate $\boldsymbol{dscale}^{AB}$, where $\boldsymbol{dscale}^{AB} = [dscale_1^{AB}, .., dscale_n^{AB}]$. If the difference between images A and B can be fully described using the text embedding $\vec{e}$, we can approximate $\vec{\boldsymbol{v}}^A$ as

$$\vec{\boldsymbol{v}}^A = \vec{\boldsymbol{v}}^B + \boldsymbol{dscale}^{AB} \tag{5.9}$$

In our observations, we noticed that the local and CLS tokens exhibit different directional changes when projected onto the text embedding. We find that for the given text "a girl

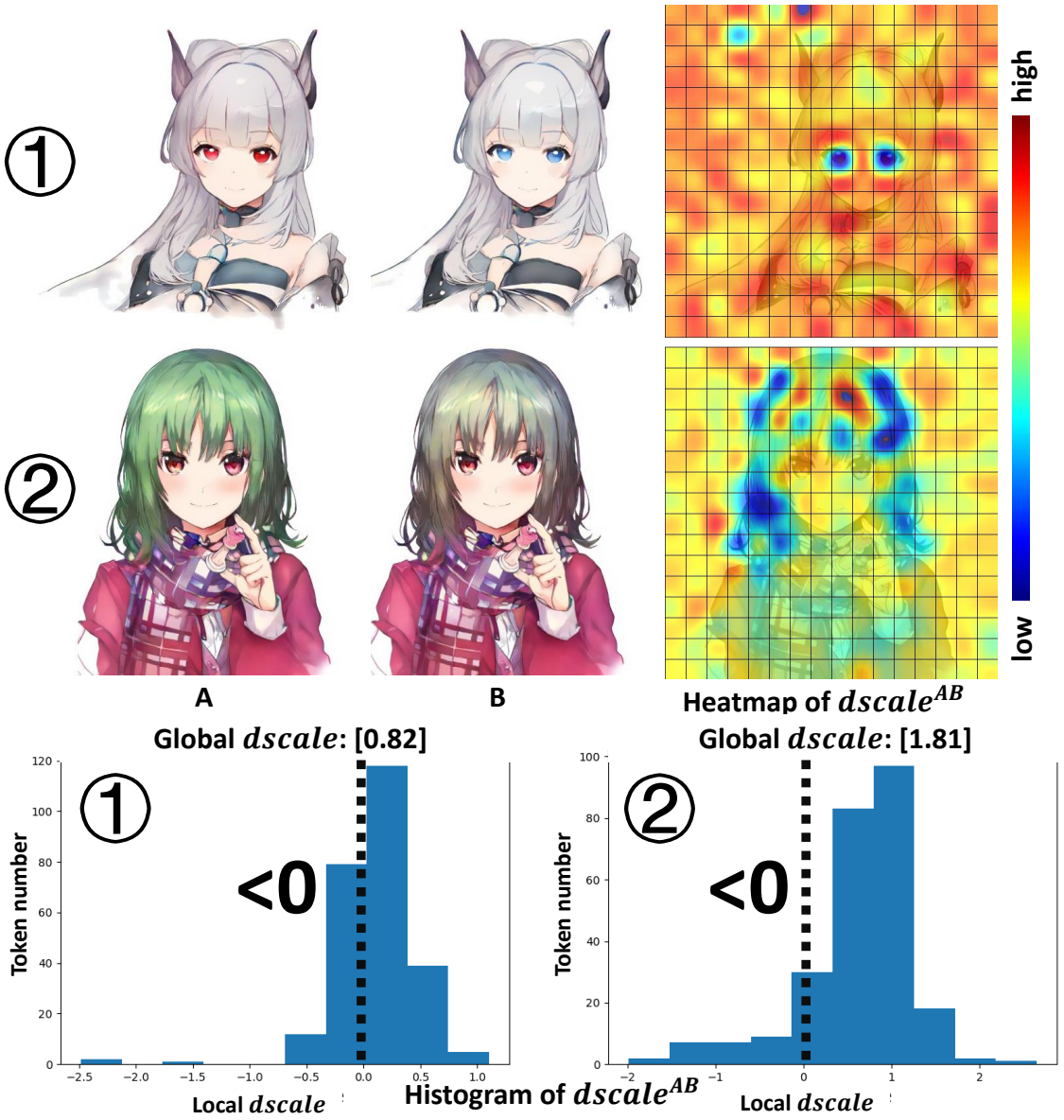


Figure 5.21: Visualization of $\boldsymbol{dscale}^{AB}$ corresponding to the texts "the girl's red eyes" (upper) and "the girl's green hair" (lower), respectively.

with green hair," as the hair becomes greener, the projection of the CLS token along the text embedding direction lengthens, which is labeled as *corr* on top of the histograms in Figure 5.21. Conversely, the projections of the most relevant local tokens decrease, while those of irrelevant tokens increase. These dynamics can be observed from the heatmaps of $\boldsymbol{dscale}^{AB}$, where regions closely related to the text are marked in blue. Given that blue is used to represent lower values, the heatmaps clearly indicate that the $\boldsymbol{dscale}^{AB}$ values for these regions are negative, as corroborated by the histograms.

We use the control prompt whose embedding is denoted as $\vec{c}$ to locate the region of local manipulation and calculate the PWV $\boldsymbol{m}$ as

$$\boldsymbol{m} = \mathcal{F}(\vec{\boldsymbol{v}} \cdot \vec{c}), \tag{5.10}$$

where $\mathcal{F}$ indicates the min-max normalization. By leveraging the correlation PWV $\boldsymbol{m}$, we

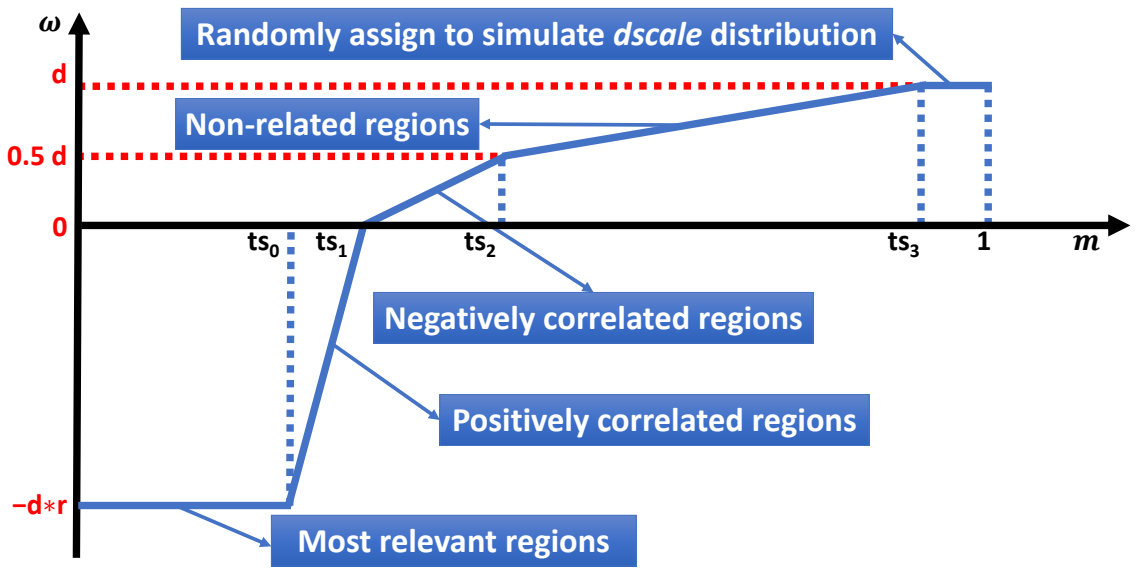


Figure 5.22: Plotting ω_i as a function of m_i in Eq. 5.11. We divide the domain into five intervals to reduce the influence of the manipulation on unrelated attributes.

formulate the PWV $\boldsymbol{\omega}$ as

$$\omega_i = \begin{cases} -d * r, & m_i \leqslant ts_0 \\ -d * r + d * r * \frac{m_i - ts_0}{ts_1 - ts_0}., & ts_0 < m_i \leqslant ts_1 \\ 0.5 * d * \frac{m_i - ts_1}{ts_2 - ts_1}, & ts_1 < m_i \leqslant ts_2 \\ 0.5 * d + 0.5 * d * \frac{m_i - ts_2}{ts_3 - ts_2}, & ts_2 < m_i \leqslant ts_3 \\ d, & m_i > ts_3 \end{cases} \tag{5.11}$$

where m_i and ω_i represent the i-th element of $\boldsymbol{m}$ and $\boldsymbol{\omega}$, respectively, with $i \in \{1, .., n\}$. This function is illustrated in Figure 5.22. In this equation, d is computed as

$$d = \begin{cases} target\_scale - \vec{v}_{cls} \cdot \vec{a}, & enhance \\ target\_scale - \vec{v}_{cls} \cdot \vec{e}. & not\ enhance \end{cases}. \tag{5.12}$$

The hyperparameters r and ts_i in Eq. 5.11 denote the strength ratio for the most pertinent areas and the thresholds for differentiating all areas of the image, respectively. The rough definitions of different threshold intervals are given in Figure 5.22. The default settings for the hyperparameters r 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. Accordingly, we can formulate the adjustment equation for the local tokens as

$$\vec{\boldsymbol{v}}^m = \vec{\boldsymbol{v}} + (\boldsymbol{\omega} + \boldsymbol{\beta} * \vec{\boldsymbol{v}} \cdot \vec{a}) * (\vec{e} - \vec{a}), \tag{5.13}$$

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

$$\vec{\boldsymbol{v}}^m = \vec{\boldsymbol{v}} + \boldsymbol{\omega} * \vec{e}. \tag{5.14}$$

Algorithm 2: Sequential local manipulation.

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 5.11
    $\vec{\boldsymbol{v}} \leftarrow \vec{\boldsymbol{v}} + (\boldsymbol{\omega} + \boldsymbol{\beta} * \vec{\boldsymbol{v}} \cdot \vec{a}) * (\vec{e}[i] - \vec{a}[i])$
  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 5.11
    $\vec{\boldsymbol{v}} \leftarrow \vec{\boldsymbol{v}} + \boldsymbol{\omega} * \vec{e}[i]$
  end
end
return $\vec{\boldsymbol{v}}$

This formulation is similar to Eq. 5.9. This calculation can also 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.

5.8.3 Experimental validation of local manipulation

Since the proposed local manipulation necessitates a PWV to adjust local tokens adaptively according to their association with the control text, leading to a more difficult manipulation. Figure 5.23 demonstrates that local manipulation can progressively adjust a specific visual attribute and showcases sequential manipulation that alters backgrounds and hair color in sequential steps. Both figures adopt real sketch images.

Although our method effectively adjusts visual attributes, a significant challenge arises from the proposed local manipulation. Observing the heatmaps in Figure 5.23, which were generated from projections on the control text embedding, reveals substantial errors in segmentation, which complicates the manipulation process.

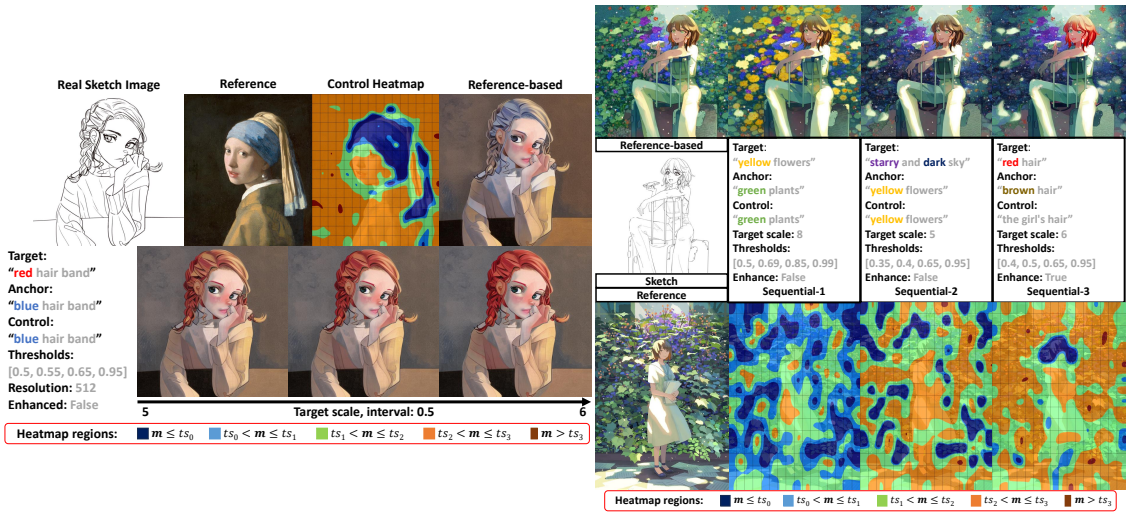


Figure 5.23: Visualization of the proposed local manipulation and its corresponding sequential editing. The stratified heatmap displays the correlation vector $\boldsymbol{m}$ calculated on the basis of the control text.

Chapter 6

Extensive discussion on training paradigm

As introduced in Section 2.4.4 and Section 3.1, reference-based sketch colorization frameworks fall into two categories defined by whether the reference encoder is trainable. This chapter analyzes the rationale for the design adopted in this thesis, which freezes a large pre-trained reference encoder, and explains the underlying mechanisms by which this scheme yields significant gains in colorization performance and generalization. In particular, I show that fixing the reference feature space promotes mapping in a stable high-level representation, reduces co-adaptation and overfitting, and improves optimization stability.

6.1 Representation levels of conditional input

Based on experimental analysis, this thesis categorizes the hidden representations of different information involved in the sketch colorization into five levels according to the semantic granularity of the conditional inputs involved in the sketch colorization, illustrated in Figure 6.1. From the highest conceptual abstraction to the lowest visual detail, these five classes are defined as follows:

1. **Discrete embedding level**. This is the highest level, with most representations at this level being text representations encoded from natural-language prompts. In practice, these are the token embeddings produced by text encoders that have been pre-trained on various language tasks. Text embeddings cannot describe the visual properties precisely with degrees.

2. **Continuous embedding level** is the level of image embeddings, which are encoded by image encoders trained for visual understanding tasks. Unlike lower-level representations, these embeddings can often be approximated as a linear combination of approximately disentangled, normalized text embeddings corresponding to visual attributes. This combination can be reflected by formulating an embedding e_{img} from an image as

$$e_{img} = \sum_{attr} \frac{e_{txt}^{attr}}{\|e_{txt}^{attr}\|} \cdot scale_{attr}, \tag{6.1}$$

where e_{img}, e_{txt}^{attr} denote the image embedding, text embedding of visual attribute

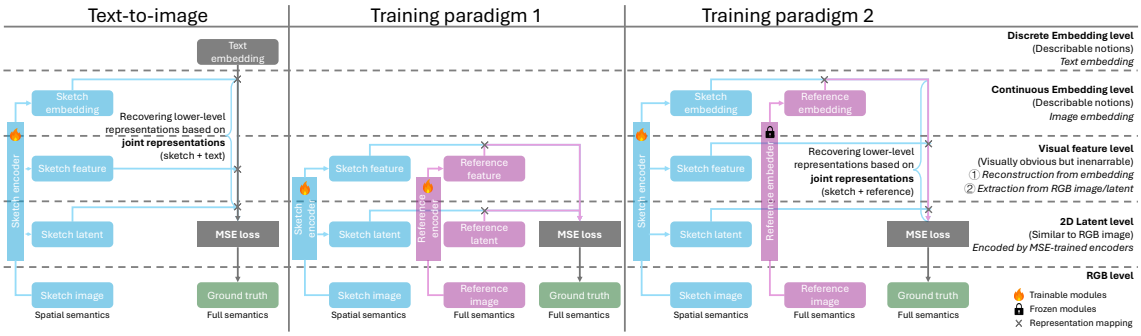


Figure 6.1: Illustration of representation transition in T2I-based colorization and the two training schemes for reference-based sketch colorization; with the conceptual role of each level shown in brackets and its usual expression source given in italics on the right.

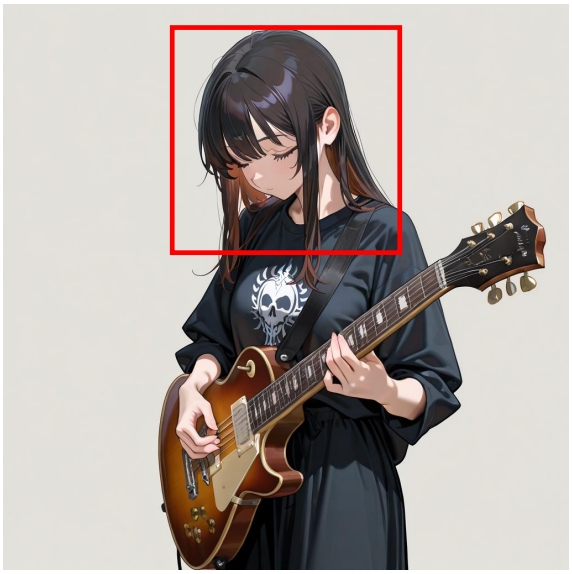


Figure 6.2: The image embedding of the highlighted region $e_{img}^{\text{hair color}}$ can be approximately represented as $\frac{e_{txt}^{\text{black hair}}}{\|e_{txt}^{\text{black hair}}\|} \cdot 7 + \frac{e_{txt}^{\text{brown hair}}}{\|e_{txt}^{\text{brown hair}}\|} \cdot 6.5$.

attr of the image, and $scale_{attr} = \frac{e_{img} \cdot e_{txt}^{attr}}{\|e_{txt}^{attr}\|}$. An example is given in Figure 6.2.

3. **Visual feature level** is for intermediate activations within a visual network (whether trained for understanding or for generation). Representations of this level capture mid-level cues, typical texture, shape, and pattern motifs. For example, an elliptical patch that could either be the mouth/eye of a character, or a rugby ball, depending on context.

4. **Latent level**, a compact representation specific to latent diffusion models. To curb computational cost, most diffusion steps are carried out in the latent space learned by a variational auto-encoder (VAE), as detailed in Chapter 2. Because the VAE is trained for near-lossless reconstruction, the information at this level is largely equivalent to that of the full-resolution RGB image.

5. **RGB level**, which is the raw pixel domain representing images or video frames in their most granular form. It contains the direct per-pixel color values and thus provides the lowest-level visual expression.

With the definition of representation levels, we can better distinguish the differences among colorization methods. This section takes the most popular generation scheme, text-to-image (T2I) colorization, to illustrate how color information is transferred from prompt conditions to sketch inputs in data-driven sketch colorization. A standard method for incorporating text guidance is to use a frozen, pre-trained text encoder, such as CLIP [68] or T5 [70], which converts natural-language prompts into embedding vectors serving as guidance representations. Freezing the encoder during the generation optimization preserves its high-level semantic knowledge. These text embeddings are then input to the generative backbone, typically a U-Net or, in recent designs, a Diffusion Transformer (DiT) [65, 72], as key–value (KV) inputs via cross-attention modules.

Meanwhile, a notable feature of diffusion models is the utilization of mean squared error (MSE) used for restricting the grounding truth with the prediction on the latent level in latent diffusion models (LDMs). This MSE loss forces the diffusion models to inherently pursue fine-grained correspondence between outputs and ground-truth images while lacking an explicit constraint for semantic-level alignment. Since natural language prompts in T2I generation are highly abstract and lack detailed visual specificity, the generative backbone has to model the training-data distribution to project text representations from the discrete embedding level into lower levels in order to minimize the latent-level MSE loss.

In T2I-based sketch colorization, where sketch images are introduced to provide fine-grained, per-pixel details, it's foreseeable that gradients would strongly bias the sketch inputs, especially for lower-level representations at early training steps, due to the pixel-level misalignment between (generated output, ground truth). Furthermore, since the color/texture/stroke-related information is only guided by the text prompts, the network is forced to map sketch and text representations at the text embedding level. Consequently, sketch representations exert a much stronger influence than text prompts on the synthesized semantics.

6.2 Reference-based colorization with trainable reference encoder

Unlike text-guided methods, most reference-based colorization baseline methods utilize a jointly trained encoder to encode reference images into intermediate representations. The key idea of the training scheme, which jointly trains the reference encoder and the generative backbone, is to collect specific training data whose pairing patterns are similar to inference input pairs as much as possible. In this training scheme, every reference image portrays the *same* identity, typically the same character, as its corresponding ground-truth image. Because the reference encoder is optimized jointly with the generator in most cases, the model learns to transfer low-level color and texture cues. However, as emphasized in Section 3.1 and further confirmed by the diffusion-model experiments in Section 5.7, baseline methods trained under this scheme suffer dramatic performance drops at inference time. A related discussion for GANs appears in Section 4.2.3.

The deterioration that happens in baseline methods can be regarded as the specific failure mode of reference-based sketch colorization models. In the context of reference-based sketch colorization, "over-fitting" does not imply that the target distribution of the

generator collapses onto a narrow subset of the ground truth distribution. Instead, the network over-specializes to the pairing rules embedded in the training data. Formally, each rule can be viewed as a mapping

$$f_{x,r} : \mathcal{X} \longrightarrow \mathcal{R} \tag{6.2}$$

where $\mathcal{X}$ denotes the space of sketches and x is the lowest-level semantic representation extracted by the sketch encoder, while $\mathcal{R}$ denotes the space of reference images and r is the analogous representation extracted by the reference encoder. Because the dataset provides only a *finite* set of pairings

$$F = \{ f_{x_1,r_1},\ f_{x_2,r_2},\ \ldots,\ f_{x_n,r_n} \}. \tag{6.3}$$

When the *reference encoder* is **trainable** rather than frozen, it is typically optimized jointly with the generator under low-level objectives (for example, per-pixel MSE, diffusion reconstruction loss, or related pixel-wise criteria), as visualized in Figure 6.1. Although these losses stabilize visual generation performance, they can hardly provide gradients for learning embedding-level abstractions in the encoders. This inability to capture abstract semantics significantly limits the generalization of the resulting colorization framework.

At inference time, as long as a query (x_i, r_i) conforms to some $f_i \in F$, color transfer is accurate and visually coherent. However, the combinatorial space of plausible pairings between low-level representations is *unbounded* and far more than those we could collect as training data, guaranteeing that the model will eventually face an unseen mapping $f^* \notin F$. More importantly, there are many query couples (x, r) that are not possible to collect to support this training scheme. As shown in Figure 5.17, different from character colorization solved in many baseline methods [8,55,110], the input schemes for such cross-content colorization are impractical to simulate when constructing training data.

In this training scheme, the learned hypothesis is tightly coupled to F and much harder to be extrapolated to f^*; results would suffer from dramatic deterioration in generative quality, manifesting as hue bleeding across object boundaries, texture distortion, or a complete loss of semantic alignment between sketch and reference.

6.3 Reference-based colorization with frozen reference encoder

To enforce embedding-level correspondence between the reference input and the colorized results, while maximizing the generalization of the colorization system, both the GAN and DM frameworks in this thesis follow the training scheme 2 introduced in Chapter 2 and Chapter 3. The reference encoder is a vision–language image embedder that remains frozen throughout colorization training. Because its weights are locked, the ground-truth color image can serve as the reference during training, eliminating the need for an external reference image as required in the baseline scheme. This streamlined setup, together with the frozen pre-trained encoder, delivers two concrete advantages

High-level mapping. By freezing the reference encoder, its representational space

is fixed at the embedding level, making the reference-based sketch colorization framework similar to a T2I framework but utilizing continuous embeddings as guiding representations. The sketch encoder is thus compelled to produce embeddings that align with this space, preventing the generator from overfitting to low-level, stroke-specific patterns in the training data. With correspondence enforced only in this abstract domain, the framework becomes far less sensitive to stylistic variations in sketch strokes and remains robust even when sketch–reference regions do not align perfectly.

Noise-free correspondence. Because every training triplet $\big(x,\, y,\, r = y\big)$ achieves perfect identity alignment, the generator can focus on learning a robust sketch-to-color transfer function without being distracted by mismatched textures, lighting, or pose variations. This clean supervision signal, combined with the encoder's fixed features, promotes a representation that generalizes well to *unseen* references at test time.

The ablation study in Section 4.8 and qualitative comparisons shown in Figures 5.17 and A.1 validate that with the proposed training scheme, which utilizes a pre-trained reference embedder and freezes it during MSE training, frameworks can achieve much better results with strong generalization ability.

6.4 Conclusion

To sharpen the contrast between text-guided and reference-based colorization and explain why baseline reference methods deteriorate, this chapter classified conditional inputs into five representation levels. Because diffusion training relies on a pixel-level MSE objective, supervision concentrates at the lowest visual levels and provides little pressure to learn semantic, embedding-level alignments. Consequently, when the reference encoder is trained jointly, both sketch and reference pathways default to low-level correspondences, yielding pair-specific mappings and weak generalization.

To overcome this, this thesis fixes the semantic space by using a frozen, pre-trained image encoder to provide embedding-level reference representations. This setup encourages the sketch encoder to learn compatible embeddings and shifts mapping from low-level to high-level embedding space, delivering substantially better generalization to unseen references and cross-content pairs.

Chapter 7

Conclusion

This thesis introduced an end-to-end sketch-colorization framework powered by deep generative models. Different from grayscale image color recovery or flat-color colorization algorithms, the system faithfully colorized a target line drawing with both **chromatic colors** and **achromatic cues**, including texture, tonal shading, and stroke style, extracted from a single reference image. Building on this reference transfer, text-guided editing algorithms are also proposed to introduce refinement of high-level visual attributes such as hair color, global hue, or lighting through natural language prompts.

Using the stricter evaluation protocol established in Chapter 3, the framework consistently yields visually coherent and aesthetically pleasing results across a wide spectrum of sketches and reference styles, demonstrating strong generalization. Together, reference-based transfer and intuitive semantic control form an efficient pipeline that advances the state of automatic sketch colorization.

Throughout Chapters 2 to 6, I revealed that striving for high-fidelity transfer exposes a distinctive overfitting pattern, and all these solutions proposed in this thesis follow a simple principle: **disentangle specific visual attributes by explicitly restricting the representations extracted from reference images.** Yet, this simple principle needs various modifications to the training strategy and network architecture.

Chapter 4 presented a GAN-based framework that delivers high-fidelity, reference-guided colorization while enabling flexible tag-based style control. Beyond the visual gains, the study yielded a key methodological insight: freezing the reference encoder throughout backbone training significantly strengthens both performance and generalization by compelling the sketch pathway to align with the reference in a high-level embedding space. This insight establishes a practical design principle for subsequent reference-based colorization systems and substantially mitigates the overfitting-driven degradation observed in jointly trained baselines.

Chapter 5 analyzed the distinctive over-fitting problem in reference-based sketch colorization, reframing it as a "distribution shift" that arises from mismatched optimization conditions. To counteract this shift, I propose three complementary techniques that successively reduce its adverse effects and restore reliable generalization across diverse input styles.

Section 5.4 introduced a two-stage training regime anchored by a novel noisy-training phase. By injecting controlled noise into the high-frequency channels during early optimization, the method decouples detail reconstruction from color-and-segmentation trans-

fer. This disentanglement sharply reduces the spatial entanglement documented in Section 5.2 and yields markedly more faithful propagation of fine stylistic details.

Section 5.5 pinpointed the residual spatial-entanglement artifacts, most evident in non-sketch areas, as a consequence of foreground reference embeddings leaking into the background via the cross-attention layers. To halt this leakage, I introduced a foreground–background, decoupled cross-attention mechanism that channels foreground features only to sketch-bounded regions while strictly gating background tokens to background pixels. This targeted separation eliminates background entanglement artifacts and further enhances perceptual fidelity.

Section 5.6 introduced the foreground–background split by disentangling the reference embeddings themselves. It decomposes the exemplar features into multiple semantic levels and routes each level to the region where it is most informative: low-level, texture-rich cues go to the background, while higher-level semantic signals refine the foreground. This hierarchical routing further enriches fine-grained detail transfer and yields an additional boost in reference-based colorization fidelity.

Expanding on the tag-based interpolation devised for the GAN framework (Section 4.3), Section 5.8 introduced a complementary, text-driven manipulation technique. The method factorizes each CLIP local token into two components, a spatial weight matrix and a semantic embedding vector, so that natural-language prompts can selectively modulate color and texture across different regions of the sketch with high precision.

An extensive discussion is provided in Chapter 6 to analyze the fundamental challenge of reference-based colorization and to justify why a pre-trained encoder should be employed and kept frozen. Since the prompt condition (reference image) lies at a lower level of representation compared to the objective function used in diffusion training, the diffusion loss alone cannot train the reference encoder to extract semantic information effectively. Consequently, employing a pre-trained reference encoder substantially enhances the generalization ability of the colorization framework, and freezing its parameters is essential to prevent degradation of the semantic level in the encoded representations.

Central to every solution presented in this dissertation is a single guiding insight: reference embeddings produced by a frozen encoder should not be treated as a monolithic signal, but rather as a set of semantically distinct channels that can be selectively routed, weighted, or suppressed to match the needs of each spatial region and task. By explicitly disentangling and managing these channels through different "**masking**" strategies, such as adopting and freezing a pre-trained reference embedder to **block low-level reference representations** (Chapter 4), adding noise to reduce the **color/segmentation semantics adaptively** (Section 5.4), spatial masks to **explicitly separate foreground and background** (Section 5.5) or selectively **applying reference representations in different levels for different visual attributes** (Section 5.6), the proposed framework eliminates the spatial and semantic entanglement artifacts that typically undermine reference-guided synthesis. This cue-isolation paradigm furnishes a scalable blueprint for a wide range of multi-conditioned generative problems that require **disentangling signals within the same modality**, paving the way for more controllable, artifact-free synthesis across diverse applications.

Although these contributions markedly elevate both quality and controllability, several limitations remain, as outlined in the following section.

7.1 Limitation

The main limitations of this thesis fall into four core areas:

1. Residual distribution-shift errors

The two-stage training regime reduces the tendency of reference features to corrupt structural segmentation from the sketch input and lowers the frequency of spatial-entanglement artifacts. Foreground–background separation further suppresses entanglement in non-sketch regions by routing coarse and fine reference cues to appropriate spatial channels. Nevertheless, entanglement can still arise within the foreground (i.e., the sketch-constrained region), indicating that distribution shift is mitigated but not yet eliminated.

2. Lack of objective evaluation/visualization on the distribution shift

Although this thesis defines the notion of distribution shift, it does not yet offer a quantitative metric or visualization tool for assessing it. The absence of an objective standard makes the reported improvements harder to verify and raises the barrier for researchers who wish to extend the work.

3. Inadequate illumination control

Illumination is pivotal to producing convincing, stylistically consistent colorizations, yet the current framework offers only indirect control through the choice of reference image or vague text prompts. Without an explicit lighting handle, otherwise similar outputs can differ markedly in overall appearance.

4. Limited and cumbersome text manipulation

The text-guided editing algorithm relies on ad-hoc vector arithmetic rather than end-to-end learned conditioning, making it less intuitive and less powerful than state-of-the-art T2I interfaces. Complex attribute combinations often require manual tuning and yield inconsistent results.

7.2 Future work

The future works of this thesis can be divided into two parts: 1. extensive improvement focusing on industrial applications in the animation production pipeline; 2. further exploration in research for solving the limitations outlined above.

7.2.1 Animation production

This thesis has addressed the acceleration of the colorization process for key visual illustrations—an auxiliary yet important step in the animation production pipeline. While the proposed approach substantially enhances the performance and generalization capability of data-driven frameworks, further progress is required before it can be widely adopted across diverse industrial contexts in animation production. In particular, ongoing research continues to address several limitations mentioned in the previous section for the following topics.

1. Visual understanding models for anime-style image/video

Because the sketch encoder struggles to effectively extract semantic information and easily

entangle with representations from the reference image, a practical strategy is to incorporate semantic maps derived from sketch images as conditional inputs to the colorization framework. However, current visual understanding models, such as segmentation networks or fine-tuned vision-language models, remain limited in their accuracy and robustness for anime-style images/videos, thereby constraining the development of such approaches. Consequently, advancing more reliable and domain-adapted semantic understanding models represents a promising direction for enabling more effective and scalable sketch-based colorization.

2. Temporal consistency

Automatic colorization in animation further demands temporal consistency across consecutive frames, a factor not tackled in this thesis. Once the disentanglement between sketch and reference image features has been effectively resolved, enhancing temporal coherence in the generated sequences will constitute a key next step toward more practical and production-ready solutions.

3. Eliminate residual foreground entanglement

With background artifacts largely resolved, the next step is to tackle semantic errors that still occur inside sketch-guided regions. A dedicated training scheme, focused solely on colorizing the foreground mask, could further disentangle character features and fully remove remaining artifacts.

7.2.2 Research exploration

In addition to future work driven by industrial demands, this thesis leaves several open challenges whose resolution could yield significant contributions to the research community.

1. Create statistical evaluations for distribution shift

Distribution shift plagues many transfer tasks, especially when the source and target belong to the same modality. Defining a robust evaluation protocol (e.g., divergence scores or diagnostic visualizations) would make progress verifiable and transferable, benefiting not only sketch colorization but a broad range of generative applications.

2. Unify image-guided and text-guided editing in a multimodal framework

A seamless approach is to adopt a multimodal tokenizer that feeds both reference images and textual prompts into the same latent space. Existing vision–language models are promising but currently lack degree-specific interpolation. Extending or fine-tuning these models to enable precise, reference-aware editing would yield more powerful and streamlined user interactions.

Taken together, these directions chart a promising path from mitigating residual shift to enabling explicit lighting, robust anime semantics, temporal coherence, and unified multimodal control—transforming the framework into a scalable, verifiable, and artist-friendly system for both anime production and disentanglement inside visual generation research.

Appendix A

Supplementary materials

Additional qualitative comparison with baseline methods.

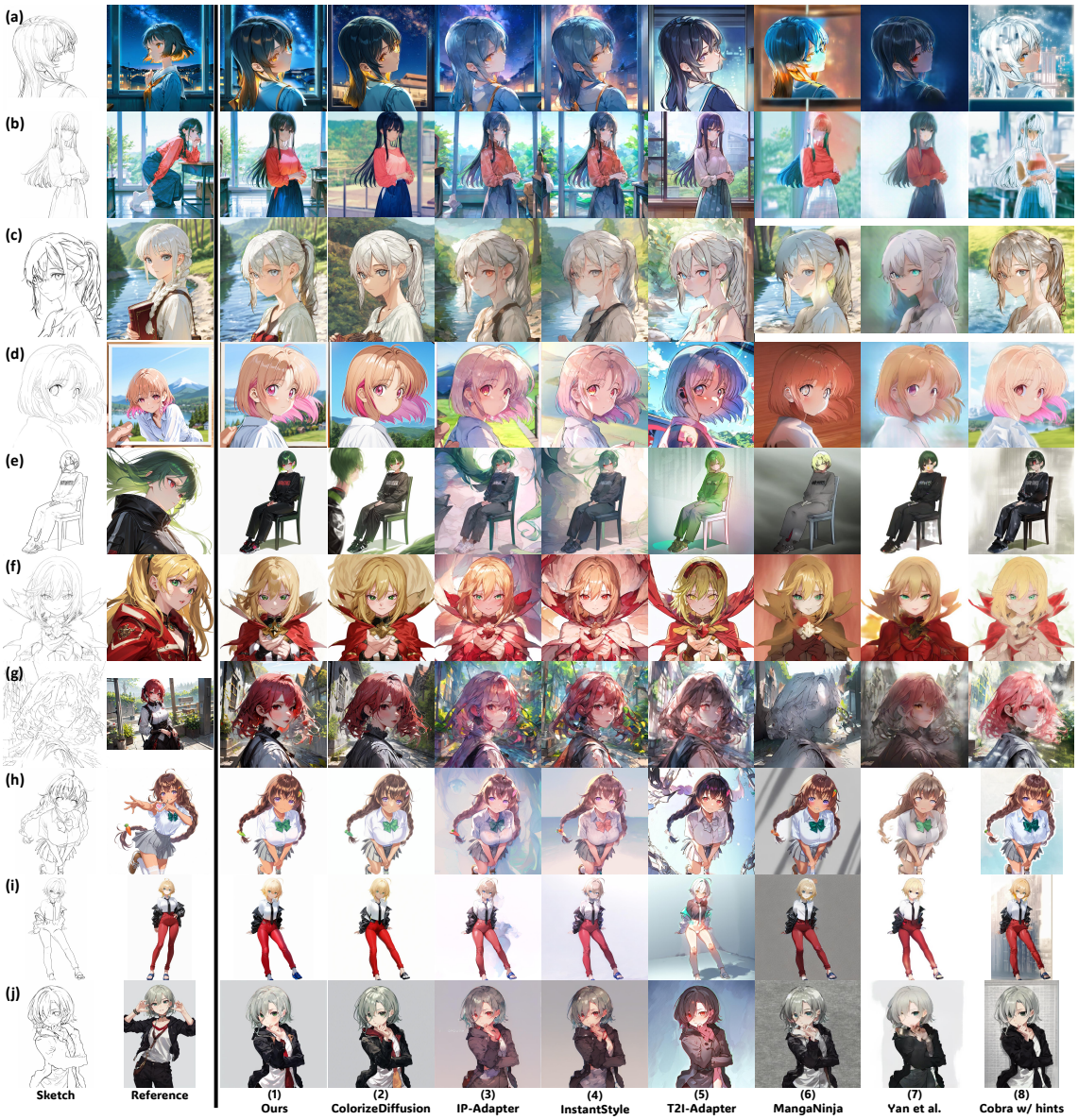


Figure A.1: Qualitative comparisons regarding figure colorization. Different from recent colorization baselines [55, 60, 98, 101, 110] and the GAN-based framework (Chapter 4, the proposed methods based on this thesis are demonstrated to be superior in the quality and similarity of colorization without having spatial entanglement and requiring inputs to have semantically or spatially similarity.

Figure A.2: Additional comparison with baseline methods.

Figure A.3: Additional comparison with baseline methods.

Figure A.4: Additional comparison with baseline methods.

Acknowledgment

I would first like to thank my mother for her unwavering encouragement. My greatest gratitude goes to my academic supervisor, Prof. Suguru Saito, whose insightful guidance and steady support over the past five years have been indispensable. I am also thankful to Dr. Liang Yuan, Prof. Issei Fujishiro, Prof. Erwin Wu, and Prof. Hideki Koike for their generous advice and assistance. I owe special thanks to Ryogo Ito and Ryo Moriai for their help with the user studies, to Xinrui Wang for his invaluable editorial support, and to Fuminori Shibasaki and Ayumu Sato for providing the hand-drawn sketches used in this work.

References

[1] Pytorch documentation. https://docs.pytorch.org/docs/stable/, 2025. Accessed: 2025-06-18.

[2] M. Abadi, P. Barham, J. Chen, and et al. Tensorflow: A system for large-scale machine learning. In *Proc. OSDI*, 2016.

[3] Automatic1111. stable-diffusion-webui. https://github.com/AUTOMATIC1111/stable-diffusion-webui/tree/master, 2023. Accessed: DATE 2023-06-25.

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. arXiv:1607.06450, 2016.

[5] S. Babbar. What are generative adversarial networks (gans)? understanding and implications. LinkedIn Articles, https://www.linkedin.com/pulse/what-generative-adversarial-networks-gans-sushant-babbar-qpc9c/, 2025. Accessed: 2025-06-18.

[6] Shane T. Barratt and Rishi Sharma. A note on the inception score. arXiv:1801.01973, 2018.

[7] J. Bradbury, R. Frostig, P. Hawkins, and et al. Jax: Autograd and high-performance machine learning in python. arXiv:2102.04600, 2021.

[8] Yu Cao, Xiangqiao Meng, P. Y. Mok, Tong-Yee Lee, Xueting Liu, and Ping Li. Animediffusion: Anime diffusion colorization. *TVCG*, pages 1–14, 2024.

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, 2020.

[10] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[11] comfyanonymous. Comfyui. https://github.com/comfyanonymous/ComfyUI, 2024. Accessed: DATE 2024-05-21.

[12] 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.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE Computer Society, 2009.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16×16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[15] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[16] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[17] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. OpenReview.net, 2024.

[18] Sébastien Fourey, David Tschumperlé, and David Revoy. A fast and efficient semi-guided algorithm for flat coloring line-arts. In Fabian Beck, Carsten Dachsbacher, and Filip Sadlo, editors, *International Symposium on Vision, Modeling, and Visualization, VMV*, pages 1–9, 2018.

[19] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[20] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri. Comicolorization: semi-automatic manga colorization. In Diego Gutierrez and Hui Huang, editors, *SIGGRAPH Asia Technical Briefs*, pages 12:1–12:4. ACM, 2017.

[21] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 2672–2680, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[25] Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Trans. Graph.*, 37(4):47, 2018.

[26] D. Hendrycks and K. Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. arXiv:1606.08415, 2016.

[27] 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 Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017.

[28] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[29] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, pages 6840–6851, 2020.

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

[31] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2022.

[32] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[34] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1510–1519. IEEE Computer Society, 2017.

[35] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11207, pages 179–196. Springer, 2018.

[36] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[37] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 4904–4916, 2021.

[39] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9906, pages 694–711. Springer, 2016.

[40] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[41] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] KichangKim. Deepdanbooru. `https://github.com/KichangKim/DeepDanbooru`, 2023. Accessed: DATE 2023-03-19.

[44] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9055–9064, 2019.

[45] D. P. Kingma and M. Welling. Auto-encoding variational bayes. arXiv:1312.6114, 2013.

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, pages 1097–1105, 2012.

[48] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[49] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5800–5809, 2020.

[50] Chengze Li, Xueting Liu, and Tien-Tsin Wong. Deep extraction of manga structural lines. *ACM Trans. Graph.*, 36(4):117:1–117:12, 2017.

[51] J. Li, D. Li, C. Xiong, and S. C. H. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 12888–12900, 2022.

[52] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[53] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *Int. Conf. Mach. Learn. (ICML)*, 2023.

[54] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[55] Zhiheng Liu, Ka Leong Cheng, Xi Chen, Jie Xiao, Hao Ouyang, Kai Zhu, Yu Liu, Yujun Shen, Qifeng Chen, and Ping Luo. Manganinja: Line art colorization with precise reference following. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[56] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[57] Yihao Meng, Hao Ouyang, Hanlin Wang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Zhiheng Liu, Yujun Shen, and Huamin Qu. Anidoc: Animation creation made easier. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[58] Midjourney, Inc. Midjourney. `https://www.midjourney.com`, 2022. AI text-to-image generator; open-beta release on 12 July 2022.

[59] L. Mosser, O. Dubrule, and M. J. Blunt. Stochastic reconstruction of an oolitic limestone by generative adversarial networks. *Transport in Porous Media*, 124(1):81–103, 2018.

[60] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arxiv:2302.08453, 2023.

[61] Avinash Navlani. Multi-Layer Perceptron Neural Network using Python. `https://machinelearninggeek.com/multi-layer-perceptron-neural-network-using-python/`, 2025. Accessed: 2025-06-18.

[62] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 2022.

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

[64] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182. IEEE, 2023.

[65] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952, 2023.

[66] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Pergamon Press, 1962.

[67] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, 2022.

[68] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[69] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

[70] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[71] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. arxiv:2204.06125, 2022.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

[73] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[74] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510. IEEE, 2023.

[75] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[76] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[77] Kazuhiro Sato, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Reference-based manga colorization by graph correspondence using quadratic programming. In *SIGGRAPH*, pages 15:1–15:4. ACM, 2014.

[78] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016.

[79] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[80] SkyTNT, infoengine1337, and not lain. anime-segmentation. `https://github.com/SkyTNT/anime-segmentation`, 2022.

[81] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 2256–2265, 2015.

[82] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[83] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11895–11907, 2019.

[84] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[85] Tsai-Ho Sun, Chien-Hsun Lai, Sai-Keung Wong, and Yu-Shuen Wang. Adversarial colorization of icons based on contour and color conditions. In Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo, and Wei Tsang Ooi, editors, *Proceedings of the 27th ACM International Conference on Multimedia (ACM MM)*, pages 683–691. ACM, 2019.

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

[87] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[88] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. arXiv:1607.08022, 2016.

[89] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 6000–6010, 2017.

[90] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[91] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[92] B. Widrow and M. E. Hoff. Adaptive switching circuits. In *IRE WESCON Convention Record*, pages 96–104, 1960.

[93] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12863–12872. Computer Vision Foundation / IEEE, 2021.

[94] Xiaoyu Xiang, Ding Liu, Xiao Yang, Yiheng Zhu, Xiaohui Shen, and Jan P. Allebach. Adversarial open domain adaptation for sketch-to-photo synthesis. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV*, pages 944–954. IEEE, 2022.

[95] Jinbo Xing, Hanyuan Liu, Menghan Xia, Yong Zhang, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Tooncrafter: Generative cartoon interpolation. *ACM Trans. Graph.*, 43(6):245:1–245:11, 2024.

[96] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. arXiv:1505.00853, 2015.

[97] Huiting Yang, Liangyu Chai, Qiang Wen, Shuang Zhao, Zixun Sun, and Shengfeng He. Discovering interpretable latent space directions of gans beyond binary attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12177–12185, 2021.

[98] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arxiv:2308.06721, 2023.

[99] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Trans. Mach. Learn. Res.*, 2022, 2022.

[100] Haichao Zhang, Jianchao Yang, Yanning Zhang, and Thomas S. Huang. Non-local kernel regression for image and video restoration. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 6313, pages 566–579. Springer, 2010.

[101] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[102] Lvmin Zhang. Sketchkeras. `https://github.com/lllyasviel/sketchKeras`, 2017.

[103] Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier GAN. In *Asian Conference on Pattern Recognition, ACPR*, pages 506–511. IEEE Computer Society, 2017.

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

[105] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9907, pages 649–666, 2016.

[106] 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.

[107] Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Prompt spectrum for attribute-aware personalization of diffusion models. *ACM Trans. Graph.*, 42(6):244:1–244:14, 2023.

[108] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 2024.

[109] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2242–2251. IEEE Computer Society, 2017.

[110] Junhao Zhuang, Lingen Li, Xuan Ju, Zhaoyang Zhang, Chun Yuan, and Ying Shan. Cobra: Efficient line art colorization with broader references. arXiv:2504.12240, 2025.

[111] Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu. Language-based colorization of scene sketches. *ACM Trans. Graph.*, 38(6):233:1–233:16, 2019.