POLITECNICO DI TORINO

Department of Computer Engineering

Master Thesis

From Narrative to Frames: AI-Assisted Storyboarding with personalized Diffusion Models

Supervisors

Prof. Tania CERQUITELLI

Dr. Bartolomeo VACCHETTI

Candidate

Luisa OCLEPPO

Academic Year 2024-2025

Abstract

This thesis presents a framework for AI-assisted storyboarding that leverages state-of-the-art text-to-image diffusion models and efficient fine-tuning techniques to generate visually coherent and narrative-consistent storyboards. The work begins with a comprehensive review of image synthesis architectures—from VAEs, to GANs to diffusion models—and explores critical components such as attention mechanisms, latent diffusion, and CLIP-based conditioning, establishing a solid technical foundation for the study. Building on this background, the thesis surveys contemporary text-to-image systems (Stable Diffusion, GLIDE, DALL-E, Imagen, MidJourney) and fine-tuning methodologies such as Dreambooth, LoRA, Textual Inversion, Custom Diffusion, ControlNet.

The work then delves into storyboarding by investigating how shot types shape visual narratives and by synthesizing insights from recent approaches like StoryGAN, AR-LDM and StoryDALL-E. These findings directly inform the design of an interactive storyboard generation system that aims to maintain character consistency and shot type fidelity across frames. To achieve these goals, the proposed approach combines efficient Dreambooth LoRA fine-tuning with a targeted prompt engineering and inpainting strategy. High-quality training datasets are constructed from curated movie stills and synthetic character images to refine a pre-trained Stable Diffusion model. The interactive storyboarding system proposed in the work integrates automated prompt refinement via ChatGPT, user control mechanisms, and an inpainting-based module for post-generation adjustments, enabling iterative enhancement of storyboard frames.

Experimental evaluations, including quantitative metrics and human assessments, demonstrate that the proposed method effectively preserves the stylistic characteristics of various shot types and the identity consistency of characters. Overall, this work leverages personalized text-to-image generation to offer a practical, accessible, and open-source tool for pre-production and creative storytelling, providing a solution that bridges the gap between high-level narrative intent and detailed visual execution.

Table of Contents

Chapter 1

Introduction

Storyboarding is a cornerstone of visual storytelling, providing filmmakers, animators, and content creators with a critical pre-production tool. Traditionally, storyboards have been crafted by hand or with limited digital assistance, requiring extensive time resources and laborious manual input. In today's fast-paced creative environment, new methods for efficient, automated storyboard generation are finding fertile ground.

Recent advancements in the domain of text-to-image generation —particularly through diffusion models— offer a promising solution. These models can synthesize visuals directly from textual descriptions and have rapidly progressed in their ability to generate highly detailed, semantically rich images. When combined with novel personalization techniques, diffusion models show remarkable potential in learning new concepts and reproducing them across diverse scenes. Building on these advances, this thesis proposes a framework for AI-assisted storyboarding that bridges the gap between high-level narrative intent and frame-level visual output.

The proposed system leverages personalized diffusion models to generate visually coherent and narratively consistent storyboards. By integrating fine-tuning techniques with prompt engineering and iterative refinement, the system can produce sequences that adhere to specific cinematic shot types—ranging from extreme close-ups to extreme long shots—while maintaining character consistency across multiple frames, including multi-character scenes. These capabilities are essential for achieving visual and narrative continuity, a core requirement for effective storyboarding.

The work is motivated by the key challenges in both traditional storyboard creation and contemporary text-to-image generation models. Manual creation is time-consuming and lacks scalability. At the same time, diffusion models struggle to accurately reproduce varied shot compositions or maintain consistent character identity across contexts without dedicated fine-tuning. Furthermore, even after successful fine-tuning of individual concepts, combining them in multi-concept scenes often leads to concept bleeding or concept confusion.

Addressing these limitations, this work explores the development of an interactive system for generating storyboards, aimed at assisting creatives in the pre-production process. The system is based on Stable Diffusion 1.5 and is designed to generate consistent and contextually coherent storyboards through the following key components:

- **Learning shot type characteristics:** We created dedicated datasets for eight distinct shot types - from extreme close up to extreme long shot- by collecting high-quality movie stills. Then we employed Dreambooth LoRA finetuning to teach the model the framing and compositional logic specific to each type.
- **Ensuring character consistency:** We created dedicated datasets for seven distinct character identities through Midjourney. Then we applied Dreambooth LoRA finetuning to teach the model each character's unique visual features, ensuring consistent appearance across different frames and scenes.
- **Enhancing narrative coherence through prompt engineering:** We created a prompt template that, by leveraging large language models, transforms broad scene descriptions into detailed, structured prompts that specify shot type, environment, character identities and their arrangement for each frame of the storyboard. These prompts play a crucial role in producing coherent and narratively aligned visuals.
- **Providing interactive refinement:** The system supports user-in-the-loop editing by allowing to generate multiple images per prompt, select preferred outputs, regenerate alternatives, or adjust the prompt during the process — thereby supporting an iterative creative process.
- **Mitigating challenges in multi-character scenes**: An inpainting-based refinement module allows users to correct inconsistencies in character representation, addressing common limitations of diffusion models when merging multiple LoRA concepts in a single frame.

To systematically address these objectives, this thesis is organized as follows:

Chapter 2: Technical Background This chapter reviews the evolution of image synthesis methods -from early generative adversarial networks (GANs) and variational autoencoders (VAEs) to diffusion models. It covers key components such as attention mechanisms and ViTs, latent diffusion, and CLIP-based conditioning.

Chapter 3: State-of-the-art Text-to-Image Models This chapter provides a survey of state-of-the-art text-to-image models, including Stable Diffusion, GLIDE, DALL-E, Imagen, and MidJourney.

Chapter 4: Fine-tuning techniques This chapter explores modern fine-tuning approaches such as Dreambooth, LoRA, Textual Inversion and ControlNet, which allow to customize and tweak the output of diffusion models to specific needs.

Chapter 5: Storyboarding This chapter provides an overview of story visualization research. It begins by outlining the fundamentals of shot types in visual storytelling and their role in shaping narrative focus. It then reviews key works in the domain of AI-assisted story generation — such as StoryGAN, StoryDALL-E, StoryDiffusion — analyzing their methodologies. The chapter also presents core challenges in multi-LoRA generation, such as concept bleeding, confusion, and omission, introducing mitigation strategies and surveying recent multi-subject consistency frameworks developed to address these limitations.

Chapter 6: Experiments and Results This chapter describes the full pipeline of the proposed framework, starting with the collection and preparation of training datasets for the eight shot types and the seven distinct characters. It then details the fine-tuning process of Stable Diffusion 1.5 using DreamBooth LoRA, followed by the design of a structured prompt template for automated storyboard prompt generation and refinement using ChatGPT. The full storyboard generation pipeline is presented, including the inpainting-based refinement module for correcting inconsistencies. The chapter also provides a comparative analysis on the impact of descriptive captions during training, and showcases a selection of fully generated storyboards. Finally, it reports both quantitative and qualitative evaluation results, using metrics such as CLIPScore, ICA, and DINOv2, alongside manual annotations and human assessment.

Chapter 7: Discussion and Future Work This chapter discusses our findings, strengths and limitations of the proposed approach. Particularly, it outlines potential directions for future works to enhance the expressiveness, control, and versatility of the proposed framework for AI-assisted storyboard generation.

In summary, this thesis analyzes the integration of artificial intelligence within the visual storytelling pre-production step, proposing an approach to AI-assisted storyboarding. By integrating advancements in text-to-image models, fine-tuning techniques and interactive refinement methods, this work aims to bridge the gap between narrative intent and visual execution through AI. Beyond the experimental validation of the proposed system, the research also provides an examination of the technological landscape, including the evolution of generative models and existing works in the field of storytelling. By addressing key challenges such as shot type consistency, character fidelity, and user interactivity, this thesis contributes to the development of AI-driven tools that aim to support the pre-production creative process, laying groundwork for future improvements.

Chapter 2

Background

2.1 Overview of Image Sythesis Models

The field of image synthesis has experienced a transformative journey, with significant advancements across multiple generations of architectures.

2.1.1 Variational Autoencoders (VAEs)

Early breakthroughs were marked by the introduction of Variational Autoencoders (VAEs) in 2013 by Kingma and Welling [1].

Autoencoders are a class of unsupervised neural networks that can represent data in a lower-dimensional space, also known as latent space, to learn efficient representations. Applications include compression, denoising, feature extraction, and generative models. Autoencoders are trained by first encoding data into a latent space and then decoding them back into the original representation, a process known as reconstruction, while minimizing the difference between the original input and the reconstructed data.

The model consists of two neural networks: an encoder, which maps the input to a distribution over latent variables, and a decoder, which reconstructs the original data from samples drawn from this latent space. While autoencoders are excellent at near-perfect reconstruction, they lack the ability to generate entirely new data.

Variational Autoencoders address this limitation by treating the latent space probabilistically. Unlike traditional autoencoders, which map inputs to fixed

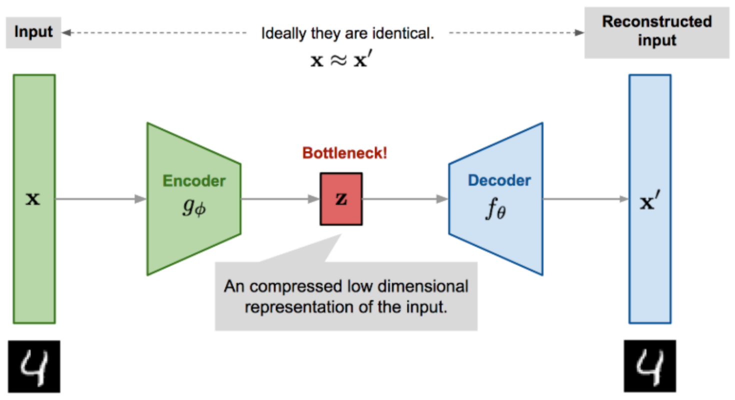

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Figure 2.1: Autoencoder architecture from [2]

vectors, VAEs represent the input as a probability distribution over the latent space. This probabilistic representation allows VAEs to generate entirely new data while sacrificing their ability to perfectly reconstruct existing data.

Each point in the latent space is represented by the mean μ and standard deviation σ of a Gaussian distribution, meaning that every latent variable is defined by two values instead of one. An image can then be reconstructed by sampling from this distribution:

$$z = \mu + r \cdot \sigma \tag{2.1}$$

where r is a random value drawn from a standard normal distribution $\mathcal{N}(0,1)$.

Variational autoencoders differ from standard autoencoders by encoding input data into two distinct vectors: one capturing the mean and another the standard deviation of the latent representation. These vectors combined are used to sample a latent encoding, which the decoder then utilizes to reconstruct the original input.

The training process optimizes a variational lower bound (ELBO), balancing two objectives: minimizing the reconstruction loss to make the generated image as close as possible to the original while ensuring a structured latent space so that similar latent representations produce similar outputs. This shifts the model between perfect reconstruction and the ability to generate new images.

However, while VAEs excel at learning structured and interpretable latent spaces, the images they generate often appear blurry. This occurs because VAEs sacrifice reconstruction precision by learning average images—represented by their mean—rather than perfectly fitting the input data, leading to blurred outputs.

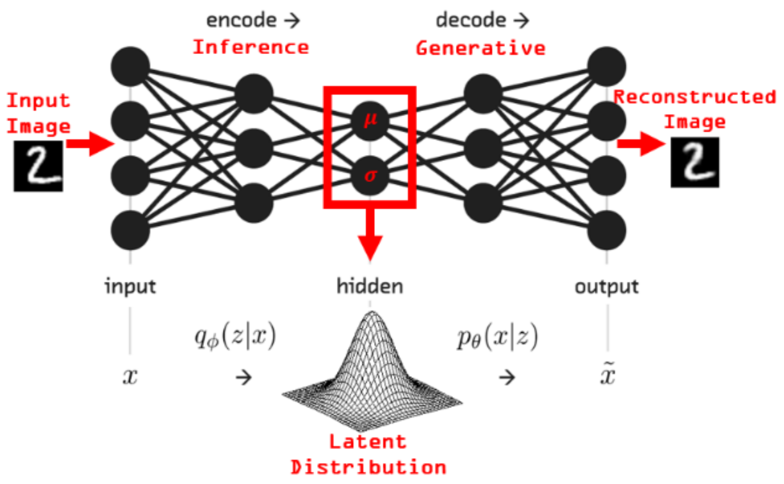


Figure 2.2: VAE flow from [3]

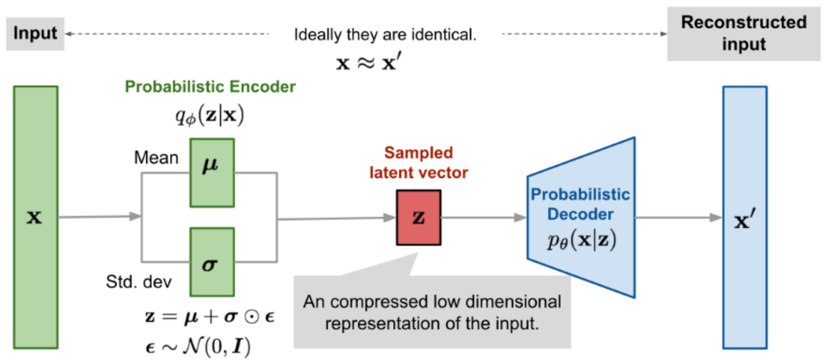

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
$\mathbf{x}$
Std. dev
$\boldsymbol{\sigma}$
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
Sampled latent vector
$\mathbf{z}$
An compressed low dimensional representation of the input.
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$

Figure 2.3: Illustration of variational autoencoder model with the multivariate Gaussian assumption from [2]

To address this limitation, Vector Quantized Variational Autoencoders (VQ-VAE) were introduced in 2017 by van den Oord et al. [4].

VQ-VAEs replace continuous latent variables with discrete categorical codes, using a vector quantization (VQ) mechanism to enforce structured latent representations. Instead of encoding data into a probabilistic latent space, VQ-VAEs assign each input to the closest entry in a learned codebook of discrete embeddings. This approach allows VQ-VAEs to capture meaningful patterns in data while reducing redundancy. The training process ensures that the encoder effectively utilizes the learned codebook by optimizing two additional loss terms:

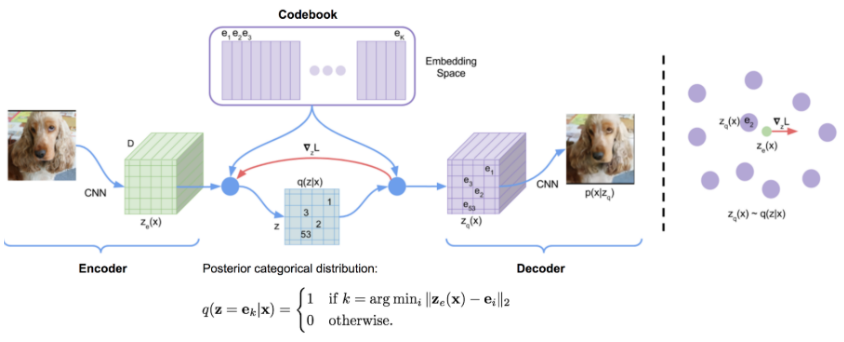

Codebook
$e_1 e_2 e_3$
e_K
Embedding Space
D
CNN
$z_e(x)$
$\nabla_z L$
$q(z|x)$
1
3
2
53
z
e_1
e_3
e_2
e_{53}
$z_q(x)$
CNN
$p(x|z_q)$
$z_q(x)$ e_2
$\nabla_z L$
$z_e(x)$
$z_q(x) \sim q(z|x)$
Encoder
Posterior categorical distribution:
Decoder
$q(\mathbf{z} = \mathbf{e}_k|\mathbf{x}) = \begin{cases} 1 & \text{if } k = \arg\min_i \|\mathbf{z}_e(\mathbf{x}) - \mathbf{e}_i\|_2 \\ 0 & \text{otherwise.} \end{cases}$

Figure 2.4: The architecture of VQ-VAE from [4]

- **Commitment loss:** encourages the encoder to make consistent use of the discrete codes by ensuring embeddings do not change drastically.
- **Codebook loss:** ensures that the codebook entries remain as close as possible to the encoder embeddings.

As a result, VQ-VAEs provide a more compact and interpretable latent space. The reconstructions generated by VQ-VAEs are significantly less blurry than those from VAEs, making them well-suited for high-quality image generation tasks.

2.1.2 GANs

In 2014, the advent of Generative Adversarial Networks (GANs) by Ian Goodfellow et al. [5] marked an important advancement in the field of image synthesis, employing an adversarial training framework. This framework consists of two neural networks: a **generator**, which synthesizes data, and a **discriminator**, which evaluates the authenticity of the data. The generator aims to produce data that can deceive the discriminator, while the discriminator tries to distinguish between real and synthetic data, creating a dynamic and competitive learning process. In the training process, whenever the discriminator incorrectly classifies a generated (fake) image, it receives a penalty, while the generator is rewarded. Additionally, the discriminator is either rewarded or penalized based on its accuracy in recognizing real images. This competitive dynamic drives both the discriminator and generator to progressively enhance their performance.

The adversarial training of GANs can be formulated as a *two-player minmax game*: the generator G and discriminator D are adversaries optimizing the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \qquad (2.2)$$

Here:

- $x \sim p_{\text{data}}(x)$ represents samples from the true data distribution.
- $z \sim p_z(z)$ represents samples from a prior distribution, typically Gaussian or uniform noise, which serves as input to the generator.
- $G(z)$ represents the synthetic data generated by G.
- $D(x)$ is the probability assigned by the discriminator that x is a real sample from the data distribution.

The generator G aims to minimize this function by producing images that deceive the discriminator, while the discriminator D attempts to maximize it by correctly distinguishing between real and fake images.

Despite their success, GANs present several training challenges:

- **Training instability:** the adversarial nature of GANs makes optimization difficult, often causing the models to oscillate without converging.
- **Mode collapse:** the generator may learn to produce a limited variety of outputs, leading to low sample diversity.
- **Vanishing gradients:** if the discriminator becomes too strong early in training, the generator may receive insufficient gradient updates.

Since the introduction of the original GAN model, numerous variants have been proposed to expand its capabilities. One example is the Conditional GAN (cGAN)[6], proposed soon after GANs first emerged. which incorporates additional context —like class labels or textual descriptions—to guide and condition the image generation process. One notable application of this framework was StackGAN (2016)[7], which leveraged text conditioning to generate high-quality images from textual descriptions through a two-stage process of coarse-to-fine generation. In 2015, Deep Convolutional GANs (DCGANs)[8] incorporated convolutional layers into both the generator and discriminator, making GANs more stable and capable of generating higher-quality images. Later, in 2017, Wasserstein GANs (WGANs) by Arjovsky

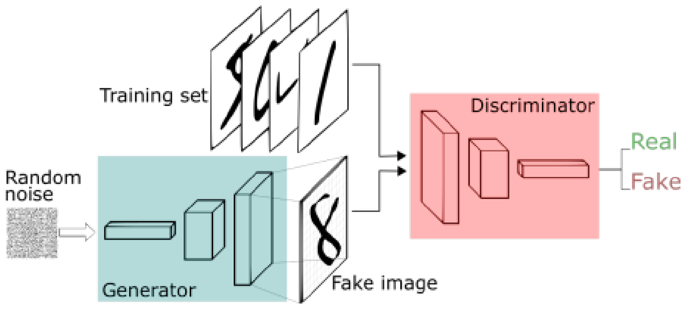


Figure 2.5: The GAN framework.

et al. [9] addressed challenges such as training instability and mode collapse by replacing the traditional GAN loss function with the Wasserstein distance, providing smoother gradients and more stable training dynamics and leading to improved performance in GAN training. Building on these advancements, StyleGAN (2018) [10] enabled control over image styles and structures by manipulating the latent space, setting a new standard for high-resolution and stylized image generation.

2.1.3 Autoregressive Models

During the same period when VAEs were being refined and GANs were gaining attention, autoregressive models like PixelRNN and PixelCNN [11] [12] presented a different approach to image synthesis. Rather than generating entire images at once, these models modeled the conditional distribution of each pixel based on its neighbors and sequentially generated images, pixel by pixel, ensuring that each pixel is informed by its predecessors, leading to high-quality and coherent images. However, this sequential generation process is computationally intensive and time-consuming, especially for high-resolution images, making it challenging to parallelize and thus inefficient for large-scale image synthesis. While PixelCNN introduced convolutional layers to mitigate some inefficiencies of PixelRNN, it still required sequential processing during image generation, limiting its practicality for real-time applications. Additionally, PixelCNN faced challenges like the "blind spot" issue, where certain pixels were not conditioned on all relevant preceding pixels, potentially affecting image quality. [13]

2.1.4 Diffusion Models

The next major breakthrough in the field of image synthesis came with the emergence of Diffusion Models in 2020, specifically the Denoising Diffusion Probabilistic Models (DDPMs) introduced by Ho et al. [14]. These models conceptualized image generation as the reversal of a noise-adding process. In the forward step, random noise is gradually applied to an image until all structure is destroyed, while the reverse step learns to progressively denoise the image to recover its original form. Unlike GANs, diffusion models produce diverse and high-quality outputs without suffering from mode collapse, although their early implementations were computationally expensive [15]. In 2022, the introduction of Latent Diffusion Models (LDMs) [16] like Stable Diffusion [17] marked the next major advancement: these models perform the diffusion process within a compressed latent representation instead of directly operating on pixel-level data, therefore allowing the generation of high-resolution images at a fraction of the computational cost.

2.1.5 Transformers

Parallel to these developments, transformer architectures [18], originally designed for natural language processing, began influencing the field of image synthesis. Transformers excel at capturing long-range dependencies and relationships, which makes them ideal for modeling complex interactions between different modalities, such as text and images. This foundation paved the way for attention mechanisms, which allow models to focus dynamically on the most relevant parts of their inputs, greatly enhancing performance of generative models. For instance, cross-attention mechanisms, inspired by transformer designs, are integral to diffusion models like Stable Diffusion, where they align text embeddings with image features during the denoising process. At the same time, models like DALL · E 2[19] and Imagen[20] combine transformer-based text encoders with diffusion-based image generation, leveraging multi-head self-attention to process textual prompts and encode semantic information to condition the diffusion process.

This progression from VAEs and GANs to diffusion models and transformers illustrates a trajectory toward models that not only produce photorealistic images but also incorporate semantic understanding and enable unprecedented user-guided control and coherence in text-to-image generation tasks.

The next sections cover more in depth Attention, Vision Transformers, Diffusion and Latent Diffusion Models, CLIP and Classifier Free Guidance.

2.2 Attention

Attention mechanisms are a cornerstone of modern deep learning architectures, enabling models to dynamically focus on the most relevant parts of their input. Introduced as part of the Transformer architecture in the landmark paper *Attention is All You Need* by Vaswani et al. (2017)[18], attention mechanisms have since become a key driver of advancements in fields like natural language processing, computer vision, and multimodal learning. At their core, attention mechanisms allow models to assign different levels of importance to input elements, enabling them to capture long-range dependencies and relationships in sequences effectively.

The attention mechanism revolves around three key concepts: **Query (Q)**, **Key (K)**, and **Value (V)**. These concepts define how relationships between elements are evaluated and used to determine relevance:

- **Query (Q):** the query represents the element that seeks relevant information from the input. It serves as a request for context, determining how much attention should be given to different parts of the input. For instance, in a translation model, a word in the decoder (e.g., "ate" in an English sentence) acts as the query when deciding which words in the input (e.g., corresponding French words "a" and "mangé") should be attended to.
- **Key (K):** the key acts as an identifier for each element in the input, helping the model evaluate which parts are most relevant to the query. Continuing with the translation example, each word in the source sentence (e.g., "a", "mangé") has an associated key that determines its importance when responding to the query "ate."
- **Value (V):** the value contains the actual content or representation that will be used in the output. It is unchanged by the attention mechanism and is combined in a weighted sum once the relevance scores have been determined. For example, if "mangé" is considered the most relevant source word for "ate," its value contributes more significantly to the translated output.

The process works as follows: given a set of queries, the attention mechanism computes the relevance of each query to all keys using a dot product, resulting in a set of alignment scores. To maintain numerical stability, these alignment scores are scaled by the square root of the key dimensionality ($\sqrt{d_k}$). Subsequently, the scaled scores are transformed into a probability distribution via the softmax function. Finally, the weighted sum of the values is computed, where elements with higher alignment scores contribute more to the result:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

In this way, attention makes the model focus dynamically on the most relevant parts of the input for each query, producing a context-sensitive output. To enhance computational efficiency and scalability, the attention mechanism processes multiple queries simultaneously in a matrix form. Queries are organized into a query matrix Q, while the associated keys and values are represented as matrices K and V, respectively. This parallelized, matrix-based computation significantly improves the speed and scalability of the attention mechanism

Multi-Head Attention

Multi-head attention extends the basic attention mechanism by allowing the model to focus on information from multiple representation subspaces simultaneously. Instead of applying one attention mechanism to the full-dimensional input vectors of size d_{model}, the input is split into multiple smaller subspaces of size d_k, and each head independently processes a subspace, focusing on different aspects or patterns in the data. By processing multiple attention heads in parallel, the model becomes able to explore diverse regions of the input simultaneously and seize intricate relationships and deeper patterns within the data. The outputs from each "head" are then concatenated and transformed back to the original dimensionality:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W_O,$$

where:

- The model learns separate weight matrices W_Q^i, W_K^i, and W_V^i for each attention head i, where $i = 1, \ldots, h$ and h is the number of heads. These matrices project the input into h subspaces:

$$Q^i = XW_Q^i, \quad K^i = XW_K^i, \quad V^i = XW_V^i.$$

- Each weight matrix has dimensions $d_{\text{model}} \times d_k$, with $d_k = d_{\text{model}}/h$.
- Attention is computed independently for each head:

$$\text{head}_i = \text{Attention}(Q^i, K^i, V^i).$$

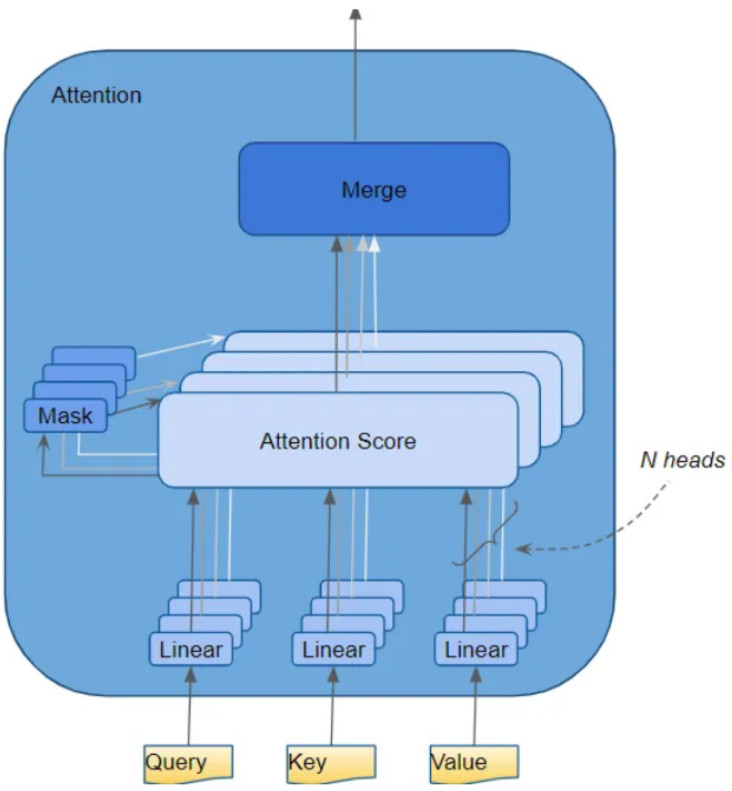


Figure 2.6: Multi-head attention mechanism.

- Once each attention head produces its output, these outputs are concatenated into a single vector. This combined representation is then projected back to the original model dimension d_{model} using a learned weight matrix $W_O \in \mathbb{R}^{(h \cdot d_k) \times d_{\text{model}}}$, effectively merging the information from the different subspaces.
- The matrix $W_O \in \mathbb{R}^{hd_k \times d_{\text{model}}}$ combines the outputs of the h heads into a single representation.

To ensure that multi-head attention remains computationally efficient:

- The size of each subspace (d_k) is reduced such that $d_k = d_{\text{model}}/h$, keeping the total computational cost similar to single-head attention.
- The projections W_Q^i, W_K^i, and W_V^i are adjusted to dimensions $d_{\text{model}} \times d_k$, minimizing the overhead of processing h heads in parallel.

For example, with $d_{\text{model}} = 512$ and $h = 8$, each head processes subspaces of size

$d_k = 512/8 = 64$, ensuring that the total number of operations remains comparable to single-head attention while providing the additional benefits of multi-head diversity.

Self-Attention

Self-attention [21] is a particular form of the attention mechanism in which the queries, keys, and values are all derived from the same input sequence. This mechanism allows the model to dynamically evaluate dependencies within a sequence. For example, in text-based applications, self-attention can capture long-range dependencies, such as subject-verb relationships in sentences, irrespective of their position. In vision tasks, it enables models to directly model long-distance spatial interactions, offering an alternative to convolutional operations.

Unlike convolutions, which apply fixed local filters, self-attention dynamically aggregates information from adaptive neighborhoods, allowing models to better capture spatial relationships and improve feature representation [21]. Instead of computing attention over the entire image, local self-attention restricts computation to a spatial neighborhood around each pixel. This local region, referred to as the memory block, is denoted as $\mathcal{N}(i, j)$, where (i, j) is the spatial position of the target pixel. Each pixel x_{ij} (with dimensionality d_{in}) and its neighbors are linearly

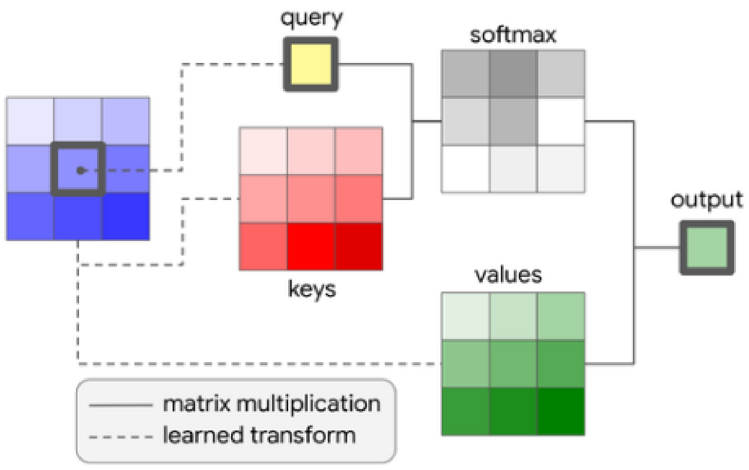


Figure 2.7: Self-attention mechanism in a Transformer model.

projected into queries (q_{ij}), keys (k_{ab}), and values (v_{ab}) using learned weight matrices:

$$q_{ij} = W_Q x_{ij}, \quad k_{ab} = W_K x_{ab}, \quad v_{ab} = W_V x_{ab}, \quad \forall (a,b) \in \mathcal{N}(i,j).$$

These projections determine which neighboring pixels the target pixel should attend to and how much influence each should have.

The similarity between the query q_{ij} and each neighboring key k_{ab} is computed using a dot product, followed by a softmax normalization to obtain attention weights:

$$y_{ij} = \sum_{a,b \in \mathcal{N}(i,j)} \text{softmax}_{ab} \left(q_{ij}^\top k_{ab} \right) v_{ab}.$$

This weighted sum determines the updated representation of the pixel (i,j), integrating contextual information from its local neighborhood.

A limitation of standard self-attention is its lack of positional information, making it permutation equivariant—a property that limits expressivity in vision models. To address this, relative position embeddings are introduced, encoding spatial relationships between pixels.

Instead of computing attention scores based solely on pixel content, the similarity between a query q_{ij} and a neighboring key k_{ab} is now influenced by their relative spatial position:

$$y_{ij} = \sum_{a,b \in \mathcal{N}(i,j)} \text{softmax}_{ab} \left(q_{ij}^\top k_{ab} + q_{ij}^\top r_{a-i,b-j} \right) v_{ab}.$$

Here, $r_{a-i,b-j}$ represents the relative distance encoding, capturing both the row offset $a - i$ and column offset $b - j$. This ensures that attention weights are modulated by both the feature content and the relative displacement of the element from the query.

By infusing relative position information, self-attention achieves translation equivariance, similar to convolutions, meaning that learned representations remain stable even when objects shift within the image. This property is crucial for vision tasks where spatial consistency is necessary, such as segmentation and object detection.

Attention in the Transformer Architecture

The Transformer architecture, built entirely on attention mechanisms, revolutionizes sequential data processing by handling input data in parallel, unlike traditional recurrent networks that process sequences step by step. Transformers consist of two main components: the encoder and the decoder stacks, which work together to transform an input sequence into an output sequence (e.g., a sentence into a translation). These stacks are composed of multiple identical layers, with each layer including self-attention, feedforward layers, normalization, and residual connections.

Tokenization and input processing: before entering the Transformer, the input sequence (e.g., a sentence) is first tokenized—split into smaller units such as words, subwords, or characters, depending on the task and tokenizer used. Each token is then mapped to a numerical vector through a process called embedding. These embeddings are fixed-size dense vectors (of dimension d_{model}) that represent the semantic meaning of each token. Since attention mechanisms do not inherently capture the order of tokens in a sequence, positional encodings are added to these embeddings. Positional encodings are numerical values that encode the position of each token in the sequence, enabling the Transformer to distinguish between tokens at different positions.

Encoder: the encoder's role is to process the input sequence and capture relationships between all parts of the input. For example, in a sentence, it learns how words relate to one another. Each encoder layer performs the following operations:

- Self-Attention: compares each token in the input sequence to every other token, computing relevance scores that help capture dependencies and context.
- Feedforward layers: apply transformations to the representations output by the self-attention mechanism, processing each position independently to refine token representations.
- Layer normalization: ensures stable and efficient training by regulating the range of outputs at each layer.
- Residual connections: shortcuts that bypass each layer, ensuring that information from earlier layers is preserved.

After processing the input, the encoder generates a sequence of embeddings enriched with contextual information, which are subsequently fed into the decoder to guide the generation process.

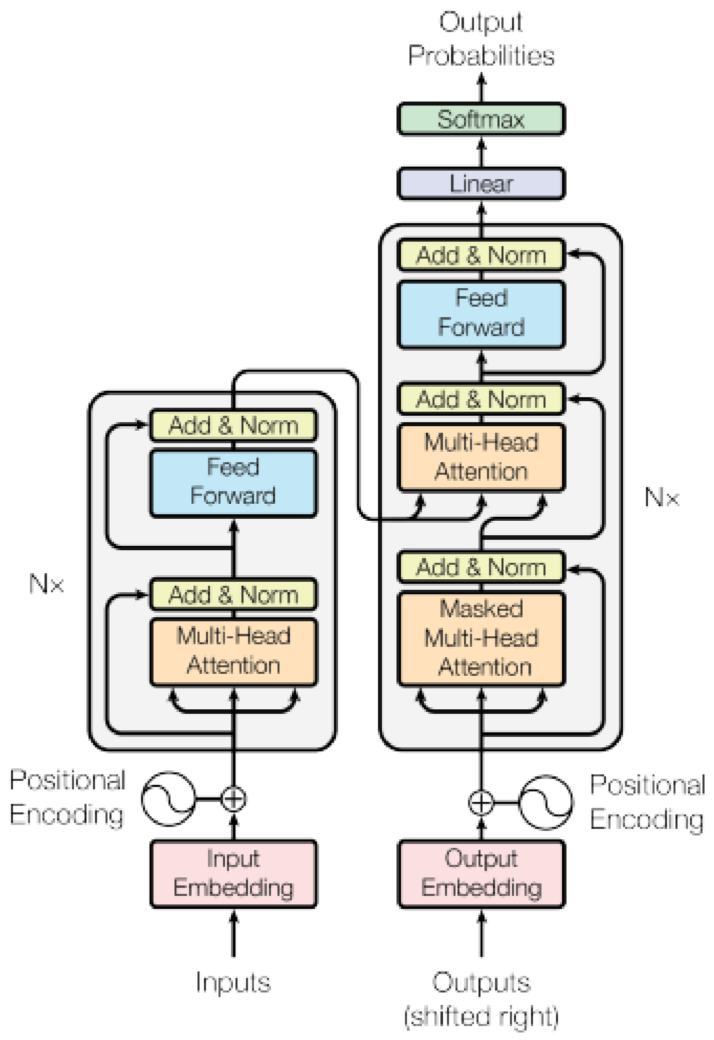


Figure 2.8: Architecture of the transformer model

Decoder: The decoder generates the output sequence (e.g., a translated sentence or generated text) by attending to both:

- The encoder output representations (through cross-attention), aligning the information from the input sequence with the corresponding positions in the output.
- The previously generated tokens (via masked self-attention), ensuring that the decoder cannot access future tokens and thus preserving the autoregressive property.

Each decoder layer has a structure similar to the encoder, but with the addition of cross-attention. Specifically:

- Masked Self-Attention: like the encoder, the decoder uses self-attention to evaluate relationships within the output sequence. However, masking ensures that each position can only attend to itself and earlier positions, preserving the autoregressive nature of sequence generation.
- Cross-Attention: this mechanism integrates information from the encoder's output, allowing the decoder to align its generated sequence with the input sequence.
- Feedforward Layers: These layers refine the token representations further, applying transformations at each position.
- Layer normalization and residual connections: as in the encoder, these mechanisms ensure stable training and effective gradient flow.

The final output of the decoder is passed through a linear projection followed by a softmax function, generating a probability distribution over the vocabulary for each position in the sequence.This allows the model to predict the next token in the sequence.

In essence, the encoder models the global dependencies within the input sequence, producing a comprehensive representation. The decoder then leverages this information to generate the output sequentially, ensuring contextual consistency at each step. The combination of self-attention, cross-attention, and feedforward processing makes the Transformer architecture powerful for sequence-to-sequence tasks, such as machine translation and text generation.

Applications of Transformers include:

- NLP: capturing word dependencies for translation, summarization, and question answering (e.g., BERT, GPT).
- Vision: modeling spatial relationships in images (e.g., Vision Transformers).
- Multimodal Models: aligning text and images in tasks like text-to-image synthesis (e.g., DALL · E, Stable Diffusion).

2.3 Vision Transformers (ViTs)

Vision Transformers (ViTs) represent a groundbreaking application of the Transformer architecture in the field of computer vision, introduced by Dosovitskiy et al. in the paper "An Image is Worth 16x16 Words" (2020)[22].

Computer vision has traditionally relied on convolutional neural networks (CNNs) as the foundational architecture. Attention mechanisms have typically been integrated alongside CNNs or used to substitute specific components while preserving the overall convolutional structure. This paper, however, shows that the transformer architecture—initially developed for processing sequential data such as text—can successfully capture both local and global patterns in images without the need for any convolutional layers.

ViTs apply the Transformer architecture to image data by treating an image as a sequence of patches. Each patch acts as a token, analogous to words in NLP, allowing the self-attention mechanism to directly model long-range dependencies and global context across the image.

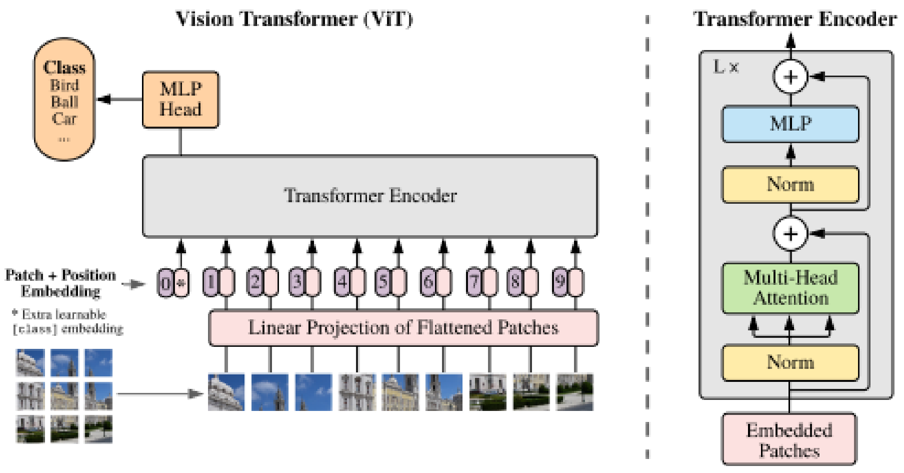


Figure 2.9: ViT architectecute.

Input representation: an image x of size $H \times W \times C$ (height, width, channels) is split into a sequence of fixed-size non-overlapping patches x_p of size $P \times P \times C$, resulting in

$$N = \frac{HW}{P^2}$$

patches.

To prepare the image patches for further processing, each one is mapped into a latent embedding space of dimension Dthrough a learnable linear transformation:

$$z_p = x_p E, \quad E \in \mathbb{R}^{(P^2 \cdot C) \times D}, \quad z_p \in \mathbb{R}^{N \times D}$$

Class token: the sequence of patch embeddings is augmented by prepending a learnable class token, denoted as

$$z_0^0 = x_{\text{class}}$$

After passing through the Transformer encoder, the final state of this token, z_L^0 is used as the resulting image representation y, allowing the model to aggregate information from all patches. The class token is updated during the attention process and is used as the final representation for classification tasks

Adding positional encodings: since attention mechanisms lack inherent positional awareness, positional encodings $E_{pos} \in \mathbb{R}^{(N+1) \times D}$ are introduced to retain the spatial structure of the image. These encodings are added to the corresponding patch embeddings, enabling the model to differentiate between positions. As a result, the Transformer receives as input the element-wise sum of the patch embeddings and their associated positional encodings

The input to the Transformer becomes:

$$z_0 = [x_{\text{class}}; x_1^p E; x_2^p E; \dots; x_N^p E] + E_{\text{pos}}, \quad E \in \mathbb{R}^{(P^2 \cdot C) \times D}, \quad E_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$$

The Transformer encoder, adapted from the original Transformer model, processes the input embeddings through multiple layers of multi-headed self-attention (MSA) and feedforward networks (MLPs):

Multi-Headed Self-Attention: to capture global relationships within the image, each patch embedding—including the class token—attends to all others in the sequence. Through self-attention, the model computes a weighted combination of all patch embeddings for each individual patch, where the weights reflect the contextual relevance between patches. This mechanism enables the model to recognize long-range dependencies and dynamically focus on the most informative regions of the image.

For each layer ℓ:

$$z'_\ell = \mathrm{MSA}(\mathrm{LN}(z_{\ell-1})) + z_{\ell-1}, \quad \ell = 1, \ldots, L$$

Feedforward network (Multi-Layer Perceptrons): each token embedding, after self-attention processing, is passed through a feedforward neural network (typically an MLP) to introduce non-linearity and model complex dependencies between patches. This component generally consists of two fully connected layers with a GELU activation in between. The output is processed with layer normalization and added back to the input via a residual connection:

$$z_\ell = \mathrm{MLP}(\mathrm{LayerNorm}(z'_\ell)) + z'_\ell, \quad \ell = 1, \ldots, L$$

Layer normalization and residual connections: layer normalization is applied before both the self-attention and feedforward sub-layers, stabilizing training and improving convergence, while residual connections are introduced to preserve information across layers and prevent vanishing gradients.As a result, the encoder stack processes the input sequence iteratively through multiple layers, producing a sequence of refined embeddings.

After L encoder layers, the output embedding z_L^0 corresponding to the class token is extracted and normalized to obtain the final image representation:

$$y = \mathrm{LN}(z_L^0).$$

ViTs and CNNs: the paper also discusses ViTs compared to CNNs.

Vision Transformers (ViTs) differ significantly from Convolutional Neural Networks (CNNs) in their architectural inductive biases. CNNs naturally encode locality, spatial hierarchy, and translation equivariance through convolutional layers and pooling. In contrast, ViTs are largely agnostic to the two-dimensional structure of images.

ViTs rely on global self-attention, which lacks inherent locality or spatial continuity. The only explicit 2D structure is introduced during preprocessing—when the input image is divided into fixed-size patches—and during fine-tuning with positional embeddings. Unlike CNNs, where spatial relationships are encoded at every layer, ViTs must learn these relations from data, making them more data-hungry but also more flexible once sufficiently trained. Because of their lack of strong priors, ViTs tend to require large-scale datasets to achieve competitive performance. This limitation is often addressed by pretraining on massive datasets (e.g., ImageNet-21k or JFT-300M), or by using hybrid architectures that combine convolutional layers with Transformer blocks to improve local feature extraction.

Fine-tuning at higher resolutions. It is common practice to fine-tune ViTs using input images of higher resolution than those used during pre-training. While the patch size is kept constant, increasing the image resolution leads to a larger number of patches, and consequently, a longer input sequence for the Transformer. Although the Transformer architecture can handle sequences of variable length, this change introduces complications with positional embeddings, because, since the positional embeddings learned during pre-training are tied to a fixed sequence length, they may not align properly with the extended sequence during fine-tuning, potentially impairing model performance due to positional misalignment.

To address this, the pre-trained positional embeddings are adjusted using two-dimensional interpolation based aligning them with the new patch grid based on their original positions in the image. This adjustment ensures that the spatial relationships learned in the pre-trained model remain meaningful even after a change in resolution.

Resolution scaling and patch extraction introduce the only explicit inductive bias related to the two-dimensional structure of images, as the Vision Transformer does not inherently encode spatial priors.

2.4 Diffusion Models

Diffusion models were introduced in 2020 with the paper *Denoising Diffusion Probabilistic Models (DDPMs)* by Ho et al [14]. These models conceptualize image generation as the reversal of a process that gradually adds noise to data, destroying its structure. By learning to iteratively denoise a starting sample of pure noise, diffusion models can reconstruct highly realistic and diverse outputs.

Diffusion models rely on the concept of a Markov chain, a mathematical framework for modeling systems where the next state depends only on the current state, not on the full history. In the context of diffusion models, the forward process is modeled as a Markov chain in which Gaussian noise is gradually added to the data over a series of steps, eventually transforming the data into pure noise.

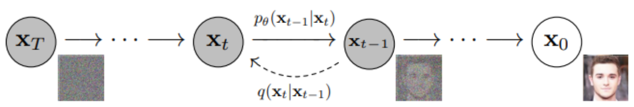

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

Figure 2.10: Diffusion process as a Markov chain

During generation (the reverse process), the model learns to reverse this Markov chain by denoising step-by-step, reconstructing the original data starting from pure noise. This iterative refinement process allows diffusion models to generate diverse and coherent outputs, and with recent advancements, they now produce highly realistic and detailed images, suitable for applications such as text-to-image synthesis and art generation.

2.4.1 Forward process

The forward diffusion process consists of iteratively adding Gaussian noise to a real sample $x_0 \sim q(x)$ over T discrete time steps until the data distribution becomes indistinguishable from pure Gaussian noise. The transition between steps follows a Markov chain, where each step applies a controlled amount of noise determined by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I).$$

where:

- $\beta_t \in (0,1)$ is a variance schedule controlling the amount of noise added at each step.
- $\sqrt{1-\beta_t}\,x_{t-1}$ represents the retained signal from the previous step.
- $\beta_t I$ is the variance (noise) added at step t.

By iteratively applying this process, the sample is transformed into pure noise as $T \to \infty$. The entire sequence of noisy samples can be described as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}).$$

To generate new samples, the model must learn to reverse this diffusion process, sampling from $q(x_{t-1}|x_t)$ to reconstruct the original data from a Gaussian noise input $x_T \sim \mathcal{N}(0, I)$. However, directly estimating the true posterior distribution $q(x_{t-1}|x_t)$ is intractable, as it requires computing the entire data distribution. Instead, the model learns a reverse process $p_\theta(x_{0:T})$, which consists of a sequence of Gaussian transitions that approximate $q(x_{t-1}|x_t)$:

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)).$$

This learned denoising process consists of a sequence of Gaussian transitions that progressively reconstruct the original sample from noise.

The variance $\Sigma_\theta(x_t, t)$ in the reverse process is typically set as a trainable or predefined function, and training is performed by maximizing the variational lower bound (ELBO) on the negative log-likelihood:

$$\mathbb{E}[-\log p_\theta(x_0)] \le \mathbb{E}_{x_{0,1:T}}\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right].$$

The forward diffusion process introduces noise into the data over T timesteps using a schedule of variance parameters β_t, which can either be predefined or learned through reparameterization. An important characteristic of this process is the ability to directly sample a noised latent x_t from the original input x_0 at any arbitrary timestep t, thanks to a closed-form expression based on cumulative noise scaling.

Letting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the distribution of x_t given x_0 is:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$$

This formulation allows efficient training and denoising, as the model only needs to learn how to reverse these Gaussian transitions by predicting the noise added at each step, that is, it learns to approximate Gaussian transitions with a parameterized mean function.

The full loss function can be expressed as a sum of KL divergence terms, where each term compares a Gaussian posterior to a learned Gaussian approximation:

$$L_{\text{VLB}} = L_T + L_{T-1} + \cdots + L_0.$$

Since L_T is constant (as x_T is simply Gaussian noise), it is ignored during training. The term L_0 is modeled using a separate decoder trained to reconstruct the original data sample. The reverse process mean function μ_θ is optimized to predict a denoised estimate $\tilde{\mu}_t$, which leads to a simplified training objective:

$$L_t^{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\left[||\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2\right].$$

By using this simplified training objective, diffusion models achieve high-quality generative performance with improved stabilit, making them effective for generating realistic images, text, and other structured data.

2.4.2 U-Net architecture

The U-Net is the core architecture used in many diffusion models for denoising. The following explanation largely comes from [23]. The U-Net is a convolutional neural network with an encoder-decoder structure. The encoder progressively downsamples the noisy image x_t to extract hierarchical features, while the decoder upsamples these features to reconstruct the denoised image x_{t-1}. Skip Connections are introduced between the encoder and the decoder to preserve important information and improve gradient flow. The time step t is encoded using a sinusoidal positional embedding and injected into the network to inform it of the particular time step (noise level) it is operating.

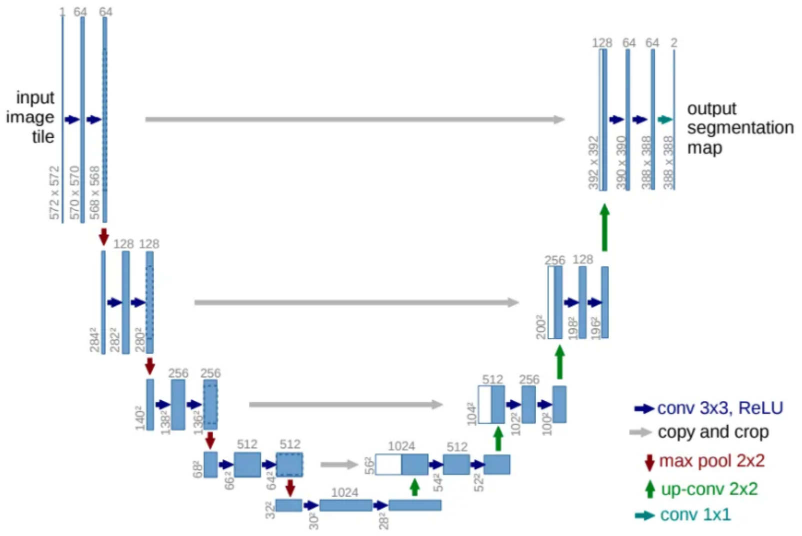


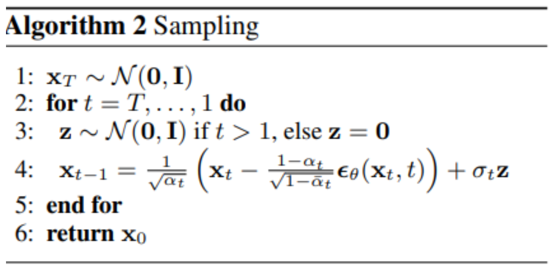
Figure 2.11: Unet architecture in Diffusion models.

As the core building block of the U-Net model, the DDPM authors employed a Wide ResNet block [24]. Next, the attention module is added to both the encoder and decoder part of the U-Net, usually between ResNet blocks, and convolutional/attention layers are interleaved with group normalization.

The network is built up as follows:

- First, a convolutional layer is applied on the batch of noisy images, and position embeddings are computed for the noise levels.
- Next, a sequence of downsampling stages are applied. Each downsampling stage consists of 2 ResNet blocks + group normalization + attention + residual connection + a downsample operation.
- At the middle of the network, again ResNet blocks are applied, interleaved with attention.
- Next, a sequence of upsampling stages are applied. Each upsampling stage consists of 2 ResNet blocks + group normalization + attention + residual connection + an upsample operation.
- Finally, a ResNet block followed by a convolutional layer is applied.

2.4.3 Sampling process



Algorithm 2 Sampling

```
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: for $t = T, \dots, 1$ do
3:   $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4:   $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: end for
6: return $\mathbf{x}_0$
```

Figure 2.12: Sampling algorithm in Diffusion models

After training, the model generates new samples by iteratively denoising a pure noise input sampled at time step T from a Gaussian distribution:$x_T \sim \mathcal{N}(0, I)$ until we end up at time step $t = 0$ with an image that ideally looks like it came from the real data distribution. The sampling process is:

1. Initialization: start with x_T.
2. Reverse Diffusion: for $t = T$ down to 1:
    (a) Predict $\mu_\theta(x_t, t)$ or $\epsilon_\theta(x_t, t)$.
    (b) Sample x_{t-1} from $p_\theta(x_{t-1} \mid x_t)$.

3. Output: the final output x_0 is the generated image.

2.4.4 Improved Denoising Diffusion Probabilistic Models (DDPMs)

While DDPMs produce high-quality images, their log-likelihood scores remain suboptimal, indicating that while the models generate visually realistic samples, they do not always capture the full diversity of the real data distribution. Nichol and Dhariwal's Improved Denoising Diffusion Probabilistic Models [25] introduced key refinements aimed at improving log-likelihood while maintaining high sample quality.

One major improvement is learning the variance term $\Sigma_\theta(x_t)$ in the reverse process rather than keeping it fixed. The original DDPMs set Σ_θ to a constant value, which limited flexibility. However, learning Σ_θ directly was found to be unstable. The improved approach parameterizes the variance as an interpolation between an upper bound β_t and a lower bound $\tilde{\beta}_t$, formulated as:

$$\log \Sigma_\theta(x_t) = v \log \tilde{\beta}_t + (1 - v) \log \beta_t$$

where v is a learnable parameter that determines the weighting of the two variance bounds. This formulation allows for stable variance learning, improving log-likelihood without degrading sample quality.

Additionally, Improved DDPMs optimized the training objective by introducing a hybrid loss function:

$$L = L_{\text{simple}} + \lambda L_{\text{vlb}}$$

where L_{simple} is the standard mean squared error (MSE) loss between predicted and real noise, and L_{vlb} is the variational lower bound (VLB) loss, which incorporates the learned variance. The weight λ (set to 0.001) ensures that the model primarily optimizes for sample quality while still improving likelihood estimation.

A significant refinement was made to the noise scheduling strategy. The original DDPMs used a linear schedule for the noise variance β_t, interpolating between 10^{-4} and 0.02. However, this approach added noise too aggressively in the early timesteps, causing unnecessary information loss. Improved DDPMs instead proposed a cosine noise schedule:

$$\bar{\alpha}_t = \cos\left(\frac{t/T + s}{1 + s}\frac{\pi}{2}\right)^2$$

where $s = 0.008$ prevents singularities at the start of the process. This adjustment allows the model to retain more image information for longer, improving training stability and sample quality. Compared to the linear schedule, the cosine schedule prevents excessive noise accumulation in the early timesteps, leading to smoother training and improved likelihood scores.

2.5 Latent Diffusion Models

Latent Diffusion Models (LDMs) address the computational inefficiency of traditional diffusion models by shifting the diffusion process from pixel space to a learned latent space. Standard diffusion models operate directly on high-dimensional pixel data, making training and inference computationally expensive, often requiring hundreds of GPU days. LDMs mitigate this by encoding images into a compact latent space where the generative process takes place, significantly reducing computational costs while maintaining high-quality synthesis.

Proposed by Rombach et al. in 2022 [12], LDMs operate by performing the diffusion process in a lower-dimensional latent space instead of directly in the pixel space. This latent space is learned through a variational autoencoder (VAE) or a similar encoding mechanism, which maps the high-dimensional input data (e.g., images) into a compressed latent representation. The latent space captures the essential structure and semantics of the data, reducing the dimensionality while preserving the quality of the generated outputs.

In the forward process, Gaussian noise is added step-by-step to the latent representation of the image, gradually destroying its structure. The reverse process then iteratively removes this noise to reconstruct the latent representation. Finally, the VAE decoder maps the denoised latent representation back into the high-dimensional pixel space, producing the final image. The efficiency of LDMs comes from performing the computationally intensive diffusion process in the compact latent space, which drastically reduces the number of required operations compared to pixel-space diffusion.

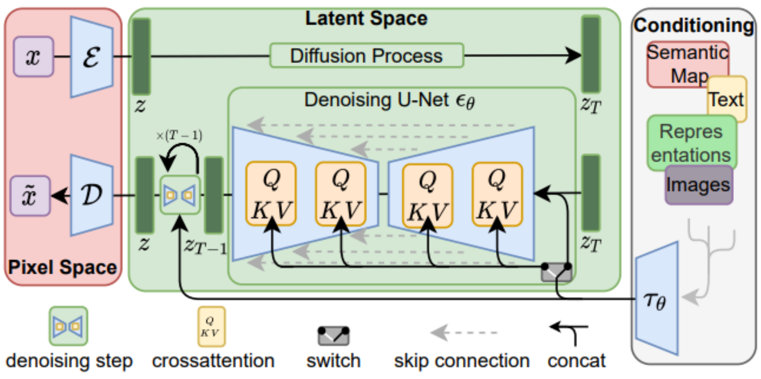


Figure 2.13: Structure of Latent Diffusion Models.

2.5.1 Perceptual image compression

The foundation of LDMs lies in learning an efficient lower-dimensional representation where diffusion can occur. This is achieved through an autoencoder consisting of an encoder E and a decoder D, which map an input image $x \in \mathbb{R}^{H\times W\times 3}$ into a latent representation z:

$$z = E(x), \quad z \in \mathbb{R}^{h\times w\times c}$$

where $h = H/f$ and $w = W/f$ represent the spatial dimensions after downsampling by a factor f. The decoder reconstructs the original image:

$$\hat{x} = D(z) = D(E(x))$$

To prevent high variance in the latent space, two regularization techniques are experimented:

- **KL regularization (KL-reg)**: encourages the latent distribution to follow $\mathcal{N}(0, I)$, similar to Variational Autoencoders (VAEs).
- **Vector Quantization regularization (VQ-reg)**: introduces a vector quantization layer, similar to VQGANs, which helps control the variance while preserving local realism.

Because the diffusion model operates within a two-dimensional learned latent space $z = E(x)$, relatively mild compression rates can be used while maintaining high-quality reconstructions. This contrasts with previous approaches that relied on aggressive spatial compression, ignoring much of its inherent structure. By leveraging a structured latent space, LDMs are able to preserve finer details of x more effectively, resulting in superior reconstructions without excessive compression artifacts.

2.5.2 Diffusion process in the latent space

After training the autoencoder, the diffusion process is applied directly within the latent space rather than on pixel-level data. The forward diffusion process progressively adds Gaussian noise to the latent representation over T discrete timesteps. At each step t, the transition is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I)$$

where β_t represents a predefined variance schedule that controls the amount of noise injected at each timestep.

The full forward process is thus a Markov chain formulated as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

As t increases, the latent representation x_t becomes progressively noisier, eventually approaching an isotropic Gaussian distribution.

To reverse this process and recover the clean latent from noise, the model learns a denoising distribution. Since the true posterior $q(x_{t-1}|x_t)$ is intractable, a neural network ϵ_θ is trained to approximate the added noise. The learned reverse process is modeled as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Here, μ_θ and Σ_θ denote the predicted mean and variance, respectively, at timestep t, both conditioned on the current noisy latent x_t. This formulation allows the model to iteratively denoise and reconstruct the original latent code from pure noise.

2.5.3 Training objective and optimization

Rather than applying an autoregressive, attention-based approach, LDMs take advantage of image-specific inductive biases. The diffusion process operates within a U-Net built primarily from 2D convolutions, making it significantly more efficient than using a transformer-based generative model.

The training centers on reconstructing the noise added during the forward diffusion process, using an objective function that is expressed as a simplified version of the Evidence Lower Bound (ELBO):

$$L_{\text{LDM}} = \mathbb{E}_{E(x),\epsilon\sim\mathcal{N}(0,I),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right]$$

where:

- z_t denotes the latent representation corrupted by Gaussian noise at timestep t,
- $\epsilon_\theta(z_t, t)$ is the model's prediction of the added noise.

To enhance computational efficiency, LDMs streamline the training process by applying the forward diffusion directly in latent space: instead of first encoding the image and then performing diffusion, both operations are integrated into a single step. Likewise, reconstruction involves decoding the denoised latent back into the pixel space in a single decoding pass.

2.5.4 Conditional generation and cross-attention

LDMs facilitate conditional image generation by learning the conditional probability distribution $p(z \mid y)$, where y denotes auxiliary information such as text prompts, segmentation maps, or other modality-specific inputs. This conditioning is incorporated into the denoising process through a conditional denoising autoencoder $\epsilon_\theta(z_t, t, y)$, which leverages the conditioning signal y to steer the generation toward semantically aligned outputs.

To achieve this, the U-Net backbone is augmented with a cross-attention mechanism. A domain-specific encoder τ_θ maps the conditioning input y into an intermediate representation:

$$\tau_\theta(y) \in \mathbb{R}^{M\times d_\tau}$$

This representation is integrated into the attention layers of the U-Net using:

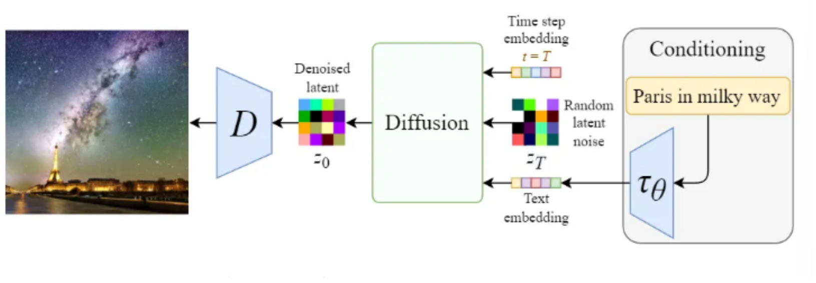


Figure 2.14: Overview of the conditioning mechanism

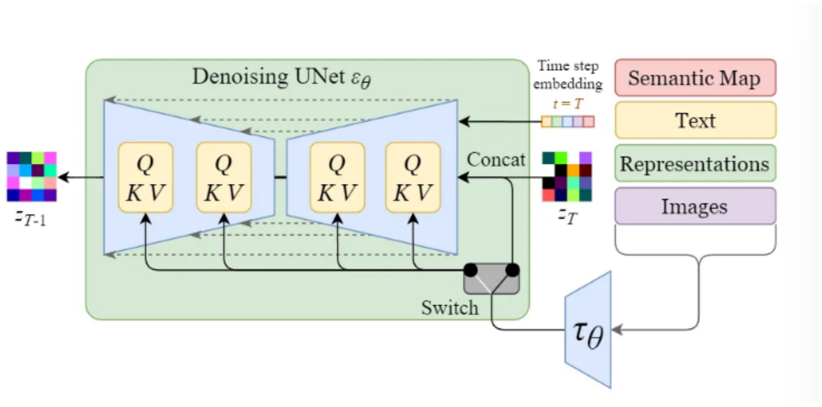


Figure 2.15: Conditioning mechanism details

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

where:

$$Q = W_Q^{(i)} \phi_i(z_t), \quad K = W_K^{(i)} \tau_\theta(y), \quad V = W_V^{(i)} \tau_\theta(y)$$

The switch in the above diagram is used to control between different types of conditioning inputs. For text inputs, they are first converted into embeddings (vectors) using a language model τ_θ(e.g. BERT, CLIP), and then they are mapped into the U-Net via the (multi-head)

$$\text{Attention}(Q, K, V)$$

layer. For other spatially aligned inputs (e.g. semantic maps, images, inpainting), the conditioning can be done using concatenation.

The training objective for the conditional Latent Diffusion Model involves minimizing the discrepancy between the true noise and the model's predicted noise across varying timesteps. Formally, the loss function is defined as:

$$L_{\text{LDM}} = \mathbb{E}_{E(x),y,\epsilon\sim\mathcal{N}(0,I),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right]$$

In this setup, both the conditioning encoder network τ_θ, which processes the input condition y, and the denoising network ϵ_θ, which predicts the noise at timestep t, are trained simultaneously.

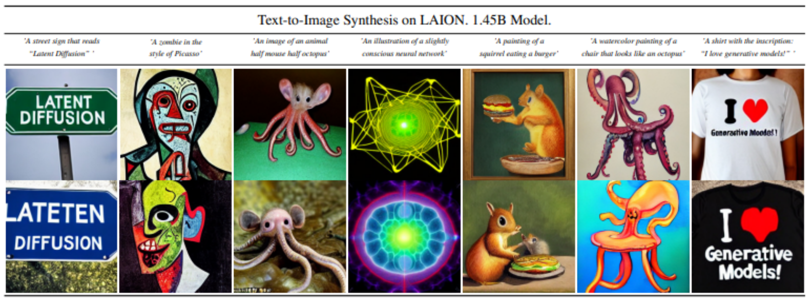

Text-to-Image Synthesis on LAION. 1.45B Model.
'A street sign that reads "Latent Diffusion" '
'A zombie in the style of Picasso'
'An image of an animal half mouse half octopus'
'An illustration of a slightly conscious neural network'
'A painting of a squirrel eating a burger'
'A watercolor painting of a chair that looks like an octopus'
'A shirt with the inscription: "I love generative models!" '
LATENT DIFFUSION
LATETEN DIFFUSION
I Generative Models!

Figure 2.16: Samples of text-to-image synthesis shown in the paper.

By performing diffusion in a lower-dimensional space, LDMs achieve substantial computational savings. Training and inference require significantly fewer operations compared to pixel-space diffusion models. Additionally, the use of U-Net architectures with 2D convolutions provides an effective balance between efficiency and quality.

Compared to autoregressive generative models, which rely on transformers operating in a highly compressed latent space, LDMs allow for a more structured approach. By leveraging spatial inductive biases and a mild compression ratio, they produce high-fidelity outputs without excessive loss of detail.

LDMs support a wide range of tasks, including:

- Text-to-Image synthesis: generating images from textual descriptions using token-based conditioning mechanisms.
- Inpainting: filling in missing regions of an image based on surrounding pixels.
- Super-Resolution: enhancing the resolution of low-quality images.
- Domain adaptation: adapting images to different styles or semantic contexts.

During inference, the entire process—encoding the input into latent space, applying the diffusion model, and decoding back to pixel space—is performed efficiently in a single pass at each stage, making LDMs highly practical for real-world applications.

2.6 CLIP

CLIP (Contrastive Language–Image Pretraining) [26] is a powerful model developed by OpenAI that bridges the gap between vision and language by aligning images with their natural language descriptions in a shared embedding space. Unlike traditional supervised models that aim to classify images into specific predefined class labels, CLIP learns to associate images with text descriptions, providing a more flexible and general approach to understanding and generating multimodal content.

The central principle behind CLIP is to learn a joint embedding space for images and text that captures semantic similarity through contrastive learning. Rather than requiring exact textual-image alignment, CLIP is trained to draw together the representations of corresponding image-text pairs while pushing apart those of unrelated ones. This is accomplished via a contrastive loss function, which maximizes the cosine similarity for positive pairs (i.e., matching images and captions) and minimizes it for negative pairs (i.e., mismatched combinations). The resulting model ensures that semantically aligned visual and textual data are positioned closely within the shared latent space.

CLIP is composed of two primary modules: an image encoder and a text encoder. The image encoder transforms visual inputs into fixed-size embeddings using either convolutional architectures such as ResNet-50 or more recent alternatives like Vision Transformers (ViT). These embeddings are then aligned with those produced by the text encoder, enabling cross-modal retrieval and zero-shot classification tasks.

The choice of encoder significantly impacts CLIP's performance and behavior. ResNet-based encoders, with their convolutional structure, introduce strong inductive biases that favor local feature extraction, similar to traditional CNNs. In contrast, Vision Transformers operate on image patches and lack these inductive biases, leading to a more flexible but data-hungry model. ViTs are often preferred for their ability to model global context by processing images as sequences of patches, or even feature maps, allowing for a hybrid architecture.

The text encoder implemented as a Transformer-based model transforms the tokenized textual inputs into vector embeddings. These embeddings are then projected into a shared latent space alongside visual representations and the correspondence between image and text pairs is evaluated by computing the cosine

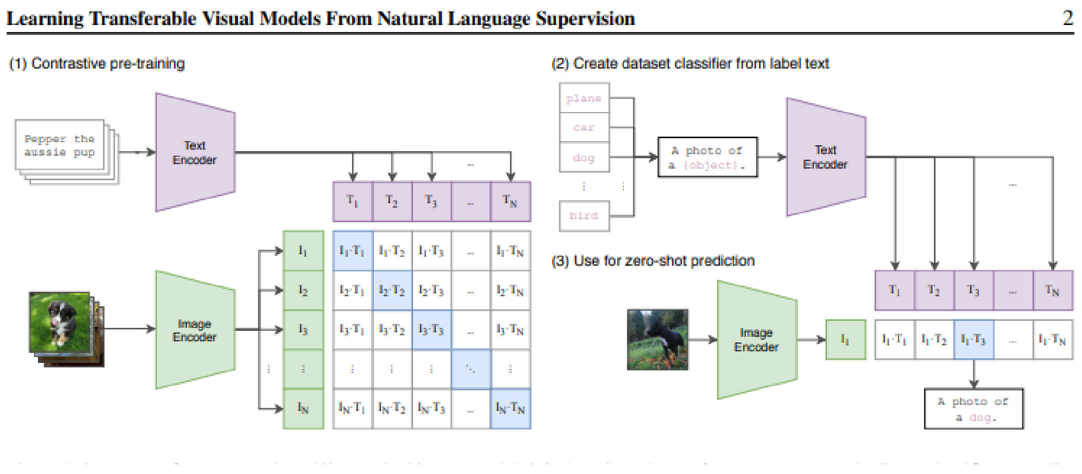

Learning Transferable Visual Models From Natural Language Supervision
2
(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
A photo of a dog.

Figure 2.17: Summary of the CLIP approach.

similarity. The encoder processes text in a manner analogous to standard large language models, utilizing special tokens such as `[SOS]` and `[EOS]` to denote the start and end of a sequence. The final embedding for a given input is obtained from the activation associated with the `[EOS]` token at the final layer of the transformer, encapsulating the semantic summary of the entire sequence.

Training involves passing batches of images and their corresponding text descriptions through their respective encoders. The model computes cosine similarities between all image-text pairs in the batch, applying a contrastive loss to pull matching pairs closer and push mismatched pairs apart. This process aligns the image and text embeddings in a way that facilitates cross-modal understanding.

A key challenge in CLIP training is ensuring that the model learns robust and generalizable representations. Unlike standard classification models that rely on predefined categories, CLIP learns from natural language supervision, meaning that it is less constrained by dataset-specific biases but more sensitive to variations in text phrasing.

CLIP's primary strength lies in its ability to generalize and excel in multimodal tasks by aligning images and text in a shared embedding space. This alignment enables zero-shot classification, where CLIP can classify images into arbitrary categories defined by textual descriptions without requiring fine-tuning. For instance, it can identify an image as "a dog playing in a park" simply by comparing the image's embedding to embeddings of descriptive text prompts.

One of the major considerations when using CLIP for classification is the impact of prompt engineering. Unlike traditional models that rely on fixed class labels, CLIP is highly sensitive to the phrasing of textual inputs. Rewriting labels into more descriptive phrases, such as replacing "cat" with "a photo of a cat," significantly improves classification accuracy. This sensitivity arises because CLIP was trained on internet-scale text, where descriptions often include contextual information rather than single-word labels. Additionally, CLIP's reliance on large-scale natural language supervision introduces challenges related to polysemy (words with multiple meanings). Effective prompt design can mitigate these issues by ensuring that descriptions are precise and contextually relevant.

In text-to-image generation systems such as Stable Diffusion, CLIP plays a pivotal role. Text inputs are converted into embeddings using CLIP (or a CLIP-like encoder), which then condition the image generation process, ensuring that the generated visuals align semantically with the input text. Beyond text-to-image tasks, CLIP's architecture supports applications like text-guided image retrieval, retrieving the most relevant images for a given textual query by comparing embeddings in the latent space. It also facilitates tasks like image segmentation and localization, where textual prompts guide the identification of objects or regions within an image.

A central innovation introduced by CLIP lies in its use of natural language as a supervisory signal, moving away from the reliance on fixed category labels typical of traditional image classification models. By training on large-scale collections of image-text pairs sourced from the web, CLIP bypasses the need for manually annotated datasets and predefined class taxonomies. This strategy enables remarkable generalization capabilities, allowing CLIP to perform zero-shot classification across a wide range of tasks. Its ability to learn visual concepts directly from textual descriptions makes it more flexible and broadly applicable than conventional models trained on datasets like ImageNet. Nonetheless, this design choice also brings certain limitations—particularly in cases involving ambiguous or polysemous language—where careful prompt formulation becomes crucial for accurate interpretation.

Additionally, tasks requiring complex or systematic reasoning, such as counting objects or estimating spatial proximity, remain difficult for CLIP. Unlike structured classification models that explicitly learn numerical relationships, CLIP's embedding space primarily captures semantic similarity, making it less effective for tasks requiring precise quantitative reasoning.

Like many large-scale AI models, CLIP also inherits biases from its training data. Since it learns from diverse internet sources, its representations can reflect social biases present in online content. This has implications for fairness and robustness, as certain concepts may be overrepresented or underrepresented depending on their prevalence in the training data.

2.7 Classifier-Free Guidance

Classifier-Free Guidance (CFG) [27] is a critical innovation in the evolution of conditional diffusion models, enabling efficient and flexible integration of conditioning inputs (e.g., text prompts) into the image generation process. Unlike earlier approaches that required a separately trained classifier to guide the generative process, CFG eliminates the need for an external classifier by integrating conditional guidance directly into the diffusion model.

In the early stages of conditional diffusion models, classifier guidance [15] was the dominant approach for steering the image generation process toward specific conditions. This method relied on a separately trained classifier $p(y \mid x)$ to predict the conditional probability of a desired condition y (e.g., a label like “cat”) given the image x. During the reverse diffusion process, the gradients of the classifier with respect to the image x served as a signal to guide the denoising process. Mathematically, this involved computing

$$\nabla_x \log p(y \mid x),$$

which adjusted the denoising trajectory to align with the specified condition.

Mathematically, the guided score function is given by:

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \, \nabla_x \log p(y \mid x),$$

where:

- $p(x)$ is the unconditional image distribution.
- $p(y \mid x)$ is the conditional probability from the classifier.
- γ is the guidance scale amplifying the conditioning signal.

Despite its effectiveness, classifier guidance introduced significant challenges. It required training a noise-robust classifier specifically for the purpose of guidance additional computational resources to train and integrate the classifier, and the resulting model often suffered from instability and artifacts. Moreover, this method was limited to predefined conditions, making it less suitable for handling flexible or descriptive inputs like free-form text prompts.

Classifier-Free Guidance (CFG) addresses these issues by integrating conditional guidance directly into the diffusion model, therefore eliminating the classifier while still providing class guidance to the model. This is achieved by training the diffusion model to handle both conditioned and unconditioned outputs through conditioning dropout, where a fraction of training steps replaces the conditioning input y with a null token. Consequently, the model learns both the conditional score function $\nabla_x \log p(x \mid y)$ and the unconditional score function $\nabla_x \log p(x)$. This dual capability enables CFG to interpolate between conditional and unconditional outputs during inference, allowing for fine-grained control over prompt adherence and diversity. By interpolating between these two, the effective score function becomes:

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\,\nabla_x \log p(x) + \gamma\,\nabla_x \log p(x \mid y),$$

where:

- γ is the guidance scale determining the balance between unconditional and conditional guidance.
- $\nabla_x \log p(x)$, unconditional score function, promotes diversity.
- $\nabla_x \log p(x \mid y)$, the conditional score function, enhances adherence to the conditioning signal.

In practice, CFG is implemented by interpolating between two predictions:

- The noise prediction conditioned on the input, denoted as $\epsilon_\theta(x_t, t, y)$, where y represents the conditioning input (e.g., a text embedding).
- The unconditional noise prediction $\epsilon_\theta(x_t, t)$, which provides a more diverse and less constrained output.

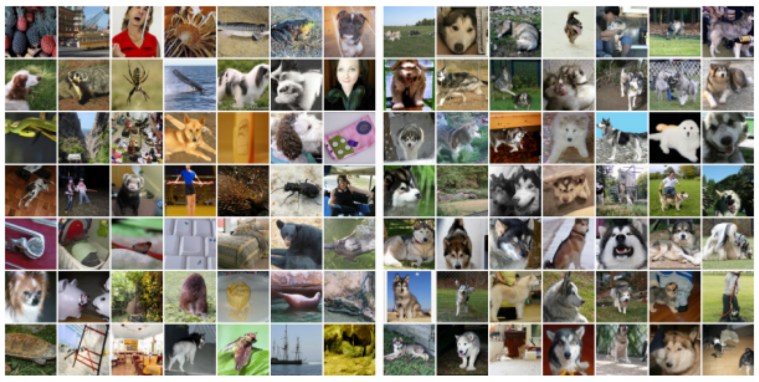
The interpolated (or guided) prediction is given by:

$$\bar{\epsilon}_\theta(x_t, t, y) = \epsilon_\theta(x_t, t, y) - \sqrt{1-\bar{\alpha}_t} w \nabla_x \log p(y \mid x_t)$$

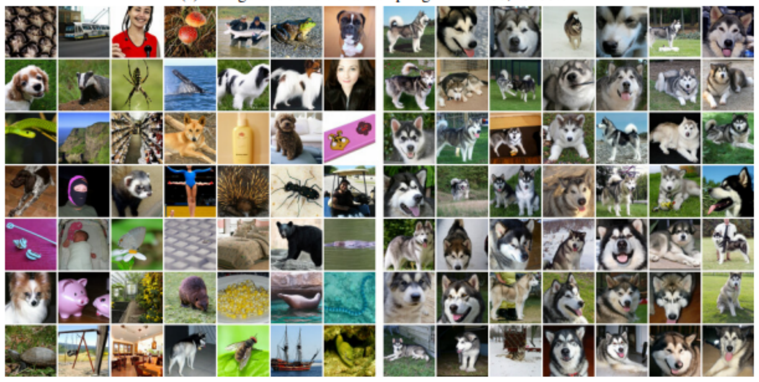
which can be rewritten as:

$$\begin{aligned}\bar{\epsilon}_\theta(x_t, t, y) &= \epsilon_\theta(x_t, t, y) + w(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t)) \\ &= (w+1)\epsilon_\theta(x_t, t, y) - w\epsilon_\theta(x_t, t)\end{aligned}$$

where $w \geq 1$ is the guidance scale. A higher w increases the influence of the conditioning prompt, ensuring that the generated image closely aligns with the input text. However, excessively high values may introduce artifacts or overly constrained results, whereas lower values allow for more creative and diverse outputs at the expense of prompt adherence.



(a) Non-guided conditional sampling: FID=1.80, IS=53.71



(b) Classifier-free guidance with $w = 1.0$: FID=12.6, IS=170.1

Figure 2.18: Comparison between non-guided conditional sampling and CFG sampling.

Classifier-Free Guidance (CFG) has become a standard technique in modern diffusion models such as Stable Diffusion and OpenAI's GLIDE. One of its key advantages is that it eliminates the need for an external classifier, enabling the diffusion model to use its own internal representations for guidance. This not only streamlines the architecture but also allows the model to better leverage its pre-trained knowledge. Additionally, CFG facilitates conditioning on complex and open-ended inputs—like natural language descriptions—which are often challenging to handle with conventional classifiers, thereby supporting more flexible and expressive generation.

For instance, text-conditioning dropout was used during training for Stable Diffusion 1.4 and 1.5 to improve robustness and enhance the classifier-free guidance mechanism. During inference, the CFG parameter (commonly referred to as the guidance scale) can be adjusted dynamically enabling users to balance between prompt adherence and output diversity.

Chapter 3

State-of-the-art Text-to-Image Models

3.1 Stable Diffusion

Stable Diffusion[17] [28]is a state-of-the-art generative model that was first released in 2022 by Stability AI in collaboration with CompVis and Runway. It is designed for efficient and high-quality image synthesis, particularly excelling in text-to-image tasks but also applied to inpainting, outpainting, and image-to-image tasks. Stable Diffusion builds on the Latent Diffusion Models (LDMs) framework, leveraging a compact latent space for computational efficiency while maintaining high-resolution and semantically rich outputs.

Stable Diffusion's architecture is composed of three primary components: a Variational Autoencoder (VAE), a denoising U-Net, and an optional text encoder for conditional generation. During training, the VAE encoder compresses input images into a latent representation that retains their semantic structure while operating in a lower-dimensional space, enabling more efficient computation. The forward diffusion process involves gradually perturbing these latent vectors by adding Gaussian noise over multiple time steps.

To reverse this process, the U-Net is trained to iteratively denoise the latent representation, effectively learning to reconstruct the original clean signal. After the denoising phase, the VAE decoder transforms the refined latent vector back into pixel space, producing the final image output.

The denoising process can be conditioned on various modalities, through the use of cross-attention mechanisms embedded within the U-Net. For text-guided synthesis, Stable Diffusion incorporates a frozen CLIP-based text encoder, which transforms textual prompts into vector embeddings. These embeddings serve as context vectors, guiding the U-Net's attention throughout the denoising steps and ensuring that the resulting image aligns with the semantic content of the prompt.

Stable Diffusion is built upon a U-Net backbone, comprising three main components: an encoder, a central processing block, and a decoder connected via skip connections. The encoder and decoder are each composed of 12 blocks, while the middle block completes the architecture, bringing the total to 25 blocks. Out of these, 8 are dedicated to spatial resolution changes through down-sampling and up-sampling convolutions. The remaining 17 blocks form the core processing units, each integrating 4 ResNet layers along with 2 Vision Transformer (ViT) modules. These ViTs embed multiple layers of cross-attention and self-attention operations, enabling rich contextual interactions and fine-grained conditioning throughout the diffusion process.

During the inference phase, the generation process starts from a latent tensor initialized with pure Gaussian noise. Through a series of iterative denoising steps, the model transforms this noise into a structured and meaningful image. This denoising is guided by conditioning inputs, which may include textual descriptions, visual references, or other modality-specific features. In the case of text prompts, Stable Diffusion utilizes a frozen, pre-trained text encoder to convert the prompt into a sequence of embeddings. These embeddings are integrated into the U-Net architecture via cross-attention layers. Specifically, queries derived from the latent representation interact with keys and values computed from the text embeddings, allowing the network to assign higher relevance to semantically important parts of the prompt. This mechanism ensures that the final output remains faithful to the intended textual guidance throughout the generation process.

3.1.1 Training datasets

Stable Diffusion is trained on large-scale datasets of image-text pairs to achieve robust generalization and high-quality outputs. Key datasets include:

LAION-5B: publicly available dataset[29] comprising over 5 billion image-text pairs sourced from the internet. This dataset offers immense diversity, including a wide range of styles, objects, and scenarios.

Filtered Subsets of LAION: for Stable Diffusion, subsets of LAION are curated and filtered to improve quality. For example, LAION-Aesthetics focuses on aesthetically pleasing images, filtering out low-quality or irrelevant samples.

The use of diverse and large-scale datasets ensures that Stable Diffusion can handle a variety of prompts, from highly specific queries to abstract or creative descriptions.

3.1.2 Released versions

Stable Diffusion has evolved significantly since its initial release, with each version aiming to improve in terms of resolution, training methodology, text-to-image alignment, and usability. Below is a detailed comparison of the key versions, highlighting their architectural features, use cases, and limitations.[30]

Stable Diffusion 1.x

The first generation of Stable Diffusion, released in 2022, includes versions 1.1 through 1.5. These models are trained on a dataset of 512×512 pixel images, with a text encoder based on CLIP ViT-L/14 for text conditioning.

Stable Diffusion v1-1:

- 237,000 steps at 256×256 on LAION-2B-en.
- 194,000 steps at 512×512 on LAION-high-resolution.

Stable Diffusion v1-2 (resumed from v1-1):

- 515,000 steps at 512×512 on LAION-improved-aesthetics.

Stable Diffusion v1-3 (resumed from v1-2):

- 195,000 steps at 512×512 on LAION-improved-aesthetics.
- Included 10% text-conditioning dropout for classifier-free guidance.

Stable Diffusion v1-4 (resumed from v1-2):

- 225,000 steps at 512×512 on LAION-aesthetics v2 5+.
- Included 10% text-conditioning dropout for classifier-free guidance.

Stable Diffusion v1-5 (resumed from v1-2):

- 595,000 steps at 512×512 on LAION-aesthetics v2 5+.

- Maintained 10% text-conditioning dropout for classifier-free guidance.

The models have 860 million parameters and excel in generating diverse outputs with relatively low computational requirements. However, the reliance on 512×512 resolution limits their performance when generating images at higher resolutions, often resulting in degradation or inaccuracies. Additionally, poor data quality for human limbs and faces in the LAION database contributed to challenges in generating anatomically correct images, especially hands and facial features. Despite these limitations, the open-source nature of 1.x models led to extensive community-driven fine-tuning, resulting in specialized checkpoint models like DreamShaper, Realistic Vision, or Juggernaut, which improve on the base model's performance for specific tasks such as photorealism or anime-style generation.

Stable Diffusion 2.x

Released in late 2022[31], the 2.x series (versions 2.0 and 2.1) introduced significant improvements in resolution and prompt alignment. These models were trained on LAION-Aesthetics v2, a filtered dataset emphasizing higher-quality, aesthetically pleasing images. The resolution was increased to 768×768 pixels, enabling sharper and more detailed outputs. Additionally, the text encoder was upgraded to OpenCLIP ViT-H/14, which allowed for more expressive and nuanced prompts. The number of parameters remained consistent with 1.x models at 860 million. In

Figure 3.1: Samples from the Stable Diffusion 2.x series.

addition to standard text-to-image generation, 2.x introduced advanced features such as depth-to-image generation and enhanced inpainting capabilities, enabling precise edits to specific regions of an image. While 2.x improved image quality and alignment with text descriptions, the changes in text encoding made it difficult to transfer prompts from the 1.x series, limiting its adoption in the open-source community.

Stable Diffusion XL (SDXL)

Released in July 2023, Stable Diffusion XL (SDXL)[32] represents a major leap forward in generative modeling, offering native 1024×1024 resolution and vastly improved handling of human anatomy, including limbs and faces. SDXL was trained on datasets with multiple aspect ratios, moving beyond the square-only focus of earlier versions. The model employs a dual text encoder system, combining OpenCLIP ViT-G/14 and CLIP ViT-L/14, which significantly enhances semantic understanding and alignment between text and image. SDXL also features a

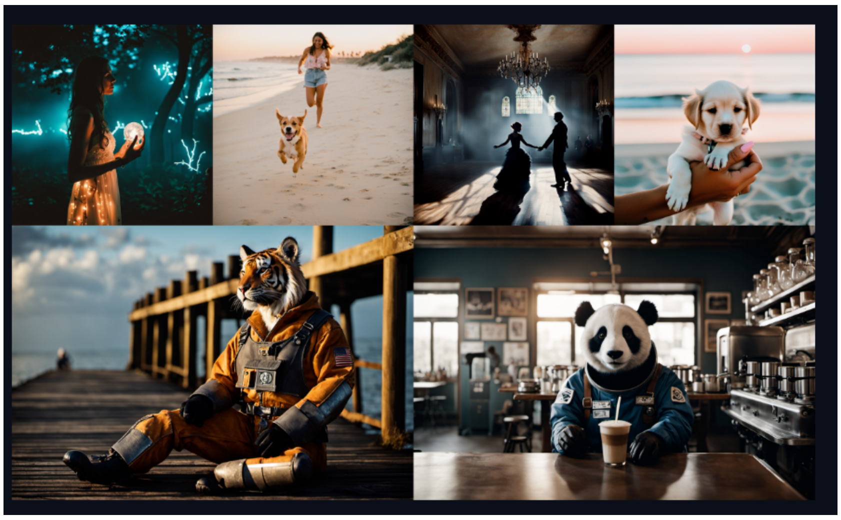


Figure 3.2: Samples generated by SDXL.

much larger architecture, with 3.5 billion parameters for its base model and a 6.6 billion parameter ensemble pipeline, enabling improved photorealism, color depth, and compositional accuracy. While SDXL delivers outputs comparable to systems like MidJourney and DALL-E, it requires more computational resources, making it challenging to run on consumer-grade hardware without optimizations. Open-source fine-tuning has led to specialized models such as Juggernaut XL and RealVisXL, which cater to specific artistic or photorealistic styles.

SDXL Turbo

SDXL Turbo [33] is a lightweight, high-speed variant of SDXL specifically optimized for real-time image synthesis at a resolution of 512×512. Despite having a smaller footprint—approximately 3.5 billion parameters—it preserves the same text-conditioning framework as its predecessor. The core innovation lies in its use of *Adversarial Diffusion Distillation*, a training strategy that enables the model to generate coherent and high-quality images in a single forward pass. This dramatically accelerates the generation process, making SDXL Turbo particularly suited for interactive applications and rapid prototyping. However, it is distributed under a non-commercial license, restricting its use to research and personal experimentation.

Many third-party open-source interfaces exist for Stable Diffusion, each catering to different user needs. The most popular is AUTOMATIC1111 Web UI, which offers an extensive feature set, including inpainting, extensions, and advanced prompt handling, making it the go-to choice for power users. Fooocus, on the other hand, simplifies the prompting process, focusing on an intuitive experience that reduces the need for complex textual inputs, making it more beginner-friendly. ComfyUI takes a different approach by providing a node-based interface, allowing users to create complex image generation workflows akin to procedural 3D modeling software. This modular design makes ComfyUI particularly appealing to researchers and advanced users who wish to experiment with custom pipelines. The variety of these interfaces highlights the flexibility and customization options available for Stable Diffusion users.

3.2 GLIDE

GLIDE, which stands for Guided Language-to-Image Diffusion for Generation and Editing [34], introduces novel techniques to improve text-conditional image synthesis using diffusion models. GLIDE not only focuses on generating images from textual descriptions but also supports targeted inpainting, enabling fine-grained control over image modifications.

GLIDE builds on the base diffusion model described by earlier works, incorporating key improvements to enhance quality and flexibility.

Learnable variance schedule: unlike the original diffusion model presented by Ho et al. (2020) where the variance schedule is fixed, GLIDE adopted the approach proposed by Nichol & Dhariwal (2021)[25], where the variance Σ_θ is a learnable

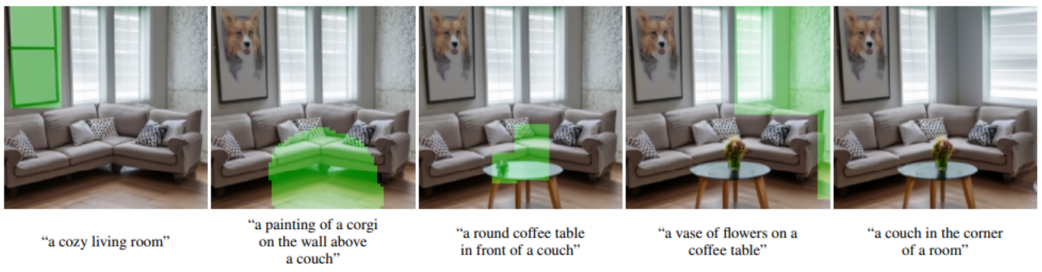

"a cozy living room"
"a painting of a corgi on the wall above a couch"
"a round coffee table in front of a couch"
"a vase of flowers on a coffee table"
"a couch in the corner of a room"

Figure 3.3: Iterative generation of a complex scene with GLIDE.

parameter during training. This was found to allow high-quality images to be generated in fewer denoising steps.

Advanced text conditioning: the model builds upon the Autoregressive Diffusion Model architecture [35], incorporating enhanced mechanisms for conditioning on textual input. At each timestep t, given a noisy image x_t and its associated caption c, the model estimates the distribution $p(x_{t-1} \mid x_t, c)$, guiding the denoising process according to the semantics of the input text. The textual prompt is first embedded using a transformer encoder, yielding a sequence of token representations denoted as K. These embeddings are then linearly projected to align with the dimensionality of the attention layers within the diffusion network, enabling effective integration of linguistic context during image generation.

These embeddings are injected into the ADM architecture through cross-attention mechanisms, where they are concatenated with the key and value components of the model's attention layers at each step of the reverse diffusion process. This design ensures that the generated image aligns closely with the semantics of the text prompt, effectively guiding the denoising trajectory.

The paper compares two primary guidance techniques—Classifier-Free Guidance (CFG) and CLIP Guidance.

Classifier-Free Guidance (CFG): CFG allows the model to learn both conditioned and unconditioned outputs during training, without using an explicit classifier during the diffusion process. This is achieved by randomly replacing the conditioning text c with a null token ($\emptyset$) in a fraction of the training steps (typically 10-20%).

During inference, the conditioned and unconditioned predictions are interpolated as follows:

$$\hat{\epsilon}_\theta(x_t|c) = \epsilon_\theta(x_t|\emptyset) + s \cdot (\epsilon_\theta(x_t|c) - \epsilon_\theta(x_t|\emptyset))$$

where s is the guidance scale. A higher s enforces stricter adherence to the conditioning prompt, enhancing semantic alignment at the cost of reduced diversity. CFG eliminates the need for a separate classifier, simplifying the architecture and reducing computational overhead.

CLIP Guidance: CLIP guidance leverages a noise-aware version of the CLIP model to align the generated image with its textual description. During the denoising process, the model assesses how well each intermediate image x_t corresponds to the input text c by comparing their respective CLIP embeddings. This is done by computing the cosine similarity between the visual representation $f(x_t)$ and the textual embedding $g(c)$, and using the gradient of their inner product to influence the denoising trajectory. The reverse diffusion step is thus modified by adjusting the predicted mean based on this alignment objective, encouraging the model to generate content that more closely matches the semantic intent of the prompt:

$$\hat{\mu}_\theta(x_t|c) = \mu_\theta(x_t|c) + s \cdot \Sigma_\theta(x_t|c)\nabla_{x_t}(f(x_t) \cdot g(c))$$

Regular CLIP models (not explicitly trained to be noise-aware) can still be used for guidance but with limitations, such as susceptibility to adversarial gradients, which can lead to artifacts in generated images.

The experimental setup involved training a 3.5 billion parameter diffusion model conditioned on text inputs at a base resolution of 64×64, along with an additional 1.5 billion parameter model for upsampling to 256×256. Additionally, a noised CLIP model using a ViT-L architecture was trained at 64×64 to provide guidance during generation.

The experiments trained a 3.5 billion parameter text-conditional diffusion model at 64×64 resolution, and another 1.5 billion parameter text-conditional upsampling diffusion model to increase the resolution to 256×256. For CLIP guidance, a noised 64×64 ViT-L CLIP model was trained.

Classifier-free guidance yielded more photorealistic and semantically accurate outputs compared to CLIP-based guidance. It is hypothesized that with CLIP guidance, the diffusion model may generate adversarial samples—images that maximize the CLIP score but fail to align with the actual semantics of the prompt. These adversarial gradients can lead to artifacts or incoherent outputs.

While GLIDE excels in generating photorealistic images, it faces challenges with complex prompts that require a high level of detail and nuance. To address this, GLIDE supports targeted image editing, allowing users to remove or modify specific portions of an image while preserving the surrounding context.

3.3 DALL-E

DALL-E 1[36], introduced in January 2021, marked the first successful attempt at directly generating images from textual prompts. The model combines a two-stage training process with autoregressive modeling to generate high-quality images from textual descriptions. The methodology is structured to balance efficiency and precision, leveraging advanced encoding techniques to map the relationship between text and image data effectively.

In the first stage, a discrete Variational Autoencoder (dVAE) learns to compress high-resolution RGB images ($256 \times 256 \times 3$) into a compact grid representation of 32×32 discrete tokens, each capable of assuming one of 8192 unique values. This compression reduces the spatial resolution of the input while preserving key visual features. The resulting 1024 tokens act as a latent representation of the image. The encoder function (q_ϕ) maps the original image x to the discrete latent space of tokens, while the decoder function (p_θ) reconstructs the image from these tokens during generation.

In the second stage, the input text is tokenized into up to 256 Byte Pair Encoding (BPE) tokens. These tokens are concatenated with the 1024 image tokens produced by the dVAE from the first stage, forming a sequence of 1280 tokens. This joint sequence serves as input to an autoregressive transformer, which is trained to model the joint distribution $p_\psi(y, z)$, where y and z denote the text and image tokens, respectively. The training objective is to predict the next token in the sequence, enabling the model to generate tokens for the image that align with the textual description. This is achieved using a transformer with 12 billion parameters trained on a dataset of 250 million image-text pairs.

The learning objective is formulated to optimize the evidence lower bound (ELBO), which provides a tractable approximation of the joint probability distribution $p_{\theta,\psi}(x, y)$ over an image x and its corresponding caption y. This joint distribution is modeled by introducing a set of latent image tokens z, and can be decomposed as:

$$p_{\theta,\psi}(x, y, z) = p_\theta(x|y, z)p_\psi(y, z)$$

where:

- $p_\theta(x|y,z)$: is the likelihood of reconstructing the image x from the image tokens z and text tokens y, modeled via the dVAE decoder.
- $p_\psi(y,z)$: defines the joint distribution over the text and image tokens, learned via an autoregressive transformer.

The ELBO is expressed as:

$$\ln p_{\theta,\psi}(x,y) \geq \mathbb{E}_{z\sim q_\phi(z|x)}\left[\ln p_\theta(x|y,z) - \beta D_{\mathrm{KL}}(q_\phi(y,z|x)||p_\psi(y,z))\right]$$

where:

- q_ϕ is the posterior distribution over discrete image tokens, as produced by the dVAE encoder.
- p_θ is the distribution over the RGB images generated by the dVAE decoder given the image tokens.
- D_{KL} is the Kullback-Leibler divergence between the approximate posterior and the learned prior
- β is a scaling factor to balance reconstruction fidelity and prior regularization.

During inference, the model first encodes the text prompt into 256 tokens using the BPE tokenizer. Then, it predicts the next 1024 image tokens autoregressively using the transformer. Finally, it decodes the full sequence of 1024 image tokens into an image using the dVAE decoder.

3.3.1 DALL-E 2

Released in April 2022, DALL-E 2 [37] refined the methodology established in its predecessor by incorporating CLIP into a two-stage generative process. Unlike DALL · E 1, which relied on a fully autoregressive approach, DALL · E 2 shifted to a diffusion-based image synthesis pipeline, improving realism and semantic coherence. The model generates images through a two-step process: first predicting a CLIP image embedding conditioned on the text prompt, then generating an image from this embedding. The probability distribution of image generation in DALL · E 2 is formulated as:

$$P(x|y) = P(x|z_i,y)P(z_i|y)$$

where:

- $P(z_i|y)$ predicts the CLIP image embedding z_i conditioned on the textual description y, leveraging CLIP model.
- $P(x|z_i, y)$ produces the final image x from the predicted CLIP image embedding z_i, optionally using the text y as an additional condition.

In the first stage, the model predicts an image embedding z_i from the text y. Two different methods were explored for modeling this prior: Autoregressive (AR) prior, which employs a causal transformer and PCA-based quantization to predict z_i as discrete tokens; and Diffusion prior, which models z_i as a continuous vector using a Gaussian diffusion process trained to predict the unnoised z_i.

The diffusion prior was found to be more effective at capturing the semantics of the text and was ultimately favored in later refinements.

In the second stage, once the image embedding z_i is obtained, the final image is generated through a cascaded diffusion model. The process begins at a low resolution of 64×64 pixels, which is then progressively upsampled to 256×256, and finally to 1024×1024. Each upsampling stage uses a dedicated diffusion model trained to reconstruct finer details from slightly corrupted images. By iteratively refining the image through diffusion-based denoising, DALL · E 2 significantly improves upon the sharpness and alignment of generated images with textual prompts. Compared to its predecessor, DALL · E 2 demonstrated substantial

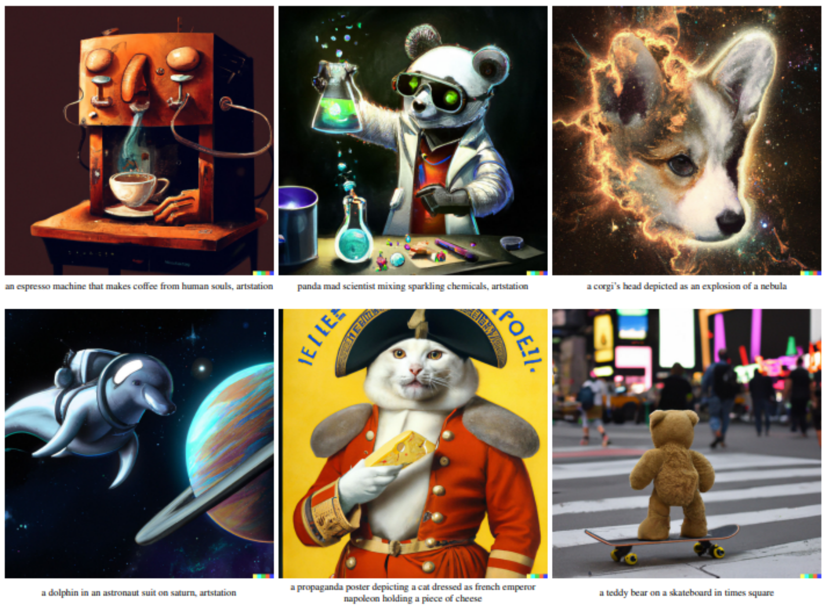

an espresso machine that makes coffee from human souls, artstation
panda mad scientist mixing sparkling chemicals, artstation
a corgi's head depicted as an explosion of a nebula
a dolphin in an astronaut suit on saturn, artstation
a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese
a teddy bear on a skateboard in times square

Figure 3.4: Samples from DALL · E 2

advancements in:

- Text-Image consistency: the integration of CLIP embeddings improved semantic alignment between the generated image and the input text.
- Compositional reasoning: the model better understands and combines multiple visual elements specified in the prompt.
- Style transfer & refinement: CLIP embeddings allow conditioning on artistic styles, enhancing creativity and fine-grained control over image attributes.
- Image quality: the diffusion-based generation method introduced smoother gradients, reducing unnatural artifacts commonly seen in purely autoregressive models.

3.3.2 DALL-E 3

DALL-E 3 [38], launched in 2023, builds on DALL-E 2, addressing the challenge of prompt adherence by introducing dataset recaptioning to improve caption fidelity and image quality. Earlier models, including DALL · E 2, occasionally struggled to interpret complex prompts accurately, sometimes omitting crucial details or misrepresenting the intended concepts. To overcome this, DALL · E 3 introduced a novel recaptioning approach, allowing the model to generate synthetic captions that enriched training data and improved alignment between text descriptions and visual outputs. The training framework of DALL · E 3 incorporated a CLIP-based

JOIN THE GALACTIC ADVENTURE!
Explore new worlds, build a brighter future!

Ancient pages filled with sketches and writings of fantasy beasts, monsters, and plants sprawl across an old, weathered journal. The faded dark green ink tells tales of magical adventures, while the high-resolution drawings detail each creature's intricate characteristics. Sunlight peeks through a nearby window, illuminating the pages and revealing their timeworn charm.

A vibrant 1960s-style poster depicting interplanetary migration, with a retro rocket ship blasting off from earth towards a distant, colorful planet. Bold typography announces "Join the galactic adventure!" with smaller text underneath reading "Explore new worlds, build a brighter future." The background features a swirling galaxy of stars and constellations.

A mischievous ferret with a playful grin squeezes itself into a large glass jar, surrounded by colorful candy. The jar sits on a wooden table in a cozy kitchen, and warm sunlight filters through a nearby window.

Figure 3.5: Samples from DALL · E 3

architecture where a language model $L(t, i)$, augmented by image features, was optimized to generate high-quality synthetic captions:

$$L(t, i) = \sum_j \log P(t_j | t_{j-k}, \ldots, t_{j-1}; F(i); \Theta)$$

where $F(i)$ represents the CLIP image embedding, and Θ denotes the model parameters. This recaptioning approach helped the model better understand nuanced user prompts, reducing instances where generated images failed to capture the full semantic meaning of the input text.

The training process was divided into two key phases:

1. **Short synthetic captions**: focused on the primary subject of an image.
2. **Descriptive synthetic captions**: included detailed surroundings, styles, and object placements.

Beyond improving textual alignment, DALL · E 3 introduced optimizations for image quality. The model was trained on a dataset of 1 billion images, leveraging a T5-conditioned diffusion architecture across 500,000 training steps. The diffusion process was further refined to generate high-resolution images at multiple aspect ratios, including 1024×1024, 1792×1024, and 1024×1792, providing greater flexibility for practical applications.

A major breakthrough in DALL · E 3 is its native integration into ChatGPT, which enables to leverage ChatGPT as a brainstorming partner and refiner for the user's prompts. This integration enables a more interactive and coherent image generation experience, as the language model can suggest prompt modifications to enhance clarity and specificity. The synergy between natural language processing and image synthesis also makes the system more intuitive for non-technical users, broadening its accessibility.

Compared to earlier iterations, DALL · E 3 significantly improved image-text alignment and visual quality, producing more coherent, detailed, and aesthetically pleasing results. The model's ability to interpret complex prompts accurately and generate high-resolution outputs positioned it as a leading advancement in generative AI.

3.4 MidJourney

MidJourney[39], first released in 2022, is a generative AI model that has gained immense popularity for its ability to convert natural language prompts into highly stylized and visually captivating images. Positioned alongside major players like

DALL-E and Stable Diffusion, MidJourney distinguishes itself by focusing on artistic creativity and abstraction rather than strict photorealism. Unlike open-source models, MidJourney operates on closed-source proprietary technology, with its underlying architecture and methodology kept confidential.

MidJourney is accessible through its Discord server, where users interact with the bot by typing prompts in designated channels, and, more recently, also through its new user-friendly web-based platform. Key features and capabilities include:

- **Image variations:** after generating an initial set of images, users can select specific outputs to refine further, exploring nuanced differences in style, color, or composition. This iterative feature allows for a high degree of customization.
- **Style conditioning:** MidJourney excels in applying artistic styles to generated content, and prompts can specify particular artistic movements or predefined aesthetics, enabling users to achieve highly tailored outputs.
- **Image referencing:** users can upload their own images as references, allowing MidJourney to base its generation on the uploaded content, which is particularly useful for maintaining visual consistency.
- **Character referencing:** by feeding character reference images, the model supports prompts that develop consistent characters across multiple generations.
- **Stylization and detail control:** users can specify the level of abstraction or realism in the output.
- **Aspect Ratio Flexibility:** MidJourney supports custom aspect ratios, enabling users to generate images optimized for specific formats.
- **Blend mode:** this unique feature allows users to combine multiple prompts or reference images, resulting in blended outputs that integrate styles, themes, or subjects.
- **Zoom out and pan:** introduced in recent versions, this feature enables users to extend the canvas of an existing image, creating a broader scene while preserving the coherence of the original content.

Since its debut, MidJourney has undergone continuous improvement, with new versions released every few months introducing enhanced features, improved image quality, and better prompt adherence. As of now, the latest released version is 6.1, released in July 2024.

MidJourney also focuses on the community, creating a space for users to actively share their creations, discuss techniques, and exchange tips, making the platform as much a social hub as a creative tool.

3.5 Imagen

Imagen[20], introduced by Google Research in 2022, is a state-of-the-art text-to-image synthesis model that innovatively combines the semantic power of large pre-trained language models with the high-fidelity generation capabilities of diffusion models.

Unlike earlier models that rely on multi-modal embeddings (e.g., CLIP), Imagen leverages a purely text-based large language model -a pre-trained T5-XXL text encoder-, to convert textual prompts into dense semantic embeddings. This choice allows for a deeper and more nuanced understanding of complex, compositional prompts thanks to the LLM's exposure to vast textual corpora during training. At the core of Imagen's design is a diffusion model framework that generates

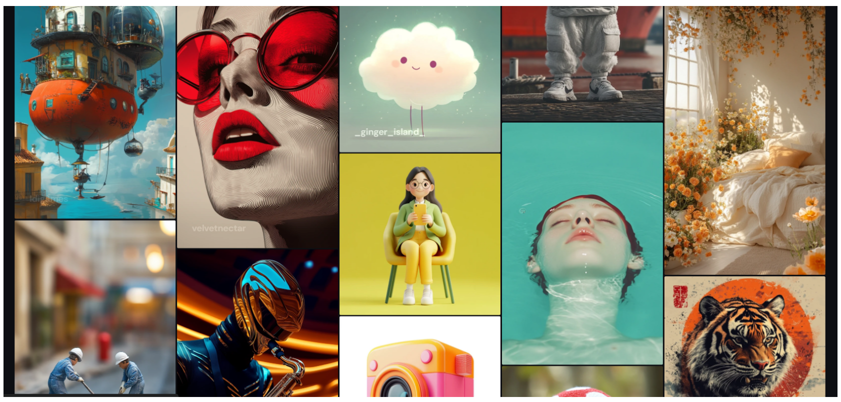


Figure 3.6: Samples from Midjourney

images through an iterative denoising process. The synthesis pipeline begins with a base diffusion model that produces low-resolution images (64×64) from random Gaussian noise. These initial outputs are then refined via a cascaded super-resolution process, where a first upsampling phase increases the resolution to 256×256, and a subsequent phase further refines the image to a high resolution of 1024×1024.

During each step, the diffusion model is trained to minimize the noise prediction error defined by:

$$\mathcal{L} = \mathbb{E}_{x,c,\epsilon,t}\left[w_t \left\| x_\theta\big(\alpha_t x + \sigma_t \epsilon, c\big) - x \right\|_2^2\right], \tag{3.1}$$

where x is the image, c represents the text embedding, α_t and σ_t are time-dependent scaling factors, and ϵ is Gaussian noise.

To enhance image quality and ensure faithful adherence to text prompts, Imagen employs classifier-free guidance (CFG). In this approach, the noise prediction is interpolated between conditional and unconditional predictions:

$$\epsilon_\theta(z_t, c) = w\,\epsilon_\theta(z_t, c) + (1 - w)\,\epsilon_\theta(z_t), \tag{3.2}$$

where the guidance scale w balances prompt adherence against diversity in the generated images.

A critical aspect of the Imagen framework is the management of output pixel ranges, which is achieved through thresholding techniques. Two approaches are considered:

- **Static thresholding:** in this method, the predicted image x_0 is directly clipped to a fixed range (typically $[-1, 1]$). Clipping refers to limiting the pixel intensity values to a predefined range, to prevent extreme pixel values from causing visual artifacts or distortions. Static thresholding is simple and computationally efficient, but it can be overly rigid, potentially discarding fine details and leading to artifacts if many values are clipped.
- **Dynamic Thresholding:** Here, the threshold value is adaptively determined at each sampling step based on the distribution of pixel intensities (e.g., by computing a specific percentile). If the computed threshold s exceeds 1, the pixel values in x_0 are clipped to $[-s, s]$ and normalized by s. This adaptive approach is more flexible, preserving subtle variations while effectively preventing oversaturation, and was found to yield superior photorealism and better text-to-image alignment.

Another key innovation in Imagen is the freezing of the text encoder during training. This strategy allows for precomputed embeddings and reduces computational costs while emphasizing that scaling up the text encoder significantly improves both image quality and text alignment.

Recent advancements in the Imagen series further refine these techniques. Imagen 2, launched in December 2023, introduced enhancements in image fidelity, contextual understanding, and multilingual support. Building on these improvements, Imagen 3, introduced in August 2024, offers even sharper photorealism, simplified prompt interpretation, and stronger safety features. Together, these iterations underscore Imagen's pioneering role in text-to-image generation and its continuous push toward improved creative and technical capabilities.

Chapter 4

Fine Tuning Techniques

4.1 Textual Inversion

Textual Inversion[40] is a fine-tuning method designed to personalize text-to-image diffusion models, such as Stable Diffusion, by introducing new visual concepts without altering the underlying model weights. This technique works by introducing a novel pseudo-word, denoted as S^*, into the text-to-image diffusion model's vocabulary. This pseudo-word corresponds to a specific subject or concept that the user wishes to integrate into the generative model. Instead of retraining the entire model, the method leverages the model's existing text embedding space and injects the new concept by optimizing a unique embedding v^*, for S^*, using only a few example images, typically 3–5, making it efficient and accessible. The result is a mechanism that enables users to include custom subjects in textual prompts. The training process aims to optimize the token embedding v^* so that it effectively binds the desired subject to the pseudo-word within the model's latent space, all while keeping the pre-trained model's weights frozen. The goal is therefore to minimize the reconstruction error between the generated images and the provided samples, enabling the model to map the unique identifier to the desired visual concept effectively. Mathematically, this involves optimizing the embedding v^* to ensure alignment with the model's broader and pre-existing latent space. The process leverages the model's existing understanding of generic concepts (e.g., "chair" or "style"), enabling it to capture the new concept's specificity while maintaining generalization.

The optimization process is grounded in the standard Latent Diffusion Model (LDM) framework. Using text-image pairs as input, the model applies random

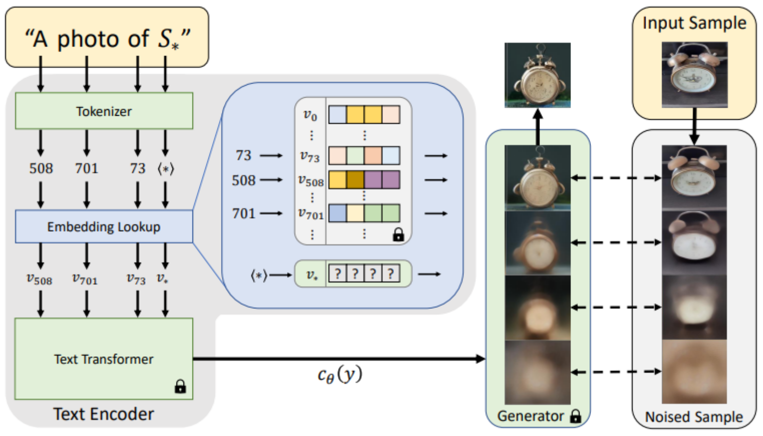


Figure 4.1: Textual Inversion process

noise to the image embeddings and then learns to reconstruct the original image using only textual conditioning. The objective function for this process is as follows:

$$v^* = \arg\min_v \mathbb{E}_{z\sim E(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2\right]$$

where z_t represents the noisy latent representation, t is the time step in the noise schedule, $\epsilon \sim \mathcal{N}(0,1)$ is the noise, and $c_\theta(y)$ is the text conditioner derived from CLIP embeddings. During training, random context texts are sampled to diversify the conditioning and encourage broader generalization of the embedding.

This approach aligns the learned embedding v^* with the model's existing knowledge, ensuring the new concept integrates seamlessly into its vocabulary. By leveraging the existing text embedding space and optimizing only the embeddings for the unique identifier, Textual Inversion avoids the computational overhead of fine-tuning the entire model and maintains the robustness of the pre-trained system. The resulting embeddings allow for creative applications, enabling users to generate images featuring specific objects, styles, or concepts by simply including the corresponding pseudo-word in their textual prompts.

Textual Inversion represents a significant step forward in making diffusion models more personalizable and flexible. However, the limitations mentioned in the original paper primarily regard two areas: precision in capturing shapes and optimization efficiency. While Textual Inversion successfully encodes the "semantic essence" of a concept, it can struggle to reconstruct precise shapes or more complex relational prompts such as spatial arrangements between objects. This trade-off makes the method particularly suitable for artistic applications, where capturing the general feel of a subject is often sufficient. However, it is more evident for tasks requiring exact replication of structural details. Additionally, the process is time-intensive, with the current setup requiring roughly two hours to learn a single concept. The authors suggest that this could be significantly reduced by developing an encoder trained to directly map a set of images to their corresponding textual embedding, which could streamline the process.

4.2 DreamBooth

DreamBooth is a fine-tuning technique designed to personalize image generation models like Stable Diffusion, presented in the paper [41]. By default, these models can generate a wide variety of subjects within a given class, but they struggle to consistently reproduce the same subject in various contexts or accurately mimic an existing subject within the generated image. DreamBooth, developed by Google Research, addresses this limitation by enabling the model to generate highly customized and contextually adaptable images based on a specific subject or object from a limited number of reference images—typically as few as 3 to 5. The method introduces a unique identifier token (e.g., `"[V]"`) into the model's vocabulary, associating it with the subject during training. This token enables the model to recognize and reproduce the subject in various contexts while maintaining its overall coherence and style diversity.

DreamBooth employs a class-specific loss function to ensure that the fine-tuned model retains its general knowledge and does not overfit to the reference images. For example, if fine-tuning a model on images of a specific dog, the training process compares outputs not only against the subject-specific images but also against a broader class of dogs. This dual-focus approach prevents the model from "forgetting" how to generate other dogs while still excelling at generating the specific target dog in diverse scenes and settings.

To guide this process, each training image is annotated with the format `"a [identifier] [class noun]"`: the identifier is a unique token chosen to represent the specific subject—typically a rare or unused token to avoid semantic interference—while the class noun corresponds to the broader category of the subject. In this way, the model is able to anchor the new concept within its existing knowledge space, facilitating accurate and context-aware synthesis of the personalized subject across diverse scenes.

The process of selecting unique identifiers in fine-tuning aims to minimize the

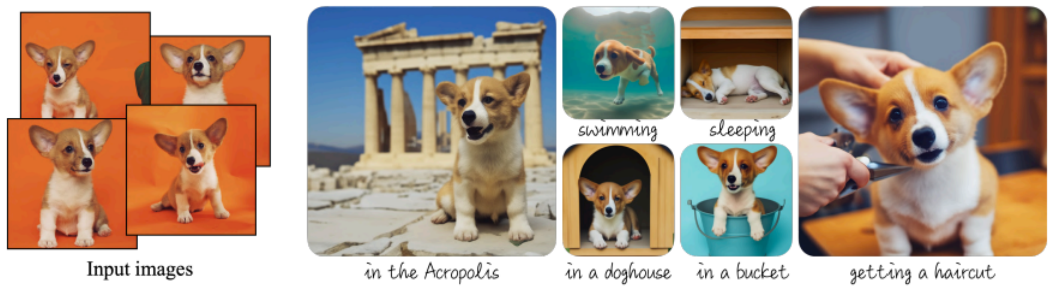


Figure 4.2: Dreambooth fine-tuning example

influence of pre-existing linguistic or model priors. Using common English words is generally discouraged, as these tokens are already semantically loaded, making them less effective for learning a new, specific concept. naive alternative might involve generating random alphanumeric strings,but such sequences are often tokenized into smaller units or individual characters, each of which may still carry their own prior meanings, therefore this kind of identifiers can still introduce noise and ambiguity. The preferred approach is to identify rare tokens in the model's vocabulary that have minimal prior associations. These rare tokens are identified through a lookup in the model's tokenizer, focusing on sequences of 1 to 3 tokens that are relatively infrequent (e.g., tokens with index values in the range of 5000–10000 for the T5-XXL tokenizer, which tend to occur less frequently and thus carry weaker semantic associations). These tokens are then inverted into text space using a detokenizer, generating unique sequences of characters that form the identifier. By selecting tokens with weak priors and avoiding spaces or lengthy sequences, this method ensures that the identifier is both distinct and effectively integrated into the fine-tuning process without unintended biases.

DreamBooth fine-tunes all layers of the model, which allows it to tightly integrate the new subject into the generative process. However, this comprehensive fine-tuning can lead to challenges such as language drift—where the model gradually loses some of its general textual understanding—and reduced output diversity, as the model may overfit to the limited subject images. To mitigate these issues, DreamBooth employs a class-specific prior preservation loss. This loss function ensures that, while the model learns to generate the specific subject, it also retains broader class knowledge by supervising the model on both subject-specific images and class-level samples. The overall loss is defined as:

$$\mathbb{E}_{x,c,\epsilon,\epsilon',t}\left[w_t\|x^\theta(\alpha_t x+\sigma_t\epsilon,c)-x\|_2^2+\lambda w_t'\|x^\theta(\alpha_t' x_{pr}+\sigma_t'\epsilon',c_{pr})-x_{pr}\|_2^2\right]$$

where the first term enforces high fidelity to the target subject and the second

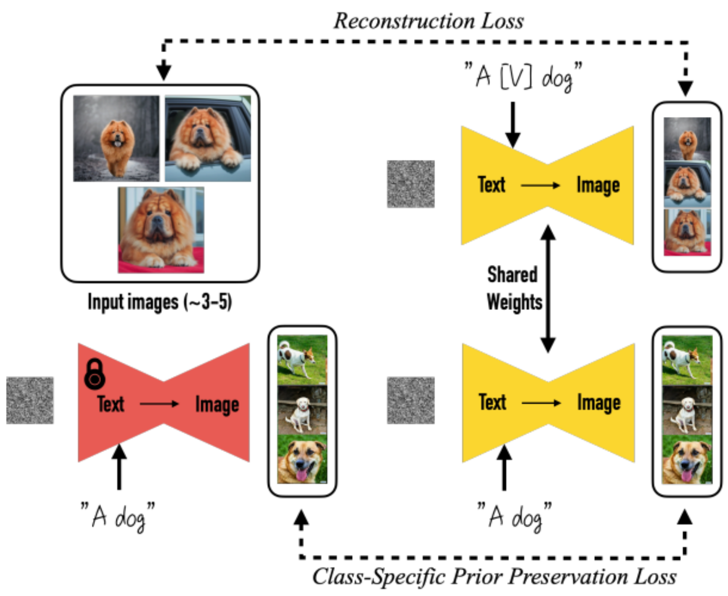


Figure 4.3: Dreambooth fine-tuning process

term, weighted by λ, is the prior-preservation term that encourages diversity and prevents overfitting by retaining knowledge of the original class through generated samples x_{pr}.

DreamBooth has found applications in personalized art generation, character design, and even photorealistic customization of individuals, making it an invaluable tool for creators and designers. Its strength lies in its ability to blend the unique features of a subject with the creative flexibility of Stable Diffusion, opening up new possibilities in custom content creation.

The limitations noted in the DreamBooth approach primarily concern its ability to handle rare or complex subject-context combinations. When such combinations are underrepresented in the training data, the model may generate inaccurate or weak contextual details, resulting in less reliable priors. Additionally, the model can sometimes entangle the subject's appearance with the contextual prompts, leading to unintended alterations—such as shifts in color or style—that compromise the subject's identity. Overfitting is another significant challenge, especially when the prompts closely mirror the subject's original training context, which limits the diversity of the generated outputs. While DreamBooth performs effectively for common subjects like dogs and cats, rarer subjects often yield fewer variations and lower fidelity, with instances of hallucinated features further degrading subject accuracy. These issues highlight the need for improved training data diversity.

In comparison, Textual Inversion optimizes only the embedding vector for a new pseudo-word while keeping the rest of the model frozen. This lighter-weight approach reduces the risk of overfitting and language drift, but it may not capture as much detail or achieve the same level of subject and prompt fidelity as DreamBooth. Quantitative evaluations and user studies indicate that DreamBooth generally provides superior performance, making it the preferred method when high-fidelity subject reproduction and contextual adaptability are critical.

4.3 LoRA

LoRA (Low-Rank Adaptation)[42] is a technique that enables efficient fine-tuning of large pre-trained models by introducing small, trainable parameter matrices into specific layers, rather than updating the entire weight matrices. In traditional approaches for adapting large language models and diffusion models, full fine-tuning is often employed to develop downstream applications. However, as models scale to hundreds of billions of parameters, such fine-tuning becomes increasingly resource- and time-intensive.

To address these challenges, alternative methods have been proposed—such as limiting the number of trainable parameters or employing external networks (known as hypernetworks[43]) to adapt inputs toward desired outputs. Although these techniques reduce computational demands, they often involve a trade-off between efficiency and quality, rarely matching the performance achieved by full fine-tuning.

LoRA leverages the insight that the changes required during model adaptation tend to have a low intrinsic rank. This means that the essential modifications can be captured by a small number of parameters. Instead of updating full weight matrices, LoRA optimizes low-rank decomposition matrices that encapsulate the necessary weight changes. By doing so, it drastically reduces the number of trainable parameters while maintaining high adaptation quality.

Instead of updating the full weight matrix $W_0 \in \mathbb{R}^{d\times k}$ of a pre-trained model during fine-tuning, LoRA injects two low-rank matrices A and B such that:

$$W = W_0 + \Delta W, \quad \text{where} \quad \Delta W = B \times A$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, and the rank $r \ll \min(d, k)$. This decomposition significantly reduces the number of trainable parameters, as the rank r is much smaller than the dimensions of the original weight matrix d.

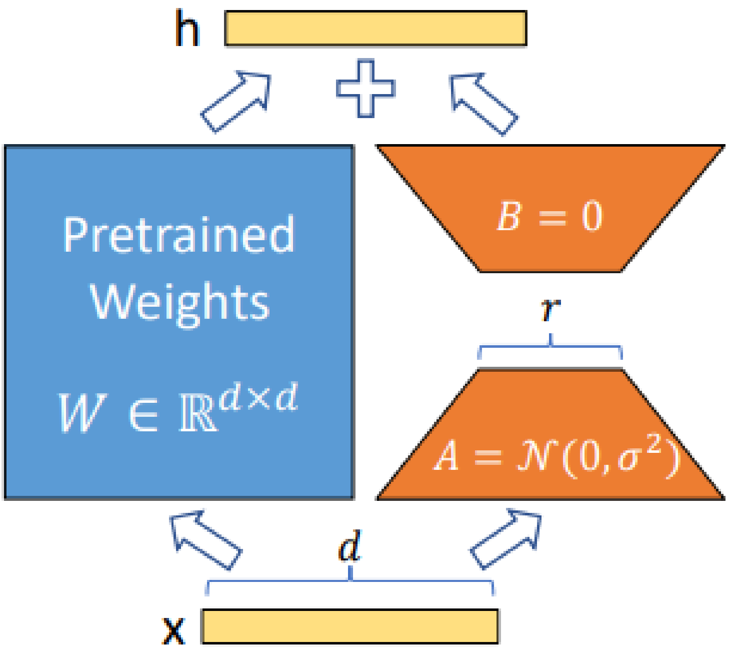

h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 4.4: LoRa reparametrization.

During fine-tuning, only A and B are optimized, while the original pre-trained weights W_0 remain frozen. The modified weights W effectively incorporate the adaptation required for the specific task, while the low-rank nature ensures that the updates are computationally lightweight.

As a result, LoRA significantly reduces the computational and hardware demands of the fine-tuning process, making it more accessible and energy-efficient. Moreover, the fact that the original weight matrix W_0 remains untouched during training allows for seamless reversion to the base model if needed. At the same time, learned low-rank matrices can be easily interchanged or composed across different tasks, enabling a modular and versatile adaptation strategy without the need to retrain the entire model. An additional advantage is that the method generates compact files containing only the low-rank updates, which can be easily distributed and integrated into any compatible pre-trained model—minimizing both storage needs and deployment complexity.

As a practical example of LoRA benefits, the paper describes the results obtained on GPT-3 175B. LoRA reduces VRAM usage during training from 1.2TB to 350GB, allowing training on fewer GPUs and avoiding I/O bottlenecks. Moreover, the storage requirements for task-specific checkpoints are dramatically lowered—from 350GB down to just 35MB, representing a 10,000$\times$ reduction. This compact representation allows for rapid task switching by simply loading the lightweight LoRA weights over the pre-trained model. Additionally, LoRA accelerates training by 25% compared to full fine-tuning, as gradients are computed only for the small subset of trainable parameters. This efficiency makes LoRA both scalable and cost-effective when adapting extremely large models.

In the transformer architecture, low-rank adaptation is usually applied to the weight matrices in the self-attention modules, such as W_q, W_k, W_v, and W_o, enabling efficient fine-tuning across various tasks. More specifically, the paper examines which weight matrices benefit most from LoRA under a fixed parameter budget (e.g., 18M trainable parameters in GPT-3 175B). Experimental results reveal that applying LoRA to all self-attention weight matrices simultaneously yields high performance only when a very low rank (e.g., $r = 2$) is used to stay within the budget. In contrast, selectively adapting the query (W_q) and value (W_v) matrices strikes the best balance between performance and efficiency, achieving superior results with a slightly higher rank ($r = 4$). Adapting only W_q or W_k in isolation, however, leads to significantly lower performance, indicating that these weights alone are insufficient to capture the required information for task-specific adaptations while the joint adaptation of W_q and W_v is more effective at capturing the relevant task-specific transformations under constrained conditions.

The optimal rank for LoRA has also been a subject of investigation. The analysis shows that even extremely low ranks, such as $r = 1$, can achieve competitive performance when applied to both W_q and W_v. This suggests that the weight update matrix ΔW inherently has a low-dimensional structure, meaning that the most crucial adaptations for the downstream task reside in a small subspace. Further experiments involving subspace similarity reveal that the dominant singular vector directions of adaptation matrices (for example, when comparing $r = 8$ to $r = 64$) overlap substantially, while the remaining directions contribute little or merely capture noise. This confirms that low-rank adaptation matrices efficiently encode the critical adjustments required for effective fine-tuning, ensuring high performance with minimal additional computational overhead.

4.4 Custom Diffusion

Custom Diffusion[44] is a fine-tuning method built on Stable Diffusion, designed to adapt pre-trained text-to-image diffusion models to incorporate new user-specified concepts efficiently. Similarly to LoRA, Custom Diffusion updates only a limited set of parameters, thus retaining the pre-trained model's broad generative capabilities while drastically reducing computational and memory overhead. Moreover, it specifically focuses on compositional fine-tuning, the ability to extend beyond a single concept and compose multiple concepts together.

Custom Diffusion selectively fine-tunes the key (W_k) and value (W_v) projection matrices in the cross-attention layers of the U-Net architecture used in Stable Diffusion. This design choice minimizes computational and memory requirements while proving sufficient to encode new concepts. By restricting training to these parameters (approximately 3% of the total model weights), the method avoids the overhead of full-model fine-tuning while achieving high-quality results.

Each target concept is associated with text captions. When a descriptive caption exists, it is used directly. In personalized scenarios—such as generating a specific individual or pet that belongs to a broader category—a unique identifier token V^* is introduced. This token is initialized with a rarely occurring embedding from the model's vocabulary—minimizing interference from existing concepts—and is optimized alongside the cross-attention parameters. This joint optimization ensures that the fine-tuned model can generate the target concept accurately across various contexts. STarting from a few images of the new concept, the method

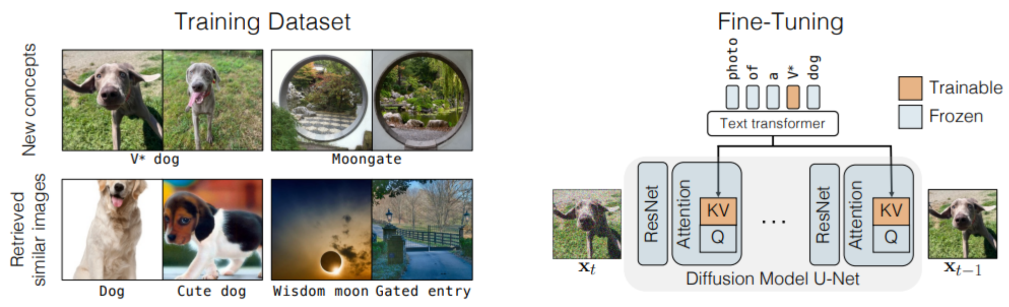


Figure 4.5: Custom diffusion process: training dataset and fine-tuning.

first retrieves additional real images with from a large dataset (e.g., LAION-400M) with captions similar to the target concept (measured via CLIP feature similarity). These retrieved images are used to create a regularization dataset that helps prevent overfitting and preserves the model's prior knowledge of related concepts during fine-tuning. In addition, data augmentation is applied by resizing target images and appending modifiers such as "zoomed in" or "far away" to the text prompts. This augmentation improves generalization and variation in the generated outputs.

Custom Diffusion is also designed to handle multiple concepts simultaneously. In multi-concept fine-tuning, training datasets for each concept are combined and each concept is assigned its own unique modifier token V_i^*. These tokens and the selected cross-attention parameters are optimized jointly during training. Alternatively, separately fine-tuned models for individual concepts can be merged via a closed-form constrained optimization:

$$W^* = W_0 + v^\top d, \quad d = C(C_{\text{reg}}^\top C_{\text{reg}})^{-1}, \quad v^\top = (V - W_0 C^\top)(dC^\top)^{-1}$$

where W_0 represents the pre-trained model weights, C denotes the combined text features of all target concepts, and C_{reg} represents regularization features. This approach avoids retraining while ensuring compatibility of multiple concepts.

While Textual Inversion typically adds a new token (or optimizes a new token embedding) while keeping the rest of the model unchanged. It focuses solely on learning the embedding that represents the new concept. In contrast, Custom Diffusion not only optimizes a new token but also fine-tunes part of the model—the cross-attention parameters—to better integrate that concept with the image-generation process. DreamBooth fine-tunes the entire diffusion model (or a large subset of its parameters) to adapt to a specific concept, which can be very resource-intensive and sometimes lead to overfitting. Custom Diffusion, on the other hand, restricts fine-tuning to only the cross-attention layers and the new token to achieve greater efficiency.

4.5 ControlNet

Despite their general capabilities, large text-to-image models often fail to precisely capture the specific user's intent or the nuanced idea behind the prompt. However, enhancing such models to specialize in task-specific generation is challenging, given their scale—often trained on datasets comprising over 5 billion data points—and the computational demands of retraining or fine-tuning them. ControlNet[45] aims to tackle this problem by introducing spatial conditioning controls to large, pretrained architectures such as Stable Diffusion, allowing users to provide specific image-based conditioning inputs—like edge maps, depth maps, segmentation masks, and human pose skeletons.

Instead of fine-tuning the original model directly, ControlNet creates a "trainable copy" of the pretrained model: gradients are updated exclusively on the trainable copy, while the original weights are retained in a frozen state. This approach allows

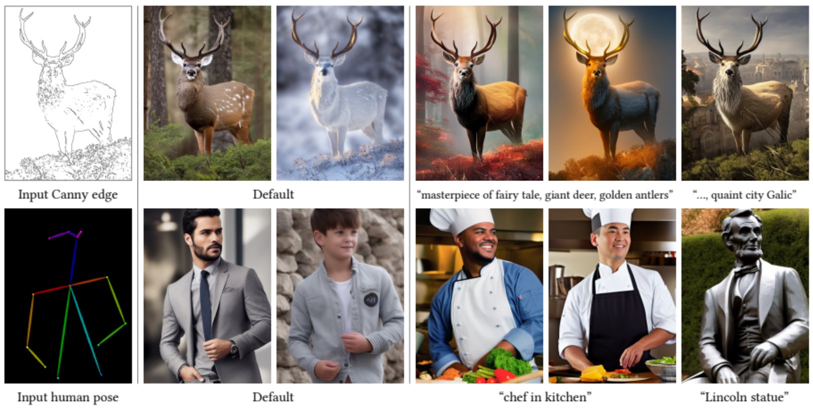


Figure 4.6: ControlNet: controlling Stable Diffusion with learned conditions.

leveraging the generalization capabilities of the original while enabling targeted learning on the cloned copy for task-specific requirements.

The core architecture of ControlNet leverages the structure of pretrained text-to-image diffusion models while introducing a trainable mechanism for spatial control. This is achieved by freezing the parameters of the pretrained model and adding a parallel, trainable copy of its encoding layers, which are connected via zero convolution layers. These zero-initialized convolution layers ensure that no noise is introduced into the pretrained model during the initial stages of training, preserving its capabilities while enabling the gradual incorporation of new conditional inputs. For each conditioning input (e.g., depth map, pose, or edge map), the architecture

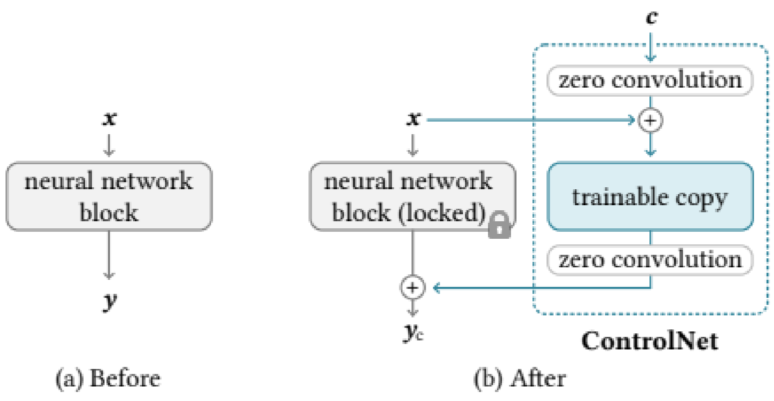


Figure 4.7: ControlNet architecure scheme

introduces a lightweight encoder to preprocess the input into a latent representation compatible with the Stable Diffusion latent space. The resulting conditioning vector is integrated into the trainable branch of ControlNet, influencing the output through the carefully designed connections between the frozen and trainable components.

Within the Stable Diffusion framework, ControlNet is implemented by augmenting the U-Net architecture. The U-Net consists of an encoder, a middle block, and a skip-connected decoder, with 25 primary blocks spanning four spatial resolutions: (64×64, 32×32, 16×16, and 8×8). ControlNet introduces parallel, trainable versions of the encoder and middle block, which are linked to the original network through zero convolution layers and process spatial conditioning inputs to steer the generation process. Since the original encoder remains frozen during training, this setup minimizes computational overhead by avoiding gradient updates on the locked parameters, allowing for efficient fine-tuning.

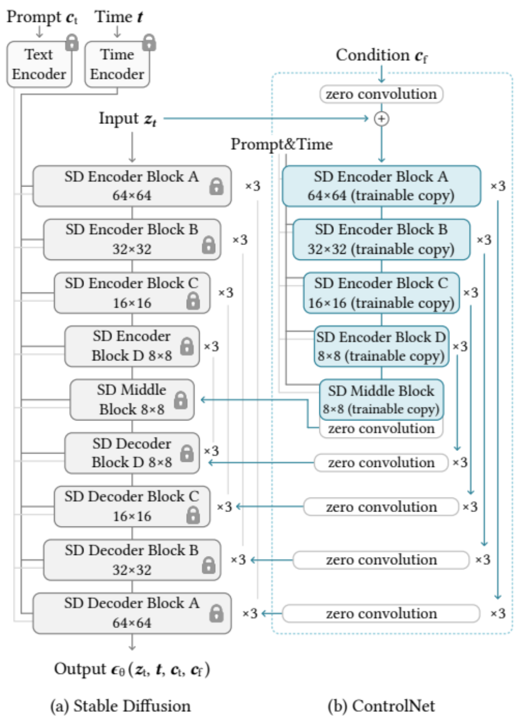


Figure 4.8: ControlNet integrated into Stable Diffusion's U-Net at the encoder and middle blocks.

To ensure compatibility, ControlNet preprocesses the conditioning images into latent representations matching the U-Net's input resolution. This preprocessing is achieved using a lightweight encoder with a sequence of convolutional layers that encode the spatial conditions into compact representations.

Stable Diffusion uses latent images, converting 512×512 pixel-space images into a 64×64 latent space. ControlNet integrates condition-specific input c_i into this latent space using an encoder network $E(\cdot)$:

$$c_f = E(c_i),$$

where c_f is the feature-space conditioning vector.

The encoder $E(\cdot)$ consists of four convolution layers with kernel size 4×4 and stride 2×2, activated by ReLU, and progressively increasing channel dimensions (16, 32, 64, 128). This ensures that input conditions are downscaled and aligned with the latent space dimensions of Stable Diffusion.

ControlNet training involves the same objective as traditional diffusion models, where the model learns to predict noise progressively added to an image during the forward diffusion process. The key difference lies in the inclusion of additional spatial conditions during training. The loss function is:

$$L = \mathbb{E}_{z_0,t,c_t,c_f,\epsilon\sim\mathcal{N}(0,1)} \left[\|\epsilon - \epsilon_\theta(z_t, t, c_t, c_f)\|^2 \right]$$

where z_t represents the noisy latent at timestep t, c_t the text condition, c_f the spatial condition, and ϵ_θ the noise predictor.

To improve robustness, the training process involves replacing 50% of the text prompts with empty strings. This compels the model to depend more strongly on spatial conditioning signals, ensuring reliable performance even when textual input is absent at inference time.

ControlNet also employs Classifier-Free Guidance Resolution Weighting (CFG-RW). As we have seen, CFG in Stable Diffusion balances between unconditional ϵ_{uc} and conditional ϵ_c outputs, controlled by a weight β_{cfg}. When ControlNet adds conditioning images, directly applying them to both outputs can overly weaken or strengthen guidance. To resolve this, CFG-RW scales the influence of ControlNet's conditioning based on block resolution by a connection's weight (w_i) that is inversely proportional to the block size (h_i):

$$w_i \propto \frac{1}{h_i}$$

This adjusts guidance strength dynamically across model layers, ensuring adherence to spatial inputs without over-dominating the text-based conditions.

ControlNet also supports compositional conditioning, allowing multiple ControlNets (e.g., for depth and pose) to be applied simultaneously in the generation process by simply summing the outputs of the respective ControlNets without additional weighting or interpolation.

This modularity allows users to specify multiple spatial conditions, effectively guiding the generative process while preserving high fidelity to the provided constraints. The ability to integrate multiple spatial conditions makes ControlNet particularly useful for applications requiring high levels of control, such as pose-guided character generation, architectural visualization, or depth-aware image synthesis.

Chapter 5

Storyboarding

5.1 Storyboarding and Shot Types

Storyboarding is the process of visually planning a narrative by creating a sequence of drawings or images that represent each shot or key moment in a scene. Each panel of a storyboard typically includes sketches of the scene's composition, camera angles, character positions, and actions, often accompanied by notes on dialogue, movement, lighting, or sound.

Storyboarding is extensively utilized in industries such as film, animation, advertising, and multimedia to plan and pre-visualize how the narrative will unfold before shooting begins, serving as a blueprint for technical execution.

Shot types in particular are a key element in storyboarding, determining how subjects and objects are framed within a scene. By varying camera placement and screen size, different effects can be achieved—guiding the viewer's focus and perception. This makes shot types a critical tool for shaping narratives and evoking specific responses in visual storytelling.

Our work focuses on 8 common shot types based on the field size, defined as follows:

- **Extreme Close Up:** The human figure is framed from the chin up.
- **Close Up:** The human figure is framed from above the shoulder up.
- **Medium Close Up:** The human figure is framed from half the torso up.
- **Medium Shot:** The human figure is framed from the waist up.
- **American Shot:** The human figure is framed from above the knee up.

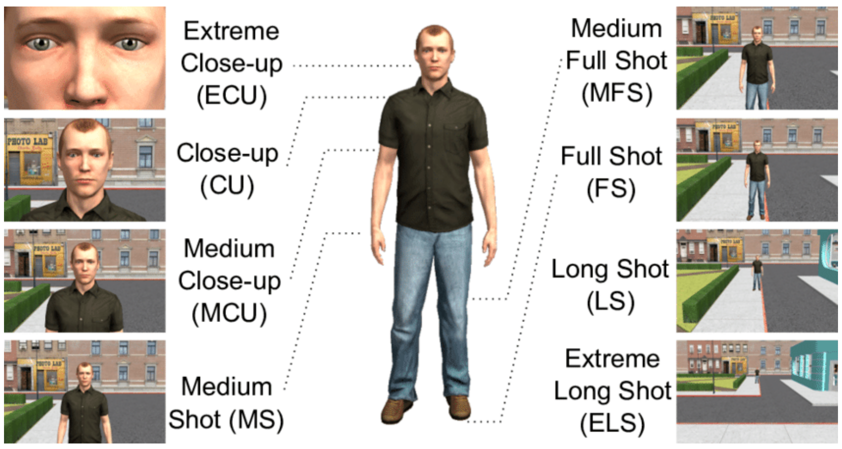


Figure 5.1: Eight shot types

- **Full Shot:**The human figure is framed entirely (or almost entirely) and occupies more than $\frac{2}{3}$ of the frame in height.
- **Long Shot** The human figure is framed entirely and occupies between $\frac{2}{3}$ and $\frac{1}{3}$ of the frame in height.
- **Extreme Long Shot** The human figure occupies less than one third of the frame in height, or is absent.

5.2 Storyboarding Works

In recent years, an increasing number of studies have focused on leveraging generative AI for story visualization and continuation. The following subsections provide an overview of key methodologies and systems in this domain.

5.2.1 StoryGAN

The StoryGAN paper [46] introduces the task of story visualization, which consists in generating a sequence of images from a multi-sentence narrative, with each image corresponding to a sentence in the story. The key challenge is to ensure that each image faithfully represents its sentence while maintaining global consistency across the full sequence—preserving continuity in characters, scenes, and narrative flow.

StoryGAN addresses this with a sequential conditional GAN architecture composed of a generator and two discriminators. The generator includes three main components: the Story Encoder, the Context Encoder, and the Image Generator.

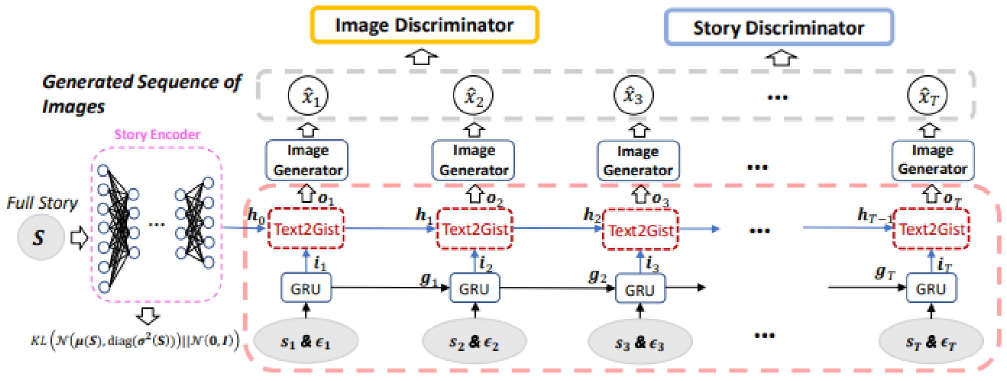


Figure 5.2: The StoryGan framework.

Given a full story $S = [s_1, s_2, ..., s_T]$, the Story Encoder compresses it into a global context vector $h_0 \in \mathbb{R}^d$. This vector is sampled from a learned Gaussian distribution $h_0 \sim \mathcal{N}(\mu(S), \Sigma(S))$ and is used to initialize the hidden state of the Context Encoder. It captures the semantic summary of the entire story and introduces variation during generation.

The Context Encoder is a two-layer RNN that dynamically tracks the progression of the story: at each timestep t, the model receives a sentence embedding s_t and random noise ϵ_t, which are combined and processed to produce a local feature vector i_t. This vector is then merged i_t with the evolving context hidden state h_{t-1}, producing an updated context h_t and an output vector o_t. o_t encodes both local (sentence-level) and global (story-level) information, thus preserving the broader narrative's continuity while capturing the information relevant to the current moment in the story.

The Image Generator uses this vector o_t to produce an image $\hat{x}_t$ that visually represents the corresponding sentence s_t, grounded in the overall story context.

To ensure the quality and consistency of these images, StoryGAN utilizes two discriminators:

- **Image discriminator**: this module evaluates whether each generated image aeach image $\hat{x}_t$ is realistic and semantically aligned with its sentence s_t and the global initial story context h_0. It compares a triplet of inputs—the sentence, the encoded story context, and the generated image—and assigns a "real" or "fake" label based on how well the generated image matches the sentence description and the broader context.

- **Story discriminator**: this component assesses whether the sequence of generated images $[\hat{x}_1, ..., \hat{x}_T]$ is coherent with the full story S. It evaluates the coherence of the entire sequence, ensuring that the images collectively represent a unified and logical narrative. The Story Discriminator encodes both the image sequence and the story text into separate fixed-length vectors, combines them via element-wise product, and scores their alignment using a fully connected layer with sigmoid activation.

The generator and discriminators are trained adversarially, with the generator aiming to fool both discriminators. A KL divergence regularization term is also applied to the Story Encoder to encourage smoothness in the latent space.

The feedback from these discriminators is backpropagated through the generator network, allowing it to refine its parameters via gradient descent. This iterative training process enables the generator to produce images that are not only more aligned with the sentence descriptions at the image level but also more coherent across the sequence at the story level. As training progresses, the model improves its ability to maintain consistency in characters, settings, and actions across frames, making StoryGAN a foundational approach in the field of story visualization.

5.2.2 AutoStory

AutoStory[47] combines Large Language Models (LLMs) and text-to-image diffusion models to transform user input into coherent visual narratives.

The process begins with converting the input (either a detailed story or a short description) into a structured, multi-panel narrative using an LLM such as GPT-4. The LLM segments the story into panel-wise descriptions, then generates a global prompt for each panel and localized object prompts paired with sparse spatial layouts in the form of bounding boxes. These form the backbone of the scene layout and help decouple complex storytelling into simpler compositional tasks.

These sparse bounding box layouts are further refined into dense control conditions, such as sketches or keypoints, to accurately extract object boundaries. Techniques like segmentation and edge detection are used to enhance visual precision and ensure that important details are preserved. These object-level control signals are composed into a full-scene dense condition, aligned with the original bounding box layout in order to achieve spatial coherence while allowing detailed control during generation.

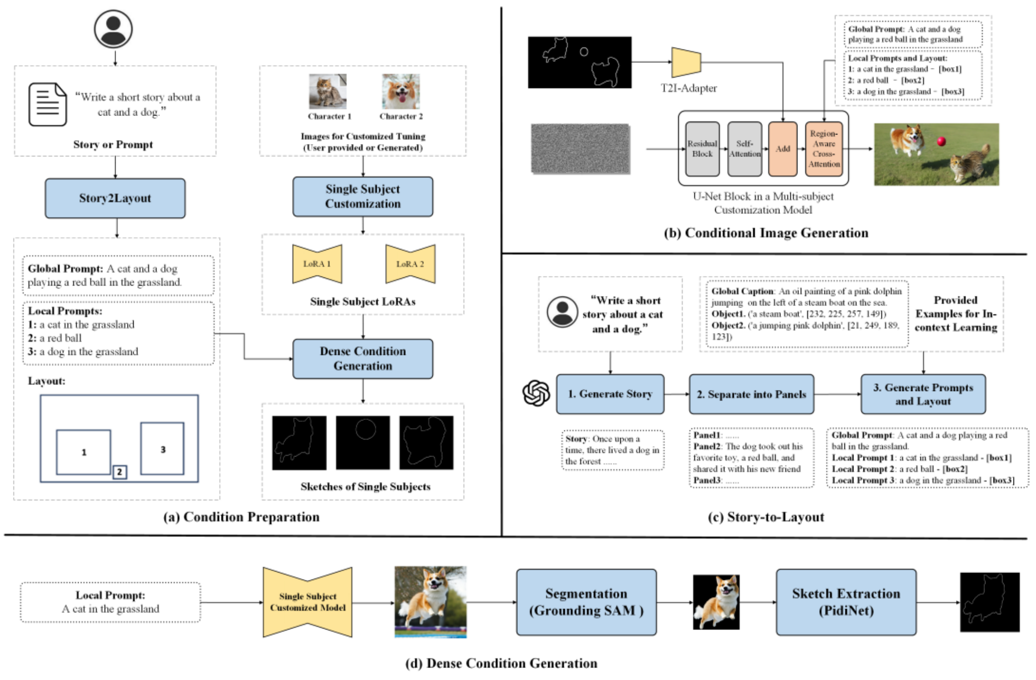

"Write a short story about a cat and a dog."
Story or Prompt
Story2Layout
Global Prompt: A cat and a dog playing a red ball in the grassland.
Local Prompts:
1: a cat in the grassland
2: a red ball
3: a dog in the grassland
Layout:
1
2
3
Character 1
Character 2
Images for Customized Tuning (User provided or Generated)
Single Subject Customization
LoRA 1
LoRA 2
Single Subject LoRAs
Dense Condition Generation
Sketches of Single Subjects
(a) Condition Preparation
T2I-Adapter
Global Prompt: A cat and a dog playing a red ball in the grassland
Local Prompts and Layout:
1: a cat in the grassland - [box1]
2: a red ball - [box2]
3: a dog in the grassland - [box3]
Residual Block
Self-Attention
Add
Region-Aware Cross-Attention
U-Net Block in a Multi-subject Customization Model
(b) Conditional Image Generation
"Write a short story about a cat and a dog."
Global Caption: An oil painting of a pink dolphin jumping on the left of a steam boat on the sea.
Object1. ('a steam boat', [232, 225, 257, 149])
Object2. ('a jumping pink dolphin', [21, 249, 189, 123])
Provided Examples for In-context Learning
1. Generate Story
2. Separate into Panels
3. Generate Prompts and Layout
Story: Once upon a time, there lived a dog in the forest ......
Panel1: ......
Panel2: The dog took out his favorite toy, a red ball, and shared it with his new friend
Panel3: ......
Global Prompt: A cat and a dog playing a red ball in the grassland.
Local Prompt 1: a cat in the grassland - [box1]
Local Prompt 2: a red ball - [box2]
Local Prompt 3: a dog in the grassland - [box3]
(c) Story-to-Layout
Local Prompt: A cat in the grassland
Single Subject Customized Model
Segmentation (Grounding SAM)
Sketch Extraction (PidiNet)
(d) Dense Condition Generation

Figure 5.3: The Autostory pipeline.

For identity preservation of characters, AutoStory employs Embedding-Decomposed LoRA (ED-LoRA) for lightweight fine-tuning, ensuring that character appearances remain consistent across frames. Given only a few reference images, this method learns identity-specific adaptations without full model retraining. Gradient fusion is then applied to merge multiple ED-LoRAs for multi-character scenes. In cases where no reference images are provided, AutoStory supports automatic character generation using a training-free method. It treats multi-view generation as a video modeling problem by applying extended self-attention across frames: each frame's latent features attend to those in the first and previous frames, enforcing identity consistency. To enhance variation, 3D-aware image translation methods (e.g., One-2-3-45) are used to generate diverse character poses and viewpoints.

Once the multi-view character images are synthesized, sketches or keypoints are extracted to serve as detailed structural control signals. These are injected into the latent space of the diffusion model to further refine image quality and ensure alignment with the intended character design.

In the image generation stage, AutoStory uses Stable Diffusion enhanced by T2I-Adapter to incorporate both sparse layout information and dense control signals.

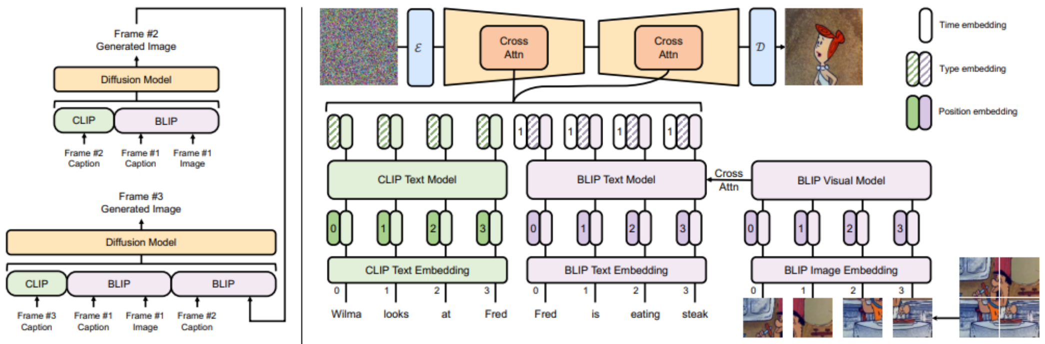
The generation process:

$$I_i = \mathrm{DM}(p_i^{\mathrm{global}};\ \sigma_i, \{A,\ C_i\},\ \Delta W)$$

where σ_i denotes the layout, C_i the dense condition, A the T2I-Adapter, and ΔW the customized LoRA weights for character identity preservation. The layout controls the spatial positioning of objects, while the dense conditions guide structural and stylistic fidelity.

5.2.3 AR-LDM Methodology

The AR-LDM methodology[48] addresses the challenge of maintaining frame-to-frame consistency in visual story synthesis, a critical factor for generating coherent narratives. Unlike single-caption text-to-image tasks—where each image is generated independently—story synthesis requires the model to incorporate contextual information from preceding frames. Captions are interdependent, and the model must ensure continuity in both visual and narrative elements across the sequence. To achieve this, the paper introduces a history-aware latent diffusion approach that incorporates both past captions and previously generated images into the generation process, ensuring visual and narrative coherence throughout a multi-frame sequence. The method generates each frame conditioned not only



(a) Auto-regression process. (b) Model Architecture.

Figure 5.4: The AR-LDM process and architecture.

on its current caption but also on a structured multimodal history of preceding caption-image pairs. This conditioning is achieved via a dedicated history-aware encoding module. The module uses CLIP to encode the current sentence and BLIP to jointly encode each prior frame along with its associated caption. These representations are then fused—along with type and position embeddings—into a comprehensive multimodal condition vector φ_j, which guides the reverse diffusion process for the current latent $z_0^{[j]}$.

The generation process proceeds sequentially: for each frame j, the model estimates the posterior $p(x_j \mid x_{<j}, c_{\leq j})$ and samples the image latent z_0 in the compressed space. This latent is then decoded using a pretrained decoder D to produce the final image $\hat{x}_j$. This strategy allows AR-LDM to ground each image in both the current narrative input and the evolving visual context of the story, resulting in coherent characters, spatial relationships, and scene progression across frames.

To support generalization to unseen characters—especially those referenced vaguely or via pronouns—AR-LDM introduces a special placeholder token $\langle char \rangle$. This token is initialized using an embedding of a semantically similar word (e.g., "man" or "woman") and is fine-tuned using 3–5 reference images of the new character. During adaptation, the full AR-LDM model is fine-tuned (except for the encoder and decoder), allowing it to learn character-specific visual traits without retraining from scratch.

By integrating history-aware conditioning and auto-regressive generation, AR-LDM provides a robust solution for synthesizing multi-frame visual stories while preserving character identity, spatial coherence, and contextual alignment across all frames.

5.2.4 StoryDALL-E

StoryDALL-E [49] adapts the DALL-E text-to-image model for the task of story continuation, where the goal is to generate a coherent sequence of images given a series of captions and an initial (source) image. Unlike standard text-to-image models that treat each image independently, StoryDALL-E ensures visual and narrative consistency across frames by conditioning each image on both the caption and a shared context, as well as copying relevant elements from a previously seen source frame.

To achieve this, the model introduces two main architectural components: a Global Story Encoder, which utilizes self-attention to process all story captions in parallel, creating a global story embedding that encodes overarching context. To maintain sequence structure, sinusoidal positional embeddings are added, encoding the frame's position within the narrative. Additionally, retro-fitted cross-attention layers are added to each self-attention block of the DALL-E transformer, which allow the model to integrate visual cues from the source frame during the generation of target frames.

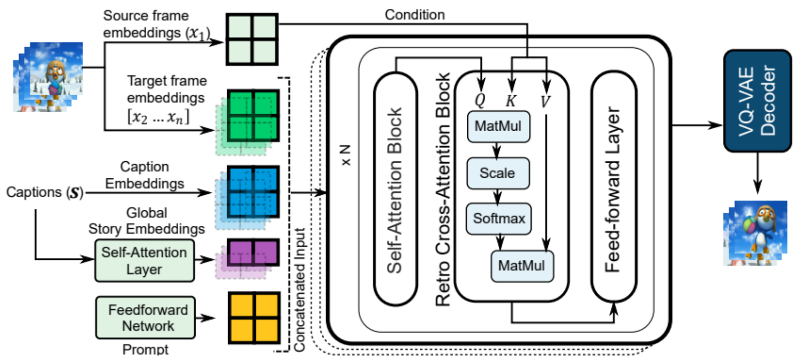


Figure 5.5: Illustration of the StoryDALL-E architecture for the prompt-tuning setting

To optimize efficiency, StoryDALL-E supports both full model fine-tuning and a parameter-efficient prompt-tuning strategy. In the latter, the pretrained model weights remain frozen, and only task-specific prompt embeddings (i.e., learnable virtual tokens), the global story encoder, and cross-attention modules are trained. These learned prompt embeddings are prepended to the caption embeddings and processed alongside global story embeddings and source frame embeddings to guide generation.

The generation pipeline processes these concatenated inputs—caption embeddings, global story embeddings, source frame embeddings, and prompt embeddings—through the modified DALL-E transformer. Within this transformer, self-attention layers textual components (caption + prompt + story embeddings), while cross-attention layers integrate source frame embeddings, allowing the model to attend to the source frame. The model then predicts image tokens for the current frame, modeling the joint distribution of text and image tokens. The predicted tokens are decoded into the RGB image using a pretrained VQVAE decoder.

Through this combination of pretrained knowledge, global context modeling, and source-frame conditioning, StoryDALL-E significantly improves visual consistency and semantic alignment in generated image sequences.

5.2.5 StoryDiffusion

StoryDiffusion[50] is a methodology designed to convert textual narratives into detailed and visually coherent storyboards through a structured process that combines text segmentation, parameterized prompt generation, image synthesis, and user-driven refinement, with a specific focus on supporting user experience (UX) design workflows.

The process begins with GPT-4 analyzing a narrative provided by the user—ranging from abstract ideas to detailed descriptions—and extracting key elements such as characters, actions, objects, emotional tone, setting, and artistic style. The narrative is then segmented into a fixed number of frames (typically six), each corresponding to a panel in the storyboard. For each frame, GPT-4 generates structured prompts composed of multiple parameters that control specific visual aspects, including general description, objects, characters, actions, emotions, background, style, and shot type.

To enhance consistency and clarity across the generated sequence, StoryDiffusion employs a prompt parameterization strategy, ensuring element preservation across frames and logical progression throughout the story. GPT-4 uses task-specific system prompts and intermediate steps to refine each generated prompt, aligning them more closely with the intended story elements before submission to Stable Diffusion.

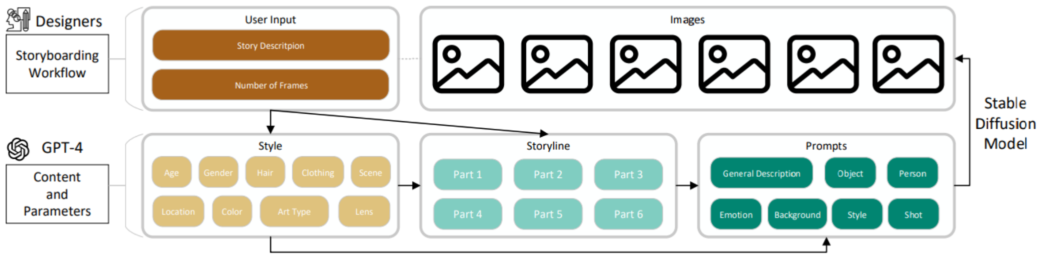


Figure 5.6: StoryDiffusion framework: parameters and co-creation pipeline

The final prompts are fed into Stable Diffusion, which produces the storyboard images. Users can then engage in interactive refinement, editing the narrative, individual prompts, or visual styles directly through a natural language interface. The system supports frame-by-frame editing and regeneration, enabling designers to adjust visuals incrementally without restarting the entire process, making it a flexible tool for dynamic storytelling.

StoryDiffusion's process is tailored for design workflows, incorporating features like parameter visibility toggling, prompt editing in natural or structured formats, and integration with tools like Figma for post-processing. A key insight from user studies is that the system supports both AI-directed and user-directed creative strategies—allowing designers to either lead the creation process or iterate based on AI-generated suggestions. This flexibility makes StoryDiffusion effective for both concept ideation and concept illustration tasks, enabling fast generation of design ideas as well as refined visual documentation of user journeys.

5.2.6 StoryGPT-V

StoryGPT-V[51] is a two-stage framework for generating multi-frame visual stories from narrative text while maintaining character consistency, background coherence, and accurate reference resolution. It combines the generative capabilities of Latent Diffusion Models (LDMs) with the contextual reasoning power of Large Language Models (LLMs), enabling coherent visual storytelling over extended sequences.

The first stage fine-tunes a Stable Diffusion model into a Character-Aware LDM by integrating visual features of characters directly into the text conditioning process. Using CLIP, text tokens corresponding to character names are fused with their image features via an MLP to create augmented embeddings. To further enhance accuracy, the model introduces cross-attention control, where segmentation masks (generated by SAM) guide the attention maps to focus on the correct spatial regions for each character. This supervision ensures that each character token influences the appropriate area of the image during generation. While this stage improves image quality and faithfulness, it is limited to generating images from isolated captions without inter-frame continuity.

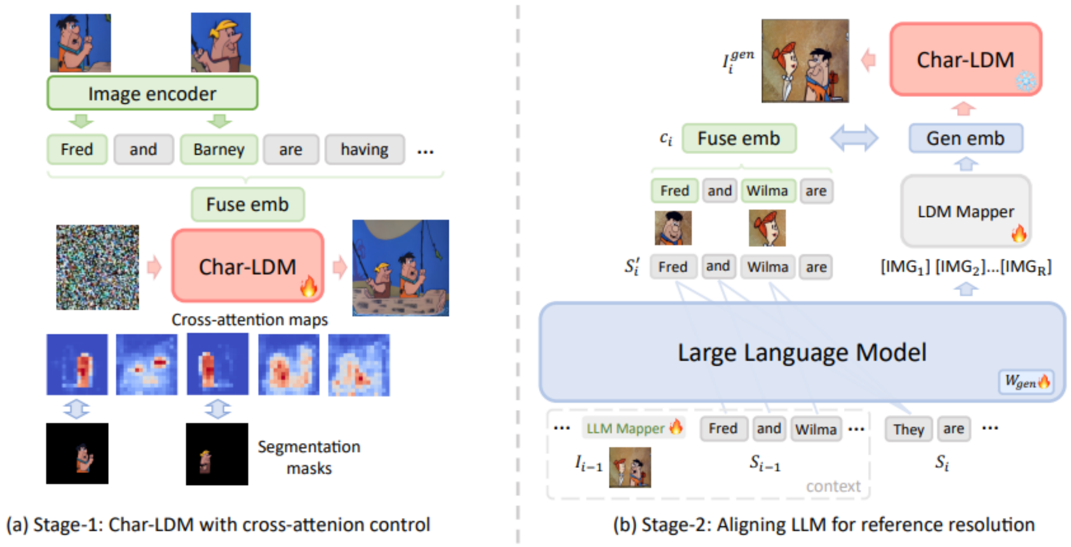


Figure 5.7: StoryGPT-V Framework

To ensure temporal consistency and resolve co-referential ambiguity (e.g., pronouns like "he" or "they"), StoryGPT-V introduces a second stage where a pretrained LLM (e.g., OPT or LLaMA2) processes interleaved sequences of text and visual embeddings from earlier frames. Using causal language modeling, the LLM tracks narrative flow and resolves ambiguous references by reasoning over the context. For instance, if a later caption says "they entered the room," the LLM uses the preceding visual-textual history to determine whether "they" refers to specific characters previously introduced.

As the story unfolds, the LLM predicts special placeholder tokens, denoted as [IMG], which indicate where a new frame should be generated. These tokens are not images themselves, but abstract representations of the next visual content to be rendered, serving as placeholders for upcoming frames. To convert these placeholders into image-guiding inputs, StoryGPT-V uses a Transformer-based LDM Mapper.

The LDM Mapper is trained to translate the [IMG] tokens into fused embedding vectors that align with the conditioning format used by the Character-Aware Latent Diffusion Model (Char-LDM) in Stage 1. These fused embeddings encapsulate the necessary character, scene, and context information for the next frame. By mapping LLM outputs into the visual generation space, the system seamlessly integrates reasoning and diffusion-based synthesis, enabling coherent multi-frame visual storytelling.

StoryGPT-V's structured two-stage approach enhances the quality and consistency of AI-generated visual stories, enabling more immersive and coherent storytelling.

5.2.7 DreamStory

DreamStory [52] is a training-free framework for open-domain story visualization that transforms a textual narrative into a series of visually consistent and semantically aligned story scenes. The system is built upon two primary components: (1) a Large Language Model (LLM), which acts as a *Story Director* and (2) a Multi-Subject Consistent Diffusion Model (MSD), which ensures subjects and scenes remain visually coherent across frames. DreamStory begins with an LLM (e.g., GPT-4 or Yi) performing story comprehension. It parses the input narrative to extract detailed descriptions of key subjects (e.g., characters or objects) and scenes. Through Chain-of-Thought (CoT) prompting, the LLM generates structured prompts and performs necessary rewrites—for example, replacing ambiguous names like "Kondo" with descriptive phrases such as "a towering gorilla." This enhances compatibility with text-to-image diffusion models, which may struggle

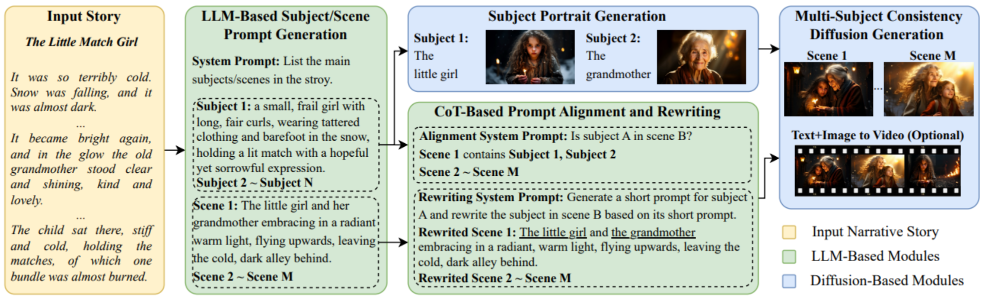

Input Story
The Little Match Girl
It was so terribly cold. Snow was falling, and it was almost dark.
...
It became bright again, and in the glow the old grandmother stood clear and shining, kind and lovely.
...
The child sat there, stiff and cold, holding the matches, of which one bundle was almost burned.
LLM-Based Subject/Scene Prompt Generation
System Prompt: List the main subjects/scenes in the stroy.
Subject 1: a small, frail girl with long, fair curls, wearing tattered clothing and barefoot in the snow, holding a lit match with a hopeful yet sorrowful expression.
Subject 2 ~ Subject N
Scene 1: The little girl and her grandmother embracing in a radiant warm light, flying upwards, leaving the cold, dark alley behind.
Scene 2 ~ Scene M
Subject Portrait Generation
Subject 1: The little girl
Subject 2: The grandmother
CoT-Based Prompt Alignment and Rewriting
Alignment System Prompt: Is subject A in scene B?
Scene 1 contains Subject 1, Subject 2
Scene 2 ~ Scene M
Rewriting System Prompt: Generate a short prompt for subject A and rewrite the subject in scene B based on its short prompt.
Rewrited Scene 1: The little girl and the grandmother embracing in a radiant, warm light, flying upwards, leaving the cold, dark alley behind.
Rewrited Scene 2 ~ Scene M
Multi-Subject Consistency Diffusion Generation
Scene 1
Scene M
Text+Image to Video (Optional)
Input Narrative Story
LLM-Based Modules
Diffusion-Based Modules

Figure 5.8: DreamStory Framework.

with novel or unfamiliar names.

Next, the system uses a pretrained text-to-image diffusion model to generate subject portraits from the rewritten prompts. These portraits are paired with their corresponding text and serve as multimodal anchors, capturing the appearance and semantic attributes of each subject. These anchors provide both visual and textual grounding for consistent subject rendering in the following scene generation step. To maintain character fidelity across scenes, DreamStory introduces a Multi-

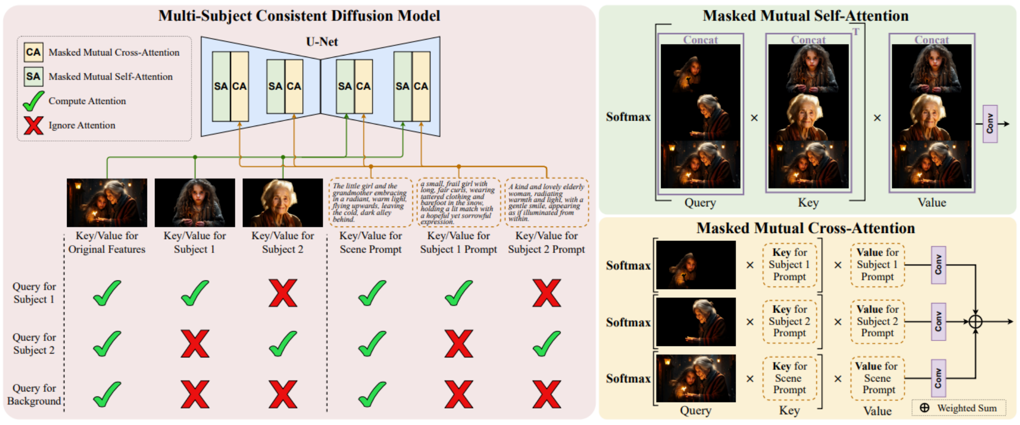

Multi-Subject Consistent Diffusion Model
CA Masked Mutual Cross-Attention
SA Masked Mutual Self-Attention
Compute Attention
Ignore Attention
U-Net
SA CA
The little girl and the grandmother embracing in a radiant, warm light, flying upwards, leaving the cold, dark alley behind.
a small, frail girl with long, fair curls, wearing tattered clothing and barefoot in the snow, holding a lit match with a hopeful yet sorrowful expression.
A kind and lovely elderly woman, radiating warmth and light, with a gentle smile, appearing as if illuminated from within.
Key/Value for Original Features
Key/Value for Subject 1
Key/Value for Subject 2
Key/Value for Scene Prompt
Key/Value for Subject 1 Prompt
Key/Value for Subject 2 Prompt
Query for Subject 1
Query for Subject 2
Query for Background
Masked Mutual Self-Attention
Concat
Softmax
Query
Key
Value
Conv
Masked Mutual Cross-Attention
Key for Subject 1 Prompt
Value for Subject 1 Prompt
Key for Subject 2 Prompt
Value for Subject 2 Prompt
Key for Scene Prompt
Value for Scene Prompt
Weighted Sum

Figure 5.9: Multi-Subject Consistent Diffusion Model in DreamStory

Subject Consistent Diffusion Model (MSD) module that includes two key attention mechanisms:

- **Masked Mutual Self-Attention (MMSA)**: enforces appearance consistency by allowing each subject in the target image to query only the corresponding subject in the reference images. A subject-specific attention mask ensures that visual features are not mistakenly shared across unrelated subjects.

- **Masked Mutual Cross-Attention (MMCA)**: injects subject-specific semantic attributes by allowing each subject to selectively attend to its own reference text. This ensures that rich details (e.g., clothing, posture) are accurately and independently transferred from the prompt to the final image.

These mechanisms are applied at all decoding layers of the diffusion model and are guided by segmentation masks. The masks are generated using GroundingSAM and refined during the denoising process via a fusion of self-attention and cross-attention maps. This ensures precise subject boundaries and avoids unwanted blending between multiple characters.

As a result, DreamStory generates a series of narrative-aligned, visually coherent images, ensuring a seamless and immersive storytelling experience.

5.2.8 Online storyboarding tools

Several online storyboarding tools leverage AI-driven image generation and provide functionalities for editing, customization, and character consistency. Notable platforms include Storyboarder.ai[53], Boords[54], and Katalist.ai[55], each offering unique features tailored to streamlining the storyboarding process.

These platforms enable users to generate AI-assisted visuals while providing tools to refine and adapt images according to their needs. Beyond simple image generation, some of them also address character consistency, ensuring that the same characters maintain a cohesive appearance throughout different frames and projects.

For instance, Boords and Katalist.ai offer a predefined, pre-made set of characters to choose from, but also provide additional tools, allowing users to create and store custom AI-generated characters that can be reused across multiple storyboarding projects. This feature ensures that characters remain visually recognizable and consistent, improving continuity in long-form storytelling.

By integrating AI-based generation with manual customization, these platforms bridge the gap between automation and creative control, making storyboarding more efficient, flexible, and accessible to artists, filmmakers, and content creators.

The field of storyboarding with generative AI is rapidly evolving. Each methodology—whether based on GANs, diffusion models, or hybrid LLM-Diffusion approaches—addresses the dual challenges of visual quality and narrative coherence. As research progresses, improvements in model architectures, training techniques, methodologies and computational efficiency are expected to further push the boundaries of automated story visualization.

5.3 Concept bleeding, omission, and confusion in multi-LoRA image generation

Fine-tuning diffusion-based text-to-image models like Stable Diffusion 1.5 using LoRA (Low-Rank Adaptation) enables the efficient addition of new visual concepts, such as specific characters, without full retraining. However, generating multi-character scenes by loading multiple LoRAs simultaneously introduces challenges, most notably: concept bleeding, concept omission, and concept confusion. These issues compromise visual fidelity, as models fail to maintain clear separations between distinct concepts.

5.3.1 Overview of the phenomena

Concept bleeding occurs when features from one concept undesirably transfer to another. For example, if LoRA A encodes a character with blue hair and LoRA B encodes a character with glasses, both characters might end up with glasses or blue hair.This phenomenon is formally known as attribute leakage, where one concept's learned visual attributes apply to another concept, and typically stems from overlapping feature representations and attention conflicts within the model.

Concept omission happens when one or more intended concepts are entirely missing from the output. A typical case involves only one character being rendered in a scene where two LoRAs are prompted. This may occur because one concept dominates the image generation process, and the omitted concept is essentially overshadowed or "forgotten" by the model during generation.

Concept confusion refers to misassigned attributes or identities— meaning the model might merge features into a hybrid, swap their attributes, or duplicate one concept in place of the other. The output might technically contain both subjects, but not in the correct form or relationship that the prompt intended. For example, if LoRA A is meant to produce character Alice and LoRA B produces character Bob in a scene, a confused result could be that the image shows two copies of Alice (with no Bob), or a single person who has a mix of Alice's and Bob's features, or Bob wearing Alice's clothing. his phenomenon is closely related to (and often co-occurs with) concept bleeding, additionally covering cases of mistaken identity or duplicates.

5.3.2 Underlying causes

Linear composition overlapping updates: In Stable Diffusion, multiple LoRAs are applied by linearly adding their weight deltas to the base model at inference time. While simple, this method doesn't account for the nonlinear dynamics of the diffusion process. As a result, concepts that work well in isolation may interfere when combined. For example, if Concept A modifies early U-Net layers (affecting composition or layout) and Concept B alters later layers ((affecting textures or details), the dominance of A can suppress B. Additionally, when LoRAs are trained on similarly structured datasets (e.g., centered characters, similar lighting), they tend to update overlapping channels of the model—especially in cross-attention—leading to blending and identity merging.

Nonlinear interference: due to the nonlinear nature of diffusion, combining LoRAs can produce emergent effects. Due to the model's nonlinear denoising dynamics, applying LoRA A and B together can result in an unintended third output—neither A nor B, but a hybrid concept (C) Each LoRA shifts the predicted noise differently, and the joint result might converge toward a visual compromise that does not cleanly express either concept, but rather a blended or diluted version that misrepresents both.

Imbalanced influence (weight dominance): if one LoRA has a higher weight multiplier or encodes more visually dominant features (e.g., high contrast, bright colors), it can overshadow the other. Because inference runs through a single pipeline, the stronger LoRA often steers the generation process, resulting in partial or full omission of the weaker concept. This imbalance is frequently observed when adjusting weights—raising one often causes the other to disappear.

Feature space entanglement: LoRA-injected concepts are embedded in the same latent space as the base model, which lacks mechanisms for isolating them. If two characters share similar styles, proportions, or training captions, their learned representations are likely to occupy overlapping dimensions in the model's latent space. In these cases, the denoiser may interpret both concepts as variants of the same visual entity, resulting in visual traits bleeding between concepts.

Shared latent space during diffusion: the denoising process operates within a shared latent image that evolves over time. All concept features—regardless of which LoRA they originate from—are fused into a single latent vector field that the denoiser refines iteratively. There is no native partitioning between "concept A's latent" and "concept B's latent," which would allow isolated control over their evolution. Instead, all information coexists and interacts at every step.This interaction causes the features of both concepts to influence one another.

Entangled attention maps: cross-attention layers in the U-Net map text tokens to spatial image features. When two concept tokens attend to overlapping regions (e.g., both heavily influence the central area of the image where the characters are positioned), the attention maps blend their visual attributions, producing hybrid or duplicated characters. Self-attention compounds this across the image, letting different parts of the image influence each other. This is especially common when LoRAs are trained on similarly composed images, and without explicit spatial guidance, the model often fails to distinguish between them.

Attention map competition: even when attention maps do not overlap spatially, one concept may dominate the attention maps if its features are more distinctive or its LoRA exerts stronger influence. This can occur regardless of equal weight settings, simply due to one LoRA having more salient or well-trained features, causing the model to focus on one concept while underrepresenting or omitting the other.

Lack of disentanglement in training: Stable Diffusion was trained to interpret scenes holistically rather than decomposing them into separable entities. Therefore, fine-tuned LoRAs inherit this limitation, embedding new concepts into preexisting subspaces without enforcing independence, which increases the risk of feature blending or overwriting.

Implicit inter-concept relationships: the model often assumes relationships between co-present subjects based on patterns from pretraining. This can override prompts and produce stereotyped interactions like hugging or kissing, even when instructed otherwise. The entanglement here is not purely representational, but also semantic and structural: the model is implicitly deciding how these two concepts "belong together" in the image, often at the cost of clarity or separability.

Prompt ambiguity: when a prompt lacks clear structure or fails to assign attributes explicitly to each subject, the model struggles to distinguish between concepts. For example, "Alice (a young woman with blue hair) stands next to Bob (a tall man with glasses)" provides more clarity than the vague "Alice and Bob stand together." Without differentiating descriptors or spatial cues, the model often merges or misassigns features. Instead, using structured phrasing (e.g., "on the left/right," "the first/second person," or separate sentences) helps direct attention more accurately. Also using identical phrases for both concepts—such as describing both characters with "a red shirt"—can lead to attribute confusion, as the model may incorrectly assign shared features to both characters, duplicate one of them, or omit the attribute from one concept due to ambiguity in token-to-feature mapping.Therefore, while prompt design alone cannot fully prevent interference, it plays an essential role in guiding the model's attention to distinguish concepts.

Trigger token overlap: many character LoRAs are trained with tags that include generic terms like “woman”, “man” or “1girl.” When multiple LoRAs share these tags, the model may interpret them as redundant instructions, often resulting in visually similar outputs or duplicated characters instead of distinct identities.

Embedding similarity between tokens: even with different trigger words, similar concepts can occupy nearby positions in CLIP’s embedding space. This proximity causes the model to treat tokens as related or interchangeable. For instance, two LoRAs trained on young female characters may both cluster near “woman” in CLIP space, leading to feature mixing, loss of distinction, or generation of an averaged, generic identity.

5.3.3 Mitigation strategies

To address concept interference phenomena in multi-character image generation, a range of mitigation strategies has emerged. These strategies target different stages of the text-to-image generation pipeline—from prompt design to latent denoising—aiming to improve the model’s ability to render distinct, coherent, and identity-consistent subjects within the same image. Below, we outline key general approaches used to reduce these challenges.

Prompt structuring. Carefully constructed prompts play a foundational role in guiding the attention mechanism. Describing each subject in a distinct clause with unique attributes—such as “on the left,” “with blue hair,” or “wearing glasses”—helps the model allocate separate visual space and attribute sets to each concept. Furthermore, assigning unique and unambiguous trigger tokens to each concept (e.g., character-specific tags) helps avoid embedding-level semantic overlap in the text encoder, particularly within CLIP, which is known to cluster similar embeddings closely. Negative prompts can complement a well structured prompt and serve as a soft constraint against undesirable outcomes. FOr example, by including terms like “fused face,” “hybrid creature,” or “extra limbs” in the negative prompt, the model is nudged away from common failure modes typically associated with concept confusion or bleeding. While not a precise control mechanism, this can reduce the likelihood of generating mixed or malformed subjects.

LoRA weight balancing. Fine-tuning the strength (multiplier) of each LoRA is important to ensure balanced representation. Excessive weight for one concept can dominate the denoising trajectory, suppressing the appearance of the other. Empirically, moderate multipliers (e.g., 0.6–0.8) often strike a better balance, even though, in some cases, uneven influence may still occur due to internal feature dominance, so careful experimentation remains necessary.

Joint representation learning. When separate LoRAs exhibit persistent interference during co-inference, an alternative is to train a unified module that learns both concepts simultaneously. By conditioning on joint occurrences (e.g., co-tagged training images), the model learns the relationship between the concepts in a consistent way. Although this reduces modularity and reusability of individual concepts, it can improve compositional fidelity by embedding multi-concept priors directly into the learned representation.

Cross-Attention guidance and separation mechanisms. Several approaches modify the cross-attention mechanism during inference to prevent token interference. Techniques include spatial masking, region-aware token injection, and contrastive conditioning, all of which aim to allocate unique attention maps or regions to each concept. Disentangling the attention paths aims to mitigate visual overlap and help preserve identity separation across subjects.

Spatial conditioning and layout-aware generation. Assigning spatial constraints—such as bounding boxes or masks—to each subject can dramatically reduce feature interference. By enforcing region-specific attention or latent updates, the model is guided to render each concept in its intended zone. This approach reduces the risk of concept overlap, attribute leakage, and positional ambiguity, particularly in scenes involving close proximity or interaction.

Latent space isolation during denoising. Another class of strategies focuses on isolating the denoising process of each concept. By generating latent features for each subject independently—either through segmented passes or controlled conditioning—and then compositing them, models can avoid entanglement in both feature and attention space, maintaining clean visual boundaries and reducing mutual influence between subject representations.

Modular and sequential concept injection. Instead of applying all LoRA modules simultaneously, staggered or scoped injection techniques introduce each concept with limited or sequential influence. For instance, selectively applying one concept's weights to specific spatial regions, time steps, or attention heads helps to preserve modularity while avoiding global interference, enabling more controlled multi-concept synthesis without needing retraining.

Below, we present a selection of recent frameworks that implement these types of strategies through various architectural and inference-time modifications.

5.4 Multi-Subject Consistency Frameworks

5.4.1 Mix of Show

The Mix-of-Show method[56] is a comprehensive approach to address the challenges of multi-concept customization in text-to-image diffusion models.These models can produce high-quality text-conditioned images and, through low-rank adaptations, they can easily learn new concepts, yet they face limitations when trying to integrate multiple concept LoRAs into a single image. The main challenges are concept conflict, where different concepts interfere with one another resulting in overlaps or unintended modifications, and identity loss, where the unique characteristics of each concept are diluted or overwritten, leading to less coherent outputs. To

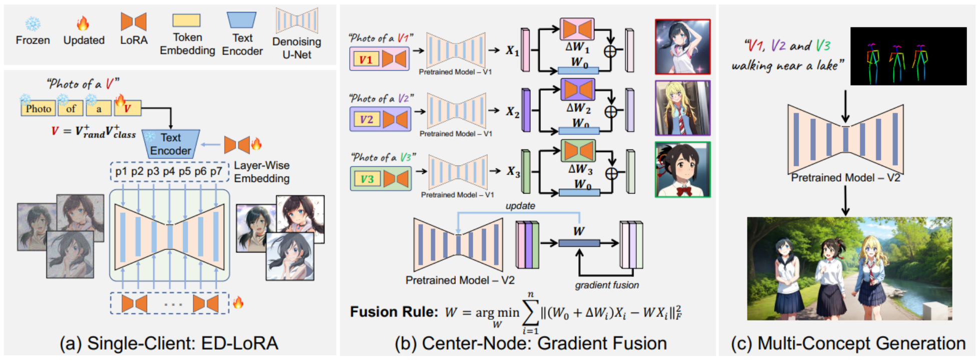


Figure 5.10: Pipeline of Mix-of-Show

address these challenges, Mix-of-Show introduces two key innovations. First is the use of Embedding-Decomposed LoRA (ED-LoRA) for single-client concept tuning. Traditional LoRA embeddings concentrate concept identity into shared weights, making it difficult to preserve each concept independently during fusion. ED-LoRA instead decomposes concept embeddings into layer-wise, learnable vectors. This allows embeddings to encode in-domain characteristics, while LoRA weights capture fine-grained or out-of-domain features. This structure offloads fine-grained and out-of-domain attributes to LoRA weights, minimizing conflicts when merging multiple concepts later. The concept token v is decomposed across layers as:

$$v = [v_1, v_2, ..., v_L], \quad v_\ell \in \mathbb{R}^d, \text{ for } \ell = 1, ..., L,$$

where L is the number of layers and v_ℓ is injected into the ℓ-th layer of the diffusion U-Net.

Secondly, Mix-of-Show introduces Gradient Fusion, a method for combining multiple concepts at the central node. Traditional fusion methods simply average the weights of different LoRA models, which often causes characters or objects to lose their unique features. In contrast, Gradient Fusion aligns the behavior of individual concepts by optimizing input and output features from their respective LoRA layers. This means that Gradient Fusion looks at how each LoRA model behaves — specifically, how it transforms input features into output features during generation- and then adjusts the shared model so that it can reproduce the behavior of all individual LoRAs.

$$\min_{w} \sum_{i=1}^{N} \left\| f_i^{(l)}(x; w) - f_i^{(l)}(x; w_i) \right\|^2,$$

where: - w_i is the weight of LoRA for concept i, - $f_i^{(l)}(x; w_i)$ is the feature produced at layer l using weight w_i, - w is the unified (fused) weight to be optimized.

This approach ensures that the fused model retains the distinct characteristics of each concept while maintaining consistency in the output. Gradient Fusion is particularly effective in reducing identity loss, allowing the model to handle multiple customized concepts without significant degradation in quality.

An additional essential feature of Mix-of-Show is regionally controllable sampling, which addresses common issues in direct multi-concept generation, such as missing objects or attributes being incorrectly assigned to the wrong concept. This feature enables users to define global and regional prompts associated with bounding boxes, mapping each concept to a specific spatial zone in the image. Through a region-aware cross-attention mechanism, the model aligns each subject's visual and textual representations with its assigned spatial location in the image.

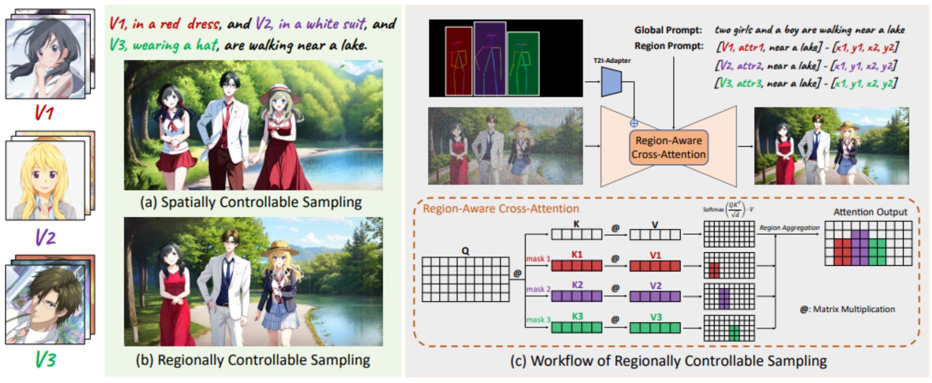

V1
V2
V3
V1, in a red dress, and V2, in a white suit, and V3, wearing a hat, are walking near a lake.
(a) Spatially Controllable Sampling
(b) Regionally Controllable Sampling
Global Prompt: two girls and a boy are walking near a lake
Region Prompt: [V1, attr1, near a lake] - [x1, y1, x2, y2]
[V2, attr2, near a lake] - [x1, y1, x2, y2]
[V3, attr3, near a lake] - [x1, y1, x2, y2]
T2I-Adapter
Region-Aware Cross-Attention
Region-Aware Cross-Attention
Q
K
V
mask 1
K1
V1
mask 2
K2
V2
mask 3
K3
V3
@
Softmax(QK^T/√d)·V
Region Aggregation
Attention Output
@: Matrix Multiplication
(c) Workflow of Regionally Controllable Sampling

Figure 5.11: Multi-concept image generation with regionally controllable sampling.

Given a global prompt P^g and n regional prompts P_r^*, the model first incorporates the global prompt via cross-attention with the latent feature z:

$$h = \text{softmax}\left(\frac{Q(z)K(P^g)^\top}{\sqrt{d}}\right)V(P^g)$$

Next, for each region i, a binary mask $M_i \in \{0,1\}^{H\times W}$ is used to extract the masked latent representation:

$$z_i = z \odot M_i$$

A region-specific attention is then performed using the regional prompt $P^*_{r_i}$:

$$h_i = \text{softmax}\left(\frac{Q(z_i)K(P^*_{r_i})^\top}{\sqrt{d}}\right)V(P^*_{r_i})$$

Finally, the model integrates the regionally refined features by substituting the corresponding areas in the global output, ensuring localized control while preserving overall coherence.

$$h[M_i] = h_i$$

This procedure ensures that each subject or object is rendered with high fidelity in its designated area while maintaining a coherent global context. It effectively addresses issues such as attribute leakage and concept overlap, which are common in multi-concept generation.

The full pipeline begins with per-client tuning with ED-LoRA for each concept. Then, each client's concept is sent to the central node, where Gradient Fusion combines the LoRAs into a unified model. During the generation process, Regionally Controllable Sampling is employed to ensure the accurate integration of multiple concepts into a cohesive image.

Mix-of-Show's contributions are substantial in advancing the field of multi-concept generation. It provides a scalable framework for integrating diverse concepts, offering flexibility for creative applications such as storytelling, game design, and animation. Despite its strengths, the framework is not without limitations. Attribute leakage, where characteristics from one region influence others, remains a challenge, although it can be mitigated with carefully crafted prompts. Also, the computational cost of Gradient Fusion, especially for large-scale models, can be significant, and the generation of small facial details remains difficult due to information loss in the underlying model architecture.

5.4.2 MuDI

The MuDI (Multi-subject Personalization for Decoupled Identities) framework[57] is designed to address the challenge of generating high-quality images containing multiple distinct subjects in text-to-image diffusion models. Current models often struggle with identity mixing, where attributes of different subjects blend, especially for visually or semantically similar subjects. MuDI provides a solution by decoupling the identities of multiple subjects during both training and inference. The core idea is to leverage segmented subjects extracted through segmentation models to train text-to-image models. A major innovation of MuDI is the introduction of Seg-Mix,

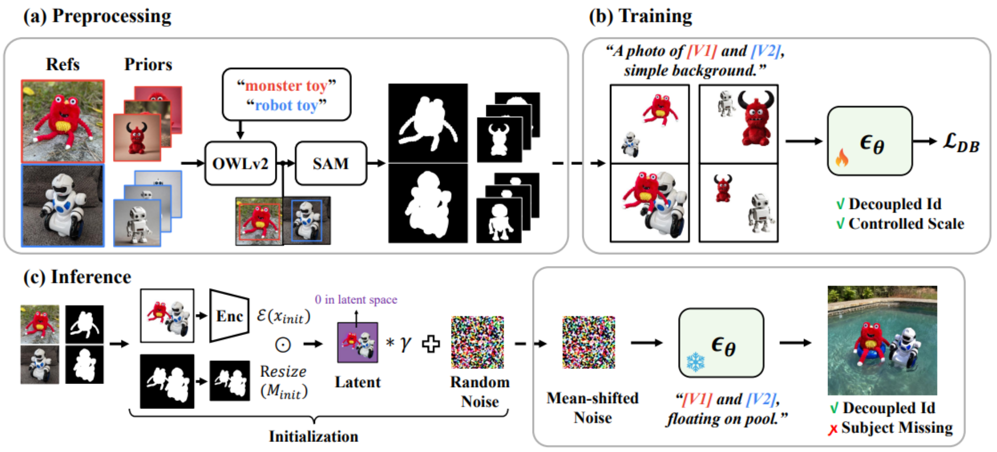


Figure 5.12: The MuDI framework.

a data augmentation method that relies on segmenting subjects and randomly positioning them within a composition. This approach eliminates background details and irrelevant artifacts, allowing the model to focus on the subjects' identity. The training process consists of three phases: first, reference images are processed through tools like the Segment Anything Model (SAM) to extract segmentation masks of individual subjects. After subject segmentation, augmented compositions are created using the extracted masks; segmented subjects are resized and placed randomly within a scene, allowing overlaps to create realistic interactions. Finally, the framework uses DreamBooth to fine-tune the pre-trained diffusion model on the augmented dataset. During this process, prior preservation is enabled to ensure that the model retains its general image-generation abilities while fine-tuning it for the new subjects.

To further enhance differentiation between subjects, descriptive classes are used during training. Instead of generic labels like "dog" or "cat," detailed descriptors (e.g., "brown robot toy") are employed, providing richer contextual information.

For inference, MuDI introduces a novel **mean-shifted initialization** technique. Rather than starting from standard Gaussian noise, MuDI initializes the diffusion process using a combination of noise and subject-specific latent embeddings, which provides a signal for separating identities without missing. This initialization prevents certain subjects from dominating the composition and reduces the chances of missing subjects during generation. It helps align the generation process with the specific layout and characteristics of the desired subjects, providing a better starting point for multi-subject compositions.

The MuDI framework also supports control over the relative size and positioning of each subject by adjusting segmentation scale during Seg-Mix. Furthermore, Seg-Mix enables modular customization, allowing separately trained single-subject LoRA modules to be composed into one coherent image through controlled spatial arrangement.

5.4.3 Isolated Diffusion

The Isolated Diffusion method[58] is a training-free approach designed to address the issue of "concept bleeding" in text-to-image diffusion models. Concept bleeding refers to the unintended merging or interference of multiple concepts within generated images, such as mixing colors, shapes, or subjects when handling multi-concept prompts. This problem is particularly evident in state-of-the-art models like Stable Diffusion.

The central idea of Isolated Diffusion is to isolate the generation process for each concept, thereby reducing mutual interference and improving text-image consistency. The method leverages pre-trained object detection and segmentation models to process and separate the individual elements of a prompt, ensuring that each component is synthesized independently before combining them into a coherent image.

Modern text-to-image diffusion models encode complex text prompts into fixed-length tokens through pre-trained text encoders. This encoding process can compress multiple concepts into overlapping latent representations, leading to unintended interactions between elements in the image. For example, a prompt like "a baby penguin wearing a blue hat, a red scarf, and a green shirt" might produce an image where colors are incorrectly assigned to attachments. Similarly, in prompts involving multiple subjects, such as "a dog next to a cat," models may produce results with concept mixing or entirely incorrect representations (e.g., "a cat next to a cat").

Isolated Diffusion splits the synthesis process into isolated steps for individual attachments or subjects. The text prompt is decomposed into simpler components using GPT4, where each attachment is individually bound to a base subject. For example, the prompt "a baby penguin wearing a blue hat, a red scarf, and a green shirt" is split into "a baby penguin," "a baby penguin wearing a blue hat," "a baby penguin wearing a red scarf," and "a baby penguin wearing a green shirt."

- Base prompt: p_{base} (e.g., "a baby penguin")
- Attachment prompts: $p_1, p_2, \ldots, p_k$ (e.g., "a baby penguin wearing a blue hat")

The noise prediction during denoising is computed as:

$$\hat{\epsilon}(x_t, t) = (1-\lambda)\epsilon_\theta(x_t, t, c_{\text{ucon}}) + \lambda\epsilon_\theta(x_t, t, c_{\text{base}}) + \sum_{i=1}^{k} \lambda \left(\epsilon_\theta(x_t, t, c_i) - \epsilon_\theta(x_t, t, c_{\text{base}})\right) \tag{5.1}$$

Here, λ is a guidance scale, c_{ucon} is the unconditional embedding, and c_i are the CLIP-encoded attachment prompts. During the denoising process, the variance between the noise predicted for the base subject and each attachment is calculated and added sequentially, ensuring that each attachment is synthesized independently, avoiding interference.

For images involving multiple distinct subjects, separate prompts for each subject are created, then the framework uses YOLO and SAM to detect subjects and segment them into masks $M_1, \ldots, M_k$. The image is then processed in two stages: first, the overall layout is synthesized using the base prompt. Secondly, the regions corresponding to individual subjects are isolated by replacing other regions with random noise.

$$x_t^{(i)} = x_t \odot (1 - M_{\text{others}}) + \epsilon \odot M_{\text{others}} \tag{5.2}$$

Each subject is then denoised separately using its corresponding text prompt, preventing attention overlap between different subjects.

$$\epsilon_i = (1-\lambda)\epsilon_\theta(x_t^{(i)}, t, c_{\text{ucon}}) + \lambda\epsilon_\theta(x_t^{(i)}, t, c_i) \tag{5.3}$$

Once all subjects or attachments have been synthesized independently, they are combined into a single image using the generated masks, and the background is refined in the final stages of denoising to ensure a harmonious and visually coherent result.

$$\hat{\epsilon}(x_t, t) = \epsilon_0 \odot (1 - \cup_i M_i) + \sum_i \epsilon_i \odot M_i \tag{5.4}$$

This method prevents attention overlap and identity confusion by isolating the denoising paths for each concept.

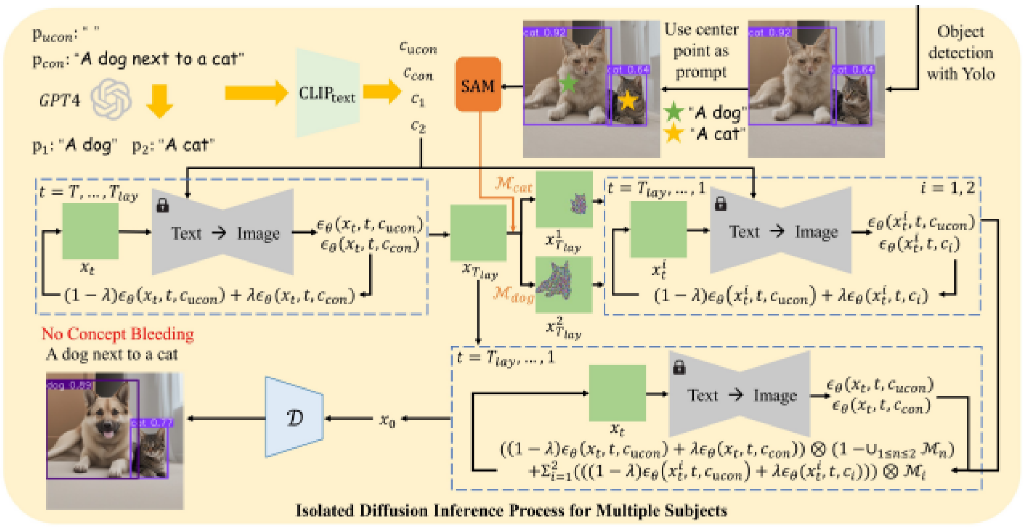

p_{ucon}: " "
p_{con}: "A dog next to a cat"
GPT4
p_1: "A dog" p_2: "A cat"
CLIP$_{text}$
c_{ucon}
c_{con}
c_1
c_2
SAM
Use center point as prompt
Object detection with Yolo
"A dog"
"A cat"
$t = T, \dots, T_{lay}$
Text → Image
x_t
$(1-\lambda)\epsilon_\theta(x_t, t, c_{ucon}) + \lambda\epsilon_\theta(x_t, t, c_{con})$
$\mathcal{M}_{cat}$
$\mathcal{M}_{dog}$
$x_{T_{lay}}$
$t = T_{lay}, \dots, 1$
$i = 1, 2$
$(1-\lambda)\epsilon_\theta(x_t^i, t, c_{ucon}) + \lambda\epsilon_\theta(x_t^i, t, c_i)$
No Concept Bleeding
A dog next to a cat
$\mathcal{D}$
x_0
$((1-\lambda)\epsilon_\theta(x_t, t, c_{ucon}) + \lambda\epsilon_\theta(x_t, t, c_{con})) \otimes (1 - \cup_{1 \le n \le 2} \mathcal{M}_n)$
$+\Sigma_{i=1}^{2}(((1-\lambda)\epsilon_\theta(x_t^i, t, c_{ucon}) + \lambda\epsilon_\theta(x_t^i, t, c_i))) \otimes \mathcal{M}_i$
Isolated Diffusion Inference Process for Multiple Subjects

Figure 5.13: Isolated Diffusion framework.

Isolated Diffusion requires no retraining and is compatible with any pre-trained diffusion model. It also integrates with widely-used segmentation and detection tools . However, its performance depends on the accuracy of these tools.

The key advantages of Isolated Diffusion are that it is training-free, working directly with pre-trained diffusion models, and is compatible with any text-to-image diffusion model. It can also be integrated with various pre-trained object detection and segmentation models like YOLO and SAM. However, it relies on the performance of the chosen models. If subjects are not detected or the base diffusion model omits elements, the final output may still be inconsistent. Additionally, it requires extra computational steps during inference due to the segmentation and isolated denoising processes.

5.4.4 FastComposer

FastComposer[59] is a framework designed for personalized multi-subject text-to-image generation that eliminates the need for subject-specific fine-tuning. It addresses two major challenges in current approaches: the computational overhead of personalization and the issue of identity blending, where visual characteristics of distinct subjects merge unintentionally. FastComposer circumvents these problems by introducing three key innovations: subject embedding augmentation, cross-attention localization, and delayed subject conditioning.

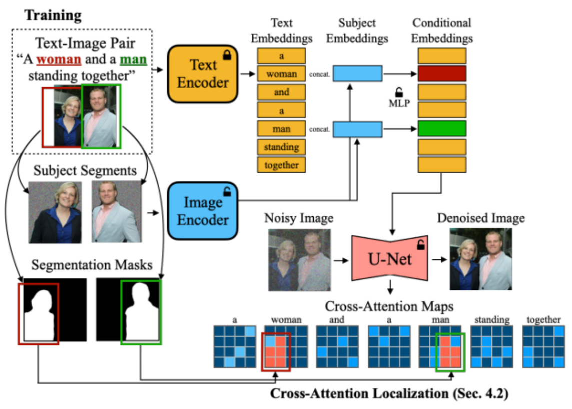


Figure 5.14: Training and inference pipeline of FastComposer.

FastComposer achieves subject-driven image generation without any additional fine-tuning, by enriching the textual prompt with visual information derived from reference images of the subject. This integration of visual cues allows the model to condition its output on specific identities in a zero-shot manner. Given a prompt $P = \{w_1, w_2, \ldots, w_n\}$, a set of reference images $S = \{s_1, \ldots, s_m\}$, and index mapping $I = \{i_1, \ldots, i_m\}$ associating each subject with a word token in the prompt, the system first encodes the prompt and subject images using CLIP text and image encoders ψ and ϕ. Subject embeddings are then concatenated with the corresponding text embeddings and processed through a multi-layer perceptron (MLP) to yield the final conditioning vector $c' \in \mathbb{R}^{n\times d}$:

$$c'_i = \begin{cases} \psi(P)_i, & i \notin I \\ \text{MLP}(\psi(P)_i \| \phi(s_j)), & i = i_j \in I \end{cases} \tag{1}$$

Training is performed on the subject-augmented image-text dataset, allowing the model to generate images featuring specific subjects directly from textual instructions and reference images, without requiring fine-tuning for each subject and thereby significantly reducing computational costs and memory requirements. At inference time, FastComposer leverages these learned associations to generate high-quality images of multiple subjects in diverse scenarios.

To address the problem of identity blending, FastComposer employs cross-attention localization during training. In diffusion models, cross-attention maps determine how textual tokens relate to specific regions of the image. Without regulation, these maps can link multiple textual tokens to overlapping regions, causing identity blending. FastComposer enforces spatial alignment by supervising cross-attention maps using segmentation masks extracted from reference images of each subject.

This mechanism ensures that the model learns to direct each subject's attention to its corresponding spatial region.

Formally, given cross-attention maps $A \in [0,1]^{(h\times w)\times n}$, where $A_i \in [0,1]^{h\times w}$ denotes the attention map for subject token i, and m_j represents the segmentation mask for subject j, a regularization loss is applied to encourage each token's attention to remain focused within its designated mask area:

$$\mathcal{L}_{\text{loc}} = \frac{1}{m}\sum_{j=1}^{m}\Big(\text{mean}(A_{i_j}[\bar{m}_j]) - \text{mean}(A_{i_j}[m_j])\Big) \tag{2}$$

This localization term is added to the denoising loss to form the total training objective:

$$\mathcal{L} = \mathcal{L}_{\text{noise}} + \lambda\mathcal{L}_{\text{loc}} \tag{3}$$

where $\lambda = 0.001$ is a weighting hyperparameter. This attention supervision ensures that each subject's identity is spatially isolated during training, thus improving generation quality in multi-subject settings.

Another innovation is delayed subject conditioning, which addresses the issue of subject overfitting. Overfitting occurs when the model overly relies on reference images, resulting in limited flexibility to edit subjects through textual instructions. FastComposer delays the application of subject embeddings in the denoising process during inference, using only text embeddings in the early stages of diffusion to establish the image layout. Once the layout is defined, subject embeddings are introduced to refine the appearance of each subject. This is formalized as:

$$\epsilon_t = \begin{cases} \epsilon_\theta(z_t, t, c), & t > \alpha T \\ \epsilon_\theta(z_t, t, c'), & \text{otherwise} \end{cases} \tag{4}$$

where c is the pure text embedding, c' is the augmented embedding with subject information, and $\alpha \in [0.6, 0.8]$ controls the transition point in the denoising steps. This approach strikes a balance between subject identity preservation and flexible editing via the text prompt.

Despite its advantages, FastComposer has some limitations, the main being that, because the dataset used for training is relatively small and predominantly composed of headshots, the model may be limited when generating complex scenes with more than three subjects or depicting a wide range of actions and scenarios with significant background interaction.

5.4.5 LoRA Composer

LoRA-Composer[60] is a framework designed for seamlessly integrating multiple LoRAs in the same image, addressing the major challenges of concept vanishing -where certain concepts fail to appear in the generated image- and concept confusion, where distinct concepts blend or interfere with each other.

Unlike earlier methods such as Mix-of-Show, which rely on fusion training and additional image-based conditions, LoRA-Composer leverages only layout and textual guidance, enhancing both usability and computational efficiency.

The framework introduces two main constraints: concept injection constraints and concept isolation constraints, along with a latent re-initialization technique. Concept injection constraints address concept vanishing by directly injecting LoRA

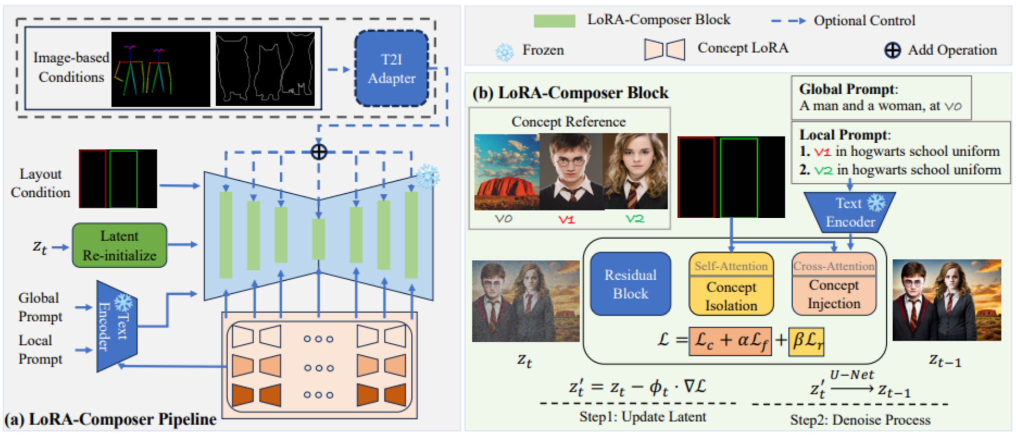

LoRA-Composer Block
Optional Control
Frozen
Concept LoRA
Add Operation
Image-based Conditions
T2I Adapter
Layout Condition
z_t
Latent Re-initialize
Global Prompt
Local Prompt
Text Encoder
(a) LoRA-Composer Pipeline
(b) LoRA-Composer Block
Concept Reference
V0
V1
V2
Global Prompt: A man and a woman, at V0
Local Prompt:
1. V1 in hogwarts school uniform
2. V2 in hogwarts school uniform
Text Encoder
Residual Block
Self-Attention Concept Isolation
Cross-Attention Concept Injection
$\mathcal{L} = \mathcal{L}_c + \alpha\mathcal{L}_f + \beta\mathcal{L}_r$
z_t
z_{t-1}
$z_t' = z_t - \phi_t \cdot \nabla\mathcal{L}$
$z_t' \xrightarrow{U-Net} z_{t-1}$
Step1: Update Latent
Step2: Denoise Process

Figure 5.15: LoRA-Composer pipeline.

features into pre-specified regions of the image using layout masks. For each concept i, the queries, keys, and values in the cross-attention layers are modified as follows:

$$Q_i = M_i \odot W_Q^0(z), \quad K_i = W_K^i(\tau_i(P_i)), \quad V_i = W_V^i(\tau_i(P_i)),$$

where M_i is the layout mask for concept i, τ_i is the CLIP text encoder with LoRA, and W_Q, W_K, W_V are projection matrices for the attention module. The cross-attention is computed as:

$$h_i = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d}}\right) V_i.$$

To encourage more detailed and region-complete synthesis, LoRA-Composer adds a concept enhancement constraint composed of two losses. The first, $\mathcal{L}_c$, applies a Gaussian-weighted top-k activation mask to ensure high response in the target region:

$$\mathcal{L}_c = \sum_{i=1}^{N} \left(1 - \frac{1}{S} \sum_{j \in E} \text{topk}(A_i^j \odot M_i \odot G, S) \right),$$

where A_i^j is the cross-attention map for concept token j, G is a Gaussian map, and S is the number of selected elements. The second loss $\mathcal{L}_f$ penalizes underfilling of the layout box:

$$\mathcal{L}_f = \frac{1}{L} \sum_{i=1}^{N} \sum_{j \in E} \left(1 - \left\{ M_i \odot a_i^j(w), M_i \odot a_{ij}(h) \right\} \right),$$

where $a_i^j(w)$ and $a_i^j(h)$ are max-pooled projections of A_i^j along the spatial axes.

On the other hand, concept isolation constraints tackle the issue of concept confusion. To prevent concept overlap, the framework introduces concept region masks in the self-attention layers to restrict attention only within a concept's designated area. To further avoid feature leakage into unintended regions, it introduces a region perceptual restriction loss:

$$\mathcal{L}_r = \frac{1}{S} \sum_{i=1}^{N} \text{topk}(\bar{A}[M_i, 1 - M_i], S),$$

where $\bar{A}$ denotes the self-attention map sliced between the concept and non-concept areas.

The total constraint loss is given by:

$$\mathcal{L} = \mathcal{L}_c + \alpha \mathcal{L}_f + \beta \mathcal{L}_r,$$

and used to update the latent at each timestep:

$$z_t' \leftarrow z_t - \phi_t \cdot \nabla \mathcal{L},$$

where ϕ_t is a decaying step size. The updated latent z_t' is then fed into the U-Net for denoising.

The latent re-initialization mechanism is a technique that adjusts the latent space before denoising begins, aligning the initialized latent features with the layout conditions provided by the user, thus improving the precision of object placement in the generated image. In particular, since standard LoRA lacks localization, LoRA-Composer refines the layout alignment by re-initializing the latent space z_t using one update step of $\mathcal{L}$. It then selects the highest scoring candidate layout region from a set of crops:

$$\hat{A}_i = \{\Phi(A_i, [x, y], W, H)\},$$

where $\Phi(\cdot)$ crops a region of size $W \times H$ from position (x, y) in the attention map A_i. This improves alignment of concepts to their designated spatial regions.

LoRA-Composer integrates constraints and control mechanisms directly into the cross-attention and self-attention layers of the diffusion model's U-Net architecture. Moreover it eliminates the need for retraining for each combination of multiple LoRAs—a requirement in prior methods such as Mix-of-Show-, because it allows for on-the-fly composition without additional training, making it both more efficient and adaptable to diverse generation scenarios.

The method's ability to handle diverse scenarios, including anime and realistic styles, highlights its versatility and robustness. However, LoRA-Composer has some limitations. When concepts are placed too closely together, their boundaries may blur due to spatial overlap, leading to potential blending. Additionally, objects can sometimes extend beyond their designated layout regions, a limitation inherited from the generalized assumptions of Stable Diffusion. Finally, the inference process involves loading multiple LoRA checkpoints and updating latent representations, which introduces a slight delay in generation time.

5.4.6 CLoRA

CLoRA (Contrastive LoRA)[61] is a training-free inference-time approach designed to reduce concept entanglement in multi-LoRA text-to-image generation. Rather than modifying model weights or retraining, CLoRA leverages contrastive learning over cross-attention maps to disentangle concept regions, ensuring that each LoRA affects only the intended parts of the image.

The key insight is that concept confusion arises when tokens from different LoRA modules activate overlapping attention regions. CLoRA addresses this by generating attention maps for each concept in isolation and then applying a contrastive attention loss to suppress overlapping activations during joint inference.

For a prompt like "a woman with an umbrella," CLoRA constructs modified

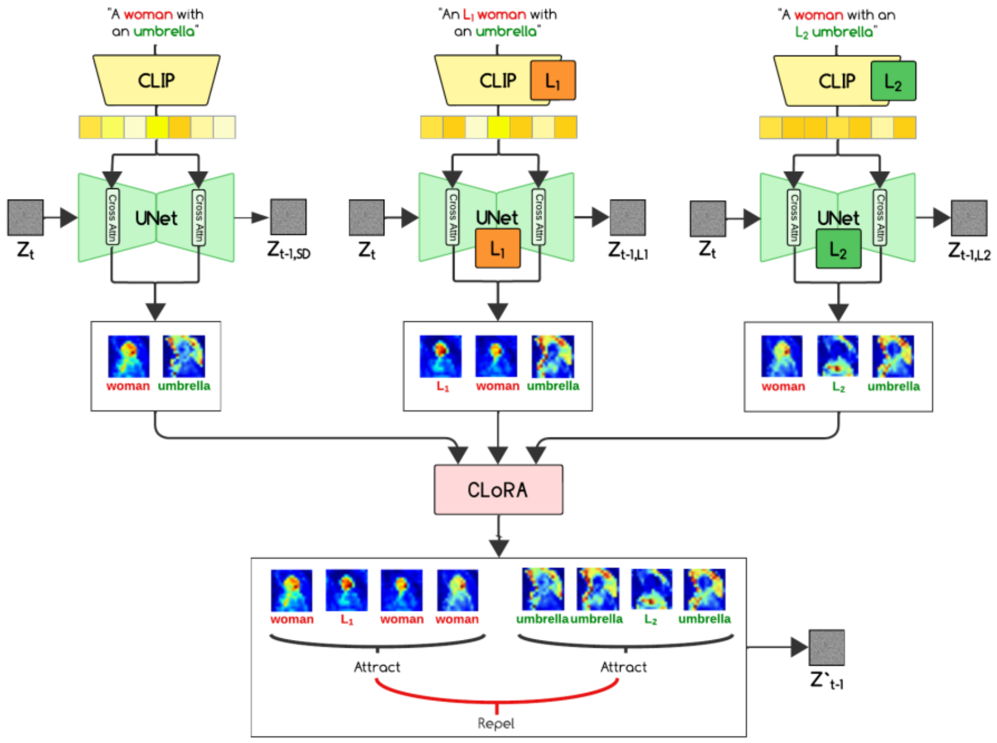

"A woman with an umbrella"
"An L1 woman with an umbrella"
"A woman with an L2 umbrella"
CLIP
UNet
Cross Attn
Z_t
Z_t-1,SD
Z_t-1,L1
Z_t-1,L2
woman
umbrella
L1
L2
CLoRA
Attract
Repel
Z`t-1

Figure 5.16: Overview of the CLoRA method.

prompts that isolate each concept via LoRA activation:

- The base prompt activates neither LoRA.
- The first variant activates only L_1 (e.g., “an L1 woman with an umbrella”) to localize the "woman" concept.
- The second activates only L_2 (e.g., “a woman with an L2 umbrella”) to localize the "umbrella" concept.

From these, CLoRA extracts cross-attention maps for each relevant token (e.g., “woman,” “L1,” and “umbrella,” “L2”) and groups them by concept. For each concept group G, attention maps are expected to be similar internally and distinct from other groups. The InfoNCE loss enforces this structure:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(A_j, A_{j^+})/\tau)}{\sum_{n \in \{j^+, j_1^-, \ldots, j_N^-\}} \exp(\text{sim}(A_j, A_n)/\tau)} \tag{5.5}$$

where $\text{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|}$ is cosine similarity and τ is a temperature parameter.

This loss updates the latent z_t during each diffusion step as:

$$z'_t = z_t - \alpha_t \nabla_{z_t} \mathcal{L}_{\text{InfoNCE}} \tag{5.6}$$

To reinforce spatial disentanglement, CLoRA employs binary attention masks. For an attention map $A[x, y]$, the mask is computed via:

$$M[x, y] = \mathbb{I}\left(A[x, y] \geq \tau \cdot \max_{i,j} A[i, j]\right) \tag{5.7}$$

These masks are aggregated per concept using union operations and applied to blend the corresponding LoRA latents into the global latent space:

$$M_{\text{LoRA}} = M_{\text{token}_1} \cup M_{\text{token}_2} \cup \dots \tag{5.8}$$

Once the binary masks are computed for each concept—by thresholding the attention maps and aggregating token-specific activations—CLoRA uses them to restrict each LoRA's influence to its intended image region. The global latent is updated by blending the base and LoRA-modified latents using the masks:

$$z_t = (1 - M_{\text{LoRA}}) \odot z_t^{\text{base}} + M_{\text{LoRA}} \odot z_t^{\text{LoRA}} \tag{5.9}$$

This mechanism ensures that each concept modifies only its relevant region, reducing identity confusion and attribute leakage. Since CLoRA operates entirely at inference time—without retraining or architectural changes—it is compatible with arbitrary LoRA modules and preserves modularity.

Its main trade-offs are increased inference time due to contrastive optimization, and reliance on attention map quality for accurate spatial disentanglement. Still, CLoRA offers a practical and effective solution for multi-LoRA composition, using attention-based contrastive learning to enforce semantic and spatial separation.

5.4.7 Subject Diffusion

Subject-Diffusion[62] introduces a zero-shot framework for personalized text-to-image generation, enabling subject-specific synthesis from just one reference image—without any test-time fine-tuning. Unlike prior methods such as DreamBooth or Custom Diffusion, which demand fine-tuning on multiple reference images, Subject-Diffusion operates in an open domain and is designed to scale efficiently to single and two-subject generation.

A key contribution of the method is the construction of the Subject-Diffusion Dataset, a large-scale multimodal dataset comprising 76M images and 222M entities. Each image is annotated with generated captions (via BLIP-2), object detection boxes (via Grounding DINO), segmentation masks (via SAM), and refined noun phrase tags (via spaCy). Such a dataset enables the model to learn rich representations of subjects and their variations in appearance, context, and style.

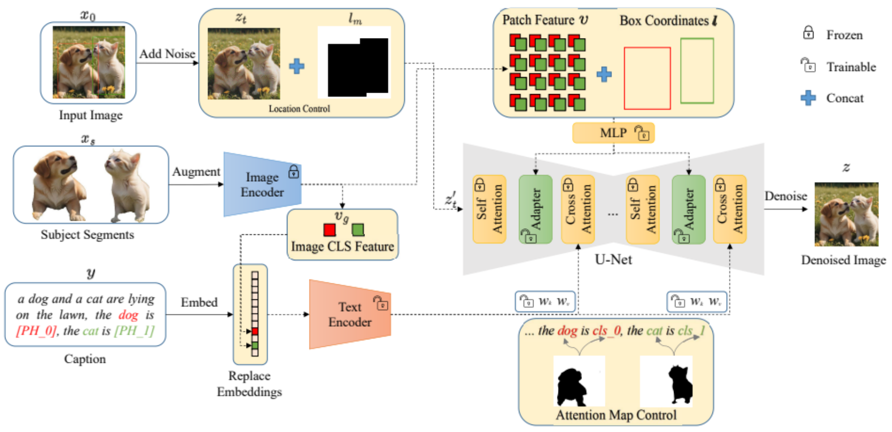


Figure 5.17: The Subject Diffusion framework.

Subject-Diffusion integrates textual and visual information through a structured prompt and multi-modal conditioning pipeline. The prompt is reformulated to include placeholders that represent subject identities—e.g., a prompt like *"a dog on a cobblestone street"* becomes *"a dog is [PH_0] on a cobblestone street"*, where `[PH_0]` is a placeholder linked to a reference image. During encoding, the embedding of this placeholder token is replaced with the corresponding subject's image representation, specifically the "CLS" token extracted from a CLIP image encoder. This substitution allows the model to associate the subject reference image directly with the placeholder in the text, enabling identity-aware generation in a zero-shot setting.

These combined inputs—text with image-conditioned tokens—are passed through a U-Net-based diffusion model. Unlike prior methods that freeze the text encoder, Subject-Diffusion jointly trains the text encoder to improve alignment between text descriptions and visual features. Within the U-Net, lightweight adapter modules are inserted between self-attention and cross-attention layers to inject dense subject-specific visual features. These adapters guide the network to focus on detailed visual identity cues from the reference images while preserving semantic alignment with the text prompt.

Fine-grained image patch features v are extracted using a CLIP image encoder from segmented subject images. These are fused with subject location coordinates l using a Fourier transformation and a learnable MLP:

$$h_e = \mathrm{MLP}([v, \mathrm{Fourier}(l)]),$$

The adapter layer updates the intermediate latent L_a as:

$$L_a := L_a + \beta \cdot \tanh(\gamma) \cdot S([L_a, h_e]),$$

where β is a fixed balance constant, γ is a learnable scalar, and $S(\cdot)$ is a self-attention operator.

To ensure spatial separation of multiple subjects, Subject-Diffusion supervises the cross-attention maps to match ground-truth segmentation masks. For each subject k, the attention map $\mathrm{CA}_\ell(z_t, y_k)$ at layer ℓ is defined as:

$$\mathrm{CA}_\ell(z_t, y_k) = \mathrm{Softmax}(Q_\ell(z_t) \cdot K_\ell(y_k)^T),$$

and the attention loss is defined as:

$$\mathcal{L}_{attn} = \frac{1}{N} \sum_{k=1}^{N} \sum_{\ell} \left\| \mathrm{CA}_\ell(z_t, y_k) - M_k \right\|_1,$$

where M_k is the segmentation mask for subject k. penalizing attention deviations from the expected subject regions.

The overall training loss includes the standard denoising loss and attention alignment:

$$\mathcal{L} = \mathbb{E}_{x_0, y, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(z_t, t, y, x_s, l, l_m)\|^2 \right] + \lambda_{\mathrm{attn}} \cdot \mathcal{L}_{attn}.$$

Subject-Diffusion also supports controlled interpolation between text and image guidance. During inference, text and image control are applied in different diffusion phases:

$$\epsilon_t = \begin{cases} \epsilon_\theta(z_t, t, y'), & \text{if } t > \alpha T, \\ \epsilon_\theta(z_t, t, y), & \text{otherwise}, \end{cases}$$

where y' is a blended prompt embedding that incorporates image placeholder tokens.

Subject-Diffusion achieves a balance between fidelity, flexibility, and generalization in open-domain, zero-shot personalized image generation. The main limitations of this framework are in handling attributes and accessories in user-provided images and challenges in generating coherent compositions for more than two subjects.

Chapter 6

Contribution

6.1 Contribution

Diffusion models[14] have demonstrated remarkable text-to-image generation capabilities, enabling users to guide the generation process through natural language while achieving impressive levels of artistry and authenticity. These advancements have opened new possibilities for a range of creative applications, including visual storytelling.

However, large-scale text-to-image diffusion models face inherent challenges in producing consistent subjects and styles across different frames—an essential requirement for creating cohesive and meaningful narratives. This limitation has driven increasing interest in the development of efficient personalization techniques, which allow models to represent specific subjects or styles in novel contexts without requiring resource-intensive retraining.

Among the diverse application areas of text-to-image models and fine-tuning techniques, storyboarding emerges as a particularly interesting use case. Traditionally, storyboards are created manually, either through hand-drawing or digital sketching tools—a process that is both labor-intensive and time-consuming. Therefore, we decided to leverage the personalization capabilities of text-to-image models to teach Stable Diffusion[17] specific character identities and cinematic shot types.

Shot types play a crucial role in establishing the emotional tone and spatial framing of a scene, while character consistency across frames is vital for preserving narrative continuity. Teaching the model to internalize these two components—via DreamBooth LoRA fine-tuning—enables the generation of storyboards that are actually meaningful for planning and visualizing shots ahead of their practical execution.

As discussed in the previous chapter, an additional key challenge arises in the generation of multi-character scenes. Merging multiple LoRA modules often leads to concept bleeding, confusion, or omission, which severely impacts the reliability of generated outputs. Several frameworks have been introduced in recent years to tackle this limitation and to address the task of story generation. However, a widely adopted standard approach has yet to emerge, and each method presents its own set of limitations. Among the methods discussed, only a few have made their code publicly available for practical experimentation, and many of the approaches for story visualization have been trained primarily on cartoon-style datasets with a fixed roster of characters, making them less adaptable to broader domains and more diverse visual styles.

Additionally, while online tools for AI-assisted storyboarding are being developed, they often lack open-source accessibility, offer limited free trials, and provide no technical insights into their implementations.

In response to these challenges, this work presents an approach based on efficient fine-tuning, prompt engineering, and iterative refinement to enable AI-assisted storyboard creation. As fine-tuning technique, we decided to employ **Dreambooth LoRA**[41][42], which is widely regarded in the online community for its ability to dramatically reduce not only training time and computational requirements compared to full-fledged fine-tuning, but also the size of the output, producing model weight files that are only a few MBs in size and easy to store and share.

Recognizing that fine-tuning performance is heavily dependent on the quality of training data, a key focus of this work is to present a structured process for data gathering and preparation. Additionally, to address the challenges posed by combining multiple character LoRAs in a single image, we developed an inpainting-based approach that enables users to generate masks for inpainting characters automatically, mitigating issues such as concept bleeding or confusion.

Therefore, our contributions are as follows:

- Outlining a process for building high-quality and diverse datasets for fine-tuning: shot type datasets curated from real movie stills, and character identity datasets generated using MidJourney's character reference feature.
- Applying Dreambooth LoRA to Stable Diffusion to efficiently fine-tune the model on eight distinct shot types and seven character identities, enabling personalized and consistent image generation.

- Introducing a structured prompt template to transform broad scene descriptions into detailed, frame-level prompts, leveraging ChatGPT—improving narrative consistency and control across storyboard sequences.
- Developing an interactive storyboard generation system that enables users to generate, review, and refine storyboard frames in real time, including features for prompt revision, image selection, and inpainting-based character correction.

Project files will be available at the following GitHub Repository.

6.2 Method

6.2.1 Training data collection and preparation

The first step to effectively fine-tune the model is to build a high-quality training dataset. This is crucial to obtain good results later on when generating images with the desired shot type and/or character.

Specifically, we needed to create a dataset for each shot type and each character.

Shot type dataset

We built the dataset for the shot types from FILMGRAB[63], a popular online portal that features a large collection of stills from a wide range of movies. To automate the data collection process, we first analyzed the HTML structure of the FILMGRAB website pages and then developed a web scraper to download the still galleries of movies, filtering those released from 2013 to 2024, in order to ensure good image resolution and quality. The resulting number of downloaded images was 63,482 from 1,075 movies.

When determining the ideal number of training images per shot type, we considered insights from the community and literature. While there is no universally agreed-upon number, fine-tuning diffusion models typically benefits from curated datasets ranging from dozens to a few hundred images per concept. Using too many samples, especially without sufficient diversity, can lead to overfitting, where the model memorizes specific visual details instead of learning general patterns. On the other hand, using too few samples may result in underfitting, where the model fails to learn the concept effectively and generalizes poorly.

After testing several dataset sizes, we decided to use 200, manually selected images

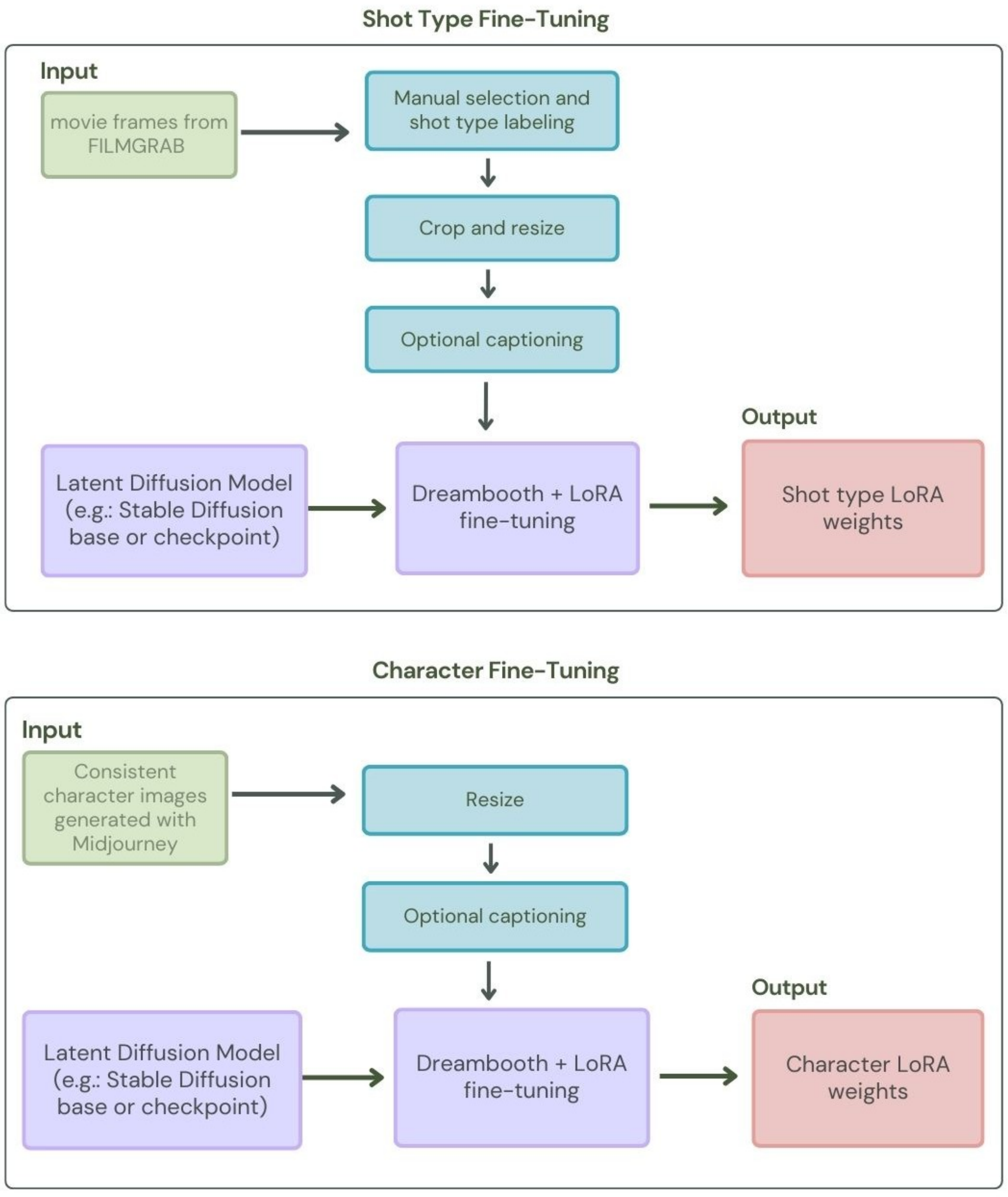

Shot Type Fine-Tuning
Input
movie frames from FILMGRAB
Manual selection and shot type labeling
Crop and resize
Optional captioning
Output
Latent Diffusion Model (e.g.: Stable Diffusion base or checkpoint)
Dreambooth + LoRA fine-tuning
Shot type LoRA weights
Character Fine-Tuning
Input
Consistent character images generated with Midjourney
Resize
Optional captioning
Output
Latent Diffusion Model (e.g.: Stable Diffusion base or checkpoint)
Dreambooth + LoRA fine-tuning
Character LoRA weights

Figure 6.1: Shot type and character Fine-tuning

per shot type. We selected images with a diversity in terms of movie genres,

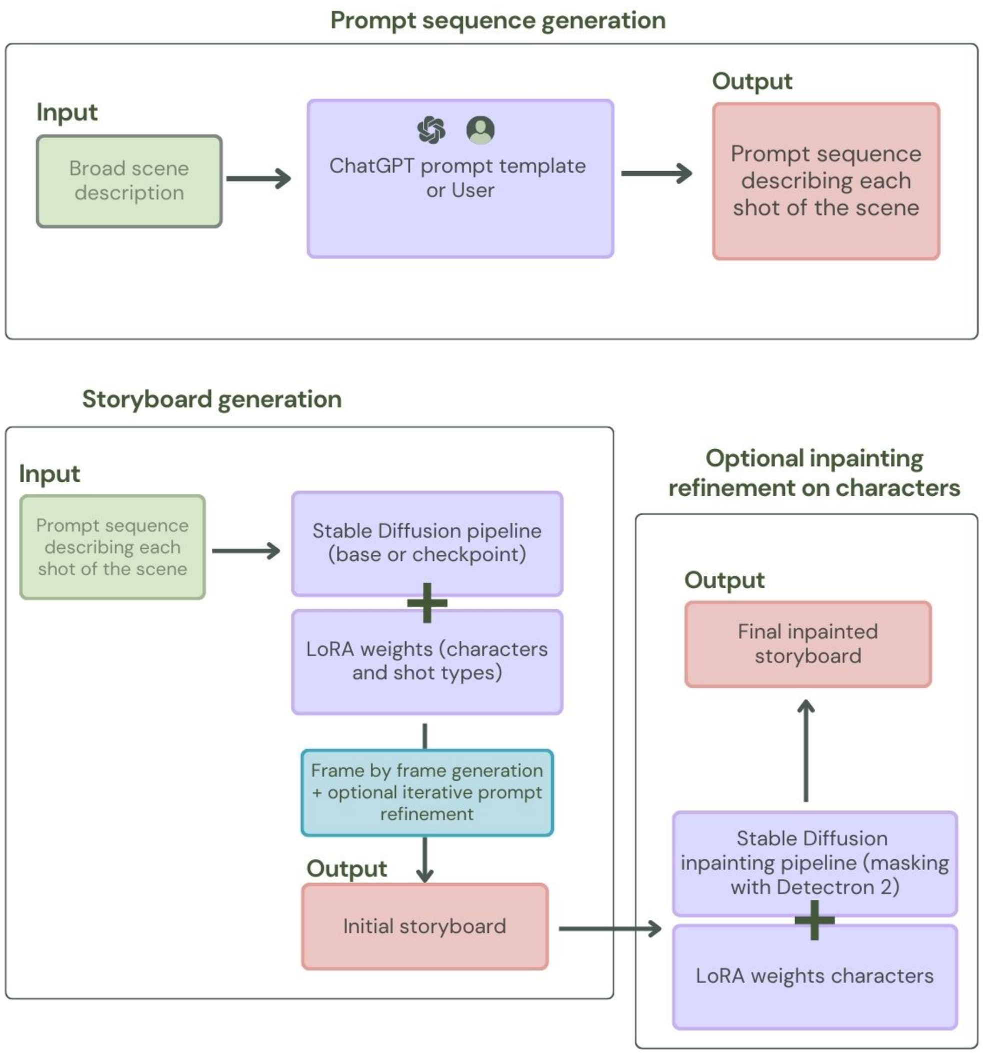

Prompt sequence generation
Input
Broad scene description
ChatGPT prompt template or User
Output
Prompt sequence describing each shot of the scene
Storyboard generation
Input
Prompt sequence describing each shot of the scene
Stable Diffusion pipeline (base or checkpoint)
LoRA weights (characters and shot types)
Frame by frame generation + optional iterative prompt refinement
Output
Initial storyboard
Optional inpainting refinement on characters
Output
Final inpainted storyboard
Stable Diffusion inpainting pipeline (masking with Detectron 2)
LoRA weights characters

Figure 6.2: Method overview

character poses, backgrounds, and lighting conditions, all while ensuring consistency in the framing specific to each shot type. This curated selection step was essential to avoid overfitting on specific lighting conditions, poses or styles, and instead help the model internalize the general framing rules that define each shot type.

We focused on eight canonical cinematic shot types, defined as follows:

- **Extreme Close-Up:** The human figure is framed from the chin up.
- **Close-Up:** The human figure is framed from above the shoulders up.

- **Medium Close-Up:** The human figure is framed from mid-torso upward.
- **Medium Shot:** The upper half of the human figure is framed.
- **American Shot:** The human figure is framed from above the knee or mid-thigh upward.
- **Full Shot:** The human figure is framed entirely or almost entirely, occupying more than 2/3 of the image in height.
- **Long Shot:** The human figure is framed entirely and occupies between 2/3 and 1/3 of the screen height.
- **Extreme Long Shot:** The subject is either very small (less the 1/3) in the frame or entirely absent, with the environment dominating the composition

After selecting 200 images for each category, we used the online tool `birme.net` to batch resize them to a resolution of 512×512 pixels. In some cases, we manually adjusted the crop to avoid incorrect framing of the human figure, such as having heads or limbs unintentionally cut off.

This uniform resolution is important for compatibility with the Stable Diffusion 1.5 model, which was chosen for its wide adoption, support for LoRA-based fine-tuning and elevated number of specialized checkpoints available, allowing our proposed framework to extend to a variety of already available styles.

Character dataset

After preparing the datasets for shot type fine-tuning, we moved on to the next stage of our pipeline: creating the training data for the individual characters that would be featured in our storyboards.

To construct a robust training set for each character, we needed images that portrayed the subject across diverse visual settings while maintaining visual identity and recognizability. However, sourcing such images from the web or real people raises potential legal and ethical concerns related to copyright and privacy. To avoid these issues entirely and retain full creative control, we chose to generate our own original characters from scratch.

At this point, a key technical decision involved selecting a tool or method capable of producing consistent visual portrayals of characters across multiple generations. Given the need for high-quality imagery, time constraints, and efficient generation at scale, we selected MidJourney as our tool of choice. In early 2024, MidJourney introduced a *Character Reference* feature, specifically designed to support the consistent depiction of fictional or stylized characters across a wide range of prompts. This feature was particularly suitable for our task.

The process began by prompting MidJourney to generate a six-panel collage (or grid) of close-up character portraits on a white background for each character. In the prompt, we specified core attributes such as age, ethnicity, skin tone, eye color, hairstyle, and hair color of the character. The neutral background ensured clarity and minimized distraction, allowing us to better evaluate each generated character design. After reviewing the four results generated by Midjourney, we selected one preferred grid and extracted the individual portraits.

From each grid, we chose four representative images to serve as *reference images* for future prompt-based generations. MidJourney's character reference feature includes a parameter called `-cw`, which adjusts the strength of the character conditioning:

- `-cw 0`: emphasizes only the character's face, ideal for wide shots or scenes with minimal facial detail.
- Higher `-cw` values: enforce stronger conditioning, preserving more detailed facial and clothing characteristics, ideal for close-ups or medium shots.

Using this approach, we generated a dataset of 70 high-resolution images for 7 distinct characters -70 images per character- portraying each character in a variety of scenes, poses, actions, lighting conditions, and shot types, in order to provide the model with sufficient diversity to be capable of generalizing to unseen compositions. To maintain temporal and visual consistency within our storyboards, we fixed the clothing style of each character throughout their respective datasets, minimizing intra-character variation while allowing for contextual and compositional diversity.

Before fine-tuning, all images were uniformly resized to 512×512 pixels using the online batch image tool available at `https://www.birme.net/`, ensuring compatibility with the Stable Diffusion 1.5 architecture, which operates on square 512-pixel inputs.

6.2.2 Training

As previously mentioned, we selected DreamBooth LoRA as our fine-tuning technique of choice after reviewing the most widely adopted methods for personalizing Stable Diffusion models. This decision was mainly informed by insights from practitioner communities and open-source projects. DreamBooth LoRA offered an ideal trade-off between training speed, memory efficiency, and quality of personalization, making it particularly suitable for our dual fine-tuning goals: encoding specific characters and shot types.

Given the limited availability of local high-performance computing resources, all training and inference tasks were carried out in **Kaggle** [64], which provides access to free GPU-enabled environments via hosted Jupyter notebooks.

DreamBooth is a fine-tuning framework developed to enable a pre-trained text-to-image diffusion model to learn how to generate new content centered around a specific subject or visual style, even when provided with as few as 3 to 5 example images. The core idea behind Dreambooth is to train the model to associate the desired subject or style with a unique identifier. This token is inserted into textual prompts during training, often alongside a general class label (e.g., "a `[V]` dog"), allowing the model to anchor new visual information to an existing semantic category. The model thereby benefits from its prior knowledge about the broader class while learning the distinguishing features of the specific instance.

To prevent catastrophic forgetting and overfitting -two common pitfalls in low-sample fine-tuning— DreamBooth also supports class regularization and prior preservation loss. These help ensure that the model retains its ability to generate generic examples from the base class, even as it learns to specialize in a new subject. In practice, this involves generating or including a set of reference images from the base class during training and penalizing deviation from the original model's outputs on those samples.

In our work, we applied DreamBooth LoRA fine-tuning in two different contexts: shot type learning and character personalization.

The training prompt for shot type fine-tuning was designed to explicitly reference the target framing style (e.g., "medium shot picture of a person"), allowing the model to associate each prompt with its corresponding visual composition. Because shot types represent stylistic camera compositions that the model already has visual priors for (e.g., the model already has a concept of how a medium shot or close-up generally look like) — we did not apply class regularization, as fine-tuning in this context does not introduce entirely novel concepts but rather reinforces pre-existing visual knowledge. The primary objective was to sharpen the model's internal understanding of these framing conventions and encourage more consistent output across generations. Additionally, the inherent diversity of our training data, spanning a broad range of visual contexts, helped mitigate the risk of overfitting. Each shot type was fine-tuned into a separate LoRA module, allowing us to switch between them as needed at inference time without retraining, offering both flexibility and modularity in the generation process.

For character fine-tuning instead, we adopted a more cautious approach, using class regularization and prior preservation loss during training to prevent overfitting and concept drift. Character fine-tuning was performed on a smaller number of images (70 per character), making it more prone to overfitting -where the model memorizes the exact appearance of the character in the training dataset rather than learning how to generalize their features in different contexts. Each character was assigned a unique, space-free name token (e.g., “cathrinannett”, “kaitoyosuke”) and linked to a general category such as “man” or “woman” within the training prompts. This strategy allowed the model to preserve its general understanding of the broader class while simultaneously learning the distinct visual features of each individual character. Regularization mitigated the tendency of the model to memorize the specific training images and helped ensure generaliztion to new poses, environments, and lighting conditions during generation.

As anticipated, we did not perform full fine-tuning, but instead combined Dreambooth with LoRA for efficient adaptation. Originally developed for large language models (LLMs), LoRA has been adapted for use in image generation models, becoming a popular technique: instead of fine-tuning the entire model, which is computationally expensive, LoRA adopts a more efficient approach by freezing the original pre-trained weights W and introducing trainable low-rank matrices into selected layers, focusing only on a residual subset of the model parameters, ΔW.

In Stable Diffusion fine-tuning, LoRA is typically applied to the attention mechanism of the U-Net component of the model, though additional LoRA layers can optionally be trained for the text encoder for improved fidelity to the subject.

The new weight set of the model is computed as:

$$W' = W + \Delta W$$

where:

- W is the original set of weights.
- ΔW represents the residual updates.

LoRA further decomposes the residual update matrix as:

$$\Delta W = AB^T$$

where A and B are low-rank matrices, significantly reducing the number of trainable parameters.

The advantages of using LoRA include:

- Drastic reduction in training time and lower computational requirements.
- Smaller output size: LoRA model weights are only a few MBs in size, nearly 1000 times smaller than the original U-Net model, making them easier to store and share.

- Flexible control over fine-tuning: the contribution of the fine-tuned weights ΔW to the overall model is controlled by a scaling factor α:

  $$W' = W + \alpha \Delta W$$

  $\alpha = 0$ corresponds to retaining the original model weights unchanged, and $\alpha = 1$represents the use of fully fine-tuned weights.
- Multiple adapters can be combined: different LoRAs can be activated simultaneously to fine-tune multiple aspects of the model at inference time.

After training, the LoRA adapter weights can be loaded into the diffusion pipeline and activated with a specific weight.

Training implementation

We performed fine-tuning using the official DreamBooth LoRA training script provided by the Hugging Face Diffusers library [65].

All training and inference were executed in the cloud using Kaggle notebooks with GPU acceleration. The training configuration and hyperparameters used for shot type and character fine-tuning respectively are outlined below.

Shot Type Training Configuration:

```
!accelerate launch train_dreambooth_lora.py \
  --pretrained_model_name_or_path=
  "stable-diffusion-v1-5/stable-diffusion-v1-5" \
  --instance_data_dir=... \  # path to the training dataset
  --output_dir=... \  # path to the output folder
  --instance_prompt="close up of a person" \
  --learning_rate=1e-4 \
  --use_8bit_adam \
  --train_text_encoder \
  --resolution=512 \
  --checkpointing_steps=1000 \
  --lr_scheduler="constant" \
  --lr_warmup_steps=0 \
  --train_batch_size=1 \
  --gradient_accumulation_steps=1 \
  --max_train_steps=3000 \
  # 4000 steps for medium, american and full shot
  --mixed_precision="fp16"
```

Each shot type (e.g., medium shot, american shot, full shot) was trained independently to produce dedicated LoRA weights. Prompts were adapted accordingly to describe the framing: *"extreme close up of a person", "close up of a person", "medium shot picture of a person", "american shot picture of a person", "full body picture of a person", "long shot picture of a person", "extreme long shot view of a person"*.

Character Training Configuration:

```
!accelerate launch train_dreambooth_lora.py \
  --pretrained_model_name_or_path=
  "stable-diffusion-v1-5/stable-diffusion-v1-5" \
  --instance_data_dir=... \  # path to the training dataset
  --output_dir=... \  # path to the output folder
  --class_data_dir=... \ # path to the class regularization dataset
  --instance_prompt="picture of <charactername> man/woman" \
  --class_prompt="picture of a man/woman" \
  --learning_rate=1e-4 \
  --use_8bit_adam \
  --train_text_encoder \
  --resolution=512 \
  --checkpointing_steps=1000 \
  --lr_scheduler="constant" \
  --lr_warmup_steps=0 \
  --num_train_epochs=4 \
  --train_batch_size=1 \
  --gradient_accumulation_steps=1 \
  --max_train_steps=2000 \
  --mixed_precision="fp16" \
  --with_prior_preservation \
  --prior_loss_weight=1.0
```

For character-specific fine-tuning, we employed prior preservation with regularization images from the base class ("man" or "woman") and set the `-prior_loss_weight` to 1.0 to balance memorization and generalization.

Contextually to fine-tuning, we investigated the impact of using specific captions for training images. By default, the `Diffusers` DreamBooth LoRA script uses a single, fixed *instance prompt* applied uniformly to all training images. However, the script can be modified to support individual captions for each training image. Due to mixed findings in the online community and a lack of consensus on best practices, we conducted comparative experiments on selected shot types (medium, american, full shot) and characters. Specifically, we tested:

1. **Single instance prompt:** first, we trained using the default script, with one uniform prompt for all images, such as `"medium shot picture of a person"` or `"picture of petkotomov man"`.
2. **Descriptive captions:** then, we modified the training script to load per-image captions. Each training image was paired with a tailored, detailed caption describing the subject's action / pose and the setting. In this case, the training dataset folder also contains a .txt file for each image (with the same filename) containing its corresponding caption.

These parallel tests were aimed at assessing whether individualized captioning would lead to improvements in generalization and prompt fidelity. The outcomes of this comparison are discussed in Section**??**.

6.2.3 Prompt generation

After obtaining the fine-tuned LoRA weights for both shot types and characters, we developed a structured approach for generating frame-level prompts to guide the storyboard creation process. This method aims to produce visually consistent frames that accurately reflect the intended cinematic composition, narrative context, and character continuity. To achieve this, we designed a prompt template to be used within ChatGPT, which transforms a general scene description—along with optional framing instructions—into a set of detailed prompts suitable for text-to-image generation using Stable Diffusion. We iteratively refined this template through empirical testing, evaluating its impact on three key areas: shot type accuracy, setting consistency, and character representation.

Final prompt template for ChatGPT

Prompt instructions: Given a general scene description, character names, and the number of frames (or, optionally, a detailed sequence of shot types), generate a series of prompts—one per frame—to be used in a storyboard generation pipeline powered by a text-to-image diffusion model. Each prompt should be formatted to accurately reflect the intended shot type, characters involved, and setting.

Example scene description: *"The detective is searching for clues in an abandoned house."*

Optional shot sequence: A numerical list indicating the shot types to be used for each frame: `2, 3, 8, 3, 3, 3, 4, 9, 3, 4, 9, 3, 4, 3, 3, 3, 9`

Shot type definition and mapping:

- 1: Extreme close-up shot (frames the character's face or part of it from the chin up).
- 2: Close-up shot (the human figure is framed from above the shoulders up).
- 3: Medium close-up shot (the human figure is framed from half the torso up).
- 4: Medium shot (the human figure is framed from the waist up).
- 5: American shot (the human figure is framed mid-thigh to head).
- 6: Full body shot (the human figure is framed entirely or almost and occupies more than 2/3 of the frame in height).
- 7: Long shot (the human figure is framed entirely or almost and occupies from 2/3 to 1/3 of the frame in height, surrounded by a larger portion of the environment).
- 8: Extreme long shot (the human figure occupies less than one-third in height or is absent within a wide environment).
- 9: Detailed shot (focus on objects, hands, or close details of an action).

If no shot sequence is provided, include a natural mix of different shot types based on the scene type (e.g., dialogue, fight, adventure, investigation...) and narrative needs.

Prompt format requirement: Each generated prompt must begin with an explicit shot-type phrase:

```
"[Shot type] of ..."
```

Comment: this formulation ensures proper use of the corresponding LoRA weights during image generation.

Character specification:

- Always use the complete character name + class (e.g., `"kaitoyosuke man"`, `"jiyeonmun woman"`) instead of generic terms such as "a man" or "a woman."

Comment: this is essential to correctly trigger the appropriate LoRA weights during inference.

- For each individual frame, only mention the character or characters that are intended to be actually present in the image. .

Framing sensitivity: To ensure correct use of the target shot type, consider their given definition and follow these guidelines:

- Avoid referencing legs or lower body parts in medium to extreme close-up shots.
- For medium close-ups or medium shots, explicitly mention shoulders or hands when needed to prevent the model from mistakenly generate close ups.
- In long and extreme long shots, prioritize environment description and do not include facial expression or specific pose details.
- For medium to long shots, avoid emotional cues or expressions, as they often bias the model toward generating closer compositions.

Comment: this instruction set helps avoiding inaccurate shot type representation.

Clarity in multi-character scenes: When multiple characters appear in the same frame, specify each character's name, position, and action in separate sentences. *Comment: this instruction proved effective to reduce concept confusion and concept omission.*

- Bad Example: "Medium close-up shot of jiyeonmun woman and petkotomov man talking and smiling, with a city street in the background."
- Improved Example: "Medium close-up shot of two people talking: jiyeonmun woman is on the left, while petkotomov man is on the right, smiling warmly as they exchange words, with a city street in the background."

Setting specification: Each prompt must include a clear and brief reference to the background or setting (e.g., "in a dusty library," "inside a car at night") and lighting condition (e.g., "soft daylight", "sunset light") .

Comment: specifying the setting and lighting proved significantly effective to enhance coherence across storyboard frames, and although exact background details may not be preserved, consistency in the general environment is sufficient for storyboarding purposes.

Conclusion

This prompt template served as an essential bridge between high-level narrative intent and frame-level image generation. Through iterative. empirical refinement, we were able to define a prompt generation pipeline that balances expressiveness, visual fidelity, shot-type accuracy and multi-character control—enabling ChatGPT to function as a powerful support tool for semantic-to-visual translation within our storyboard generation framework.

Despite the robustness of the final prompt structure, certain scenes still required minor manual adjustments during the generation phase. These included clarifying ambiguous elements, refining character placement, or simplifying overly complex descriptions. Nevertheless, we empirically consolidated the most recurrent and impactful instructions into the final template, aiming to minimize inconsistencies and make the prompt design process as complete and reproducible as possible.

6.2.4 Storyboard generation system

With both the fine-tuned LoRA weights and the prompt generation pipeline in place, our next objective was to build an interactive system to support the process of AI-assisted storyboard creation. The system needed to be both flexible and interactive, guiding step-by-step through the storyboard generation process while still offering control to make manual adjustments when needed.

All necessary code was developed within Kaggle notebooks, which currently serve as the front-end interface for the storyboard generation system. The main notebook consolidates all essential components—including prompt input, model selection, LoRA activation, negative prompt handling, inference configuration, and output management—into a unified and accessible workflow.

To start the generation process, the user needs to specify the sequence of prompts (generated either with ChatGPT through our proposed template or manually), the Stable Diffusion checkpoint to be used, the destination directory for saving outputs, and the file paths for both shot type LoRA weights and character-specific LoRA weights. This modular structure allows the user to flexibly swap different models or weights.

Given our focus on generating high-quality, realistic characters, we selected the Realistic Vision v5.1 model for inference, available via HuggingFace at `https://huggingface.co/SG161222/Realistic_Vision_V5.1_noVAE`. This checkpoint, based on Stable Diffusion 1.5, has been fine-tuned to produce photorealistic images, and aligns well with our use case involving visually consistent and cinematic outputs.

For inference, we used a configuration of `num_inference_steps` = 50

and `guidance_scale` = 7.5. Negative prompts were also carefully crafted and dynamically selected to improve adherence to the intended shot type: while we started from the base negative prompt suggested in the model card of Realistic Vision v5.1, we expanded and customized it per shot type. For instance:

- For medium shots, we included in the negative prompt terms such as "legs, feet, lower body, fullbody" to discourage the model from generating wider compositions.
- For full shots, we included keywords such as "cropped, out of frame, cut-off feet, close-up, missing feet, missing head" to promote complete subject visibility within the frame.

The code automatically selects the appropriate negative prompt to use, depending on the shot type specified in the prompt.

To automate LoRA activation, we implemented specific functions that parse the textual prompt, detect the specified shot type, and automatically load the corresponding LoRA weight into the inference pipeline. Similarly, character names are parsed from the prompt to activate one or more character LoRA weights. If multiple characters are detected, the system applies them simultaneously with appropriate balancing. Moreover, LoRA influence is adjusted based on the expected framing: for closer shots (e.g., close-up or medium close-up), character LoRA weights are increased to ensure better facial consistency, while for longer shots, the weights are reduced slightly to avoid over-constraint and promote flexibility in pose and composition.

For each prompt, the system generates a batch of candidate images, with the number of outputs customizable by the user. These candidates are displayed side-by-side in a horizontal layout within the notebook interface, allowing for easy visual comparison. The user is then prompted to select their preferred image, request a new batch, or optionally refine the prompt before re-generating. This design establishes an interactive feedback loop that supports creative control and iterative refinement throughout the storyboard generation process.

Once the user selects a preferred image, it is saved to the specified output directory using sequential naming (shot_0.png, shot_1.png, ...). This process continues iteratively until all prompts in the sequence have been processed. At the end of the session, the system compiles the selected frames into a final storyboard grid, offering a clear visual overview of the entire scene progression.

6.2.5 Inpainting and storyboard refinement

As previously discussed, one of the persistent challenges in text-to-image generative models is the accurate and consistent depiction of multiple distinct characters within the same frame.

When using LoRA-based personalization, the standard method for incorporating multiple fine-tuned weights involves merging them into a single inference pass. However, this often leads to undesirable effects such as feature blending between characters or the complete omission of one character's attributes. Moreover, preserving the distinct visual identity of each character—while simultaneously coordinating their poses, spatial arrangement, and interactions within a coherent scene—adds an additional layer of complexity to the generation process.

To address these shortcomings, we integrated an interactive inpainting refinement step as an optional final stage in the storyboard generation framework. This module enables users to selectively correct or modify specific regions of a generated image—particularly in cases where character fidelity has been compromised—by leveraging a guided masking process combined with a Stable Diffusion-based inpainting pipeline.

The inpainting system was implemented using the `StableDiffusionInpaintPipeline` from the `Diffusers` library. The pipeline was initialized with a specialized inpainting checkpoint (`krnl/stable-diffusion-v15-inpainting`), and all character-specific LoRA adapters were preloaded to allow for dynamic activation based on the prompt content during inference.

The inpainting process begins with the user selecting the image to be edited. To enable localized refinement, we incorporated a flexible and interactive mask generation module based on the Segment Anything Model (SAM) [66]. This component allows the user to define the inpainting region through two alternative methods:

- **Point-based mask generation:** The user selects a single point on the image, providing coordinates x, y,

  $$x \in [0, 511], \quad y \in [0, 511]$$

  that falls within the region of the character to be edited. This point coordinates are passed to SAM, which then generates a set of three candidate segmentation masks based on the image content around the selected point. The system presents these masks to the user for visual selection. Once a mask is chosen, it can be optionally refined using classical image processing operations—such as Gaussian blurring, dilation, and morphological closing—to avoid harsh borders and produce a clean binary mask suitable for inpainting. This method is fast and user-friendly, although its precision may vary depending on the complexity of the image content around the selected point: if the selected point falls on an ambiguous or overlapping region, the resulting segmentation may not align perfectly with the desired character silhouette and generation may result in failure cases, where the character is simply removed rather than regenerated—despite the correct activation of the corresponding LoRA.

- **Bounding box-based mask creation:** As an alternative, the user can manually define a bounding box by specifying the coordinates of the top-left and bottom-right corners. A rectangular binary mask is then generated to cover the specified region. This method offers greater control and precision, especially in cases where automatic segmentation does not yield satisfactory results or where character boundaries are visually complex or occluded. For instance, different characters' silohuettes may not align, leading point-based mask generation to poor results: in this case, specifying the broader region is more beneficial. This is because inpainting models tend to prioritize global visual coherence over localized reconstruction. If regenerating a small masked area disrupts the broader scene context, the model may respond by erasing or blending it to preserve overall harmony. Additionally, identity conditioning via LoRA tends to be more effective when applied to the full image rather than to isolated regions, where spatial attention may be weaker or diffused, therefore selecting a wider area for inpainting improves the output quality.

Once a valid mask has been finalized, it is combined with the original image and passed to the inpainting pipeline for refinement. The pipeline is configured with a pre-trained inpainting model and dynamically enhanced with character-specific LoRA weights based on the input prompt. As in the earlier stages of the system, the user is presented with a batch of refined image options to choose from. This process can be repeated iteratively until all desired corrections are completed. Once refinement is finalized, the updated set of images is saved to the specified output directory.

This iterative, user-in-the-loop process empowers the user to maintain fine-grained control over the visual narrative, while leveraging the generative capabilities of diffusion models. Importantly, inpainting only affects the masked region of the image, preserving the surrounding context and maintaining overall frame coherence, and, even though this module is focused on character correction in our specific application, it can be used as is also for adding a new character or objects to the image.

Combined with careful prompt crafting, this inpainting module offers an efficient solution to one of the core challenges in text-to-image generation, multi-character scenes. By combining automatic segmentation, prompt-driven LoRA activation, and interactive inpainting, the proposed system enables high-fidelity storyboard refinement while maintaining narrative consistency and user control.

6.3 Experiments and results

In this section, we present the results obtained from fine-tuning LoRA adapters for eight distinct shot types and seven original characters, and we analyze the performance of the resulting models under various prompt and framing conditions. We also present a selection of storyboards generated using our framework.

6.3.1 Shot type fine-tuning

We trained individual LoRA models for the following eight shot types, each corresponding to a standardized cinematic composition:

- **Extreme Close-Up**: The human figure is framed from the chin up, focusing exclusively on facial details.
- **Close-Up**: The subject is framed from just above the shoulders, capturing facial expressions and head positioning.
- **Medium Close-Up**: The subject is framed from mid-torso upward, providing a balance between facial detail and upper-body posture.
- **Medium Shot**: The upper half of the body is shown, typically from the waist up, allowing for basic gesture visibility.
- **American Shot**: The subject is framed from mid-thigh to the head, commonly used in dynamic or action-oriented contexts.
- **Full Shot**: The entire human figure is framed, occupying more than two-thirds of the frame height.
- **Long Shot**: The full human figure is shown while occupying between two-thirds and one-third of the image height, offering broader context.
- **Extreme Long Shot**: The subject occupies less than one-third of the frame height or may be absent altogether, with focus placed on the surrounding environment.

As previously discussed, each shot type was initially fine-tuned using a single instance prompt without detailed per-image captions. This approach proved effective for certain shot types —particularly *extreme close-up*, *close-up*, *medium close-up*, *long shot*, and *extreme long shot*— which produced highly consistent results in terms of framing and subject positioning.

However, other shot types, namely *medium*, *american*, and *full shot*, presented recurring challenges. With *medium* and *american shots*, the most common issue was the model generating full-body compositions when prompts included action-related verbs such as “walking,” “standing,” or “posing,” or even simply referenced more elements of the background setting. Conversely, prompts that emphasized facial expressions or emotional cues tended to bias the model toward *medium close-up shots*. Similarly, the *full shot* category occasionally failed to maintain the expected framing, resulting in shots longer or shorter than expected, depending on the prompt.

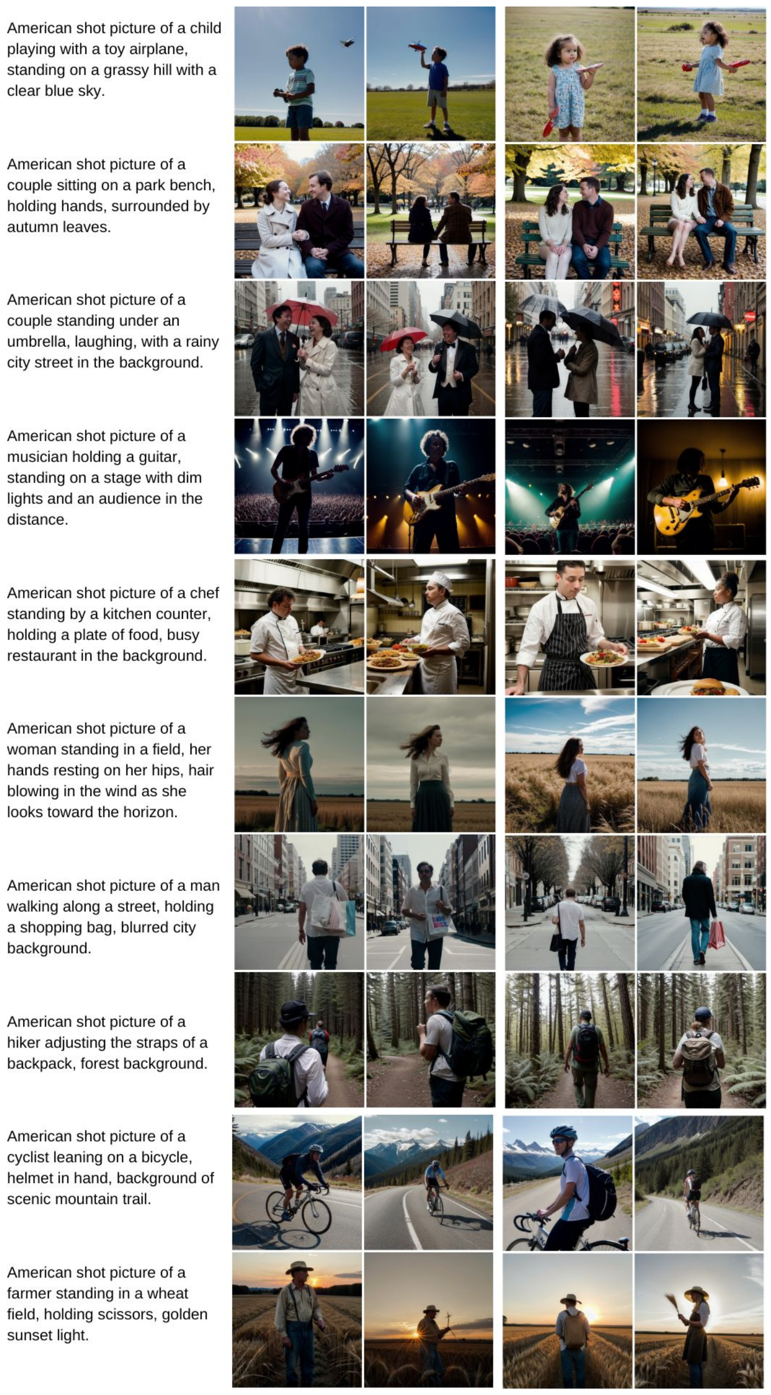
To address these limitations, we conducted a second training round for the problematic shot types, this time using detailed, manually curated captions for each training image. The goal was to reinforce shot-type-specific spatial priors and improve the alignment between the training data and the model’s output during inference.

The following figures present a comparison across twenty prompts, highlighting the impact of using captions for the three most challenging shot types, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8. In the figures, the two left columns show generations using LoRA models trained with captions, while the two right columns display results from training without captions.

Overall, the comparative results demonstrate that both captioned and non-captioned training approaches are capable of producing high-quality images and generally satisfactory outcomes. In practice, the effectiveness of each method often depends on the specific prompt. Inaccuracies in shot-type framing appear in both settings, particularly in cases involving challenging prompts—such as those describing actions like “walking” or “standing”. These actions tend to bias the model toward wider framings, regardless of whether captions were used during training.

While captioned prompts tend to yield slightly more consistent and reliable framing, especially for ambiguous or edge cases, the difference is often subtle. In many instances, non-captioned prompts perform comparably well, with generated images matching the shot type as accurately as their captioned counterparts. This suggests that the LoRA weights—once properly trained—are largely responsible for controlling framing behavior, even without detailed guidance from captions.

Captions, therefore, offer a modest but not hyper decisive improvement in framing fidelity. They encourage better adherence to the intended composition, but their



American shot picture of a child playing with a toy airplane, standing on a grassy hill with a clear blue sky.

American shot picture of a couple sitting on a park bench, holding hands, surrounded by autumn leaves.

American shot picture of a couple standing under an umbrella, laughing, with a rainy city street in the background.

American shot picture of a musician holding a guitar, standing on a stage with dim lights and an audience in the distance.

American shot picture of a chef standing by a kitchen counter, holding a plate of food, busy restaurant in the background.

American shot picture of a woman standing in a field, her hands resting on her hips, hair blowing in the wind as she looks toward the horizon.

American shot picture of a man walking along a street, holding a shopping bag, blurred city background.

American shot picture of a hiker adjusting the straps of a backpack, forest background.

American shot picture of a cyclist leaning on a bicycle, helmet in hand, background of scenic mountain trail.

American shot picture of a farmer standing in a wheat field, holding scissors, golden sunset light.

Figure 6.3: Impact of captions on american shot generations. Left two columns: with captions; right two columns: without captions.

American shot picture of a musician standing in a dimly lit jazz club, one hand resting on their guitar, waiting for their turn to play.

American shot picture of a man holding a steaming cup of coffee with both hands, standing on a balcony, lost in thought as the city wakes up behind him.

American shot picture of a woman standing at the edge of a pier, hands in her coat pockets, gazing out at the calm water with a contemplative expression.

American shot picture of a woman holding a yoga mat, standing outside a fitness studio, blurry urban background.

American shot picture of a woman standing near a market stall, holding a basket of fruit, lively marketplace background.

American shot picture of a man standing on a pier, holding a fishing rod.

American shot picture of a couple standing close together at a train station, their hands nearly touching, as if hesitating before saying goodbye.

American shot picture of a traveler standing on a dirt road, adjusting the straps of their backpack, ready to continue their journey.

American shot picture of a young woman standing in a sunlit bookstore, gently flipping through the pages of an old novel, completely absorbed in the moment.

American shot picture of a journalist standing in front of a camera, holding a microphone, poised and ready to deliver a live report.

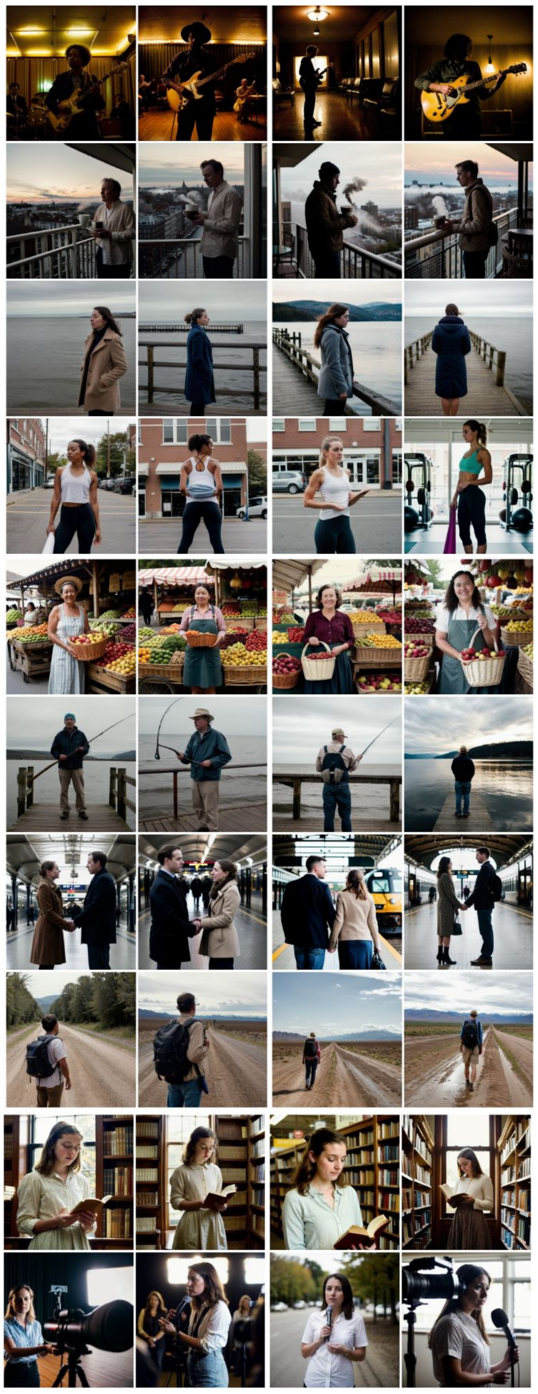


Figure 6.4: Impact of captions on american shot generations. Left two columns: with captions; right two columns: without captions.

Medium shot picture of a woman adjusting the strap of her bag, standing at a train station platform, her expression a mix of excitement and uncertainty.

Medium shot picture of a man gesturing with one hand while talking, engaged in conversation, blurred urban background.

Medium shot picture of a woman standing in front of a train schedule board, slightly tilting her head as she reads.

Medium shot picture of a woman sleeping deeply on her pillow, serene expression.

Medium shot picture of a man walking through a marketplace, holding a paper bag, glancing at the vendor stalls.

Medium shot picture of a man walking through a train station, suitcase in hand, checking his phone.

Medium shot picture of a man standing in a library, holding an open book in one hand while his other hand rests on a shelf, lost in thought.

Medium shot picture of a man walking along a waterfront promenade, hands in pockets, looking at the horizon.

Medium shot picture of a woman standing at a crosswalk, looking at the pedestrian light, hands resting on her bag.

Medium shot picture of a woman standing in front of a store display, touching her chin as she considers a purchase.

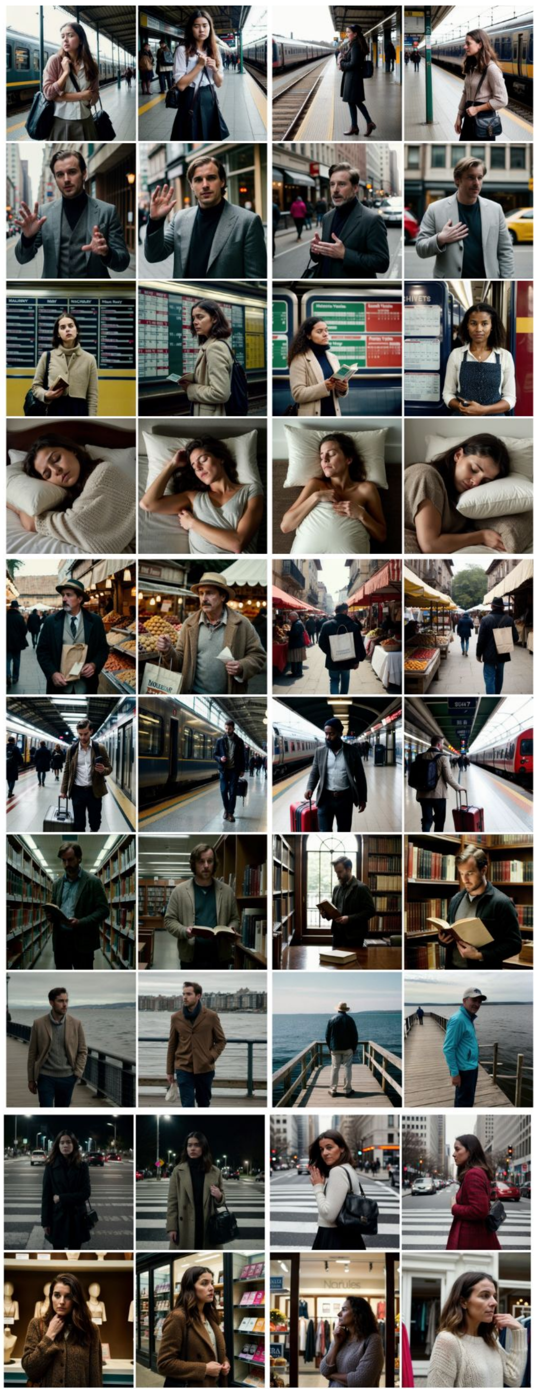
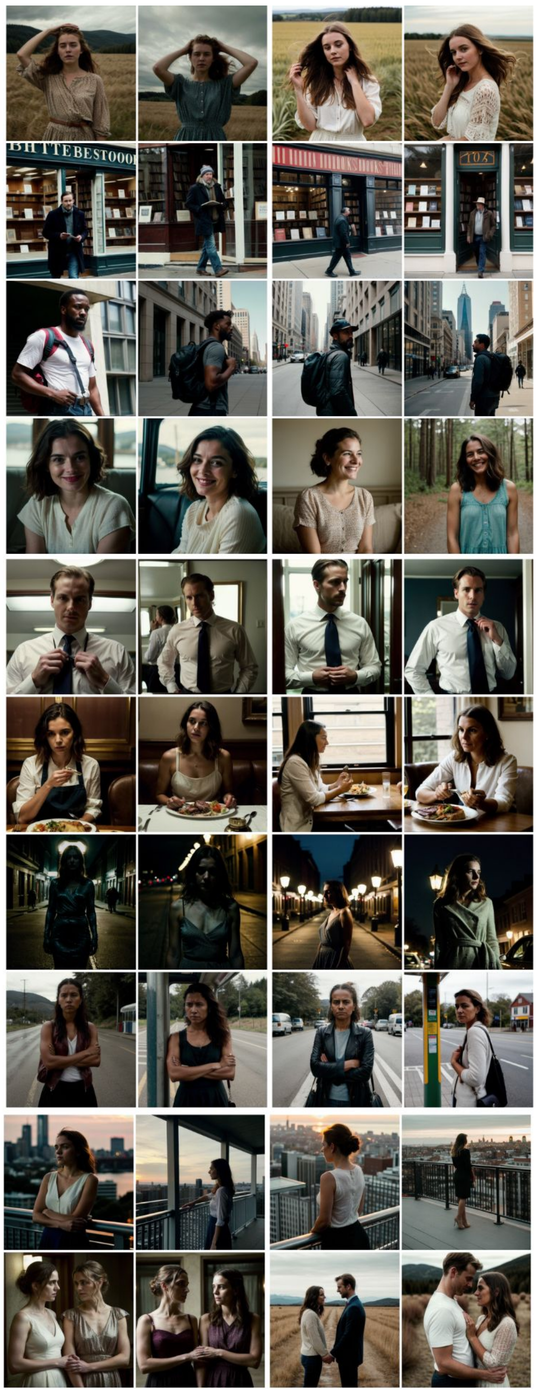


Figure 6.5: Impact of captions on medium shot generations. Left two columns: with captions; right two columns: without captions.



Medium shot picture of a young woman standing in a field, her hands to her head as the wind plays with her hair.

Medium shot picture of a man walking past a bookstore, briefly glancing at the books in the display window.

Medium shot picture of a man stepping out of a building, adjusting the strap of his backpack, city background.

Medium shot picture of a woman tilting her head slightly, smiling as she listens to someone off-camera.

Medium shot picture of a man adjusting his tie in front of a mirror, focused expression, soft indoor lighting.

Medium shot picture of a woman sitting at a dining table, cutting into a meal with a knife and fork, restaurant setting.

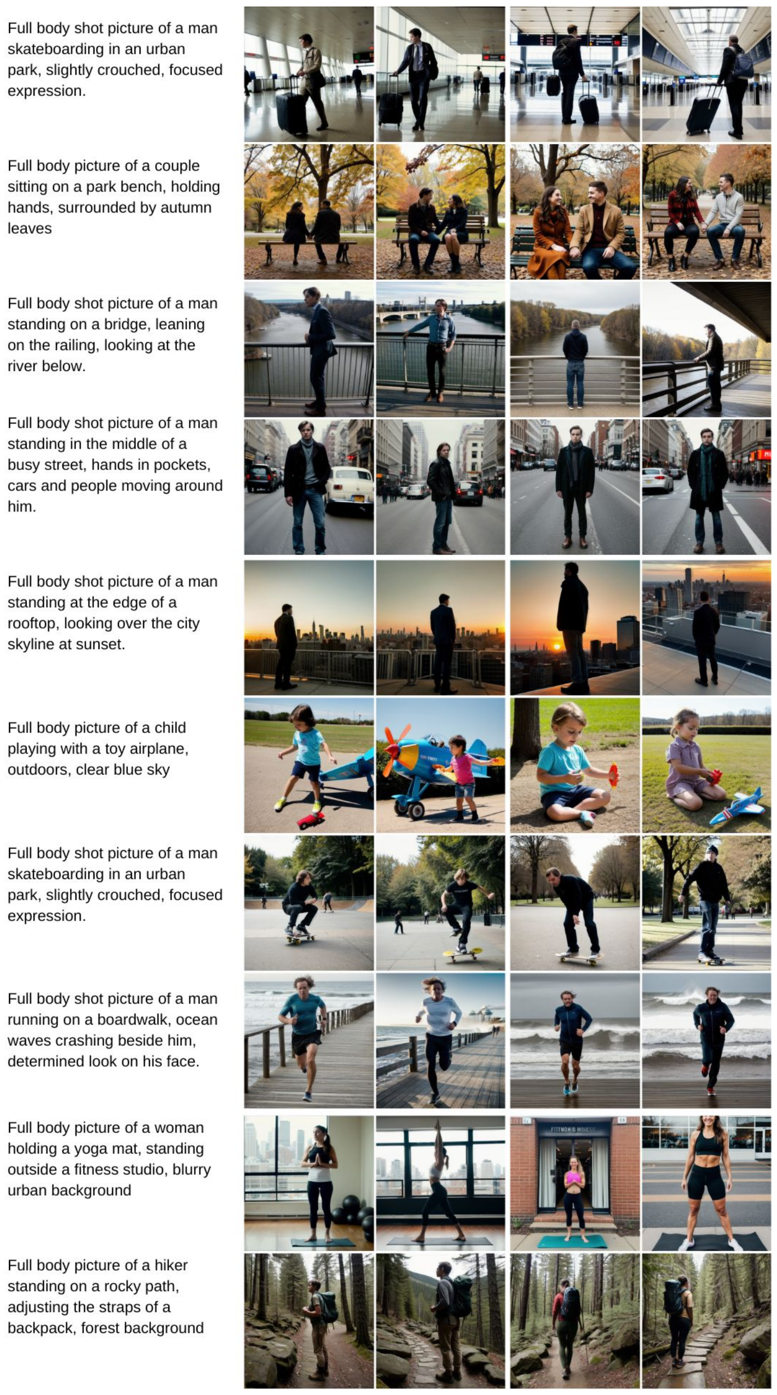
Medium shot picture of a woman standing under a streetlamp at night, arms wrapped around herself, deep in thought.

Medium shot picture of a woman standing at a bus stop, arms crossed, looking down the road with a waiting expression.

Medium shot picture of a woman standing on a balcony, arms resting on the railing, gazing at the city skyline as the sun sets.

Medium shot picture of a couple standing close together, their hands intertwined, both looking in the same direction with expressions of quiet anticipation.

Figure 6.6: Impact of captions on medium shot generations. Left two columns: with captions; right two columns: without captions.



Full body shot picture of a man skateboarding in an urban park, slightly crouched, focused expression.

Full body picture of a couple sitting on a park bench, holding hands, surrounded by autumn leaves

Full body shot picture of a man standing on a bridge, leaning on the railing, looking at the river below.

Full body shot picture of a man standing in the middle of a busy street, hands in pockets, cars and people moving around him.

Full body shot picture of a man standing at the edge of a rooftop, looking over the city skyline at sunset.

Full body picture of a child playing with a toy airplane, outdoors, clear blue sky

Full body shot picture of a man skateboarding in an urban park, slightly crouched, focused expression.

Full body shot picture of a man running on a boardwalk, ocean waves crashing beside him, determined look on his face.

Full body picture of a woman holding a yoga mat, standing outside a fitness studio, blurry urban background

Full body picture of a hiker standing on a rocky path, adjusting the straps of a backpack, forest background

Figure 6.7: Impact of captions on full shot generations. Left two columns: with captions; right two columns: without captions.

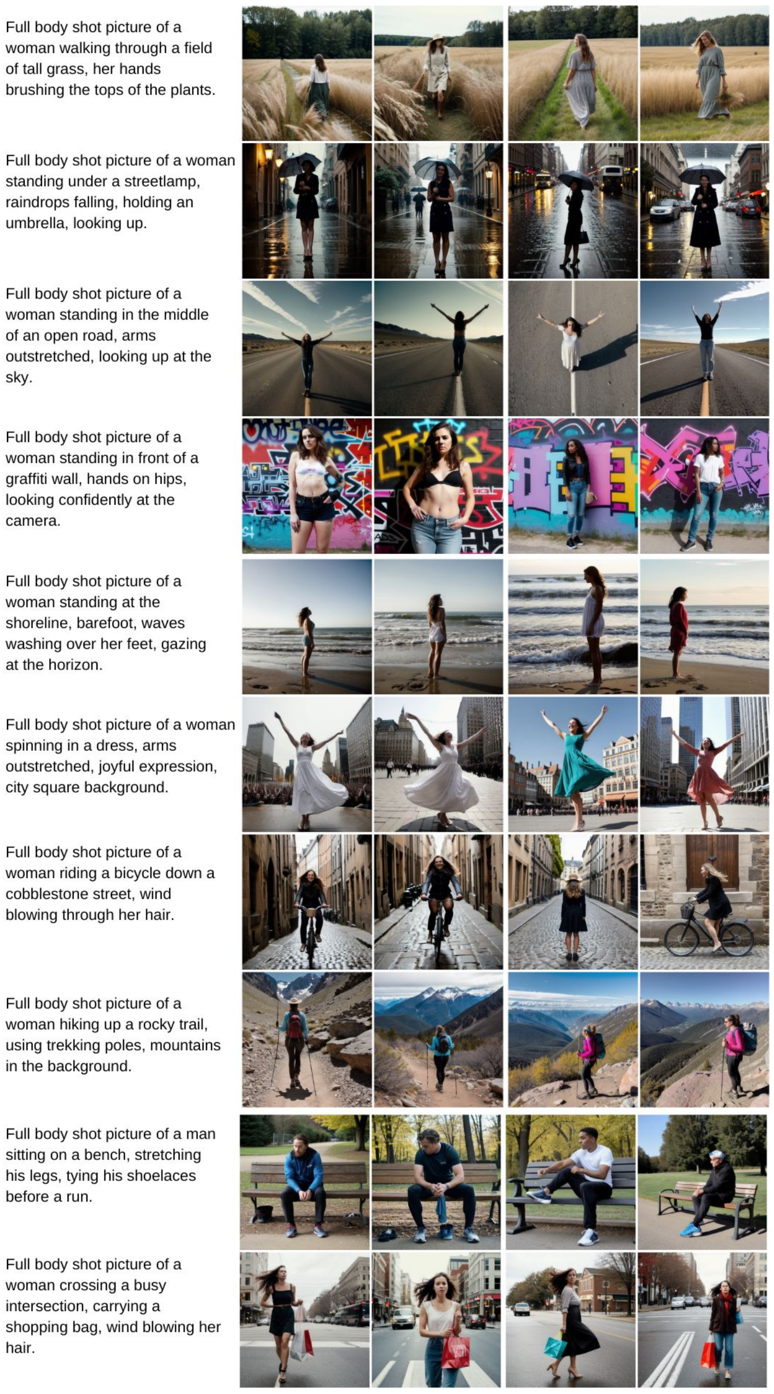

Full body shot picture of a woman walking through a field of tall grass, her hands brushing the tops of the plants.
Full body shot picture of a woman standing under a streetlamp, raindrops falling, holding an umbrella, looking up.
Full body shot picture of a woman standing in the middle of an open road, arms outstretched, looking up at the sky.
Full body shot picture of a woman standing in front of a graffiti wall, hands on hips, looking confidently at the camera.
Full body shot picture of a woman standing at the shoreline, barefoot, waves washing over her feet, gazing at the horizon.
Full body shot picture of a woman spinning in a dress, arms outstretched, joyful expression, city square background.
Full body shot picture of a woman riding a bicycle down a cobblestone street, wind blowing through her hair.
Full body shot picture of a woman hiking up a rocky trail, using trekking poles, mountains in the background.
Full body shot picture of a man sitting on a bench, stretching his legs, tying his shoelaces before a run.
Full body shot picture of a woman crossing a busy intersection, carrying a shopping bag, wind blowing her hair.

Figure 6.8: Impact of captions on full shot generations. Left two columns: with captions; right two columns: without captions.

influence remains conditional —shaped by the clarity of the shot-type LoRA, the specific actions or expressions described in the prompt, and the relative emphasis placed on the subject versus the environment. While not essential, captions can serve as a useful enhancement when higher precision is required, particularly for more nuanced or constrained scenes.

Figures 6.9, 6.10, showcase generations and their corresponding prompts for each trained shot type. For medium, american, and full shot, the model trained with captions was used.

6.3.2 Character fine-tuning

For the characters, we aimed for diversity in age, ethnicity, and physical features. Figure 6.11 presents a selection of generations for each trained character.

Initially, we trained character LoRAs using only the general instance prompt. However, we later tested and compared the results obtained by incorporating captions during training. Figures 6.12, 6.13 and 6.14 illustrate this comparison across a set of various prompts.

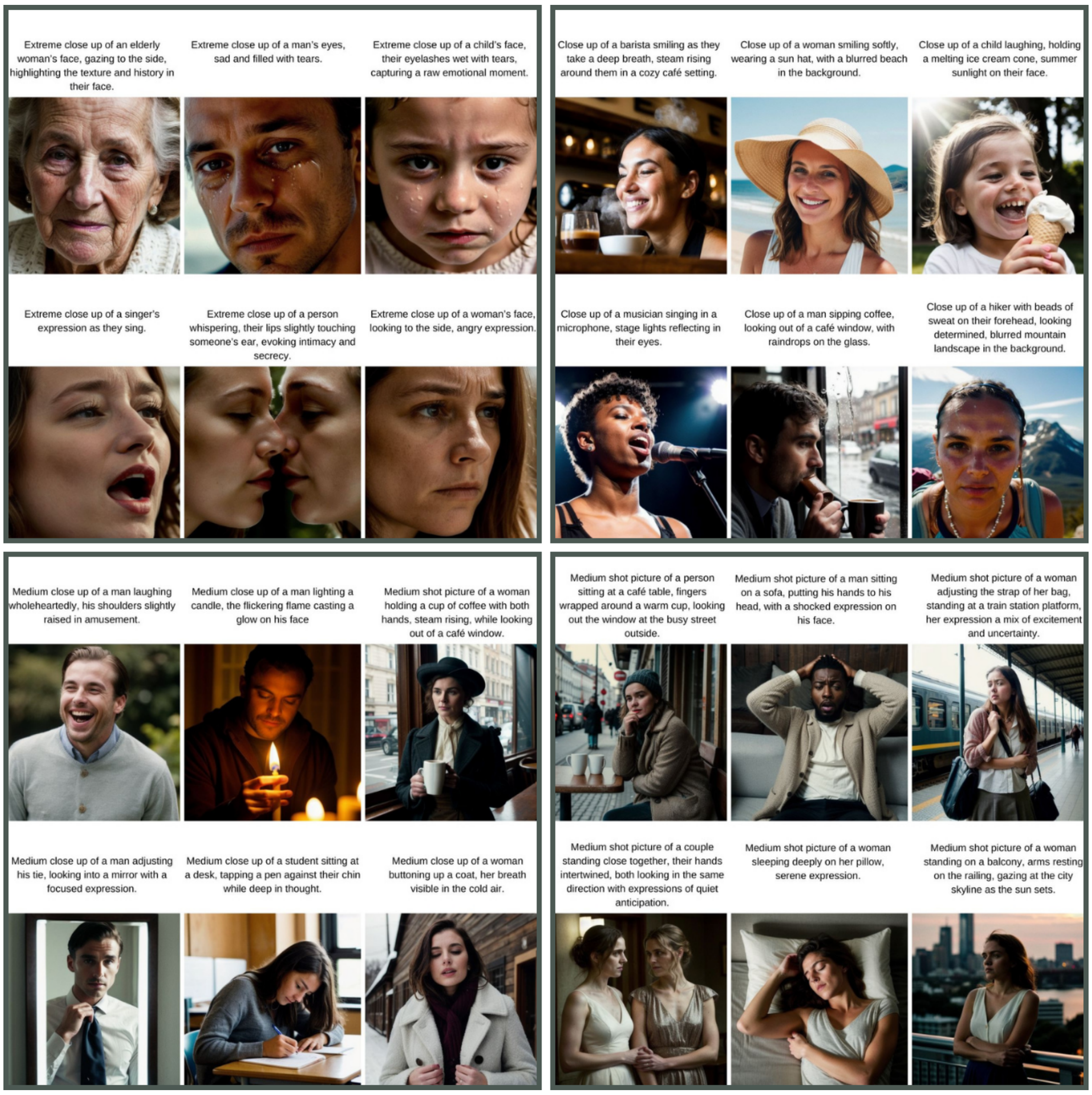
In the figures, for each prompt, the two left images were generated by the captioned model, while the two right images were generated by the non-captioned model.

Figures show that, while both captioned and non-captioned approaches are capable of generating high-quality and visually coherent character images, captions contribute to better alignment between the intended prompt and the generated output.

Prompts involving fine-grained emotional cues (such as surprise, sorrow, or joy), specific object interactions (like holding a book or camera), or spatial positioning (e.g., sitting cross-legged or leaning against a windowsill) are more consistently and accurately rendered when supported by the use of descriptive captions during training, which help guiding the model toward more precise results.

An area where captions offer value is pose accuracy and gesture fidelity. The LoRA trained with captions maintains intended body positioning and limb articulation — elements that often drift with the non-captioned LoRA, where the model might default to more generic or comfortable poses. This is particularly important for character-driven outputs that rely on subtle emotional storytelling or realistic scene dynamics.

Scene composition and object grounding also benefit from captioned training images.


Extreme close up of an elderly woman's face, gazing to the side, highlighting the texture and history in their face.

Extreme close up of a man's eyes, sad and filled with tears.

Extreme close up of a child's face, their eyelashes wet with tears, capturing a raw emotional moment.

Extreme close up of a singer's expression as they sing.

Extreme close up of a person whispering, their lips slightly touching someone's ear, evoking intimacy and secrecy.

Extreme close up of a woman's face, looking to the side, angry expression.

Close up of a barista smiling as they take a deep breath, steam rising around them in a cozy café setting.

Close up of a woman smiling softly, wearing a sun hat, with a blurred beach in the background.

Close up of a child laughing, holding a melting ice cream cone, summer sunlight on their face.

Close up of a musician singing in a microphone, stage lights reflecting in their eyes.

Close up of a man sipping coffee, looking out of a café window, with raindrops on the glass.

Close up of a hiker with beads of sweat on their forehead, looking determined, blurred mountain landscape in the background.

Medium close up of a man laughing wholeheartedly, his shoulders slightly raised in amusement.

Medium close up of a man lighting a candle, the flickering flame casting a glow on his face

Medium shot picture of a woman holding a cup of coffee with both hands, steam rising, while looking out of a café window.

Medium shot picture of a person sitting at a café table, fingers wrapped around a warm cup, looking out the window at the busy street outside.

Medium shot picture of a man sitting on a sofa, putting his hands to his head, with a shocked expression on his face.

Medium shot picture of a woman adjusting the strap of her bag, standing at a train station platform, her expression a mix of excitement and uncertainty.

Medium close up of a man adjusting his tie, looking into a mirror with a focused expression.

Medium close up of a student sitting at a desk, tapping a pen against their chin while deep in thought.

Medium close up of a woman buttoning up a coat, her breath visible in the cold air.

Medium shot picture of a couple standing close together, their hands intertwined, both looking in the same direction with expressions of quiet anticipation.

Medium shot picture of a woman sleeping deeply on her pillow, serene expression.

Medium shot picture of a woman standing on a balcony, arms resting on the railing, gazing at the city skyline as the sun sets.

Figure 6.9: Extreme Close Up, Close Up, Medium Close Up and Medium Shot images generated with the LoRA weights.

Objects referenced in the prompt (like flowers, books, or musical instruments) are more likely to be present and correctly placed in the scene. Without captions, these elements sometimes disappear entirely or appear in ambiguous ways, suggesting that object presence is more fragile in caption-free setups.

Overall, the core visual identity, clothing style, and aesthetic of the characters remain remarkably stable across both conditions. This indicates that LoRA weights trained specifically for character fidelity are sufficiently strong to preserve identity

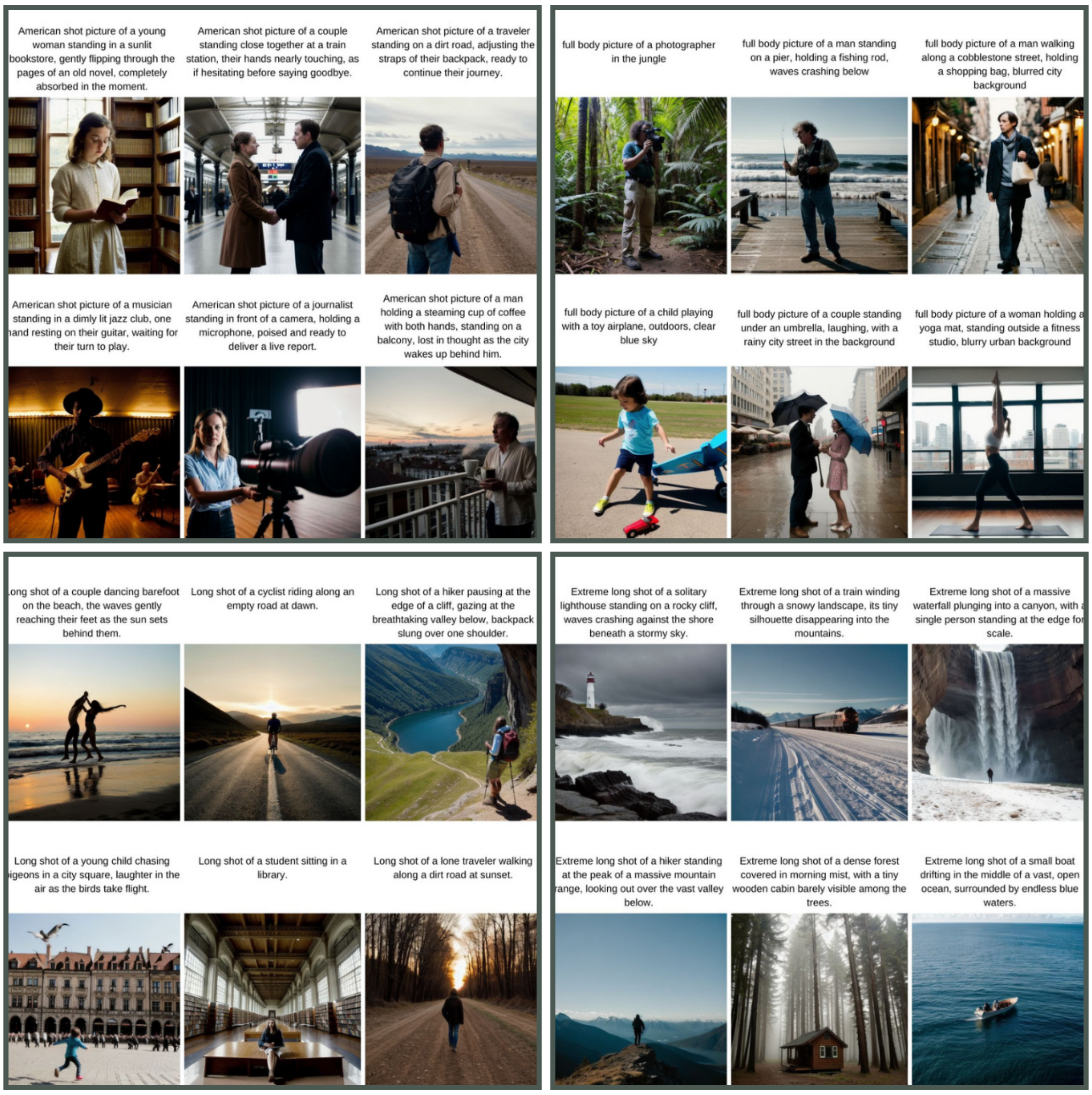

American shot picture of a young woman standing in a sunlit bookstore, gently flipping through the pages of an old novel, completely absorbed in the moment.
American shot picture of a couple standing close together at a train station, their hands nearly touching, as if hesitating before saying goodbye.
American shot picture of a traveler standing on a dirt road, adjusting the straps of their backpack, ready to continue their journey.
full body picture of a photographer in the jungle
full body picture of a man standing on a pier, holding a fishing rod, waves crashing below
full body picture of a man walking along a cobblestone street, holding a shopping bag, blurred city background
American shot picture of a musician standing in a dimly lit jazz club, one hand resting on their guitar, waiting for their turn to play.
American shot picture of a journalist standing in front of a camera, holding a microphone, poised and ready to deliver a live report.
American shot picture of a man holding a steaming cup of coffee with both hands, standing on a balcony, lost in thought as the city wakes up behind him.
full body picture of a child playing with a toy airplane, outdoors, clear blue sky
full body picture of a couple standing under an umbrella, laughing, with a rainy city street in the background
full body picture of a woman holding a yoga mat, standing outside a fitness studio, blurry urban background
.ong shot of a couple dancing barefoot on the beach, the waves gently reaching their feet as the sun sets behind them.
Long shot of a cyclist riding along an empty road at dawn.
Long shot of a hiker pausing at the edge of a cliff, gazing at the breathtaking valley below, backpack slung over one shoulder.
Extreme long shot of a solitary lighthouse standing on a rocky cliff, waves crashing against the shore beneath a stormy sky.
Extreme long shot of a train winding through a snowy landscape, its tiny silhouette disappearing into the mountains.
Extreme long shot of a massive waterfall plunging into a canyon, with a single person standing at the edge for scale.
Long shot of a young child chasing pigeons in a city square, laughter in the air as the birds take flight.
Long shot of a student sitting in a library.
Long shot of a lone traveler walking along a dirt road at sunset.
Extreme long shot of a hiker standing at the peak of a massive mountain range, looking out over the vast valley below.
Extreme long shot of a dense forest covered in morning mist, with a tiny wooden cabin barely visible among the trees.
Extreme long shot of a small boat drifting in the middle of a vast, open ocean, surrounded by endless blue waters.

Figure 6.10: American Shot, Full Shot, Long Shot and Extreme Long Shot images generated with the LoRA weights.

features even in the absence of caption guidance. This underscores an important distinction: while captions can fine-tune the content of the scene, they are not the primary driver of character coherence or visual quality.

Interestingly, in a few cases, non-captioned generations show a bit more compositional creativity, offering alternate camera angles, framing, or stylized takes that deviate slightly from the expected outcome. While this can lead to minor inaccuracies, it also reflects the model's capacity to generalize and interpret prompts with

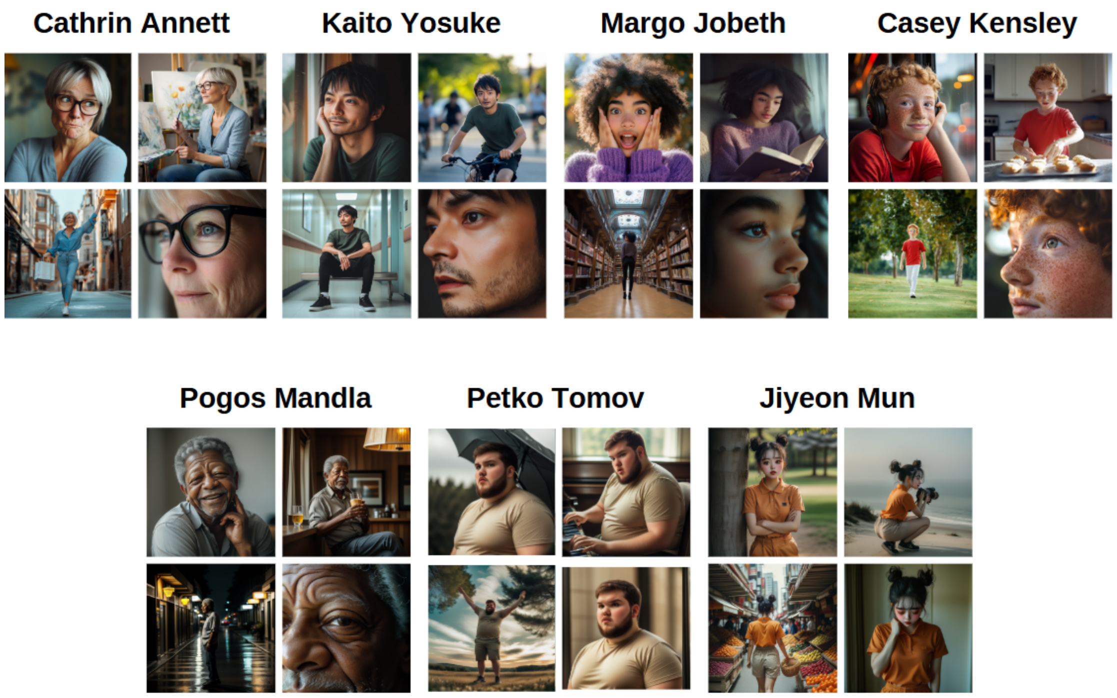


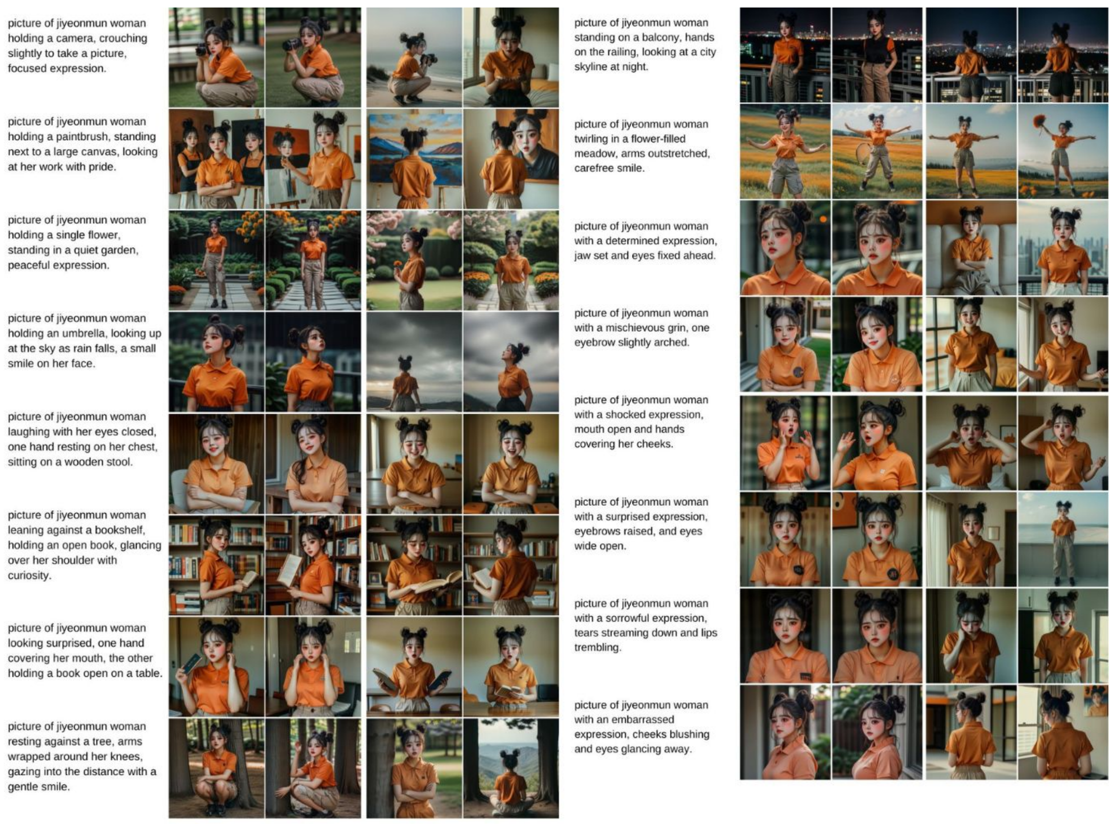
Figure 6.11: Enter Caption

some flexibility — which may be more desirable.

In summary, captions function as a powerful tool for enhancing precision, fidelity, and alignment in character image generation, especially when prompts are action-rich or emotionally nuanced. However, they are not strictly necessary for maintaining character integrity or generating visually pleasing results, particularly when strong character-specific LoRA models are used. Therefore, captions are a valuable tool for interpretability and control, but not a strict prerequisite for quality.

6.3.3 Full storyboard generation

Following model training, we conducted a comprehensive evaluation of the full storyboard generation system, testing it across a diverse range of cinematic scenes. These included dialogue, farewell, investigation, and dream scenes — each designed to test the system's ability to capture distinct narrative tones, emotional beats, and compositional dynamics.


picture of jiyeonmun woman holding a camera, crouching slightly to take a picture, focused expression.

picture of jiyeonmun woman holding a paintbrush, standing next to a large canvas, looking at her work with pride.

picture of jiyeonmun woman holding a single flower, standing in a quiet garden, peaceful expression.

picture of jiyeonmun woman holding an umbrella, looking up at the sky as rain falls, a small smile on her face.

picture of jiyeonmun woman laughing with her eyes closed, one hand resting on her chest, sitting on a wooden stool.

picture of jiyeonmun woman leaning against a bookshelf, holding an open book, glancing over her shoulder with curiosity.

picture of jiyeonmun woman looking surprised, one hand covering her mouth, the other holding a book open on a table.

picture of jiyeonmun woman resting against a tree, arms wrapped around her knees, gazing into the distance with a gentle smile.

picture of jiyeonmun woman standing on a balcony, hands on the railing, looking at a city skyline at night.

picture of jiyeonmun woman twirling in a flower-filled meadow, arms outstretched, carefree smile.

picture of jiyeonmun woman with a determined expression, jaw set and eyes fixed ahead.

picture of jiyeonmun woman with a mischievous grin, one eyebrow slightly arched.

picture of jiyeonmun woman with a shocked expression, mouth open and hands covering her cheeks.

picture of jiyeonmun woman with a surprised expression, eyebrows raised, and eyes wide open.

picture of jiyeonmun woman with a sorrowful expression, tears streaming down and lips trembling.

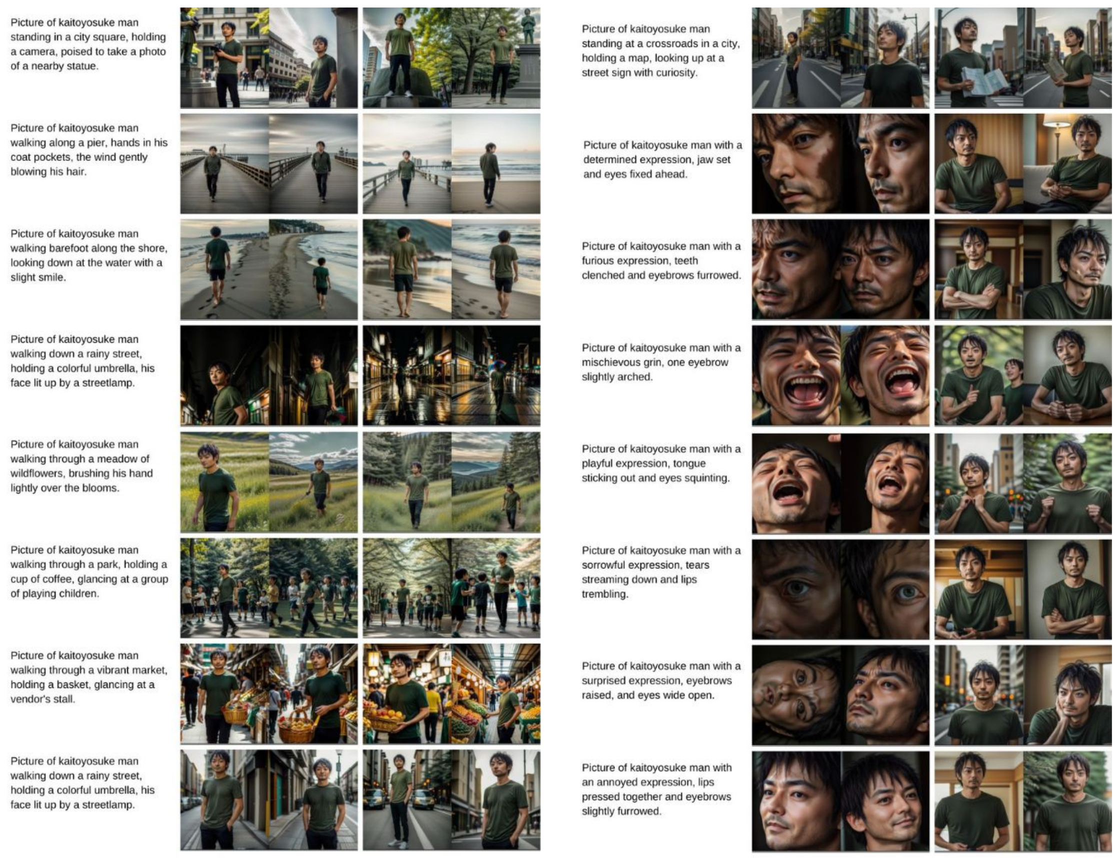
picture of jiyeonmun woman with an embarrassed expression, cheeks blushing and eyes glancing away.

Figure 6.12: Comparison of jiyeonmun woman character generations with(left) and without(right) captions.

To assess the effectiveness of our storyboard generation pipeline, we conducted a comparative analysis between our finetuned model — which leverages LoRA weights for both shot type control and character identity preservation — and the baseline Stable Diffusion 1.5 model without any finetuning applied.

Figures 6.15, 6.16, 6.17, 6.18, 6.19, 6.20 present the complete storyboards and corresponding prompts generated by both models across all scenes. The storyboards from the finetuned model have already gone through the inpainting step, the impact of which is shown in figure 6.21.

In the baseline setting, prompts were preserved except for the specific subject identities: we replaced the names of the characters with generic subject references (e.g., “man,” “woman”).



Picture of kaitoyosuke man standing in a city square, holding a camera, poised to take a photo of a nearby statue.

Picture of kaitoyosuke man walking along a pier, hands in his coat pockets, the wind gently blowing his hair.

Picture of kaitoyosuke man walking barefoot along the shore, looking down at the water with a slight smile.

Picture of kaitoyosuke man walking down a rainy street, holding a colorful umbrella, his face lit up by a streetlamp.

Picture of kaitoyosuke man walking through a meadow of wildflowers, brushing his hand lightly over the blooms.

Picture of kaitoyosuke man walking through a park, holding a cup of coffee, glancing at a group of playing children.

Picture of kaitoyosuke man walking through a vibrant market, holding a basket, glancing at a vendor's stall.

Picture of kaitoyosuke man walking down a rainy street, holding a colorful umbrella, his face lit up by a streetlamp.

Picture of kaitoyosuke man standing at a crossroads in a city, holding a map, looking up at a street sign with curiosity.

Picture of kaitoyosuke man with a determined expression, jaw set and eyes fixed ahead.

Picture of kaitoyosuke man with a furious expression, teeth clenched and eyebrows furrowed.

Picture of kaitoyosuke man with a mischievous grin, one eyebrow slightly arched.

Picture of kaitoyosuke man with a playful expression, tongue sticking out and eyes squinting.

Picture of kaitoyosuke man with a sorrowful expression, tears streaming down and lips trembling.

Picture of kaitoyosuke man with a surprised expression, eyebrows raised, and eyes wide open.

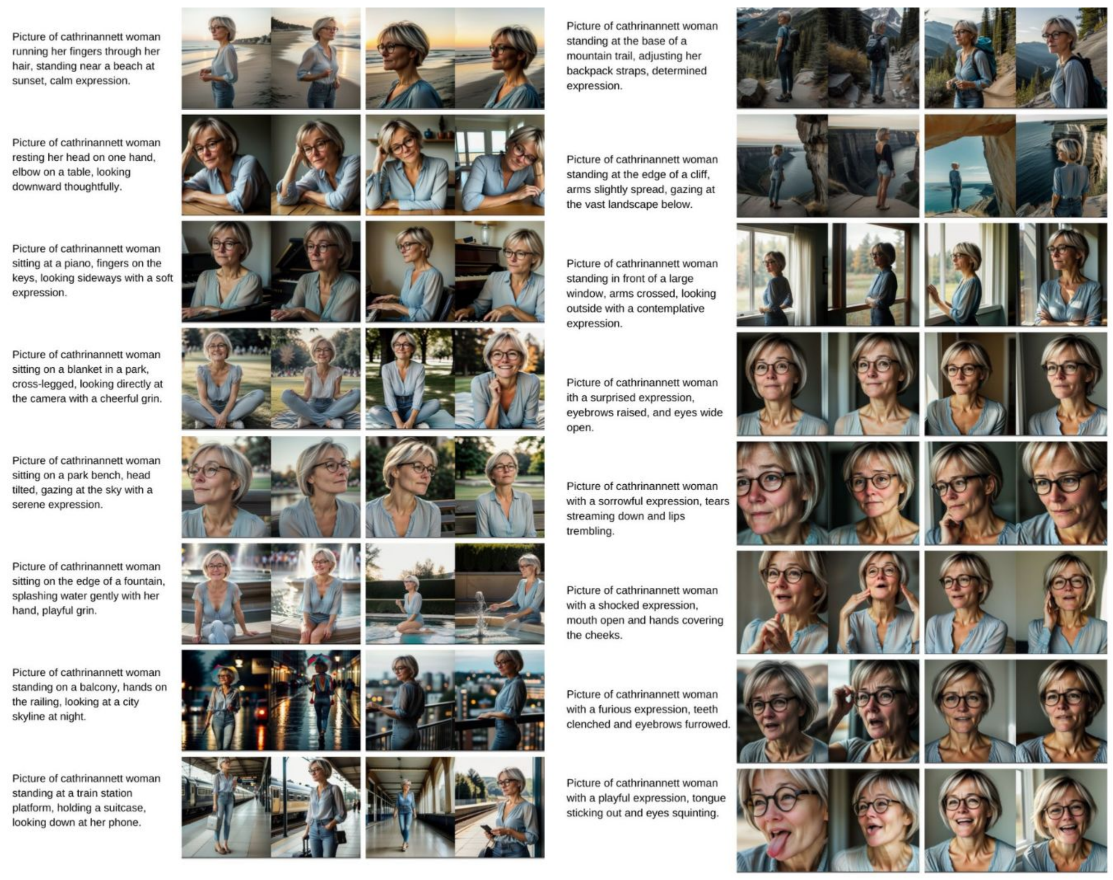
Picture of kaitoyosuke man with an annoyed expression, lips pressed together and eyebrows slightly furrowed.

Figure 6.13: Comparison of kaitoyosuke man character generations with(left) and without(right) captions.

This comparison is not intended as a direct benchmarking exercise or an adversarial evaluation aimed at proving absolute superiority. Rather, it serves to demonstrate the qualitative and measurable benefits of task-specific fine-tuning over a generic, non-specialized baseline.

The fine-tuned system is specifically designed to enforce visual consistency across frames, maintain character identity, and adhere more closely to cinematic shot conventions. The baseline, in contrast, naturally generates inconsistent characters from shot to shot as generic terms are employed, and often does not adhere to the specified shot type.

Interestingly, during the generation of the *house investigation* scene, we observed a recurring bias in the base model, which frequently rendered frames in black and white. This behavior was not explicitly prompted and likely stems from latent



Picture of cathrinannett woman running her fingers through her hair, standing near a beach at sunset, calm expression.

Picture of cathrinannett woman resting her head on one hand, elbow on a table, looking downward thoughtfully.

Picture of cathrinannett woman sitting at a piano, fingers on the keys, looking sideways with a soft expression.

Picture of cathrinannett woman sitting on a blanket in a park, cross-legged, looking directly at the camera with a cheerful grin.

Picture of cathrinannett woman sitting on a park bench, head tilted, gazing at the sky with a serene expression.

Picture of cathrinannett woman sitting on the edge of a fountain, splashing water gently with her hand, playful grin.

Picture of cathrinannett woman standing on a balcony, hands on the railing, looking at a city skyline at night.

Picture of cathrinannett woman standing at a train station platform, holding a suitcase, looking down at her phone.

Picture of cathrinannett woman standing at the base of a mountain trail, adjusting her backpack straps, determined expression.

Picture of cathrinannett woman standing at the edge of a cliff, arms slightly spread, gazing at the vast landscape below.

Picture of cathrinannett woman standing in front of a large window, arms crossed, looking outside with a contemplative expression.

Picture of cathrinannett woman ith a surprised expression, eyebrows raised, and eyes wide open.

Picture of cathrinannett woman with a sorrowful expression, tears streaming down and lips trembling.

Picture of cathrinannett woman with a shocked expression, mouth open and hands covering the cheeks.

Picture of cathrinannett woman with a furious expression, teeth clenched and eyebrows furrowed.

Picture of cathrinannett woman with a playful expression, tongue sticking out and eyes squinting.

Figure 6.14: Comparison of cathrinannett woman character generations with(left) and without(right) captions.

associations in the model's training data, where investigative or suspenseful themes are often linked to noir or grayscale aesthetics. In contrast, the fine-tuned model produced outputs that aligned more closely with the intended visual tone, reflecting the stylistic tendencies learned from the character and shot type training data. This observation further illustrates how task-specific fine-tuning can help override unintended stylistic priors, enabling more controlled and context-appropriate visual generation.

Finally, figure 6.21 below demonstrates the effectiveness of the inpainting refinement step, displaying the shots before and after inpainting was applied.

6.3.4 Metrics

To conduct a quantitative evaluation of our storyboard generation pipeline, we adopted three complementary metrics that assess different aspects of image quality and semantic relevance. These are:

- CLIPscore [67], which measures the semantic alignment between a generated image and its corresponding text prompt. It does so by computing the cosine similarity between their respective CLIP embeddings. The score ranges from 0 to 100, with higher values indicating better semantic alignment between visual content and textual description.
- Image Composition Assessment (ICA) Metric[68], a metric designed to evaluate the aesthetic and compositional quality of an image. The score ranges from 1 to 5, reflecting principles of visual balance, composition rules, object placement, and general photographic appeal.
- DINOv2 score [69], which quantifies the semantic similarity between two images. The score ranges from 0 to 1, with values closer to 1 indicating higher visual-semantic consistency.

Together, these metrics allowed us to evaluate not just the accuracy of the prompt-image alignment (via CLIPScore), but also image quality (via ICA) and image similarity between the finetuned and base models (via DINOv2). Quantitative evaluations were performed on all storyboard pairs presented in this work.

CLIP score

To assess prompt fidelity, we computed the CLIPScore for each frame in every storyboard. For each scene, we then averaged the individual scores across all shots, allowing us to compare the overall alignment performance between the fine-tuned model and the base model.

The results in Table 6.1 show that both models achieve comparable CLIPScore values across the board, with the base model slightly outperforming the fine-tuned model in most cases. However, these differences are marginal (typically within 1 point) and should be interpreted with caution.

CLIPScore evaluates alignment between the image and its prompt using global CLIP embeddings. However, it does not account for whether fine-grained elements—such as specific shot types, emotional cues, or spatial arrangements—are accurately rendered. Moreover, CLIPScore assesses each frame in isolation, without considering temporal or narrative continuity across sequences. As a result, it may fail to capture

Scene	Fine-tuned Model	Base Model
Cookie scene	34.1279	33.1410
Bench dialogue scene	30.4116	30.5454
Dream scene	31.6787	30.7279
Goodbye scene	28.1581	28.7286
Bus stop dialogue scene	30.3008	31.6678
House investigation scene	28.8912	30.7103

Table 6.1: Average CLIP score for the fine-tuned model and the base model across the different storyboard scenes.

the nuances that are critical for evaluating storyboard coherence, such as character consistency, emotional progression and narrative flow, or adherence to cinematic structure.

To illustrate the limitations of CLIPscore, figure 6.22 shows selected examples from different scenes where the base model significantly outperformed the fine-tuned model, by 5 points or more. A qualitative inspection reveals that these higher scores do not correspond to significantly better understanding of the narrative intent, image quality or nuances.

Therefore, while CLIPScore offers a first-order metric for prompt-image alignment, it is insufficient for evaluating the richer aspects of cinematic generation. It should be complemented with metrics (like DINOv2 and ICA) and especially qualitative analysis to capture the full scope of the storyboarding task..

Image Composition Assessment

Similarly to CLIPScore, we computed the Image Composition Assessment (ICA) metric for each individual shot in both the fine-tuned model and base model storyboards and then averaged the results across all shots. ICA is designed to assess the aesthetic and compositional quality of an image by analyzing spatial balance, symmetry, saliency distribution, and conformity to classical photography rules such as the rule of thirds, leading lines, and golden ratio alignment.

Table 6.2 reports the average ICA score for each scene: Across all scenes, the fine-tuned model demonstrates a consistent —although modest— advantage in compositional quality. This slight edge for the fine-tuned model suggests that it has a marginal advantage in placing characters or elements according to classical composition techniques, contributing to more visually pleasing and balanced outputs.

Scene	Fine-tuned Model	Base Model
Cookie scene	2.7575	2.6077
Bench dialogue scene	3.0920	2.9204
Dream scene	2.7865	2.7644
Goodbye scene	3.0148	2.7476
Bus stop dialogue scene	2.9126	2.5228
House investigation scene	2.7155	2.6464

Table 6.2: Average ICA score for the fine-tuned model and the base model across the different storyboard scenes.

A likely contributing factor is the use of Midjourney-generated images for fine-tuning. Midjourney is known for producing artistically high-quality images with well-balanced compositions, often following rules like the rule of thirds, symmetry, and the golden ratio. This may have implicitly taught the model to position characters and scene elements in a more deliberate and artistically pleasing manner.

Similarly to CLIPscore, the ICA metric also presents certain limitations. As a context-agnostic measure, it evaluates each image independently, without considering the narrative intent or functional role of a shot within a sequence. For example, an off-center or asymmetrical composition might be deliberately chosen to convey tension or focus, yet ICA could penalize it for deviating from classical compositional rules. As such, while ICA is useful for assessing aesthetic and structural qualities, it does not fully capture the expressive or storytelling dimensions of visual design.

In conjunction with other metrics and qualitative analysis, however, ICA reinforces the observation that the fine-tuned model offers notable advantages—producing compositions that are more aesthetically coherent and aligned with cinematic conventions.

DINOv2 score

Finally, we assessed the semantic similarity between the storyboards generated by the fine-tuned and base models by computing the average DINOv2 score for each scene. This metric was calculated by averaging the individual similarity scores for each pair of corresponding shots. DINOv2 evaluates the similarity between image embeddings, capturing high-level visual features such as object configuration, composition, and overall scene structure.

Table 6.3 reports the average DINOv2 score for each scene:

Scene	**DINO score**
Cookie scene	0.8604
Bench dialogue scene	0.7596
Dream scene	0.7861
Goodbye scene	0.8117
Bus stop dialogue scene	0.8117
House investigation scene	0.7691

Table 6.3: Average DINOv2 score for the fine-tuned model and the base model across the different storyboard scenes.

The results indicate a consistently high semantic similarity between the outputs of the fine-tuned and base models. All DINOv2 scores are above 0.75, indicating that, despite differences between the models, the generated images remain quite similar in terms of overall content and semantic structure. This aligns with the observation that the fine-tuned model does not drastically deviate from the base model in terms of high-level scene layout, but rather focuses on enhancing identity fidelity, pose realism, and contextual grounding.

The DINOv2 metric is optimized for general semantic similarity and compares individual image pairs, without understanding whether visual semantics match the intended textual meaning. Therefore, again, it is to be considered along with other metrics and qualitative evaluation.

6.3.5 Shot Type accuracy evaluation.

To evaluate how accurately the generated images matched the shot types specified in the prompts, we conducted a manual annotation of all frames where shot-type LoRAs were applied. For each image, we assessed whether the visual composition aligned with the definition of the intended shot type. Results were aggregated as correct matches over the total number of evaluated frames (excluding object-centric shots where no shot-type LoRA was applied), and reported per model and per scene. See Table 6.4.

The fine-tuned model correctly matched the intended shot types in 59 out of 75 cases, yielding an accuracy of approximately 78.7%. In contrast, the base model achieved only 26 correct matches out of 75, corresponding to an accuracy of 34.7%. These results confirm the efficacy of LoRA-based fine-tuning in teaching shot-type-specific visual patterns, nearly doubling the accuracy over the base model.

The most frequent mismatches observed in the fine-tuned model involved transitions

Scene	**Fine-tuned Accuracy**	**Base Accuracy**
Cookie	8/10	4/10
Bus Stop Dialogue	10/12	4/12
Goodbye	6/10	5/10
Bench Dialogue	9/14	4/14
House Investigation	15/16	3/16
Dream	11/13	6/13
Total	**59/75** (~78.7%)	**26/75** (~34.7%)

Table 6.4: Shot type accuracy across scenes for the fine-tuned model and the base model.

between visually adjacent categories —such as medium close-ups and medium shots, or medium shots and american shots. Conversely, the base model displayed wider and more inconsistent deviations, for example mistaking medium for full shots, american for long shots, or even medium close up for american, full or long shots, indicating a weaker grasp of shot type semantics.

To better contextualize these findings, it is worth noting that minor mismatches between visually adjacent shot types are expected and may not always be interpreted as true errors. Cinematic framing often leaves room for interpretive variation, and even human annotators may disagree on borderline cases where, strictly speaking, a shot would fall in between two categories. Additionally, prompt phrasing, inherent stochasticity in the generation process, whether the image is multi-character, or more subtle character positioning, are all contributing factors to small deviations. As such, most mismacthes observed in the fine-tuned model may still fall within an acceptable range of expressiveness rather than indicating failure.

In contrast, mismatches in the base model tend to be more erratic and pronounced, often involving jumps across non-adjacent shot types (e.g., medium to full or long shots, medium close ups to american or full shots), revealing a less structured internal representation of spatial framing.

Overall, the fine-tuned model demonstrates significantly superior shot-type fidelity, with consistent adherence to prompt-specified framing and reduced variance across scenes. This suggests that LoRA fine-tuning effectively enhances the model's ability to internalize and apply shot-type visual conventions in a structured and reliable manner.

6.3.6 Human evaluation

To complement the quantitative metrics and gain deeper insight into how human subjects perceive the quality and effectiveness of the generated storyboards, we conducted a human evaluation study comparing outputs from the fine-tuned model and the baseline Stable Diffusion 1.5 model.

A structured survey was designed, in which participants were shown pairs of storyboards —one generated by the fine-tuned model and the other by the base model— for each of the six narrative scenes presented in this work. Each pair was accompanied by a concise description of the scene to provide narrative context.

Participants were asked to respond to the following four questions for each storyboard:

- How would you score each storyboard based on visual quality (composition and aesthetics)?
- How would you score each storyboard based on overall coherence and consistency of characters and environmental objects?
- Based on the given broad description of the scene, how well do you think each storyboard communicates the implied narrative?
- How emotionally engaging do you find each storyboard?

Each question was answered on a 5-point Likert scale, where 1 indicated a very poor rating and 5 an excellent one.

We collected responses from a total of 63 participants, and the results, reported as average scores across participants, are summarized in Tables: 6.5 and 6.6.

Table 6.5: Human evaluation scores for Visual Quality and Consistency of characters and setting.

Scene	**Visual Quality**		**Consistency**	
	Finetuned	**Base**	**Finetuned**	**Base**
Cookie scene	3.73	3.14	3.71	2.71
Bench dialogue scene	2.89	3.30	3.32	2.78
Dream scene	3.70	3.30	3.96	3.02
Goodbye scene	3.37	3.20	3.62	2.67
Bus stop dialogue scene	3.24	3.21	3.63	2.55
House investigation scene	3.89	2.71	3.92	2.24
Average	3.47	3.14	3.69	2.66

Table 6.6: Human evaluation scores for Narrative Clarity and Emotional Relevance.

Scene	**Narrative Clarity**		**Emotional Relevance**	
	Finetuned	**Base**	**Finetuned**	**Base**
Cookie scene	3.51	2.63	3.37	2.36
Bench dialogue scene	2.96	2.60	2.98	2.55
Dream scene	3.75	3.19	3.49	2.76
Goodbye scene	3.68	2.73	3.30	2.71
Bus stop dialogue scene	3.54	2.60	3.21	2.54
House investigation scene	3.73	2.19	3.47	1.87
Average	3.53	2.66	3.30	2.46

The average scores were overall closer than initially expected, suggesting that the base Stable Diffusion model already possesses strong generative capabilities. Nonetheless, the fine-tuned model outperformed the base model across all four evaluation dimensions, proving that the fine-tuning pipeline provides tangible benefits.

Giving a deeper look into the results:

- **Visual quality:** the fine-tuned model was generally perceived as producing more aesthetically pleasing and compositionally balanced outputs. The average difference between the two models was the smallest among all categories (3.47 vs. 3.14). This narrower gap may be attributed to the already strong generative capabilities of the base Stable Diffusion model in producing visually pleasing compositions. Aspects such as lighting, color harmony, and photographic framing can be effectively handled even without targeted fine-tuning, which may explain why human evaluators rated both models similarly in terms of visual appeal.

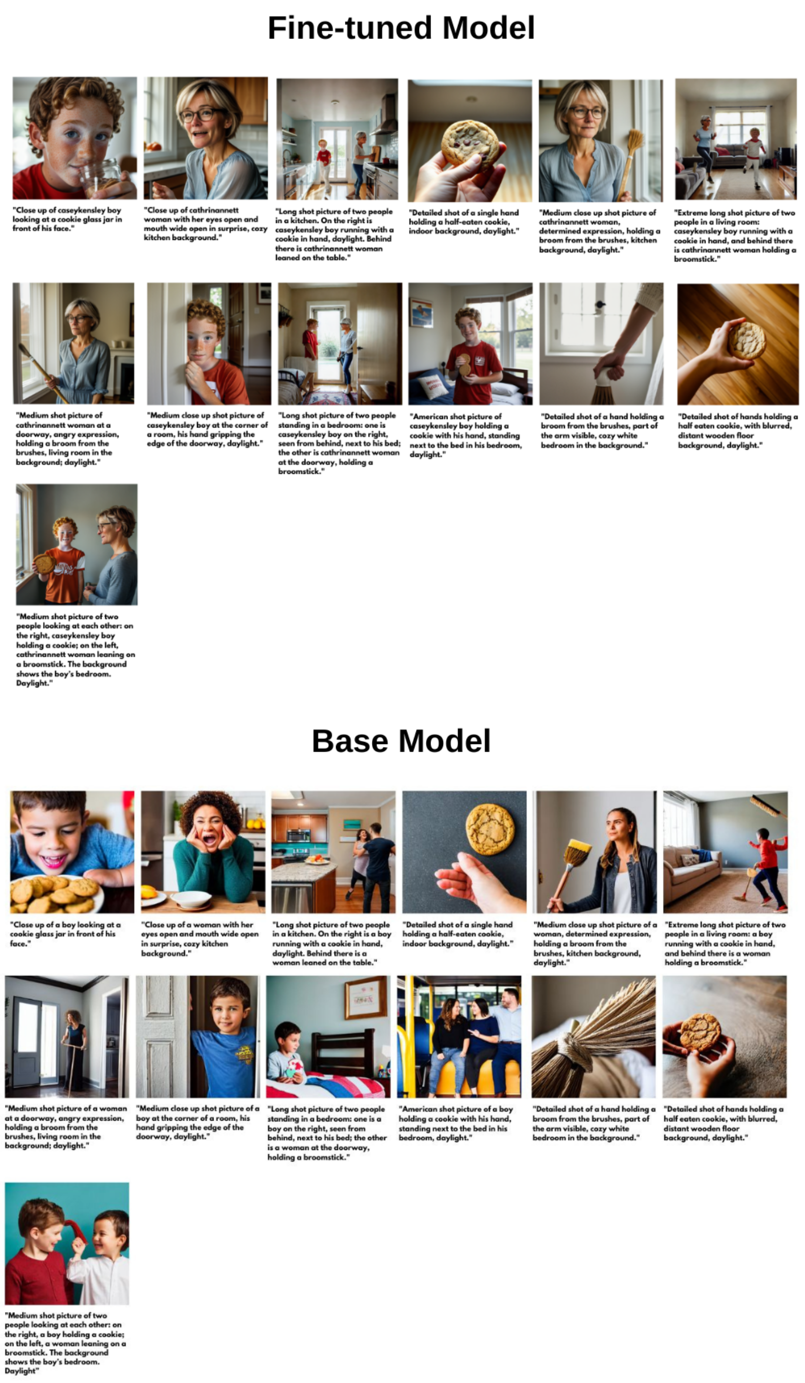
- **Consistency of characters and setting:** Here, the improvement was substantial (3.69 vs. 2.66), indicating that fine-tuning played a key role in maintaining visual consistency across frames. This aligns with one of the core challenges in storyboard generation, where character identity continuity is essential.

- **Narrative Clarity:** The fine-tuned model was also rated higher in its ability to visually communicate the story (3.53 vs. 2.66), suggesting that, beyond producing more attractive images, the storyboards benefited from better alignment with the narrative intent.

- **Emotional Relevance:** Interestingly, this was the category with the strongest relative gain (3.3 vs. 2.46), suggesting that the fine-tuned model better captured the emotional undertone of each scene, which led to more engaging and impactful storyboards and highlights the benefit of fine-tuning for affective storytelling.

These human evaluation results also help contextualize the findings and limitations of current quantitative metrics. Although CLIPScore slightly favored the base model in several scenes, participants consistently preferred the fine-tuned model —highlighting how CLIPScore, which assesses prompt-image alignment on a frame-by-frame basis, overlooks narrative structure, emotional nuance, and visual continuity. In contrast, the ICA metric showed a trend more aligned with human judgment on the dimension of visual quality, with a better performance of the finetuned model even though by a modest gap. DINOv2 scores aligned with human evaluation in that they confirmed both models produce visually similar content at a high level of abstraction; however, the fine-tuned model consistently outperformed in aspects that DINOv2 does not capture—such as character identity preservation, emotional nuance, and narrative structure. This further emphasizes the importance of complementing automated metrics with human-centered evaluations when assessing storyboards and narrative visualizations.

Overall, the human evaluation underscores the added value of targeted fine-tuning strategies in achieving better semantic alignment, expressive depth, and coherence in visual storytelling.



Fine-tuned Model

"Close up of caseykensley boy looking at a cookie glass jar in front of his face."

"Close up of cathrinannett woman with her eyes open and mouth wide open in surprise, cozy kitchen background."

"Long shot picture of two people in a kitchen. On the right is caseykensley boy running with a cookie in hand, daylight. Behind there is cathrinannett woman leaned on the table."

"Detailed shot of a single hand holding a half-eaten cookie, indoor background, daylight."

"Medium close up shot picture of cathrinannett woman, determined expression, holding a broom from the brushes, kitchen background, daylight."

"Extreme long shot picture of two people in a living room: caseykensley boy running with a cookie in hand, and behind there is cathrinannett woman holding a broomstick."

"Medium shot picture of cathrinannett woman at a doorway, angry expression, holding a broom from the brushes, living room in the background; daylight."

"Medium close up shot picture of caseykensley boy at the corner of a room, his hand gripping the edge of the doorway, daylight."

"Long shot picture of two people standing in a bedroom: one is caseykensley boy on the right, seen from behind, next to his bed; the other is cathrinannett woman at the doorway, holding a broomstick."

"American shot picture of caseykensley boy holding a cookie with his hand, standing next to the bed in his bedroom, daylight."

"Detailed shot of a hand holding a broom from the brushes, part of the arm visible, cozy white bedroom in the background."

"Detailed shot of hands holding a half eaten cookie, with blurred, distant wooden floor background, daylight."

"Medium shot picture of two people looking at each other: on the right, caseykensley boy holding a cookie; on the left, cathrinannett woman leaning on a broomstick. The background shows the boy's bedroom. Daylight."

Base Model

"Close up of a boy looking at a cookie glass jar in front of his face."

"Close up of a woman with her eyes open and mouth wide open in surprise, cozy kitchen background."

"Long shot picture of two people in a kitchen. On the right is a boy running with a cookie in hand, daylight. Behind there is a woman leaned on the table."

"Detailed shot of a single hand holding a half-eaten cookie, indoor background, daylight."

"Medium close up shot picture of a woman, determined expression, holding a broom from the brushes, kitchen background, daylight."

"Extreme long shot picture of two people in a living room: a boy running with a cookie in hand, and behind there is a woman holding a broomstick."

"Medium shot picture of a woman at a doorway, angry expression, holding a broom from the brushes, living room in the background; daylight."

"Medium close up shot picture of a boy at the corner of a room, his hand gripping the edge of the doorway, daylight."

"Long shot picture of two people standing in a bedroom: one is a boy on the right, seen from behind, next to his bed; the other is a woman at the doorway, holding a broomstick."

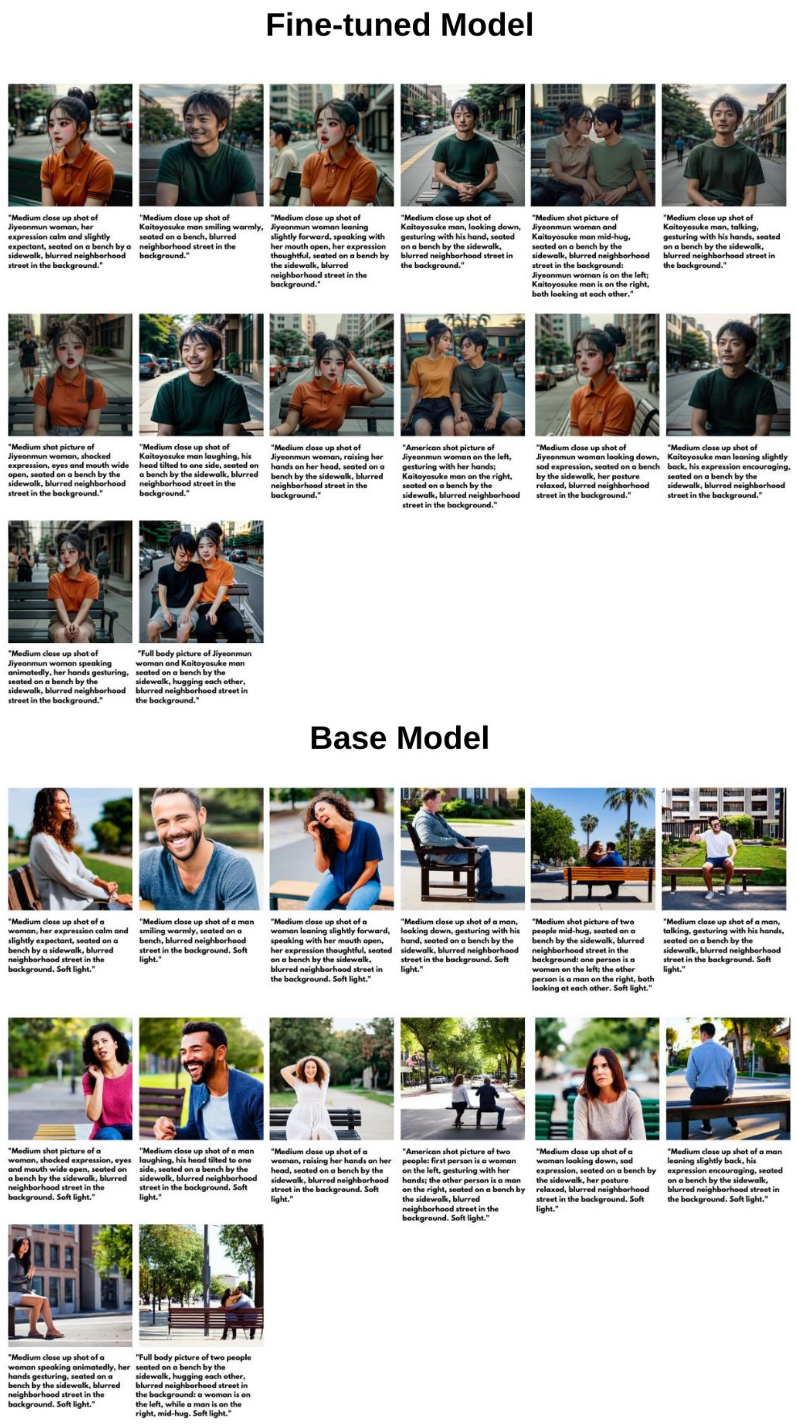
"American shot picture of a boy holding a cookie with his hand, standing next to the bed in his bedroom, daylight."

"Detailed shot of a hand holding a broom from the brushes, part of the arm visible, cozy white bedroom in the background."

"Detailed shot of hands holding a half eaten cookie, with blurred, distant wooden floor background, daylight."

"Medium shot picture of two people looking at each other: on the right, a boy holding a cookie; on the left, a woman leaning on a broomstick. The background shows the boy's bedroom. Daylight"

Figure 6.15: Cookie scene - finetuned vs base model



Fine-tuned Model

"Medium close up shot of Jiyeonmun woman, her expression calm and slightly expectant, seated on a bench by a sidewalk, blurred neighborhood street in the background."

"Medium close up shot of Kaitoyosuke man smiling warmly, seated on a bench, blurred neighborhood street in the background."

"Medium close up shot of Jiyeonmun woman leaning slightly forward, speaking with her mouth open, her expression thoughtful, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium close up shot of Kaitoyosuke man, looking down, gesturing with his hand, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium shot picture of Jiyeonmun woman and Kaitoyosuke man mid-hug, seated on a bench by the sidewalk, blurred neighborhood street in the background: Jiyeonmun woman is on the left; Kaitoyosuke man is on the right, both looking at each other."

"Medium close up shot of Kaitoyosuke man, talking, gesturing with his hands, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium shot picture of Jiyeonmun woman, shocked expression, eyes and mouth wide open, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium close up shot of Kaitoyosuke man laughing, his head tilted to one side, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium close up shot of Jiyeonmun woman, raising her hands on her head, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"American shot picture of Jiyeonmun woman on the left, gesturing with her hands; Kaitoyosuke man on the right, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium close up shot of Jiyeonmun woman looking down, sad expression, seated on a bench by the sidewalk, her posture relaxed, blurred neighborhood street in the background."

"Medium close up shot of Kaitoyosuke man leaning slightly back, his expression encouraging, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Medium close up shot of Jiyeonmun woman speaking animatedly, her hands gesturing, seated on a bench by the sidewalk, blurred neighborhood street in the background."

"Full body picture of Jiyeonmun woman and Kaitoyosuke man seated on a bench by the sidewalk, hugging each other, blurred neighborhood street in the background."

Base Model

"Medium close up shot of a woman, her expression calm and slightly expectant, seated on a bench by a sidewalk, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a man smiling warmly, seated on a bench, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a woman leaning slightly forward, speaking with her mouth open, her expression thoughtful, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a man, looking down, gesturing with his hand, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Medium shot picture of two people mid-hug, seated on a bench by the sidewalk, blurred neighborhood street in the background: one person is a woman on the left; the other person is a man on the right, both looking at each other. Soft light."

"Medium close up shot of a man, talking, gesturing with his hands, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Medium shot picture of a woman, shocked expression, eyes and mouth wide open, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a man laughing, his head tilted to one side, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a woman, raising her hands on her head, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"American shot picture of two people: first person is a woman on the left, gesturing with her hands; the other person is a man on the right, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

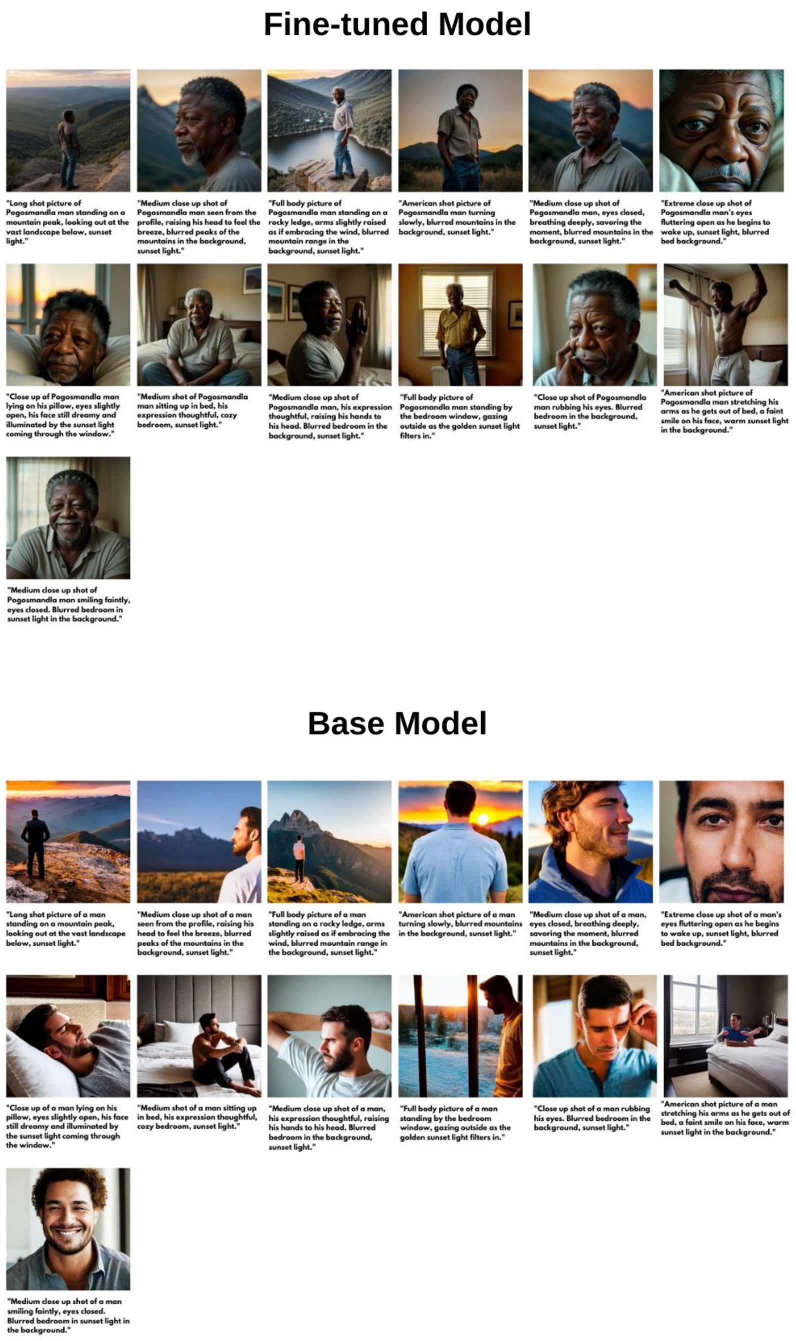
"Medium close up shot of a woman looking down, sad expression, seated on a bench by the sidewalk, her posture relaxed, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a man leaning slightly back, his expression encouraging, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Medium close up shot of a woman speaking animatedly, her hands gesturing, seated on a bench by the sidewalk, blurred neighborhood street in the background. Soft light."

"Full body picture of two people seated on a bench by the sidewalk, hugging each other, blurred neighborhood street in the background: a woman is on the left, while a man is on the right, mid-hug. Soft light."

Figure 6.16: Bench dialogue scene - finetuned vs base model



"Long shot picture of Pogosmandla man standing on a mountain peak, looking out at the vast landscape below, sunset light."

"Medium close up shot of Pogosmandla man seen from the profile, raising his head to feel the breeze, blurred peaks of the mountains in the background, sunset light."

"Full body picture of Pogosmandla man standing on a rocky ledge, arms slightly raised as if embracing the wind, blurred mountain range in the background, sunset light."

"American shot picture of Pogosmandla man turning slowly, blurred mountains in the background, sunset light."

"Medium close up shot of Pogosmandla man, eyes closed, breathing deeply, savoring the moment, blurred mountains in the background, sunset light."

"Extreme close up shot of Pogosmandla man's eyes fluttering open as he begins to wake up, sunset light, blurred bed background."

"Close up of Pogosmandla man lying on his pillow, eyes slightly open, his face still dreamy and illuminated by the sunset light coming through the window."

"Medium shot of Pogosmandla man sitting up in bed, his expression thoughtful, cozy bedroom, sunset light."

"Medium close up shot of Pogosmandla man, his expression thoughtful, raising his hands to his head. Blurred bedroom in the background, sunset light."

"Full body picture of Pogosmandla man standing by the bedroom window, gazing outside as the golden sunset light filters in."

"Close up shot of Pogosmandla man rubbing his eyes. Blurred bedroom in the background, sunset light."

"American shot picture of Pogosmandla man stretching his arms as he gets out of bed, a faint smile on his face, warm sunset light in the background."

"Medium close up shot of Pogosmandla man smiling faintly, eyes closed. Blurred bedroom in sunset light in the background."

"Long shot picture of a man standing on a mountain peak, looking out at the vast landscape below, sunset light."

"Medium close up shot of a man seen from the profile, raising his head to feel the breeze, blurred peaks of the mountains in the background, sunset light."

"Full body picture of a man standing on a rocky ledge, arms slightly raised as if embracing the wind, blurred mountain range in the background, sunset light."

"American shot picture of a man turning slowly, blurred mountains in the background, sunset light."

"Medium close up shot of a man, eyes closed, breathing deeply, savoring the moment, blurred mountains in the background, sunset light."

"Extreme close up shot of a man's eyes fluttering open as he begins to wake up, sunset light, blurred bed background."

"Close up of a man lying on his pillow, eyes slightly open, his face still dreamy and illuminated by the sunset light coming through the window."

"Medium shot of a man sitting up in bed, his expression thoughtful, cozy bedroom, sunset light."

"Medium close up shot of a man, his expression thoughtful, raising his hands to his head. Blurred bedroom in the background, sunset light."

"Full body picture of a man standing by the bedroom window, gazing outside as the golden sunset light filters in."

"Close up shot of a man rubbing his eyes. Blurred bedroom in the background, sunset light."

"American shot picture of a man stretching his arms as he gets out of bed, a faint smile on his face, warm sunset light in the background."

"Medium close up shot of a man smiling faintly, eyes closed. Blurred bedroom in sunset light in the background."

Figure 6.17: Dream scene - finetuned vs base model

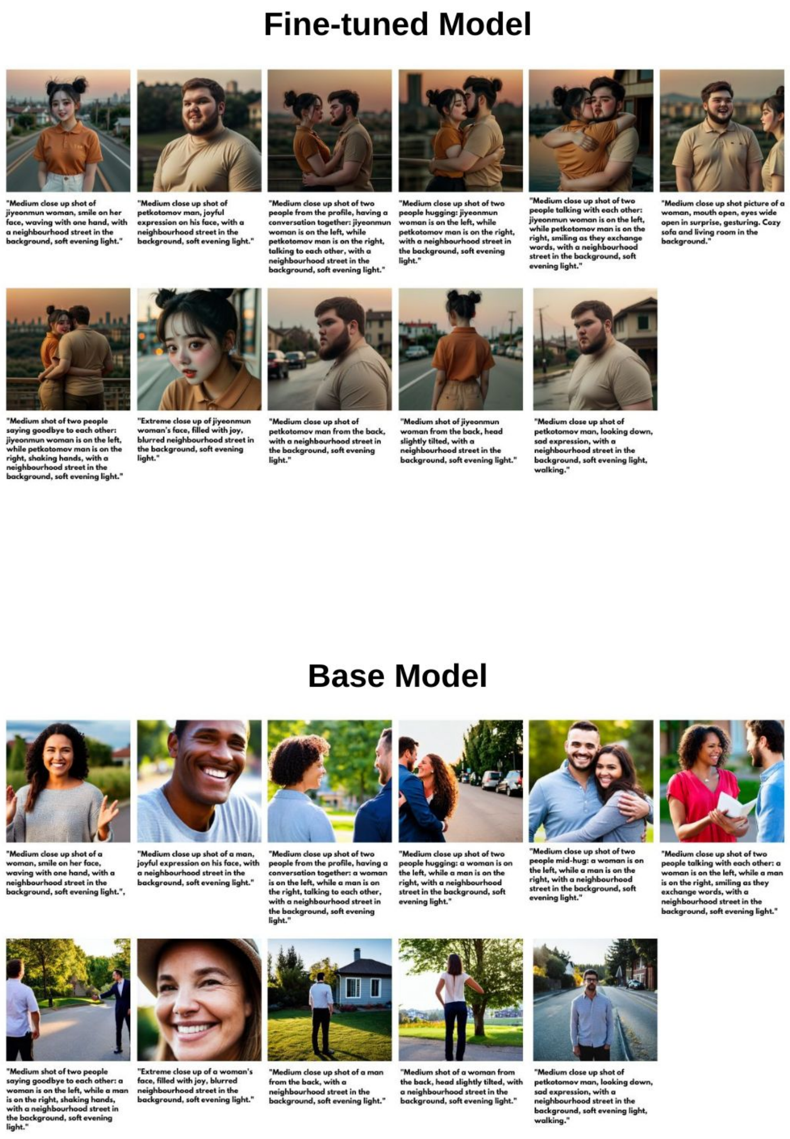
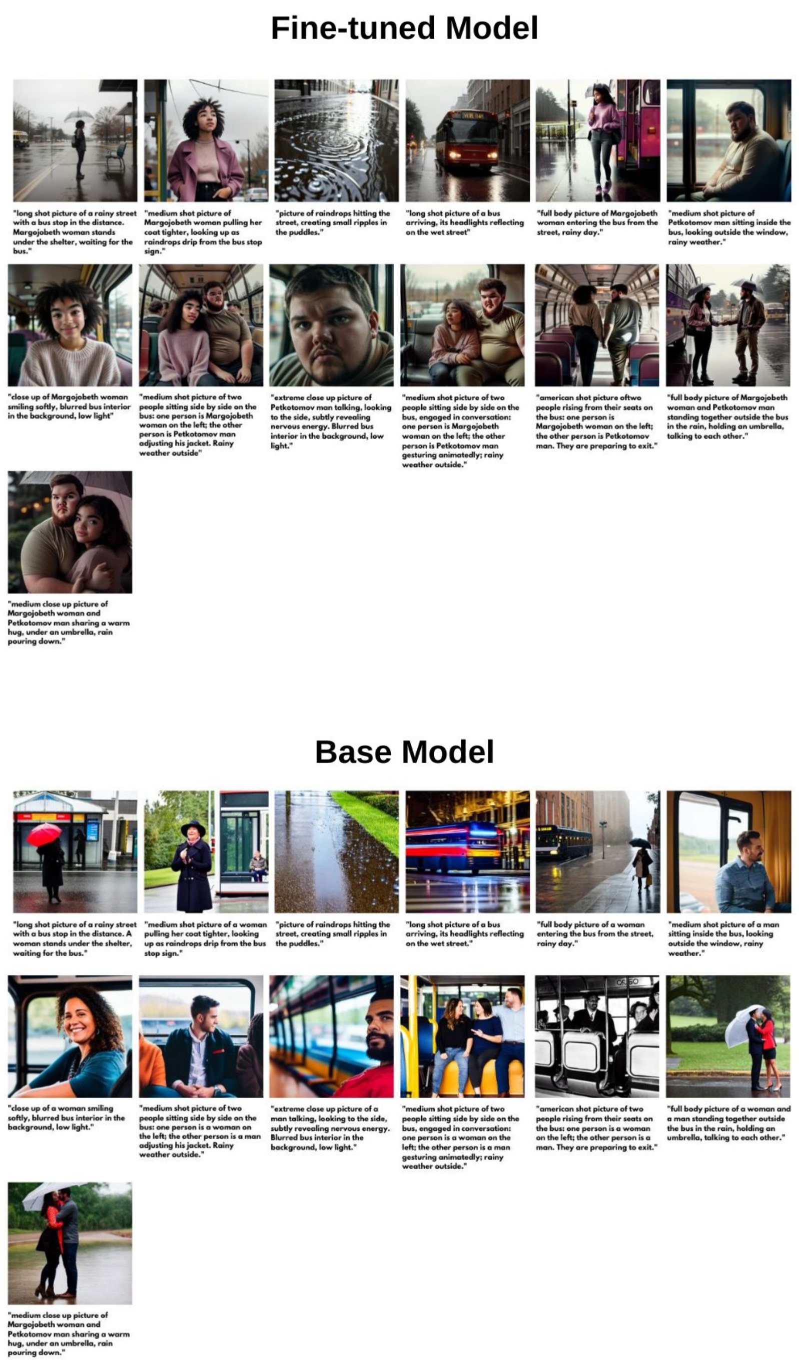

Fine-tuned Model
"Medium close up shot of jiyeonmun woman, smile on her face, waving with one hand, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of petkotomov man, joyful expression on his face, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people from the profile, having a conversation together: jiyeonmun woman is on the left, while petkotomov man is on the right, talking to each other, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people hugging: jiyeonmun woman is on the left, while petkotomov man is on the right, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people talking with each other: jiyeonmun woman is on the left, while petkotomov man is on the right, smiling as they exchange words, with a neighbourhood street in the background, soft evening light."
"Medium close up shot picture of a woman, mouth open, eyes wide open in surprise, gesturing. Cozy sofa and living room in the background."
"Medium shot of two people saying goodbye to each other: jiyeonmun woman is on the left, while petkotomov man is on the right, shaking hands, with a neighbourhood street in the background, soft evening light."
"Extreme close up of jiyeonmun woman's face, filled with joy, blurred neighbourhood street in the background, soft evening light."
"Medium close up shot of petkotomov man from the back, with a neighbourhood street in the background, soft evening light."
"Medium shot of jiyeonmun woman from the back, head slightly tilted, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of petkotomov man, looking down, sad expression, with a neighbourhood street in the background, soft evening light, walking."
Base Model
"Medium close up shot of a woman, smile on her face, waving with one hand, with a neighbourhood street in the background, soft evening light.",
"Medium close up shot of a man, joyful expression on his face, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people from the profile, having a conversation together: a woman is on the left, while a man is on the right, talking to each other, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people hugging: a woman is on the left, while a man is on the right, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people mid-hug: a woman is on the left, while a man is on the right, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of two people talking with each other: a woman is on the left, while a man is on the right, smiling as they exchange words, with a neighbourhood street in the background, soft evening light."
"Medium shot of two people saying goodbye to each other: a woman is on the left, while a man is on the right, shaking hands, with a neighbourhood street in the background, soft evening light."
"Extreme close up of a woman's face, filled with joy, blurred neighbourhood street in the background, soft evening light."
"Medium close up shot of a man from the back, with a neighbourhood street in the background, soft evening light."
"Medium shot of a woman from the back, head slightly tilted, with a neighbourhood street in the background, soft evening light."
"Medium close up shot of petkotomov man, looking down, sad expression, with a neighbourhood street in the background, soft evening light, walking."

Figure 6.18: Goodbye scene - finetuned vs base model



"long shot picture of a rainy street with a bus stop in the distance. Margojobeth woman stands under the shelter, waiting for the bus."

"medium shot picture of Margojobeth woman pulling her coat tighter, looking up as raindrops drip from the bus stop sign."

"picture of raindrops hitting the street, creating small ripples in the puddles."

"long shot picture of a bus arriving, its headlights reflecting on the wet street"

"full body picture of Margojobeth woman entering the bus from the street, rainy day."

"medium shot picture of Petkotomov man sitting inside the bus, looking outside the window, rainy weather."

"close up of Margojobeth woman smiling softly, blurred bus interior in the background, low light"

"medium shot picture of two people sitting side by side on the bus: one person is Margojobeth woman on the left; the other person is Petkotomov man adjusting his jacket. Rainy weather outside"

"extreme close up picture of Petkotomov man talking, looking to the side, subtly revealing nervous energy. Blurred bus interior in the background, low light."

"medium shot picture of two people sitting side by side on the bus, engaged in conversation: one person is Margojobeth woman on the left; the other person is Petkotomov man gesturing animatedly; rainy weather outside."

"american shot picture oftwo people rising from their seats on the bus: one person is Margojobeth woman on the left; the other person is Petkotomov man. They are preparing to exit."

"full body picture of Margojobeth woman and Petkotomov man standing together outside the bus in the rain, holding an umbrella, talking to each other."

"medium close up picture of Margojobeth woman and Petkotomov man sharing a warm hug, under an umbrella, rain pouring down."

"long shot picture of a rainy street with a bus stop in the distance. A woman stands under the shelter, waiting for the bus."

"medium shot picture of a woman pulling her coat tighter, looking up as raindrops drip from the bus stop sign."

"picture of raindrops hitting the street, creating small ripples in the puddles."

"long shot picture of a bus arriving, its headlights reflecting on the wet street."

"full body picture of a woman entering the bus from the street, rainy day."

"medium shot picture of a man sitting inside the bus, looking outside the window, rainy weather."

"close up of a woman smiling softly, blurred bus interior in the background, low light."

"medium shot picture of two people sitting side by side on the bus: one person is a woman on the left; the other person is a man adjusting his jacket. Rainy weather outside."

"extreme close up picture of a man talking, looking to the side, subtly revealing nervous energy. Blurred bus interior in the background, low light."

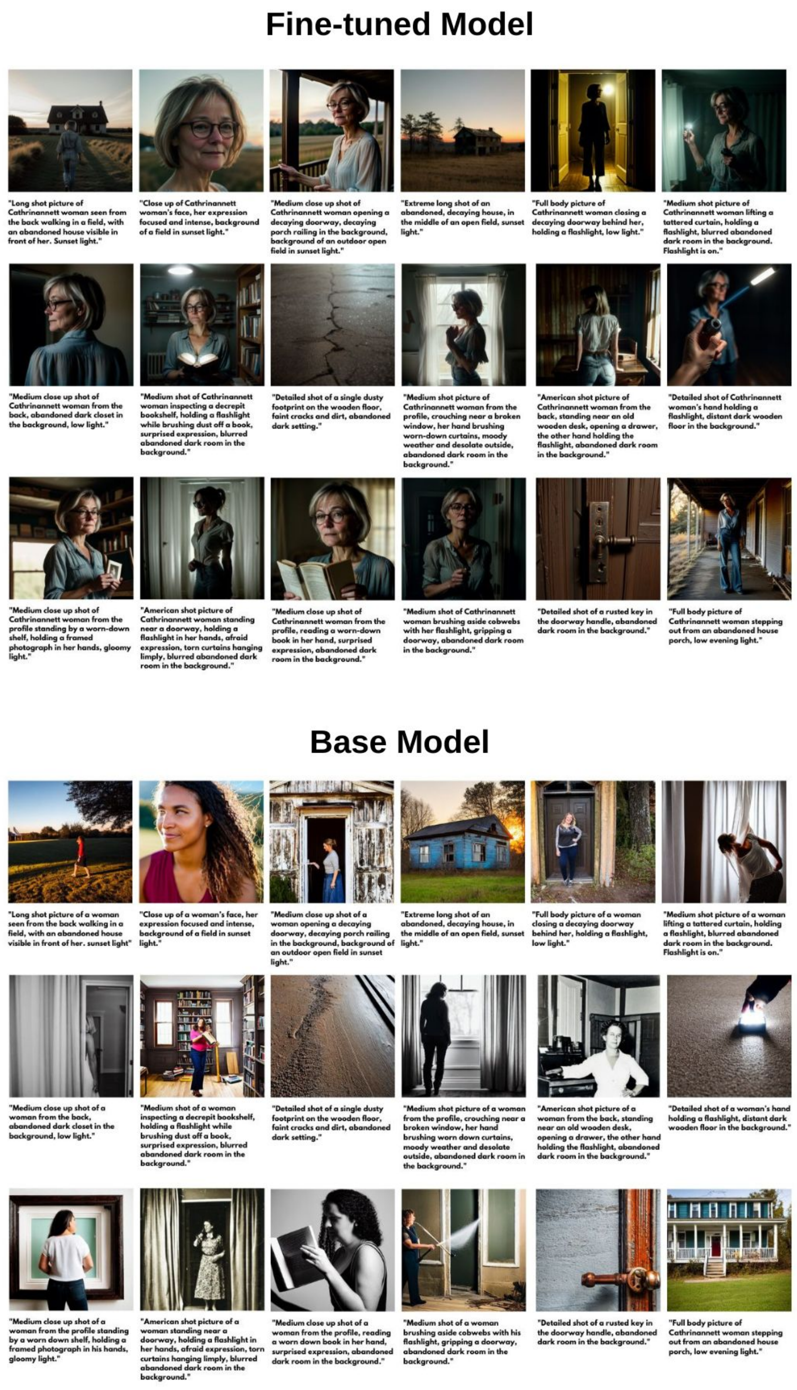
"medium shot picture of two people sitting side by side on the bus, engaged in conversation: one person is a woman on the left; the other person is a man gesturing animatedly; rainy weather outside."

"american shot picture of two people rising from their seats on the bus: one person is a woman on the left; the other person is a man. They are preparing to exit."

"full body picture of a woman and a man standing together outside the bus in the rain, holding an umbrella, talking to each other."

"medium close up picture of Margojobeth woman and Petkotomov man sharing a warm hug, under an umbrella, rain pouring down."

Figure 6.19: Bus stop dialogue scene - finetuned vs base model


Fine-tuned Model

"Long shot picture of Cathrinannett woman seen from the back walking in a field, with an abandoned house visible in front of her. Sunset light."

"Close up of Cathrinannett woman's face, her expression focused and intense, background of a field in sunset light."

"Medium close up shot of Cathrinannett woman opening a decaying doorway, decaying porch railing in the background, background of an outdoor open field in sunset light."

"Extreme long shot of an abandoned, decaying house, in the middle of an open field, sunset light."

"Full body picture of Cathrinannett woman closing a decaying doorway behind her, holding a flashlight, low light."

"Medium shot picture of Cathrinannett woman lifting a tattered curtain, holding a flashlight, blurred abandoned dark room in the background. Flashlight is on."

"Medium close up shot of Cathrinannett woman from the back, abandoned dark closet in the background, low light."

"Medium shot of Cathrinannett woman inspecting a decrepit bookshelf, holding a flashlight while brushing dust off a book, surprised expression, blurred abandoned dark room in the background."

"Detailed shot of a single dusty footprint on the wooden floor, faint cracks and dirt, abandoned dark setting."

"Medium shot picture of Cathrinannett woman from the profile, crouching near a broken window, her hand brushing worn-down curtains, moody weather and desolate outside, abandoned dark room in the background."

"American shot picture of Cathrinannett woman from the back, standing near an old wooden desk, opening a drawer, the other hand holding the flashlight, abandoned dark room in the background."

"Detailed shot of Cathrinannett woman's hand holding a flashlight, distant dark wooden floor in the background."

"Medium close up shot of Cathrinannett woman from the profile standing by a worn-down shelf, holding a framed photograph in her hands, gloomy light."

"American shot picture of Cathrinannett woman standing near a doorway, holding a flashlight in her hands, afraid expression, torn curtains hanging limply, blurred abandoned dark room in the background."

"Medium close up shot of Cathrinannett woman from the profile, reading a worn-down book in her hand, surprised expression, abandoned dark room in the background."

"Medium shot of Cathrinannett woman brushing aside cobwebs with her flashlight, gripping a doorway, abandoned dark room in the background."

"Detailed shot of a rusted key in the doorway handle, abandoned dark room in the background."

"Full body picture of Cathrinannett woman stepping out from an abandoned house porch, low evening light."

Base Model

"Long shot picture of a woman seen from the back walking in a field, with an abandoned house visible in front of her. sunset light"

"Close up of a woman's face, her expression focused and intense, background of a field in sunset light."

"Medium close up shot of a woman opening a decaying doorway, decaying porch railing in the background, background of an outdoor open field in sunset light."

"Extreme long shot of an abandoned, decaying house, in the middle of an open field, sunset light."

"Full body picture of a woman closing a decaying doorway behind her, holding a flashlight, low light."

"Medium shot picture of a woman lifting a tattered curtain, holding a flashlight, blurred abandoned dark room in the background. Flashlight is on."

"Medium close up shot of a woman from the back, abandoned dark closet in the background, low light."

"Medium shot of a woman inspecting a decrepit bookshelf, holding a flashlight while brushing dust off a book, surprised expression, blurred abandoned dark room in the background."

"Detailed shot of a single dusty footprint on the wooden floor, faint cracks and dirt, abandoned dark setting."

"Medium shot picture of a woman from the profile, crouching near a broken window, her hand brushing worn down curtains, moody weather and desolate outside, abandoned dark room in the background."

"American shot picture of a woman from the back, standing near an old wooden desk, opening a drawer, the other hand holding the flashlight, abandoned dark room in the background."

"Detailed shot of a woman's hand holding a flashlight, distant dark wooden floor in the background."

"Medium close up shot of a woman from the profile standing by a worn down shelf, holding a framed photograph in his hands, gloomy light."

"American shot picture of a woman standing near a doorway, holding a flashlight in her hands, afraid expression, torn curtains hanging limply, blurred abandoned dark room in the background."

"Medium close up shot of a woman from the profile, reading a worn down book in her hand, surprised expression, abandoned dark room in the background."

"Medium shot of a woman brushing aside cobwebs with his flashlight, gripping a doorway, abandoned dark room in the background."

"Detailed shot of a rusted key in the doorway handle, abandoned dark room in the background."

"Full body picture of Cathrinannett woman stepping out from an abandoned house porch, low evening light."

Figure 6.20: House investigation scene - finetuned vs base model

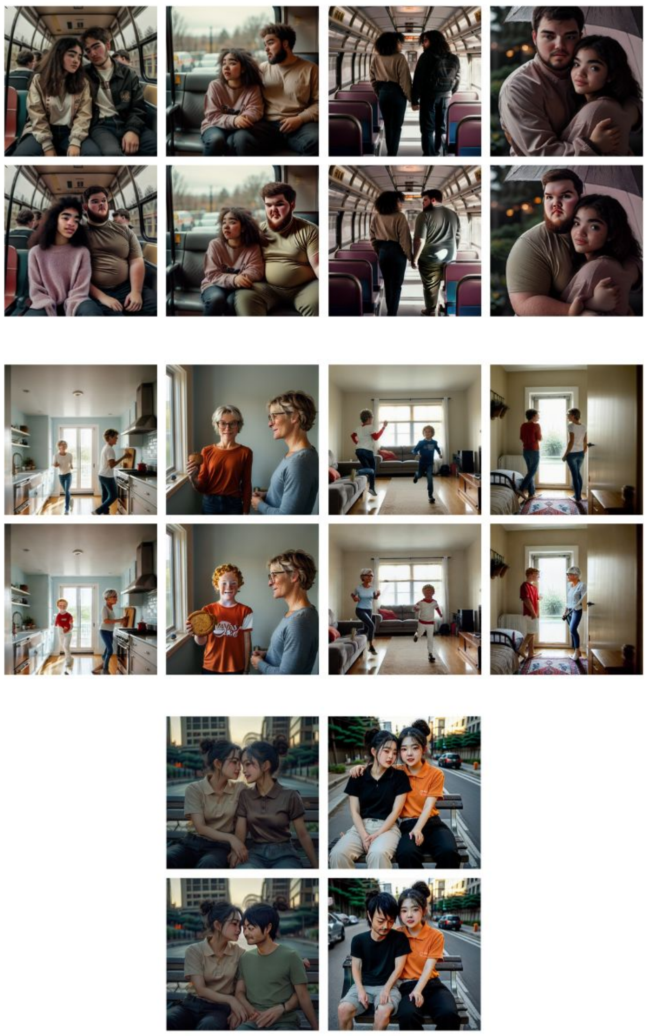
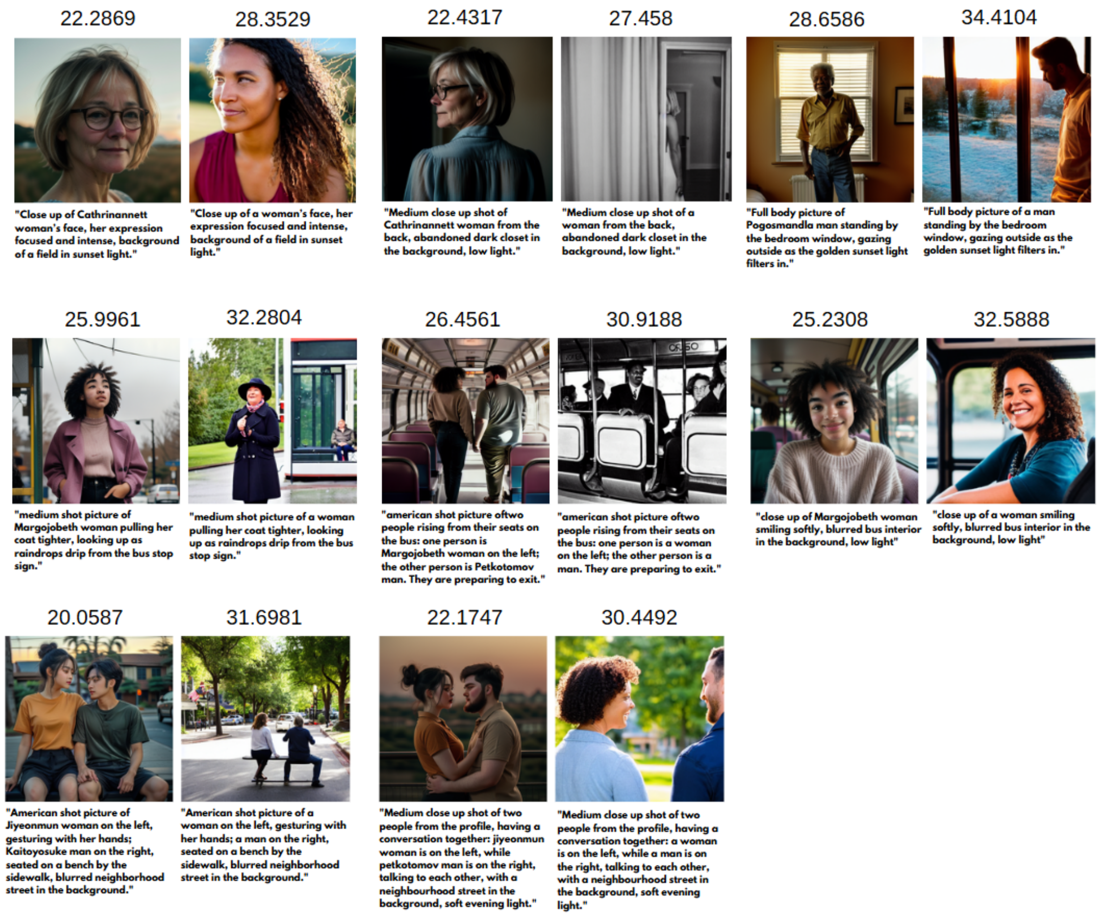


Figure 6.21: Results of the inpainting refinement step, showing shots before and after refinement.



22.2869

"Close up of Cathrinannett woman's face, her expression focused and intense, background of a field in sunset light."

28.3529

"Close up of a woman's face, her expression focused and intense, background of a field in sunset light."

22.4317

"Medium close up shot of Cathrinannett woman from the back, abandoned dark closet in the background, low light."

27.458

"Medium close up shot of a woman from the back, abandoned dark closet in the background, low light."

28.6586

"Full body picture of Pogosmandla man standing by the bedroom window, gazing outside as the golden sunset light filters in."

34.4104

"Full body picture of a man standing by the bedroom window, gazing outside as the golden sunset light filters in."

25.9961

"medium shot picture of Margojobeth woman pulling her coat tighter, looking up as raindrops drip from the bus stop sign."

32.2804

"medium shot picture of a woman pulling her coat tighter, looking up as raindrops drip from the bus stop sign."

26.4561

"american shot picture oftwo people rising from their seats on the bus: one person is Margojobeth woman on the left; the other person is Petkotomov man. They are preparing to exit."

30.9188

"american shot picture oftwo people rising from their seats on the bus: one person is a woman on the left; the other person is a man. They are preparing to exit."

25.2308

"close up of Margojobeth woman smiling softly, blurred bus interior in the background, low light"

32.5888

"close up of a woman smiling softly, blurred bus interior in the background, low light"

20.0587

"American shot picture of Jiyeonmun woman on the left, gesturing with her hands; Kaitoyosuke man on the right, seated on a bench by the sidewalk, blurred neighborhood street in the background."

31.6981

"American shot picture of a woman on the left, gesturing with her hands; a man on the right, seated on a bench by the sidewalk, blurred neighborhood street in the background."

22.1747

"Medium close up shot of two people from the profile, having a conversation together: jiyeonmun woman is on the left, while petkotomov man is on the right, talking to each other, with a neighbourhood street in the background, soft evening light."

30.4492

"Medium close up shot of two people from the profile, having a conversation together: a woman is on the left, while a man is on the right, talking to each other, with a neighbourhood street in the background, soft evening light."

Figure 6.22: Examples where the base model achieved significantly higher CLIP-Score than the fine-tuned model

Chapter 7

Conclusions and future developments

This thesis presents a comprehensive exploration of the current landscape of image synthesis, with particular focus on the evolution of diffusion-based models and their application to visual storytelling. Alongside a detailed overview of the architectures and fine-tuning techniques that have shaped modern generative pipelines, the work contributes a practical system tailored for a relatively underexplored but highly impactful task: multi-frame storyboard generation.

Unlike traditional text-to-image applications, storyboard generation requires models to produce sequences of visually consistent images, aligned not only to a prompt but also to narrative continuity, character fidelity, and cinematic conventions such as shot composition and emotional nuance. While several recent works in the field of story visualization have tackled aspects of this challenge (as discussed in Chapter 5), no single approach has yet emerged as a robust solution for cinema-style storyboard creation. This thesis addresses this gap by proposing a method that combines fine-tuning, prompt engineering, and inpainting in a coherent generation pipeline.

The core contribution of this work lies in adapting Stable Diffusion 1.5, a state-of-the-art diffusion model, towards storyboard generation through DreamBooth LoRA finetuning, LLM-based prompt generation, and inpainting-based corrections. This pipeline enables the creation of storyboard sequences that maintain character identity across shots, follow predefined shot-type constraints, and align with the intended visual narrative.

Two distinct fine-tuning phases were conducted: one targeting character identity and another for shot-type specificity. Experiments demonstrated that the model successfully learned to render coherent characters across multiple scenes and generate shots with a high degree of alignment to cinematic conventions. We also explored the use of captions on both shot type and character-specific LoRA training, showing their degree of impact on controllability and semantic accuracy.

The proposed prompt generation strategy leverages modern LLMs to further streamline the process, especially the sequence generation step. To achieve this goal, we empirically built a prompt for the LLM to convert broad scene descriptions into detailed and carefully crafted prompts, one for each storyboard frame. This proved effective to allow the system to serve as a practical pre-visualization tool for early-stage cinematic design.

Further refinement was achieved through inpainting steps, which allowed for targeted corrections in individual frames—enhancing local and global coherence in a controlled way that can easily scale to the inclusion of specific props and fine details.

Quantitative metrics such as CLIPScore, DINOv2, and the Image Composition Assessment (ICA) metric were used to evaluate the generated outputs against a Stable Diffusion 1.5 baseline. While CLIPScore results were comparable between the base and fine-tuned models, further analysis revealed that higher scores from the base model often do not correspond to significanlty better alignment. ICA results provided moderate evidence that the fine-tuned model inherited better compositional tendencies—likely influenced by the high-quality, stylistically consistent training data. DINOv2 confirmed a strong semantic similarity between corresponding frames of base and fine-tuned models, reinforcing the fact that both pipelines capture core prompt semantics.

However, key storytelling aspects such as shot-type control, character consistency, and emotional clarity—where the fine-tuned model excels—are not adequately captured by these metrics: qualitative human evaluation consistently highlighted the superiority of the fine-tuned model in generating visually expressive, narratively coherent, and stylistically unified storyboards. The presented pipeline therefore offers a foundation for controllable storyboard generation, bridging the gap between narrative intent and visual execution.

Along with the promising results, also several areas emerged for improvement.

The current system shows limitations in areas such as nuanced control over spatial layout and character positioning, accurate rendering of emotional expressions or physical interactions, and prompt interpretability for complex, multi-character compositions.

For instance, in scenes involving multiple characters, unintended interactions (e.g., hugging or kissing) were often generated even when not specified in the prompt. Moreover, while LoRA provides a lightweight and modular finetuning framework, it struggles when merging multiple concepts into a single frame, requiring careful prompt design and inpainting corrections to limit and correct concept bleeding and concept omission.

Building on these insights, future research could explore the following directions to further elevate the quality and versatility of storyboard generation systems:

- Adopt more advanced diffusion architectures: upgrading from Stable Diffusion 1.5 to more powerful models such as SDXL could enhance prompt understanding, dynamic range, and visual fidelity.
- Perform systematic hyperparameter optimization: current training and inference settings are based on empirical tuning and insights from the online community. A structured grid search could lead to significant performance improvements.
- Explore alternative fine-tuning techniques: a comparative study between DreamBooth, full model fine-tuning, and Textual Inversion may shed light on the performance of different strategies and the trade-offs between control, flexibility, and efficiency.
- Incorporate diverse visual styles and cinematic grading: our work mainly focused on photorealistic storyboards, but supporting cartoon, anime or comic-book styles alongside color grading profiles like “teal and orange” or monochrome can expand the creative potential and application domains.
- Enrich character training datasets: increasing the diversity of character training images—covering more facial expressions, body poses and interactions—may improve generalization capabilities and support the rendering of more complex emotional cues or interactions.
- Integrate ControlNet and region-based prompting: these tools can provide localized, fine-grained control over composition, enabling precise action choreography and interaction design in multi-character scenes, while also tackling the common issues of LoRA merging.
- Explore recent story visualization approaches: it would be valuable to conduct practical experimentation with recently proposed approaches in the field of

story visualization, as discussed in Chapter 5—especially as soon as open-source implementations or reproducible code become available. This would enable a direct comparison and potentially allow for hybrid solutions that combine the strengths of multiple methods.

- Expand inpainting and outpainting capabilities: beyond character fixes, inpainting can be used to integrate objects, refine environments and adjust spatial elements, while outpainting could help maintain continuity across scene transitions or larger panoramic shots.
- Introduce sketch-based or layout-guided conditioning: using sketches, segmentation masks or spatial maps to guide generation can provide more deterministic control over composition, especially in complex scenes.

An exciting direction for future work is also the transition from static images generation to dynamic video creation. Leveraging emerging text-to-video models opens the door to generating animated storyboards (animatics), or even full video clips, conditioned on script segments or scene descriptions.

Bibliography

[1] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: `1312.6114 [stat.ML]`. URL: `https://arxiv.org/abs/1312.6114` (cit. on p. 5).

[2] Lilian Weng. *From Autoencoder to Beta-VAE*. 2018. URL: `https://lilianweng.github.io/posts/2018-08-12-vae/` (cit. on pp. 6, 7).

[3] Ravislay Wade. *Visualizing MNIST using a Variational Autoencoder*. `https://www.kaggle.com/rvislaywade/visualizingmnist-using-a-variational-autoencoder`. 2020 (cit. on p. 7).

[4] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. *Neural Discrete Representation Learning*. 2018. arXiv: `1711.00937 [cs.LG]`. URL: `https://arxiv.org/abs/1711.00937` (cit. on pp. 7, 8).

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. 2014. arXiv: `1406.2661 [stat.ML]`. URL: `https://arxiv.org/abs/1406.2661` (cit. on p. 8).

[6] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. arXiv: `1411.1784 [cs.LG]`. URL: `https://arxiv.org/abs/1411.1784` (cit. on p. 9).

[7] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*. 2017. arXiv: `1612.03242 [cs.CV]`. URL: `https://arxiv.org/abs/1612.03242` (cit. on p. 9).

[8] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: `1511.06434 [cs.LG]`. URL: `https://arxiv.org/abs/1511.06434` (cit. on p. 9).

[9] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. arXiv: `1701.07875 [stat.ML]`. URL: `https://arxiv.org/abs/1701.07875` (cit. on p. 10).

[10] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2019. arXiv: `1812.04948 [cs.NE]`. URL: `https://arxiv.org/abs/1812.04948` (cit. on p. 10).

[11] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. *Pixel Recurrent Neural Networks*. 2016. arXiv: `1601.06759 [cs.CV]`. URL: `https://arxiv.org/abs/1601.06759` (cit. on p. 10).

[12] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. *Conditional Image Generation with PixelCNN Decoders*. 2016. arXiv: `1606.05328 [cs.CV]`. URL: `https://arxiv.org/abs/1606.05328` (cit. on p. 10).

[13] Saurabh Mishra Harshit Sharma. *Auto-Regressive Generative Models (PixelRNN, PixelCNN++)*. `https://medium.com/p/32d192911173`. 2017 (cit. on p. 10).

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: `2006.11239 [cs.LG]`. URL: `https://arxiv.org/abs/2006.11239` (cit. on pp. 11, 23, 105).

[15] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. arXiv: `2105.05233 [cs.LG]`. URL: `https://arxiv.org/abs/2105.05233` (cit. on pp. 11, 38).

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: `2112.10752 [cs.CV]`. URL: `https://arxiv.org/abs/2112.10752` (cit. on p. 11).

[17] Runway ML Stability AI. *Stable Diffusion Public Release*. `https://stability.ai/blog/stable-diffusion-public-release`. Accessed: 23-May-2023. 2022 (cit. on pp. 11, 41, 105).

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: `1706.03762 [cs.CL]`. URL: `https://arxiv.org/abs/1706.03762` (cit. on pp. 11, 12).

[19] OpenAI. *DALL · E 2*. `https://openai.com/index/dall-e-2/`. Accessed: March 2025. 2022 (cit. on p. 11).

[20] Chitwan Saharia et al. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. 2022. arXiv: `2205.11487 [cs.CV]`. URL: `https://arxiv.org/abs/2205.11487` (cit. on pp. 11, 55).

[21] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. *Stand-Alone Self-Attention in Vision Models*. 2019. arXiv: 1906.05909 [cs.CV]. URL: https://arxiv.org/abs/1906.05909 (cit. on p. 15).

[22] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929 (cit. on p. 20).

[23] Niels Rogge and Kashif Rasul. *The Annotated Diffusion Model*. https://huggingface.co/blog/annotated-diffusion. 2022 (cit. on p. 26).

[24] Sergey Zagoruyko and Nikos Komodakis. *Wide Residual Networks*. 2017. arXiv: 1605.07146 [cs.CV]. URL: https://arxiv.org/abs/1605.07146 (cit. on p. 26).

[25] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. 2021. arXiv: 2102.09672 [cs.LG]. URL: https://arxiv.org/abs/2102.09672 (cit. on pp. 28, 46).

[26] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020 (cit. on p. 35).

[27] Jonathan Ho and Tim Salimans. *Classifier-Free Diffusion Guidance*. 2022. arXiv: 2207.12598 [cs.LG]. URL: https://arxiv.org/abs/2207.12598 (cit. on p. 38).

[28] Onkar Mishra. *Stable Diffusion Explained*. Medium. Accessed: March 19, 2025. 2022. URL: https://medium.com/@onkarmishra/stable-diffusion-explained-1f101284484d (cit. on p. 41).

[29] Christoph Schuhmann et al. *LAION-5B: An open large-scale dataset for training next generation image-text models*. 2022. arXiv: 2210.08402 [cs.CV]. URL: https://arxiv.org/abs/2210.08402 (cit. on p. 42).

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. «High-Resolution Image Synthesis With Latent Diffusion Models». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695 (cit. on p. 43).

[31] Stability AI. *Stable Diffusion 2.0 Release*. https://stability.ai/news/stable-diffusion-v2-release. Accessed: March 2025. 2022 (cit. on p. 44).

[32] 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*. 2023. arXiv: 2307.01952 [cs.CV]. URL: https://arxiv.org/abs/2307.01952 (cit. on p. 45).

[33] Stability AI. *Stability AI Announces SDXL Turbo.* Online. Accessed: March 19, 2025. 2023. URL: `https://stability.ai/news/stability-ai-sdxl-turbo` (cit. on p. 46).

[34] Alex 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.* 2022. arXiv: `2112.10741 [cs.CV]`. URL: `https://arxiv.org/abs/2112.10741` (cit. on p. 46).

[35] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. *Autoregressive Diffusion Models.* 2022. arXiv: `2110.02037 [cs.LG]`. URL: `https://arxiv.org/abs/2110.02037` (cit. on p. 47).

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. *Zero-Shot Text-to-Image Generation.* 2021. arXiv: `2102.12092 [cs.CV]`. URL: `https://arxiv.org/abs/2102.12092` (cit. on p. 49).

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents.* 2022. arXiv: `2204.06125 [cs.CV]`. URL: `https://arxiv.org/abs/2204.06125` (cit. on p. 50).

[38] OpenAI. *DALL · E 3.* Research Report. Accessed: March 2025. OpenAI, 2023. URL: `https://cdn.openai.com/papers/dall-e-3.pdf` (cit. on p. 52).

[39] Midjourney. *Midjourney.* `https://www.midjourney.com/home`. 2023 (cit. on p. 53).

[40] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. *An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion.* 2022. arXiv: `2208.01618 [cs.CV]`. URL: `https://arxiv.org/abs/2208.01618` (cit. on p. 57).

[41] 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.* 2023. arXiv: `2208.12242 [cs.CV]`. URL: `https://arxiv.org/abs/2208.12242` (cit. on pp. 59, 106).

[42] 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.* 2021. arXiv: `2106.09685 [cs.CL]`. URL: `https://arxiv.org/abs/2106.09685` (cit. on pp. 62, 106).

[43] David Ha, Andrew Dai, and Quoc V. Le. *HyperNetworks.* 2016. arXiv: `1609.09106 [cs.LG]`. URL: `https://arxiv.org/abs/1609.09106` (cit. on p. 62).

[44] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. *Multi-Concept Customization of Text-to-Image Diffusion*. 2023. arXiv: 2212.04488 [cs.CV]. URL: https://arxiv.org/abs/2212.04488 (cit. on p. 64).

[45] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. *Adding Conditional Control to Text-to-Image Diffusion Models*. 2023. arXiv: 2302.05543 [cs.CV]. URL: https://arxiv.org/abs/2302.05543 (cit. on p. 66).

[46] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. *StoryGAN: A Sequential Conditional GAN for Story Visualization*. 2019. arXiv: 1812.02784 [cs.CV]. URL: https://arxiv.org/abs/1812.02784 (cit. on p. 72).

[47] Wen Wang, Canyu Zhao, Hao Chen, Zhekai Chen, Kecheng Zheng, and Chunhua Shen. *AutoStory: Generating Diverse Storytelling Images with Minimal Human Effort*. 2023. arXiv: 2311.11243 [cs.CV]. URL: https://arxiv.org/abs/2311.11243 (cit. on p. 74).

[48] Xichen Pan, Pengda Qin, Yuhong Li, Hui Xue, and Wenhu Chen. *Synthesizing Coherent Story with Auto-Regressive Latent Diffusion Models*. 2022. arXiv: 2211.10950 [cs.CV]. URL: https://arxiv.org/abs/2211.10950 (cit. on p. 76).

[49] Adyasha Maharana, Darryl Hannan, and Mohit Bansal. *StoryDALL-E: Adapting Pretrained Text-to-Image Transformers for Story Continuation*. 2022. arXiv: 2209.06192 [cs.CV]. URL: https://arxiv.org/abs/2209.06192 (cit. on p. 77).

[50] Zhaohui Liang, Xiaoyu Zhang, Kevin Ma, Zhao Liu, Xipei Ren, Kosa Goucher-Lambert, and Can Liu. *StoryDiffusion: How to Support UX Storyboarding With Generative-AI*. 2024. arXiv: 2407.07672 [cs.HC]. URL: https://arxiv.org/abs/2407.07672 (cit. on p. 78).

[51] Xiaoqian Shen and Mohamed Elhoseiny. *StoryGPT-V: Large Language Models as Consistent Story Visualizers*. 2023. arXiv: 2312.02252 [cs.CV]. URL: https://arxiv.org/abs/2312.02252 (cit. on p. 80).

[52] Huiguo He et al. *DreamStory: Open-Domain Story Visualization by LLM-Guided Multi-Subject Consistent Diffusion*. 2025. arXiv: 2407.12899 [cs.CV]. URL: https://arxiv.org/abs/2407.12899 (cit. on p. 81).

[53] Storyboarder AI. *Storyboarder AI - AI-Driven Storyboard Generation Platform*. https://storyboarder.ai. Accessed: 2025-04-05 (cit. on p. 83).

[54] Boords. *Boords - Online Storyboarding Software*. https://boords.com. Accessed: 2025-04-05 (cit. on p. 83).

[55] Katalist AI. *Katalist AI - AI-powered tools for storytelling and creative planning.* https://katalist.ai. Accessed: 2025-04-05 (cit. on p. 83).

[56] Yuchao Gu et al. *Mix-of-Show: Decentralized Low-Rank Adaptation for Multi-Concept Customization of Diffusion Models.* 2023. arXiv: 2305.18292 [cs.CV]. URL: https://arxiv.org/abs/2305.18292 (cit. on p. 89).

[57] Sangwon Jang, Jaehyeong Jo, Kimin Lee, and Sung Ju Hwang. *Identity Decoupling for Multi-Subject Personalization of Text-to-Image Models.* 2024. arXiv: 2404.04243 [cs.CV]. URL: https://arxiv.org/abs/2404.04243 (cit. on p. 92).

[58] Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan. *Isolated Diffusion: Optimizing Multi-Concept Text-to-Image Generation Training-Freely with Isolated Diffusion Guidance.* 2025. arXiv: 2403.16954 [cs.CV]. URL: https://arxiv.org/abs/2403.16954 (cit. on p. 93).

[59] Guangxuan Xiao, Tianwei Yin, William T. Freeman, Frédo Durand, and Song Han. *FastComposer: Tuning-Free Multi-Subject Image Generation with Localized Attention.* 2023. arXiv: 2305.10431 [cs.CV]. URL: https://arxiv.org/abs/2305.10431 (cit. on p. 95).

[60] Yang Yang et al. *LoRA-Composer: Leveraging Low-Rank Adaptation for Multi-Concept Customization in Training-Free Diffusion Models.* 2024. arXiv: 2403.11627 [cs.CV]. URL: https://arxiv.org/abs/2403.11627 (cit. on p. 98).

[61] Tuna Han Salih Meral, Enis Simsar, Federico Tombari, and Pinar Yanardag. *CLoRA: A Contrastive Approach to Compose Multiple LoRA Models.* 2024. arXiv: 2403.19776 [cs.CV]. URL: https://arxiv.org/abs/2403.19776 (cit. on p. 100).

[62] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. *Subject-Diffusion:Open Domain Personalized Text-to-Image Generation without Test-time Fine-tuning.* 2024. arXiv: 2307.11410 [cs.CV]. URL: https://arxiv.org/abs/2307.11410 (cit. on p. 102).

[63] FilmGrab. *FilmGrab.* https://film-grab.com/. Accessed: March 2025. 2023 (cit. on p. 107).

[64] Kaggle. *Kaggle.* https://www.kaggle.com/. 2023 (cit. on p. 112).

[65] Hugging Face. *Diffusers.* https://github.com/huggingface/diffusers. 2023 (cit. on p. 114).

[66] Meta AI. *Segment Anything.* https://segment-anything.com/. 2023 (cit. on p. 121).

[67] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. «CLIPScore: A Reference-free Evaluation Metric for Image Captioning». In: *EMNLP*. 2021 (cit. on p. 138).

[68] Bo Zhang, Li Niu, and Liqing Zhang. «Image Composition Assessment with Saliency-augmented Multi-pattern Pooling». In: *arXiv preprint arXiv:2104.03133* (2021) (cit. on p. 138).

[69] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision.* 2023 (cit. on p. 138).