A Systematic Evaluation of Independent Strategies for Enhancing Text-to-Image Semantic Alignment in Stable Diffusion

Xinxie Wu
Department of Computer Science
Stanford University, USA
xinxiewu@stanford.edu

Rahul Venkatesh (Project Mentor)
Department of Computer Science
Stanford University, USA
rmvenkat@stanford.edu

Abstract

Diffusion-based generative models have significantly advanced the field of text-to-image generation, yet achieving fine-grained semantic alignment between textual prompts and generated images remains challenging. In this project, we present a systematic evaluation of several independent strategies designed to enhance semantic alignment within the Stable Diffusion pipeline. Our study examines template-based prompt engineering, CLIP-based output filtering, LoRA fine-tuning, CLIP-guided latent optimization, and DiffusionCLIP fine-tuning, using the 2014 MS-COCO Captions dataset and Stable Diffusion v1.4.

Evaluated quantitatively by CLIPScore, our experiments show that the average CLIPScore improves from 0.162 (baseline) to 0.165 (+0.003) with prompt engineering, 0.173 (+0.011) with output filtering, 0.196 (+0.034) after LoRA tuning, and 0.203 (+0.041) using CLIP-guided latent optimization. For DiffusionCLIP, the qualitative visual analyses demonstrate clearer alignment after fine-tuning. Our findings validate the effectiveness of modular improvements across training and inference stages, providing actionable insights for future research and practical applications of text-to-image diffusion models.

1. Introduction

Text-to-image generation is a core task in computer vision, with extensive applications in creative media, automated content generation, assistive technologies, and interactive systems. Recent latent diffusion models, such as Stable Diffusion, have significantly improved image quality and realism. Despite these developments, challenges persist in achieving precise semantic alignment, visual coherence, and effective controllability of generated outputs. Addressing these issues is crucial not only for enhancing the usability and reliability of generated content but also for ensuring broader applicability in real-world scenarios.

This project systematically implements and evaluates multiple independent strategies to enhance semantic alignment within the Stable Diffusion pipeline, leveraging the rich textual annotations of the 2014 MS-COCO Captions dataset. Specifically, we independently implement and evaluate five distinct strategies: (1) template-based prompt engineering to provide clearer textual context; (2) CLIP-based output filtering to select the best semantic matches from multiple generated outputs; (3) lightweight LoRA fine-tuning of the Stable Diffusion model for improved semantic understanding; (4) CLIP-guided latent optimization during the inference phase to dynamically adjust generation towards higher semantic fidelity; (5) DiffusionCLIP and the corresponding fine-tuning to enhance semantic coherence.

The inputs to our methods are textual prompts randomly selected from the 2014 MS-COCO dataset, $10,000$ for fine-tuning and additional 100 for evaluation. Our outputs are synthesized images with a resolution of 512×512 pixels. To rigorously evaluate these strategies, we use CLIPScore as a quantitative metric, complemented by qualitative visual assessments. By systematically isolating and comparing the effectiveness of each enhancement strategy, we provide a comprehensive understanding of their individual and relative impacts.

Our experimental results demonstrate incremental improvements in semantic alignment from a baseline average CLIPScore of 0.1622 to 0.1649 $(+0.0027)$ with prompt engineering, 0.1734 $(+0.0112)$ with output filtering, 0.1958 $(+0.0336)$ after LoRA fine-tuning, and 0.2034 $(+0.0412)$ using CLIP-guided latent optimization. Qualitative assessments further validate these findings, particularly highlighting the visual coherence improvements achieved through DiffusionCLIP fine-tuning.

This project performs a structured and extensive comparative analysis of modular strategies for improving semantic alignment in text-to-image diffusion models, providing clear insights and guidelines for future research directions and practical applications of text-to-image generation technologies.

2. Related Work

This project builds on several recent works in text-to-image generation, image-text alignment evaluation, and efficient model adaptation:

Rombach et al. (2022)[9] introduce the *Latent Diffusion Model*, which performs diffusion in a compressed latent space rather than pixel space to drastically improve memory efficiency and resolution. This foundational idea underpins the design of Stable Diffusion $v1.4$ which we adopt. While its efficiency allows democratized high-quality image generation, the reliance on CLIP's fixed pre-trained embeddings can limit semantic precision. This motivates several of our modular improvements that target controllability and alignment without sacrificing model compactness.

Saharia et al. (2022)[10] present *Imagen*, a photorealistic diffusion model that integrates large language models like T5 for prompt understanding. *Imagen* emphasizes the importance of language-model-conditioned synthesis and deep cross-modal integration. Its reliance on powerful proprietary models also highlights the trade-off between performance and reproducibility in academic settings.

Hessel et al. (2021)[5] propose *CLIPScore*, a reference-free metric that uses CLIP embeddings to quantify how well an image aligns semantically with a given prompt. Unlike BLEU or CIDEr, CLIPScore captures perceptual similarity grounded in joint vision-language space. While highly informative, CLIPScore has limitations in edge cases, such as unusual phrasing or abstract scenes, which necessitate complementary visual inspection. It remains our primary quantitative metric, offering a convenient and reproducible benchmark to evaluate semantic alignment improvements across various methods. Additionally, we provide image comparisons in our project.

Hertz et al. (2023)[4] introduce *Prompt-to-Prompt*, a method that manipulates cross-attention weights to enable attribute-specific control in image generation. Though computationally involved, their work underscores the power of prompt-level control. Our use of templated prompt engineering stems from similar goals but is executed without modifying model internals, showing that even simple linguistic interventions can lead to measurable semantic gains. Compared to attention manipulation techniques, our approach trades off granular control for simplicity and scalability.

Hao et al. (2023)[3] explore *prompt optimization* through automated search and alignment models, proposing methods that automatically refine prompts for better generation quality. Their work confirms that textual design is a crucial and underexploited lever for improving diffusion model outputs. While their framework is automated and leverages gradient-based updates, our handcrafted templates represent a practical and lightweight alternative, especially suitable in low-resource or prototyping settings.

Hu et al. (2022)[6] propose *LoRA*, a method for low-rank adaptation of pre-trained transformers and diffusion networks. By injecting rank-constrained updates, LoRA drastically reduces the parameter count for fine-tuning. This approach enables us to selectively fine-tune Stable Diffusion's Unet layers using limited compute, making experimentation feasible and efficient while maintaining generalization. LoRA exemplifies a broader trend in efficient fine-tuning, striking a balance between flexibility, accessibility, and performance.

Gal et al. (2023)[2] present *Textual Inversion*, a technique for learning new concepts as special tokens through few-shot tuning. Though not directly applied in our project, it informs the broader context of fine-tuning approaches. Unlike LoRA, which tunes model weights, Textual Inversion updates token embeddings, making it ideal for personalization but less suited for broad semantic enhancements like those required in MS-COCO. Nonetheless, it opens avenues for integrating learned prompts into broader pipelines.

Kim et al. (2022)[7] introduce *DiffusionCLIP*, which incorporates CLIP-based loss during diffusion model fine-tuning to enhance semantic fidelity. Their method supports controlled editing and alignment, demonstrating the effectiveness of CLIP supervision. We draw from their ideas in our own fine-tuning branch, using CLIP loss to iteratively refine outputs and enable more coherent visual representations. This approach reflects a hybrid of inference-time guidance and training-time supervision, bridging two common enhancement paradigms.

Liu et al. (2022)[8] present *Composable Diffusion Models*, allowing the user to generate images from multiple prompts in a modular way, leveraging CLIP to guide and combine latent trajectories. This idea of latent refinement closely parallels our inference-time CLIP-guided generation, reinforcing the utility of CLIP in post-generation semantic corrections without retraining the model. Compositionality also highlights how fine-grained control can emerge from inference techniques rather than architectural changes.

Crowson et al. (2022)[1] propose *VQGAN-CLIP*, one of the earliest models to use CLIP for open-domain text-to-image synthesis by optimizing latent codes through gradient ascent. Though replaced by more advanced diffusion-based techniques, it remains historically important and conceptually foundational to our use of CLIP-based filtering and guidance at inference. It also demonstrates how optimization in the latent space serves as a powerful method for aligning generation in the absence of retraining.

These works form the theoretical scaffolding. By synthesizing insights, our approach is distinguished by its modularity, offering practical contributions to the ongoing discourse on the text-to-image synthesis.

3. Methods

Our approach integrates a series of modular enhancements to the Stable Diffusion framework to improve the semantic fidelity and controllability of generated images. The pipeline is composed of five key components: prompt engineering, output filtering, LoRA-based fine-tuning, CLIP-guided inference, and DifussionCLIP fine-tuning. Each module contributes independently to the generation quality while remaining compatible with the overall diffusion architecture.

3.1. Baseline Setup

The baseline is defined by a vanilla Stable Diffusion **v1.4** model, released by CompVis and accessed through the Hugging Face diffusers library. Stable Diffusion **v1.4** model is pre-trained on LAION-2B, generating images from raw prompts without any post-processing or tuning.

Although utilized the model from Hugging Face, we would like to discuss the computational graph. Stable Diffusion operates as a Latent Diffusion Model (LDM), in which the denoising process occurs in a compressed latent space z rather than directly in pixel space. This significantly reduces memory and computation requirements while preserving semantic structure.

3.1.1 Forward Process

Starting from a clean latent image z_0, noise is added over T steps using a fixed variance schedule $\beta_t{}_{t=1}^{T}$. The latent at step t is:

$$q(z_t \mid z_0) = \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t} z_0, (1 - \bar{\alpha}_t)\mathbf{I}) \tag{1}$$

where $\bar{\alpha}t = \prod s = 1^t \alpha_s$ and $\alpha_t = 1 - \beta_t$.

3.1.2 Reverse Process

A neural network $\epsilon_\theta(z_t, t)$ is trained to predict the added noise, minimizing the simplified loss:

$$\mathcal{L}\text{simple} = \mathbb{E} z_0, \epsilon, t \left[|\epsilon - \epsilon_\theta(z_t, t)|^2 \right] \tag{2}$$

3.1.3 Classifier-Free Guidance

To incorporate conditioning information c (such as a text prompt), classifier-free guidance interpolates between conditional and unconditional predictions:

$$\epsilon_{\text{guided}} = (1 + w) \cdot \epsilon_\theta(z_t, t, c) - w \cdot \epsilon_\theta(z_t, t, \emptyset) \tag{3}$$

where w is the guidance scale (7.5 in our baseline), and $\emptyset$ denotes the null condition.

3.1.4 DDIM Sampling

For efficiency, we employ Deterministic DDIM sampling. Given predicted noise $\hat{\epsilon}\theta$ and current latent z_t, the next latent $zt - 1$ is estimated as:

$$z_{t-1} = \sqrt{\bar{\alpha}t - 1} z_0 + \sqrt{1 - \bar{\alpha}t - 1} \cdot \hat{\epsilon}_\theta \tag{4}$$

This allows high-quality generation in significantly fewer steps.

In this project, each textual prompt generates a single image of resolution 512×512 using DDIM sampling with 50 steps and a guidance scale of 7.5.

Quantitative evaluation is conducted using CLIPScore, as introduced by [5], which computes cosine similarity between CLIP-encoded image and text embeddings:

$$\text{CLIPScore}(I, T) = \cos\left(\phi_{\text{image}}(I), \phi_{\text{text}}(T)\right) \tag{5}$$

where ϕ_{image} and ϕ_{text} are the respective CLIP encoders.

3.2. Prompt Engineering

We apply template-based prompt engineering to enhance the specificity and expressiveness of input prompts, thereby guiding the model toward more semantically faithful and visually coherent outputs. Inspired by [4] but applied without architectural modification, we modified the input text rather than the generation process.

Specifically, we define three stylistic prompt templates: *photorealistic*, *cinematic*, and *nature-oriented*. To select the appropriate template, we implement a lightweight keyword-based classification: each input caption is scanned for terms associated with animals, food, or natural environments, and matched to the corresponding style.

Formally, let c_i be the raw caption and P the template, then the engineered prompt $c'_i = P(c_i)$. The model then performs:

$$I_i = \text{SD}\theta(c'_i) \tag{6}$$

where SDθ denotes the Stable Diffusion model. We evaluate semantic gains by comparing CLIPScore before and after prompt engineering.

3.3. Output Filtering

Instead of altering the generation pipeline, output filtering improves alignment by re-ranking multiple candidate images. For each caption, we sample $k = 5$ outputs $I_i^1, \ldots, I_i^k$ and compute their CLIPScores $s_i^1, \ldots, s_i^k$. The final image is selected via:

$$I_i^* = \arg\max_j \text{CLIPScore}(I_i^j, c_i) \tag{7}$$

This zero-cost method directly leverages the stochasticity of diffusion outputs to choose more semantically aligned results.

3.4. LoRA Fine-Tuning

3.4.1 Architecture and Parameters

To efficiently adapt the model to our specific dataset, we adopt the Low-Rank Adaptation (LoRA) technique [6], which injects trainable low-rank matrices (A, B) into existing linear layers W as:

$$W' = W + \alpha AB \tag{8}$$

where $A \in \mathbb{R}^{d\times r}$, $B \in \mathbb{R}^{r\times d}$, and $r \ll d$. We apply LoRA modules to attention layers in the Unet component of Stable Diffusion, enabling parameter-efficient fine-tuning that preserves the core generative capacity of the pre-trained model.

3.4.2 Training Configuration

To adapt the model toward better semantic alignment, we fine-tune on $10,000$ image-caption pairs from the MS-COCO 2014 dataset using batch size 8, learning rate $5e-5$, rank $r = 4$, scaling factor $\alpha = 16$, and 10 epochs. Only the attention weights in the Unet are updated; the VAE and CLIP encoders remain frozen. Training is on RTX 3080.

3.5. CLIP-Guided Generation

3.5.1 Inference-Time Reranking

To optimize semantic consistency at inference time, we incorporate CLIP-guided reranking into the latent sampling stage. For each prompt c_i, we generate multiple latent samples $z_i^1, \ldots, z_i^k$ and decode each to its image I_i^j using the VAE. CLIPScores $s_i^j = \text{CLIPScore}(I_i^j, c_i)$ are computed and used to select the image I_i^j with the highest alignment score:

$$I_i^j = \arg\max_j \text{CLIPScore}(I_i^j, c_i) \tag{9}$$

This method guides sampling toward optimal semantic outputs without modifying the model weights.

3.5.2 CLIP-Guided Latent Optimization

Beyond post-hoc selection, we integrate CLIP feedback directly into the denoising trajectory to optimize latent codes during generation. Following approaches inspired by CLIP guidance in prior works [1], we decode the latent z_t every n steps (e.g., every 25 iterations) into an image I_t, compute its CLIP similarity score $\text{sim}(I_t, c_i)$ with the input caption c_i, and use its gradient to update the latent.

Let $f_{\text{img}}(I_t)$ and $f_{\text{text}}(c_i)$ be the normalized CLIP image and text embeddings. The CLIP-based alignment loss is:

$$\mathcal{L}\text{CLIP} = -\lambda \cdot \cos\left(f\text{img}(I_t), f_{\text{text}}(c_i)\right) \tag{10}$$

where λ is the guidance scale. The gradient $\nabla_{z_t}\mathcal{L}_{\text{CLIP}}$ is computed and used to nudge z_t toward a direction that improves semantic consistency. This yields an updated latent:

$$z_t \leftarrow z_t - \eta \cdot \nabla_{z_t}\mathcal{L}_{\text{CLIP}} \tag{11}$$

where η is an implicit learning rate. This technique enables fine-grained, inference-time control without modifying model weights or requiring additional training.

3.6. DiffusionCLIP

3.6.1 Computational Graph

DiffusionCLIP's computational graph extends that of the standard Stable Diffusion model by incorporating a differentiable loop over the entire denoising trajectory. Unlike traditional inference-only pipelines, gradients flow through UNet during both forward and reverse diffusion steps.

- Encoding reference image I_{ref} to latent z_0 via the VAE
- Performing reverse noising to obtain z_t from z_0:

$$z_t = \sqrt{\bar{\alpha}t}z_0 + \sqrt{1-\bar{\alpha}t}\cdot\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{12}$$

- Using UNet to predict noise at each t, producing z_{t+1}:

$$\hat{\epsilon}\theta = \text{UNet}\theta(z_t, t, \text{condition} = \emptyset) \tag{13}$$

$$zt - 1 = \text{Scheduler}(\hat{\epsilon}_\theta, z_t, t) \tag{14}$$

- Reconstructing intermediate outputs and comparing them to textual semantics using directional CLIP loss.
- Backpropagating gradients through CLIP, UNet, and diffusion scheduler.

This structure allows the optimization process to refine generative trajectories such that semantic shifts in the latent space mirror the desired changes in text.

3.6.2 Loss Function and Finetune Iterations

We adopt DiffusionCLIP [7] to further improve semantic alignment by injecting CLIP-based gradients into the diffusion training loop. DiffusionCLIP aligns text and image semantics by optimizing a directional CLIP loss within the diffusion framework. Let $f_{\text{img}}(\cdot)$ and $f_{\text{text}}(\cdot)$ denote CLIP encoders. Given reference image I_{ref}, generated image I_{gen}, reference caption c_{ref}, and target caption c_{tar}, we define the directional loss as:

$$\mathcal{L}\text{CLIP} = 1 - \cos\left(\Delta I, \Delta T\right) \tag{15}$$

$$\Delta I = f\text{img}(I_{\text{gen}}) - f_{\text{img}}(I_{\text{ref}}) \tag{16}$$

$$\Delta T = f_{\text{text}}(c_{\text{tar}}) - f_{\text{text}}(c_{\text{ref}}) \tag{17}$$

This loss encourages the direction of change in image space to align with the intended direction of change in semantic space.

Due to hardware and time constraints, we fine-tune on 5 image-caption pairs using $t_0 = 30$, $ft_iters = 10$ and $lr = 3e-4$ on RTX 3080.

4. Dataset and Evaluation Metrics

4.1. Dataset: MS-COCO Captions Subset

Our experiments are based on the MS-COCO 2014 Captions dataset[1], a widely adopted benchmark for tasks involving image-text alignment. The dataset contains over $120,000$ natural images, each annotated with five human-generated captions describing salient objects and scenes. To balance computational feasibility and diversity, we randomly sample $10,000$ image-caption pairs from the training split for fine-tuning, and reserve an additional $1,000$ distinct samples for evaluating inference-time strategies. For DiffusionCLIP, which involves gradient-based optimization through multiple denoising steps, we select 5 images from the fine-tuning subset and 10 samples from the evaluation subset, due to its significantly higher computational cost.

Before model input, each image is resized to a fixed resolution of 512×512 pixels using bicubic interpolation, aligning with the latent resolution expectations of Stable Diffusion $v1.4$. Pixel values are normalized to the $[-1, 1]$ range, matching the input domain of the VAE encoder. For captions, we employ the CLIP tokenizer (from ViT-L/14) to tokenize each sentence, truncating or padding to a uniform sequence length of 77 tokens as required by the text encoder. We also remove invalid samples such as corrupted images, empty captions, or samples containing special characters incompatible with the tokenizer.

No explicit data augmentation or external features (e.g., HOG, PCA, etc.) are applied, as the primary objective is to evaluate prompt-based and latent-space alignment strategies rather than train large vision-language models from scratch. Nonetheless, the inherent linguistic variability and visual richness in MS-COCO provide sufficient complexity to stress-test our semantic enhancement modules. Representative examples from the dataset span common scenes such as "a dog running through a grassy field," "a bowl of soup placed on a wooden table," and "people walking down a busy city street," allowing us to examine how well various interventions improve fine-grained visual grounding.

4.2. Evaluation Metric: CLIPScore (ViT-L/14)

We use CLIPScore [5] as the primary metric to evaluate semantic alignment between generated images and textual prompts. CLIPScore is defined as the cosine similarity between CLIP-encoded image and text representations. All evaluations are conducted using the CLIP ViT-L/14 model to ensure consistency and sensitivity to fine-grained alignment. Higher CLIPScores indicate better semantic coherence between the image and its caption. For each generation method, we report mean CLIPScore across the evaluation set, and visualize score distributions using boxplots and histograms to facilitate comparative analysis.

[1] http://cocodataset.org

5. Experiments / Results / Discussions

We present a comprehensive analysis of our experimental results, evaluating each enhancement strategy using both quantitative metrics and qualitative image comparisons. The discussion proceeds through baseline performance, incremental improvements across our pipeline, and final outcomes from fine-tuning-based methods.

5.1. Experimental Setup

All experiments are conducted on RTX 3080. We evaluate methods on a consistent set of 100 image-caption pairs sampled from the MS-COCO 2014 validation split. Each method—including Prompt Engineering, Output Filtering, CLIP-Guided Reranking, and LoRA Fine-Tuning—is tested in isolation against the baseline Stable Diffusion $v1.4$ model. DiffusionCLIP is evaluated by the step-by-step image comparison qualitatively.

We apply consistent hyperparameters across methods where applicable:

- For **Baseline** Stable Diffusion $v1.4$ model, we use 50 inference steps and a guidance scale of 7.5;
- For **Prompt Engineering**, fixed templates are applied without stochastic variation;
- **Output Filtering** generates 5 candidate images per caption and selects the one with the highest CLIP-Score, with hyperparameters the same to the baseline's default values;
- **LoRA Fine-Tuning** is performed over 10 epochs with learning rate 5×10^{-5}, batch size 8, rank $r = 4$, and scaling factor $\alpha = 16$. The fine-tuning is based on $10,000$ samples;
- **CLIP-Guided Generation** uses CLIP ViT-L/14, with the latent height and width as $\frac{512}{8} = 64$, inference steps as 50, guidance scale as 0.5 and guidance interval as 25;
- **DiffusionCLIP** applies 10 iterations of directional fine-tuning using CLIP loss, with learning rate $3e-4$ and DDIM inversion over 30 steps. The fine-tuning is based on 5 images.

Our primary quantitative evaluation metric is CLIP-Score [5], computed using the CLIP ViT-L/14 model. This metric measures the cosine similarity between the image and text embeddings in the CLIP latent space, providing an effective proxy for semantic alignment. We report mean scores, standard deviation, and statistical distribution summaries. We also include side-by-side image comparisons to highlight perceptual differences.

5.2. CLIPScore Results

Figure 1 summarize the improvements in mean CLIPScore across all methods. As a baseline, images generated by Stable Diffusion $v1.4$ achieve a mean CLIPScore of 0.1622. Structured **Prompt Engineering** slightly improves alignment, yielding an average score of 0.1649 $(+0.0027)$. **Output Filtering**, which selects the best among five candidate images per caption, increases the average score to 0.1734 $(+0.0112)$, reflecting the effectiveness of selection-based inference.

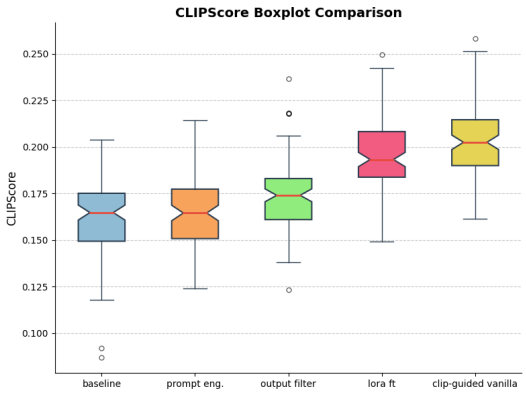

Figure 1. CLIP Score Comparison by Boxplot

We observe further gains from training-time methods. LoRA fine-tuning on $10,000$ samples improves the average CLIPScore to 0.1958 $(+0.0336)$, confirming that adapting attention layers, even with low-rank updates, leads to stronger semantic encoding. CLIP-guided generation, which applies latent-space optimization using CLIP similarity gradients during inference, achieves the highest score of 0.2034—an improvement of $+0.0412$ over the baseline.

To provide a more granular view of model performance, we report both aggregate and per-sample CLIPScore statistics. Figure 1 presents a boxplot comparison across all methods, clearly showing the progressive shift toward higher scores and tighter distributions as enhancement strategies become more advanced. Notably, CLIP-guided generation and LoRA fine-tuning not only exhibit the highest medians but also the smallest interquartile ranges, suggesting strong consistency in semantic alignment.

Methodology Distributuons					
Method	**Mean**	**Median**	**STD**	**25%**	**75%**
Baseline	0.1622	0.1647	0.0212	0.1493	0.1750
Prompt Eng.	0.1649	0.1645	0.0182	0.1507	0.1773
Output Filter	0.1734	0.1740	0.0190	0.1609	0.1830
LoRA FT	0.1958	0.1932	0.0191	0.1837	0.2082
Clip-guided	0.2034	0.2024	0.0185	0.1895	0.2156

Figure 2. Methodology Distribution

Figure 2 summarizes the full statistical breakdown including mean, median, standard deviation, and interquartile range $(25\%$ and $75\%)$. These values confirm the visual trend: all enhancement strategies yield measurable improvements over the baseline. Prompt engineering and output filtering offer meaningful gains with relatively low variance, lifting median scores while maintaining tight distributions. LoRA and CLIP-guided methods further consolidate these trends, achieving both superior average performance and high distributional robustness across diverse prompts.

To assess the consistency of improvements across individual samples, we present a per-sample CLIPScore gap analysis in Figure 3. For each method, we compute the proportion of samples that achieve better CLIPScores compared to the baseline, along with the mean and median improvement magnitudes. Prompt engineering improves CLIPScore in 59% of the samples, with a modest mean gain of $+0.003$ and median gain of $+0.002$. Output filtering proves more robust, showing 87% improvement with a mean gain of $+0.01$ and median gain of $+0.008$. LoRA and CLIP-guided generation exhibit the strongest consistency, both achieving 100% sample-wise improvement rates with substantial mean gains $(+0.03$ and $+0.04$ respectively), highlighting their reliability across diverse caption-image scenarios.

CLIPScore Gap, by each sample											
Prompt Eng.			**Output Filter**			**LoRA FT**			**Clip-guided**		
% Imprv	Mean	Median	% Imprv	Mean	Median	% Imprv	Mean	Median	% Imprv	Mean	Median
59%	0.003	0.002	87%	0.01	0.008	100%	0.03	0.03	100%	0.04	0.04

Figure 3. Improvement Stats by Methodologies

Finally, we visualize the training process of LoRA fine-tuning in Figure 4. The plot shows mean squared error (MSE) loss over 10 epochs, revealing rapid convergence in the early stages followed by stable refinement, suggesting effective learning without overfitting.

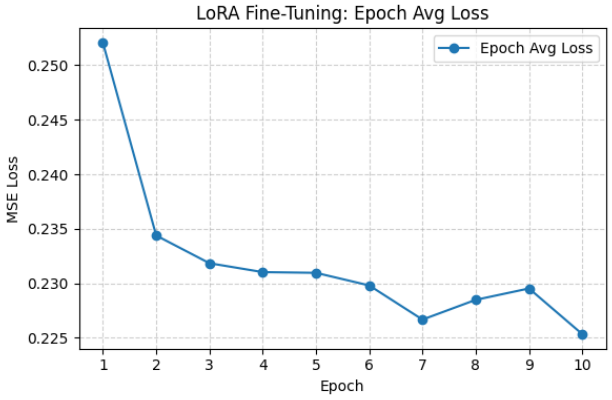

Figure 4. LoRA Training Loss by Epoch

5.3. Image Comparison

We provide direct visual comparisons to qualitatively assess the impact of each enhancement method. We select two representative examples (ID: Caption) to illustrate the evolution of image quality and semantic alignment across our enhancement pipeline.

- 91257: A seagull sitting on the pier with the light house behind him.
- 31757: A man putting on gloves standing with people going skiing and snowboarding.

5.3.1 Baseline and Improvements

Figures 5 and 6 show side-by-side comparisons of the baseline with successive improvement stages. In Figure 5, we observe that the baseline output for sample 91257 lacks spatial coherence: the red-roofed lighthouse is misaligned and blurry, the pier is warped and partially occluded, and the seagull's features are distorted with unnatural edges. Prompt engineering shows partial improvement—while the seagull and lighthouse are now more semantically aligned and the lighting is more photorealistic, the background structure (e.g., the buildings) is missing, reducing contextual richness. Output filtering yields sharper textures on the bird and improved ocean rendering, but introduces a flipped seagull orientation and omits the pier entirely, indicating a trade-off between object clarity and scene completeness.

Figure 5. Baseline, w/ Prompt and Filter

For sample 31757, the baseline output introduces visual inconsistencies: although the snowy setting is preserved, the background cabin is entirely missing, the human figures are stiffly rendered, and the snowboard in the foreground is barely distinguishable. Prompt engineering improves human pose naturalness and adds detail to winter gear such as jackets and helmets, yet the snowboard remains absent and the background building is still missing, limiting contextual depth. Output filtering yields clearer body outlines and facial features, and the snowboard becomes visibly integrated into the foreground. However, the building backdrop continues to be omitted, suggesting limitations in background fidelity across methods.

In Figure 6, LoRA fine-tuning offers substantial improvements in both samples. For sample 91257, the method restores the pier structure, brings the lighthouse into sharper focus, and improves ocean texture. However, the seagull's orientation is incorrect, facing away from the camera rather than matching the original image, and the background building appears overly reduced in scale. For sample 31757, LoRA generates more coherent human poses and clearly presents the snowboard in the foreground. The snowy terrain and mountainous backdrop are well-preserved, but the faces of the figures remain undetailed, and the background cabin is still missing, indicating that while LoRA enhances salient objects, it may overlook finer contextual elements.

Figure 6. Baseline, w/ LoRA and CLIP-Guided

CLIP-guided generation, in contrast, pushes semantic specificity further but occasionally introduces artifacts. For sample 91257, although the seagull and harbor setting are sharply defined, the bird's body is exaggerated, the lighthouse is missing, and the region near the seagull's tail becomes blurry and incoherent. For 31757, the generation includes skis and poles and captures the mountainous terrain well, but suffers from extreme warping, visual clutter, and distorted geometry—likely due to over-optimization on local CLIP features at the cost of spatial coherence and global structure.

5.3.2 DiffusionCLIP

Figure 7 illustrates the results of applying DiffusionCLIP on the same examples, showing the progression from non-finetuned to 1-step and 5-step fine-tuning.

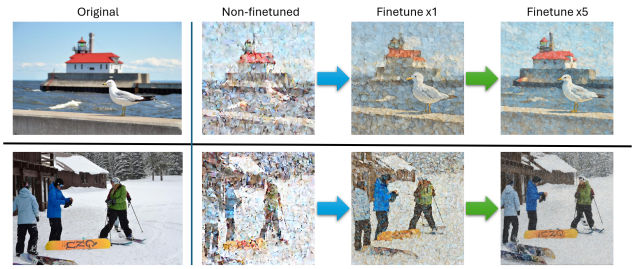

Figure 7. DifussionCLIP Image Comparison

Compared to previous methods, DiffusionCLIP offers a qualitatively different effect. Rather than altering prompt formulations or tuning layers globally, it attempts to semantically steer the output toward a known reference image through iterative guidance, effectively reproducing the original photograph.

For sample 91257, the non-finetuned result is heavily distorted, with vague color blobs and no recognizable object boundaries. After a single round of fine-tuning, we observe significant improvements: the seagull becomes discernible, the red-roofed lighthouse begins to emerge, and wave textures gain structure. However, some details such as the pier remain ambiguous, and the seagull is not perfectly integrated into the scene. After five rounds, image coherence improves dramatically—the seagull exhibits sharper contour definition and more realistic texture, the lighthouse is centered and recognizable, and the overall composition matches the original photograph. One minor flaw persists: the direction of the seagull is slightly misaligned from the original, and the background building remains undersized.

For sample 31757, the non-finetuned output is similarly noisy and fragmented, with no visible characters or winter landscape elements. After one round of fine-tuning, the snowy environment and basic human silhouettes start to appear, along with a rough outline of the foreground snowboard. The surrounding structure remains underdeveloped. After five rounds of fine-tuning, the scene becomes substantially clearer: the snowboard is correctly positioned and colored, the figures are fully visible with appropriate skiing posture, and the snowy backdrop, including trees and sky, matches the semantic expectations of the caption. Notably, the background cabin becomes identifiable, which was absent from all previous methods, suggesting that directional fine-tuning aids in retrieving global contextual features as well as local object fidelity.

5.4. Discussions

We briefly compare methods' strengths and limitations.

- **Prompt Eng. and Output Filter** are low-cost techniques, improving output without changing the model. Prompt templates guide generation style, while filtering selects better candidates. However, both lack internal model adaptation and struggle with structural fidelity.
- **LoRA Fine-Tuning** improves core alignment and object fidelity with limited updates, but often misses background details and fine textures.
- **CLIP-Guided Generation** improves semantic precision, by injecting CLIP-based gradients during inference. However, it risks introducing artifacts due to local over-optimization and requires careful tuning.
- **DiffusionCLIP** best reproduces reference images via iterative, CLIP-guided fine-tuning. It excels in alignment but is slow, memory-intensive, and less flexible for diverse generation.

To summarize, simpler methods scale well; fine-tuning and CLIP-based methods yield higher fidelity. The best results may come from hybrid approaches that combine strengths across strategies.

6. Conclusion and Future Work

In this project, we systematically explored multiple strategies to improve text-to-image generation using Stable Diffusion. Starting from a vanilla baseline, we implemented and evaluated five distinct enhancement techniques: prompt engineering, output filtering, LoRA fine-tuning, CLIP-guided generation, and DiffusionCLIP.

Our experiments demonstrate that while simple methods like prompt engineering and output filtering offer quick wins with low computational cost, they are ultimately limited in correcting deep semantic or structural flaws. LoRA fine-tuning achieves substantial gains in alignment and visual fidelity by modifying a small number of model parameters, making it an efficient and scalable solution. CLIP-guided generation pushes semantic alignment further through inference-time optimization, albeit with trade-offs in artifact risk and stability. DiffusionCLIP delivers the most faithful reconstructions of target concepts and images, confirming its strength in reference-guided scenarios, but also highlighting its high computational demands.

Looking ahead, there are several promising directions for future work. First, combining multiple methods—such as integrating LoRA fine-tuning with inference-time CLIP guidance—may yield complementary benefits. Second, improving memory and speed efficiency of CLIP-guided optimization could enable its deployment in real-time systems. Third, developing more interpretable metrics beyond CLIP-Score would help better assess alignment quality. Lastly, expanding our evaluation to include human preference studies could provide more holistic insights into perceived image quality.

Together, our findings provide a comprehensive roadmap for enhancing diffusion-based generation systems and offer actionable guidance for selecting the appropriate method based on application-specific constraints and objectives.

7. Contributions & Acknowledgements

This project is done by Xinxie Wu, under the GREAT guidance from the project mentor, Rahul Venkatesh. Special thanks to Rahul Venkatesh. This project is for the course $CS231N$ only, not shared with other courses.

References

[1] K. Crowson et al. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *arXiv preprint arXiv:2204.08583*, 2022.

[2] R. Gal, O. Patashnik, H. Maron, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[3] Y. Hao et al. Optimizing prompts for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[4] A. Hertz et al. Prompt-to-prompt image editing with cross attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[5] J. Hessel et al. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[6] E. J. Hu et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[7] Y. Kim et al. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[8] N. Liu et al. Compositional visual generation with composable diffusion models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[9] R. Rombach et al. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] C. Saharia et al. Photorealistic text-to-image diffusion models with deep language understanding. In *arXiv preprint arXiv:2205.11487*, 2022.