Training Strategy Analysis for ControlNet: Balancing Conditional Controllability and Generalization on Small-Scale Datasets

Hao Zheng[1], Zhiyi Chen[1], Rongpei Li[1]

Abstract

ControlNet provides an effective paradigm for incorporating structural conditions into pretrained text-to-image diffusion models. However, the optimal training configuration—particularly regarding which components to freeze and where to inject conditional signals—remains underexplored, especially in small-scale data scenarios. In this work, we systematically investigate the interplay between two key hyperparameters: `sd_locked` *(whether to freeze the Stable Diffusion backbone) and* `only_mid_control` *(whether to restrict condition injection to the middle block only). Through ablation studies on the Fill50K dataset, we identify four representative configurations and analyze their convergence behavior, generation quality, and generalization capability. Our experiments reveal that Configuration D (*`sd_locked=False`, `only_mid_control=True`*) achieves the best balance among conditional alignment, prior preservation, and generalization—exhibiting emergent convergence around 2400 steps while avoiding the overfitting observed in other configurations. We provide mechanistic explanations from optimization dynamics and gradient pathway perspectives, offering practical guidelines for training controllable diffusion models under data-limited conditions.*

Keywords: ControlNet, Diffusion Models, Conditional Image Generation, Training Strategies, Ablation Study

1. Introduction

Text-to-image diffusion models such as Stable Diffusion [8] have demonstrated remarkable capabilities in generating high-quality images from textual descriptions. However, precise spatial control over the generated content—such as specifying exact poses, edges, or depth layouts—remains challenging when relying solely on text prompts. ControlNet [11] addresses this limitation by introducing a trainable copy of the encoder blocks that processes structural conditions (e.g., Canny edges, human poses, depth maps) and modulates the frozen backbone through zero-initialized convolutions.

While the original ControlNet paper establishes the general framework, several practical questions arise when applying this approach to domain-specific applications with limited training data:

- Should the Stable Diffusion backbone remain frozen during training, or can selective unfreezing improve adaptation to new condition distributions?
- Is it beneficial to inject conditional signals at all decoder levels, or does restricting injection to specific layers yield better generalization?
- How do these choices interact with each other and with dataset scale?

In this paper, we present a systematic ablation study addressing these questions. Using the Fill50K dataset as a testbed, we evaluate four configurations spanning the combinatorial space of `sd_locked` $\in \{\text{True}, \text{False}\}$ and `only_mid_control` $\in \{\text{True}, \text{False}\}$. Our key contributions are:

1. **Empirical characterization** of convergence patterns, generation quality, and overfitting tendencies across configurations.
2. **Identification of Configuration D** (`sd_locked=False`, `only_mid_control=True`) as optimal for small-data scenarios, achieving high-quality generation at approximately 2400 steps.
3. **Mechanistic analysis** explaining the observed behaviors through the lens of effective capacity, gradient pathway constraints, and loss landscape geometry.
4. **Practical guidelines** for training ControlNet variants under data-limited conditions.

2. Problem Formulation

We formalize the research questions motivating our experimental investigation. Each problem statement corresponds directly to hypotheses tested in our ablation study.

2.1. Problem 1: Backbone Freezing Trade-off

The standard ControlNet training protocol freezes all Stable Diffusion parameters (`sd_locked=True`), training only the copied encoder blocks and zero-convolution connectors. This preserves the pretrained generative prior but may limit the model's ability to adapt to condition distributions that differ significantly from natural images.

Question: Under what circumstances does unfreezing the backbone (`sd_locked=False`) improve conditional alignment without catastrophic forgetting of the generative prior?

Addressed in: Configurations A vs. B (Section 5), with mechanistic analysis in Section 6.

2.2. Problem 2: Condition Injection Depth

ControlNet's default architecture injects conditional features at multiple levels of the U-Net decoder through skip connections. An alternative is to restrict injection to the middle block only (`only_mid_control=True`), hypothetically reducing the risk of low-level feature interference.

Question: Does restricting condition injection to the semantic bottleneck layer improve generalization on small datasets by acting as an implicit regularizer?

Addressed in: Configurations A vs. C and B vs. D (Section 5), with gradient pathway analysis in Section 6.

2.3. Problem 3: Emergent Convergence Dynamics

Preliminary observations suggest that ControlNet training exhibits non-monotonic behavior: models may maintain near-baseline outputs for many steps before suddenly achieving strong conditional alignment. Understanding this phenomenon is crucial for setting appropriate training schedules.

Question: What causes the "emergent convergence" pattern, and how do configuration choices affect its timing and stability?

Addressed in: Step-wise quality analysis (Section 5) and optimization dynamics discussion (Section 6).

2.4. Problem 4: Optimal Configuration for Small Data

Combining the above considerations, we seek to identify configurations that achieve:

- Strong conditional controllability (alignment with input structure)
- Preservation of generative prior (realistic textures, coherent semantics)
- Robust generalization (no overfitting to training conditions)

Question: Which combination of `sd_locked` and `only_mid_control` best balances these objectives on datasets of ∼50K samples?

Addressed in: Comparative evaluation and final recommendations (Sections 5–7).

3. Related Work

3.1. Diffusion Models for Image Generation

Diffusion probabilistic models [3, 9] have emerged as a dominant paradigm for generative modeling, achieving state-of-the-art results on image synthesis benchmarks. Latent diffusion models [8] improve efficiency by operating in a compressed latent space, enabling high-resolution generation with reduced computational cost. Stable Diffusion, built on this framework, serves as the backbone for numerous conditional generation methods.

3.2. Conditional Control in Diffusion Models

Several approaches have been proposed to enhance spatial controllability in diffusion models:

Classifier Guidance [1] steers the sampling process using gradients from an external classifier, but requires training separate classifiers for each condition type.

Classifier-Free Guidance [2] eliminates the need for external classifiers by jointly training conditional and unconditional models, now standard in text-to-image systems.

T2I-Adapter [7] introduces lightweight adapter modules that inject spatial conditions without modifying the base model, enabling efficient multi-condition composition.

GLIGEN [6] extends diffusion models with grounding capabilities, allowing bounding box and keypoint-based control through gated self-attention layers.

IP-Adapter [10] enables image prompt conditioning through decoupled cross-attention, complementing text-based control with visual references.

3.3. ControlNet Architecture

ControlNet [11] introduces a "trainable copy" paradigm where the encoder blocks of a pretrained U-Net are duplicated and connected to the frozen decoder through zero-initialized convolutions. This design offers several advantages:

- **Prior preservation:** Zero initialization ensures the model starts from the pretrained behavior.
- **Flexible conditioning:** The same architecture handles diverse condition types (edges, poses, depth, segmentation).
- **Training stability:** Freezing the backbone prevents catastrophic forgetting during fine-tuning.

The ControlNet formulation can be expressed as:

$$y_c = F(x; \Theta) + \mathcal{Z}(F(x + \mathcal{Z}(c; \Theta_{z1}); \Theta_c); \Theta_{z2}) \qquad (1)$$

where $F(\cdot; \Theta)$ denotes the frozen backbone, Θ_c the trainable copy, and $\mathcal{Z}(\cdot; \Theta_z)$ the zero-initialized convolutions.

3.4. Parameter-Efficient Fine-Tuning

Our analysis connects to broader work on efficient adaptation of large models:

LoRA [5] constrains weight updates to low-rank subspaces, reducing trainable parameters while maintaining adaptation quality.

Adapter Tuning [4] inserts small bottleneck modules between frozen layers, enabling task-specific adaptation with minimal parameter overhead.

The `only_mid_control` configuration can be viewed through this lens: by restricting condition injection to the middle block, we implicitly constrain the effective update subspace, achieving regularization effects similar to architectural bottlenecks.

4. Methods

4.1. Merging Conditional Control with the Stable Diffusion Backbone

ControlNet augments a pretrained text-to-image diffusion model by introducing a secondary, learnable pathway that injects spatial conditioning signals into the denoising network while preserving the capabilities of the original backbone (Stable Diffusion). Let $F(\cdot;\Theta)$ denote a pretrained U-Net block that maps a feature map $x \in \mathbb{R}^{h\times w\times c}$ to an output $y = F(x;\Theta)$. To incorporate structural conditions such as edges, poses, or depth maps, ControlNet constructs a *dual-path* architecture: the original block is kept intact as a frozen backbone, and a trainable copy $F(\cdot;\Theta_c)$ is created to process condition-aware features. These two paths are fused through lightweight 1×1 convolutions, denoted by $Z(\cdot;\Theta_z)$, which align feature dimensions and modulate the backbone activations with task-specific information.

Given a conditioning feature map c, the ControlNet block computes

$$y_c = F(x;\Theta) + Z\big(F(x + Z(c;\Theta_{z1});\Theta_c);\Theta_{z2}\big). \tag{2}$$

This formulation allows the trainable copy to interpret the conditioning signal and contribute corrective residuals, while the frozen backbone ensures that the pretrained generative behavior is preserved. The connector convolutions $Z(\cdot;\Theta_{z1})$ and $Z(\cdot;\Theta_{z2})$ serve as projection layers that map the conditioning features into the appropriate channel space and then inject the processed representation back into the main feature stream.

The conditioning feature map c itself is derived from a raw conditioning image $c_i \in \mathbb{R}^{512\times512\times3}$ through a small encoder $E(\cdot)$. This encoder consists of a few strided convolution layers, where the first convolution primarily aligns the spatial resolution with the latent space of Stable Diffusion (typically 64×64). The goal of $E(\cdot)$ is not to perform deep semantic extraction but to convert the conditioning image into a feature map compatible with the U-Net resolution at which ControlNet operates.

4.2. Training Considerations: Frozen Backbone and Zero-Initialized Connectors

To stabilize learning and prevent catastrophic forgetting, all pretrained parameters Θ of Stable Diffusion are frozen throughout training. Since conditional datasets are often much smaller than the large-scale corpora used to train the original model, freezing the backbone prevents overspecialization and ensures that the core generative abilities remain intact. Only the parameters of the trainable copy Θ_c, the conditioning encoder $E(\cdot)$, and the connector convolutions Θ_{z1}, Θ_{z2} are updated.

A second key design choice is the zero-initialization of the connector convolutions. At initialization, both $Z(c;\Theta_{z1})$ and $Z(\cdot;\Theta_{z2})$ output zero for any input, causing the entire ControlNet block to reduce to

$$y_c = F(x;\Theta), \tag{3}$$

which means the model behaves identically to the pretrained Stable Diffusion at the start of training. This avoids injecting untrained, potentially harmful noise into the backbone and allows the influence of the conditional branch to "grow in" smoothly as training progresses. Empirically, this strategy leads to a stable optimization process and a characteristic "sudden convergence" effect: after a period in which the model behaves like the vanilla backbone, the connector weights learn meaningful transformations that allow the network to abruptly acquire strong conditioning fidelity.

Together, the dual-path structure, frozen backbone, and zero-initialized connector convolutions form a robust and data-efficient architecture. ControlNet is thus able to learn spatially localized, task-specific controls while preserving the expressive power and visual quality of the underlying diffusion model.

4.3. Conditioning Generation

In our experiments, we evaluated two conditioning types: *Human Pose* and *HED Boundary*. The human pose condition is generated using an OpenPose-based keypoint detector, which produces a structured skeletal representation that captures the global configuration and articulation of the subject. The HED boundary condition is derived from the Holistically-Nested Edge Detection (HED) model, which extracts smooth, high-resolution edge maps that provide fine-grained structural cues. Both conditioning types are widely used in controllable generation tasks due to their ability to capture complementary spatial information: pose encodes coarse geometry, while HED preserves local boundaries.

For the condition-type experiments, we adopted publicly available pretrained ControlNet models from Hugging-

Face. These models were originally trained on large-scale general-domain datasets and have demonstrated strong generalization to diverse image conditions. In our replication, both conditioning types yielded high-quality generations, and the model consistently interpreted the structural signals provided by pose skeletons and boundary maps. The results indicate that the pretrained ControlNet weights possess robust representational capacity and that the conditioning mechanism is sufficiently expressive to guide the generation process in a semantically meaningful way.

4.4. Training Procedure

We train our models using the Fill50K dataset, a collection of 50,000 image–mask pairs originally developed for image inpainting and completion tasks. The dataset contains diverse structural patterns and object boundaries, providing meaningful supervision for conditioning-guided generation even though it is significantly smaller than the large-scale datasets used for training Stable Diffusion. As our generative backbone, we adopt Stable Diffusion v1.5, a latent diffusion model trained on LAION-5B that produces high-quality natural images through a U-Net denoiser operating in a compressed latent space. Its strong visual prior makes it an ideal foundation for studying conditional finetuning behavior.

To understand the effect of architectural choices during finetuning, we performed an ablation study following the configuration conventions in the ControlNet training documentation.[1] Specifically, we varied two key flags: `sd_locked` and `only_mid_control`. These options determine which portions of the Stable Diffusion U-Net are updated during training.

Definition of `sd_locked`. When the configuration `sd_locked = True`, all parameters of the original Stable Diffusion model are frozen. Only the ControlNet branch (trainable copy, connector convolutions, and condition encoder) receives gradient updates. This configuration preserves the pretrained generative behavior and prevents the model from drifting away from the domain of natural images. When `sd_locked = False`, the Stable Diffusion U-Net becomes trainable, greatly increasing the number of parameters updated at each iteration. This allows strong adaptation to the conditioning task, but also increases the risk of overfitting and destabilizing the pretrained backbone.

Definition of `only_mid_control`. When the configuration `only_mid_control = True`, ControlNet attaches trainable modules only to the middle block of the U-Net, rather than to all encoder blocks. This substantially reduces the number of trainable parameters, as the skip-connected encoder pathways remain unchanged. When `only_mid_control = False`, ControlNet is attached to every encoder block, maximizing control capacity but also enlarging the effective finetuning footprint.

These configurations produce four experimental variants. To ensure column compatibility, we present the results in a compact table:

ExpID	sd_locked	mid_only	Sudden Conv.	Behavior
A	True	False	2700	Medium quality
B	True	True	N/A	Low quality
C	False	False	1500	Overfitting, forgetting
D	False	True	2400	High quality, mild overfit

Table 1. Ablation results of training configurations.

Across all configurations, we consistently observe the "sudden convergence" phenomenon: for several training steps, the model behaves similarly to the pretrained backbone, and then at a specific iteration, it abruptly begins to follow conditioning signals and generate structurally aligned images. However, the step at which this convergence occurs varies significantly between training settings, as shown in Table 1.

When both the Stable Diffusion model and all ControlNet modules are trainable (`sd_locked = False`, `only_mid_control = False`), the network receives gradients across a very large parameter space. This accelerates the sudden convergence (1500 steps) and yields high-quality generations early in training. However, this flexibility also leads to severe overfitting: as training continues, images develop blurry or irregular boundaries, and catastrophic forgetting may occur, with some samples collapsing into disordered patterns.

When the backbone is frozen but all ControlNet blocks are active (`sd_locked = True`, `only_mid_control = False`), the training process becomes more stable. This is the recommended configuration in the ControlNet paper. The model maintains reasonable image quality and avoids overfitting even at high training iterations. Notably, the generated images tend to inherit aesthetic properties of Stable Diffusion v1.5 itself—such as characteristic textures and shading—indicating that the frozen backbone strongly shapes the model output.

Restricting ControlNet to the middle block while keeping the backbone frozen (`sd_locked = True`, `only_mid_control = True`) significantly limits the effective gradient pathways. As a result, we observe no sudden convergence even after 12k steps, and the model fails to learn meaningful structural patterns from the dataset. The limited backpropagation route through the single mid-block likely reduces optimization efficiency.

Interestingly, when the backbone is unfrozen but ControlNet is attached only to the middle block (`sd_locked`

[1] https://github.com/lllyasviel/ControlNet/blob/main/docs/train.md

= False, only_mid_control = True), the model becomes more stable than in the fully trainable case. Overfitting still appears but is substantially mitigated, and image quality improves. We hypothesize two contributing factors: (1) the number of trainable parameters under this configuration becomes comparable to the standard setting (sd_locked = True, only_mid_control = False), whereas the fully trainable configuration involves significantly more parameters; and (2) because only a single decoding pathway is controllable, backpropagation becomes more direct, enabling more effective learning of dataset-specific features without destabilizing the entire U-Net.

Taken together, these results highlight the trade-offs between model stability, learning efficiency, and overfitting. Allowing too many components to update accelerates convergence but risks rapid degradation, whereas restricting updates can improve robustness but slow down or even prevent meaningful learning. Based on our observations, it may be advantageous to unfreeze the backbone while attaching ControlNet only to the middle block (sd_locked = False, only_mid_control = True) when finetuning on a small dataset with a strong and distinctive style, which behaves more like a transfer-learning procedure. In contrast, when the pretrained generative properties of Stable Diffusion v1.5 are crucial—such as maintaining realism or preserving its characteristic aesthetic—and the primary goal of finetuning is to teach the model how to interpret a new conditioning modality rather than to change its visual style, the configuration (sd_locked = True, only_mid_control = False) is preferable.

5. Experimental Results

5.1. Configuration Comparison

Table 2 summarizes the four experimental configurations and their key characteristics.

Table 2. Ablation configurations and observed outcomes.

Config	sd_locked	only_mid	Steps	Quality
A	True	True	2700	Medium
B	False	False	–	Low (Overfit)
C	True	False	1500	Overfit
D	False	True	2400	**High**

5.2. Detailed Analysis by Configuration

Configuration A (sd_locked=True, only_mid_control=True): This conservative setting achieves medium quality with stable convergence around 2700 steps. The frozen backbone ensures prior preservation, while middle-only injection provides implicit regularization. However, the inability to adapt backbone features limits conditional alignment quality.

Configuration B (sd_locked=False, only_mid_control=False): Full unfreezing with multi-level injection dramatically increases effective capacity. While training loss decreases rapidly, validation metrics degrade, indicating severe overfitting. The model memorizes training condition-output pairs rather than learning generalizable mappings.

Configuration C (sd_locked=True, only_mid_control=False): Despite keeping the backbone frozen, multi-level injection still leads to overfitting, with convergence as early as 1500 steps followed by quality degradation. This suggests that injection depth, not just backbone training, contributes to overfitting risk.

Configuration D (sd_locked=False, only_mid_control=True): This configuration achieves the best results. The unfrozen backbone allows adaptation to the condition distribution, while middle-only injection constrains the effective update subspace. Emergent convergence occurs around 2400 steps, producing high-quality outputs with strong generalization.

5.3. Emergent Convergence Phenomenon

A notable observation across configurations is the nonlinear convergence behavior. Models maintain outputs close to the unconditional baseline for extended periods before exhibiting sudden alignment with input conditions. This "emergent convergence" is most pronounced in Configuration D, where:

- Steps 0–1500: Minimal deviation from pretrained behavior
- Steps 1500–2200: Gradual increase in condition sensitivity
- Steps 2200–2400: Rapid alignment with structural conditions
- Steps 2400+: Stable high-quality generation

Configuration B also shows rapid change but transitions into overfitting rather than stable alignment.

6. Discussion

We provide mechanistic explanations for the observed phenomena from three complementary perspectives: optimization dynamics, gradient pathway constraints, and diffusion sampling trajectory modulation.

6.1. Optimization Dynamics and Loss Geometry

The emergent convergence pattern can be understood through the lens of non-convex optimization dynamics.

6.1.1. Gradient Plateau and Critical Point Transition

When sd_locked=False, including backbone parameters significantly expands the optimization landscape, increasing the likelihood of traversing flat regions and saddle point neighborhoods. Combined with zero-initialized convolutions that initially contribute near-zero gradients, early training exhibits a "plateau phase" where updates accumulate without visible behavioral change.

As training progresses, coupling between the condition branch and backbone strengthens. When gradient accumulation crosses a critical threshold, the model transitions from "reusing pretrained prior" to "condition-alignment-driven" mode. This can be formalized as escaping from a flat region along Hessian-dominated directions:

$$\Delta\theta = -\eta\nabla_\theta\mathcal{L} - \frac{\eta^2}{2}H^{-1}\nabla_\theta\mathcal{L} + O(\eta^3) \tag{4}$$

Configuration D's convergence at ∼2400 steps suggests it operates near the optimal regime where gradient signal strength balances with stability.

6.1.2. Effective Capacity and Bias-Variance Trade-off

Configuration B's rapid training loss reduction coupled with validation degradation exemplifies the high-variance overfitting characteristic of small-data regimes. Full-layer injection plus backbone unfreezing dramatically increases effective capacity, causing the model to deviate from the pretrained manifold and memorize training-specific patterns.

Configuration D's `only_mid_control=True` acts as a structural constraint, implementing implicit regularization that:

- Allows backbone adaptation to new condition distributions
- Prevents shallow/deep layer perturbations that would corrupt the generative prior

This can be augmented with explicit condition consistency regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda\mathbb{E}\left[\|E(g(x)) - c\|_2^2\right] \tag{5}$$

where $E(\cdot)$ is a pretrained condition extractor and $g(x)$ the generated output.

6.2. Gradient Pathway Analysis

6.2.1. Low-Rank Subspace Adaptation Effect

When `only_mid_control=True`, gradients primarily flow through the middle block parameters W_{mid}:

$$\frac{\partial\mathcal{L}}{\partial W_{\text{mid}}} = \frac{\partial\mathcal{L}}{\partial y}\cdot\frac{\partial y}{\partial W_{\text{mid}}} \tag{6}$$

This effectively restricts updates to a lower-dimensional subspace, analogous to LoRA's low-rank constraint. By limiting which modules receive condition-driven updates, we prevent gradient-induced drift in shallow (detail/edge) and deep (semantic/texture) layers, better preserving the pretrained distribution.

6.2.2. Structural Bottleneck Effect

The U-Net middle block encodes high-level abstractions (layout, pose, object shapes), while shallow layers handle local edges/textures and deep layers refine output details. Focusing control injection on the middle layer creates a "structural bottleneck":

- Structural signals (pose, edges) are more transferable under small data
- Texture/local patterns are more easily memorized, leading to overfitting

This explains Configuration C's fast convergence but poor generalization: multi-level injection allows texture memorization despite backbone freezing.

6.2.3. Toward Dynamic Injection

Fixed injection positions may not be optimal across condition types. A learnable gating mechanism could generalize `only_mid_control`:

$$y = F(x;\Theta) + \sum_{i=1}^{L}\alpha_i\Delta_i(c, F_i(x)) \tag{7}$$

where α_i are learned injection weights, enabling condition-adaptive modulation.

6.3. Sampling Trajectory Modulation

6.3.1. Phase-Dependent Condition Influence

In diffusion sampling, early (high-noise) steps determine global layout while late (low-noise) steps refine textures. Configuration D's architecture aligns with this:

- Middle-layer injection primarily influences layout decisions.
- Unfrozen backbone retains fine-detail generation capability.

This prevents the boundary blurring and unrealistic textures observed in overfit configurations.

6.3.2. Drift Term Dominance Transition

Modeling diffusion as a conditioned SDE:

$$dx = f(x,c)\,dt + g(t)\,dw \tag{8}$$

When the condition branch is weak, trajectories follow the pretrained prior. As training strengthens condition coupling past a threshold, the condition-dependent drift term dominates, pulling trajectories toward the condition manifold. Configuration D achieves this transition while the middle-block bottleneck prevents destabilization.

6.3.3. Memory Effects in Overfitting

Configuration B's overfitting manifests as "trajectory memorization": training condition-output pairs are encoded directly into the denoising path, producing artifacts and blurred boundaries on unseen conditions. This represents contamination of the diffusion prior by training-set statistics.

7. Conclusion

We have presented a systematic investigation of ControlNet training configurations, focusing on the interplay between backbone freezing (`sd_locked`) and condition injection depth (`only_mid_control`) under small-scale data conditions.

7.1. Key Findings

1. **Configuration D is optimal**: Setting `sd_locked=False` and `only_mid_control=True` achieves the best balance of conditional controllability, prior preservation, and generalization on the Fill50K dataset.
2. **Emergent convergence is configuration-dependent**: The timing and stability of the alignment transition varies significantly, with Configuration D exhibiting stable emergence at ∼2400 steps.
3. **Injection depth is as important as backbone freezing**: Configuration C demonstrates that multi-level injection can cause overfitting even with a frozen backbone.
4. **Middle-block injection provides implicit regularization**: Restricting conditions to the semantic bottleneck constrains effective capacity while preserving adaptation flexibility.

7.2. Practical Recommendations

Based on our analysis, we recommend the following strategies for training ControlNet on limited data:

1. **Progressive unfreezing with smooth scheduling**: Begin with frozen backbone, gradually unfreeze layers, and use cosine annealing to smooth the alignment transition.
2. **Prioritize structural layer injection**: Default to `only_mid_control=True` or implement hierarchical gating with middle-layer emphasis.
3. **Add consistency regularization**: Consider cycle consistency or condition-extractable consistency losses:

$$\mathcal{L}_{\text{cycle}} = \|\text{Extract}(G(c)) - c\|_2^2 \tag{9}$$

4. **Automated hyperparameter search**: Use Bayesian optimization to explore the configuration space of unfreezing degree, injection levels, and learning rate schedules.

7.3. Future Directions

Our findings open several avenues for future work:

- Extending the analysis to other condition types (depth, segmentation, normal maps)
- Developing learnable injection gating mechanisms
- Investigating transfer across datasets with distribution shift
- Scaling analysis to larger backbone models (SDXL, SD3)

The core insight—that effective capacity and injection topology jointly shape optimization dynamics and generalization—provides a foundation for designing more robust controllable generation systems.

Team Contributions

Hao Zheng. Led the overall methodology design and execution. Conducted model training, ablation studies, and comparative experiments; evaluated results systematically; and wrote the core technical sections of the report. Coordinated the final presentation for the course project.

Zhiyi Chen. Organized and refined the presentation materials, aligned the methods with experimental results, and improved the report structure and layout. Designed and produced the project poster.

Rongpei Li. Explored extension directions and improvement strategies, and contributed to the analysis and synthesis of experimental findings.

Acknowledgments

This project was completed as part of the Diffusion Models course. We thank the course instructors and teaching assistants for their guidance and support.

References

[1] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv:2207.12598, 2022.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[4] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Qiao De Laroussilhe, Andrea Gesmundo, Mohammad Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019.

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[6] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22511–22521, 2023.

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

[8] 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)*, pages 10674–10685, 2022.

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

[10] Hu Ye, Junting Zhang, Shujie Liu, Xinghang Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv:2308.06721, 2023.

[11] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023.