Optimizer Geometry for Stable Diffusion Model Fine-Tuning: A Comparative Study of Muon and AdamW with and without LoRA

Bruno Vieira, Daniel Kao, Jin Ying, Zhaoxi Zhang

December 15, 2025

Abstract

Fine-tuning text-to-image diffusion models remains computationally expensive and highly sensitive to optimization hyperparameters. Although AdamW is the default optimizer for Stable Diffusion and is used during its original pretraining, recent work on geometry-aware optimization—particularly the Muon optimizer—suggests that alternative update rules may offer improved robustness, smoother convergence, and stronger generalization. In this project, we investigate whether Muon provides measurable improvements over AdamW when fine-tuning Stable Diffusion under both full-model and parameter-efficient LoRA adaptation. Our study analyzes training stability, gradient behavior, sensitivity to learning rate and weight decay, and image generalization quality on unseen prompts. We additionally examine how the geometry imposed by LoRA's low-rank updates interacts with Muon's orthogonalized, spectral-norm-aware update rule. Taken together, our experiments aim to provide the first systematic and controlled evaluation of optimizer choice for diffusion model fine-tuning and to clarify the practical role of geometry-aware optimization in this setting.

1 Introduction

Diffusion models have become central to modern generative modeling, achieving unprecedented photorealism and semantic consistency in text-to-image generation [1, 2, 3]. Stable Diffusion in particular provides an efficient and accessible architecture, combining a latent diffusion process with a U-Net denoiser trained on large-scale text–image datasets [3]. Despite the widespread adoption of these models, fine-tuning them for downstream tasks, such as domain-adaptation, style personalization, or small sample customization, remains challenging. Training instability, high sensitivity to hyperparameters, and rapid overfitting can undermine attempts to adapt these large pretrained systems. These issues highlight the importance of understanding the optimization behavior underlying diffusion model fine-tuning.

Stable Diffusion is trained using a simplified denoising objective, where the model predicts the Gaussian noise added to an image latent at a random diffusion timestep [1]. If x_0 is an image latent, $\epsilon \sim N(0, I)$ is a noise component, and t is a timestep sampled from a fixed schedule, the forward diffusion adds noise according to

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\epsilon$$

The U-Net predicts $\epsilon_\theta(x_t, t, c)$, conditioned on some text embedding c, and training minimizes the expected denoising error:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, c)\|_2^2\right]$$

Since this loss couples many timesteps of the diffusion chain, stable optimization requires consistent gradients across widely varying noise levels [13, 2]. Even small fluctuations in early updates can propagate through the chain and destabilize convergence. Training dynamics are therefore tightly connected to optimizer behavior, particularly during the initial steps of adaptation.

AdamW is the default optimizer used during the original training of Stable Diffusion [4, 3]. Its effectiveness in large-scale pretraining has motivated its continued use in fine-tuning settings. However, Adam-based methods exhibit several well-known limitations relevant to diffusion optimization [5]. The adaptive learning rate can have high variance early in training, oscillatory updates arise from elementwise normalization of gradient moments, and stability can degrade sharply under suboptimal learning-rate or weight-decay configurations. These challenges are magnified in low-data fine-tuning scenarios, where optimization noise is large and the model must adapt rapidly without overfitting. Although AdamW's update,

$$\theta_{k+1} = \theta_k - \eta \frac{m_k}{\sqrt{v_k} + \epsilon} - \eta\lambda\theta_k$$

remains effective in many cases, it does not explicitly control geometric properties of weight updates, such as spectral norms, that have been linked to stability in deep networks [6, 7]. This motivates examining alternatives that impose more structured constraints on updates.

Muon is a recent geometry-aware optimizer that introduces a fundamentally different update mechanism [12]. Instead of normalizing gradients coordinate-wise, Muon applies an orthogonalized update that implicitly controls the spectral norm of weight matrices [6]. If G_k denotes the gradient matrix of a layer, Muon first orthogonalizes this gradient using a projection of the form:

$$\tilde{G}_k = G_k - U_k U_k^T G_k$$

where U_k approximates the top singular directions of the weight matrix [8]. The resulting update preserves directions of high curvature but limits growth along dominant

singular modes, producing smoother and more stable optimization. The complete Muon update can be expressed as:

$$\theta_{k+1} = \theta_k - \eta \, \text{normalize}(\tilde{G}_k)$$

where the normalization depends on matrix-level statistics rather than elementwise variance. Recent theoretical analyses describe Muon as a special or limiting case of the Lion-family optimizers [9], inheriting momentum-driven sign updates but augmented with spectral-norm constraints. Empirically, Muon has demonstrated improved robustness to hyperparameters and stronger generalization across a variety of large-scale pretraining tasks [12], raising the question of whether these advantages persist in diffusion fine-tuning.

A second dimension of complexity arises from parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA). Instead of updating a full weight matrix W, LoRA injects a low-rank decomposition [10], modifying the layer through

$$W' = W + AB \quad ; \; A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times l}$$

with rank $k << min(m, l)$. This constrains updates to a low-dimensional subspace—a geometric restriction that can improve generalization and reduce overfitting, but which also interacts strongly with the optimizer. LoRA's restriction to a low-rank manifold may amplify or suppress particular update directions, and it is unclear whether Muon's orthogonalized updates align naturally with this constraint or introduce conflicting geometric biases. Moreover, LoRA is now the dominant method for diffusion fine-tuning, making it essential to understand whether the optimizer should be chosen differently when using low-rank parameterizations [11].

The goal of this project is to study how optimizer geometry shapes fine-tuning dynamics in diffusion models. We focus on robustness, convergence, and generalization. Robustness concerns how gracefully performance degrades when hyperparameters deviate from their optimal values [12, 4]. Convergence examines loss smoothness, gradient norms, and the presence of oscillatory behavior that often signals instability [5]. Generalization evaluates whether the optimizer encourages representations that transfer to unseen prompts rather than overfit the fine-tuning dataset [14]. These questions are particularly important because fine-tuning diffusion models frequently occurs in small-data regimes, where optimization noise dominates signal and stable training is difficult to achieve.

This work provides a systematic and controlled comparison between Muon and AdamW in Stable Diffusion fine-tuning, examining both full-model and low-rank adaptations. By analyzing optimization trajectories, sensitivity landscapes, and output image quality across multiple hyperparameter settings, we aim to clarify whether geometry-aware optimization offers a meaningful advantage in generative fine-tuning and whether the default reliance on AdamW should be reconsidered for diffusion-based applications.

2 Background and Related Work

Fine-tuning diffusion models has emerged as an important yet computationally delicate task, requiring both stable optimization dynamics and robustness to hyperparameter choices. Stable Diffusion and latent diffusion models [1, 2, 3] rely on denoising objectives in which a U-Net predicts Gaussian noise across a wide range of timesteps. Because small perturbations in early optimization steps can propagate through the multi-step denoising chain, fine-tuning these models is highly sensitive to optimizer behavior, especially in small-data or domain-specific adaptation settings.

2.1 Optimizer Behavior in Diffusion Models

AdamW remains the default optimizer for training and fine-tuning Stable Diffusion due to its strong empirical performance and decoupled weight decay formulation [4]. However, Adam-based methods are known to exhibit high variance in adaptive learning rates during early training, leading to unstable or oscillatory optimization trajectories [5]. These issues become more pronounced when batch sizes are small or datasets contain limited variation—conditions common in personalization and domain adaptation. Recent theoretical work shows that AdamW's element-wise normalization does not control matrix-level geometry, such as the spectral norms of weight matrices [6, 7, 8], which has been linked to generalization and training stability.

2.2 Geometry-Aware Optimization and Muon

Recent work introduces geometry-aware optimizers that operate at the matrix level rather than coordinate-wise. The Muon optimizer [12] orthogonalizes gradients using low-rank approximations of dominant singular directions, implicitly constraining spectral norms and producing smoother, more stable updates. Formally, Muon applies an update of the form:

$$G'_k = G_k - U(U^\top G_k),$$

where U contains approximate top singular vectors of the gradient matrix. This projection removes high-curvature directions, enabling more stable optimization dynamics.

Muon can be interpreted as a special case within the Lion-family sign-momentum framework [9], while also incorporating spectral regularization effects absent in AdamW. Empirical evaluations demonstrate that Muon can outperform AdamW in large-scale pretraining, especially in noisy-gradient or high-batch regimes [12]. However, whether these benefits extend to diffusion fine-tuning remains untested, as no prior work systematically explores this question.

2.3 Parameter-Efficient Fine-Tuning and LoRA

Low-Rank Adaptation (LoRA) [10] constrains updates to a learned low-rank decomposition:

$$W' = W + AB,$$

where $A \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{k \times l}$, and $r \ll \min(m, l)$. This reduces compute cost and often improves generalization by restricting updates to a low-dimensional manifold.

Practical adaptations for diffusion models such as those summarized in [11] show that LoRA can significantly reduce compute cost in fine-tuning pipelines. Yet, little is known about how geometry-aware optimizers—such as Muon—interact with LoRA's constrained update space.

2.4 Optimizers for Diffusion Training

Several analyses of diffusion training dynamics highlight the importance of optimizer behavior, noise structure, and Lipschitz constraints [13, 14]. These works collectively suggest that diffusion training is sensitive to optimizer-induced geometry. However, systematic comparisons of geometry-aware optimizers—particularly Muon—against AdamW in diffusion settings remain absent from the literature.

2.5 Fine-Tuning Diffusion Models

Personalization and domain-specific adaptation remain challenging due to overfitting, instability, and catastrophic forgetting. Prior studies highlight that diffusion models' denoising objectives can amplify training irregularities [14]. Although AdamW remains the de facto standard, the role of optimizer geometry in diffusion fine-tuning has not been systematically studied.

2.6 Summary

Existing work establishes that:
(1) diffusion models are highly sensitive to optimization dynamics;
(2) geometry-aware updates such as Muon may provide smoother and more stable training;
(3) LoRA imposes additional geometric constraints that interact with optimizer behavior;
(4) no prior work systematically evaluates Muon vs. AdamW for Stable Diffusion fine-tuning.

These gaps motivate our empirical investigation comparing Muon and AdamW under both full-model and LoRA-based fine-tuning. Our study evaluates efficiency, generalization, and training stability to determine whether geometry-aware optimization offers practical advantages for diffusion adaptation.

3 Methodology

3.1 Resources

The compute resources we use include roughly 320 hours of Nvidia T4 Tensor Core GPU compute, as well as 14 hours of full NVIDIA A100 GPU compute. We choose to fine tune over the OFA-sys Small Stable Diffusion v0 as our base model, given the compute and time limitations of this project.

3.2 Dataset

We choose the ffurfaro/PixelBytes-PokemonAll dataset, consisting of 533 50x50 px images of pixel sprite images of Pokemon, with textual description labels. This dataset fits our needs well; due to our limited compute resources, the low resolution allows the model to train faster thus allowing larger hyperparameter sweeps and a greater number of epochs. In addition, our smaller base model is more suited to learn the simple pixelated patterns in the dataset without fear of overfitting. The dataset size of 533 curated images provides enough data to highlight meaningful differences in performance and hyperparameter convergence across our chosen optimizers and fine tuning methods.

- **Training - Validation Split:** We choose a subset of 80% (480 images) of the dataset as our training set, and 20% (53 images) for our validation set. Given the smaller dataset size, reserving more of it for training increases the likelihood of model convergence.

3.3 Experimental Setup

The 4 fine tuning configurations we are interested in are: no fine tuning, AdamW with LoRA, AdamW with full fine tuning, Muon with LoRA, Muon with full fine tuning. We fine tune the model according to each configuration of optimizer and training type. Next, the maximum learning rate for the given configuration that still results in convergent fine tuning is determined, lr_{max}. Then we perform a logarithmic sweep on the next 4 orders of descending magnitude: $[lr_{max}, \ldots, lr_{max} \times 10^{-4}]$. For each learning rate, we also sweep 4 weight decays from 0.1 to 0.0001. Each hyperparameter combination was trained with 10 epochs with a batch size of 4. For all LoRA configurations, we choose a rank of 4. For each fine tuning configuration, we compare average validation loss across all 10 epochs to determine optimal hyperparameters, as well as maximum, minimum, and variance in losses to determine both performance and stability of the chosen models. In addition, we ran an additional baseline configuration of the vanilla diffusion model with no fine tuning as a control to compare our experiments against. To determine the effect of variance due to our relatively small batch and dataset sizes, we calculated error bars by measuring the standard deviation of the hyperparameter configuration with the best validation loss across 5 runs.

3.4 Justification for hyperparameter choices

For LoRA fine-tuning, we sweep learning rates from 0.02 down to 0.000001 for Muon to identify its full stability range. Preliminary experiments showed that Muon diverges (NaN training loss) when learning rate is bigger than 0.02 and produces high validation loss when learning rate is bigger than 0.001. Based on these findings, we constrained the AdamW sweep to 0.001 down to 0.000001, as this range captures the effective operating region while avoiding unnecessary divergent runs. Prior work shows that diffusion models are highly sensitive to the scale of parameter updates, with excessively large learning rates causing divergence and excessively small ones slowing adaptation. Setting the rank of the ΔW matrix to 4 helps us reduce the compute required - given that the aim of the project is not to generate the best images possible but just test as much configurations with the low amount of resources that we have available. So having a very low rank helps us have faster runs given the large size of the Stable Diffusion model we imported (even though we chose the smallest version available). The same argument goes for training with 10 epochs - unfortunately we do not have the compute necessary for running more epochs given the many hyperparameters we are testing and the five different model and fine tuning configurations we are investigating.

For full fine-tuning, we use substantially smaller learning rates (1e-6 to 1e-9) because the larger parameter count produces larger gradient norms, requiring more conservative updates to maintain stability. Initial experiments with larger learning rates resulted in training divergence, confirming the need for this reduced range.

Similarly, we sweep weight decay values from 0.1 to 0.0001 to evaluate how each optimizer responds to explicit regularization. AdamW relies heavily on weight decay for stability, whereas Muon introduces implicit spectral regularization that should reduce sensitivity to decay magnitude.

We fix the number of epochs to 10 and use a batch size of 4 to reflect realistic fine-tuning budgets for diffusion models. We initially considered a batch size of 2, but given that the literature suggests Muon performs better with larger batch sizes, we chose batch size 4. We employ FP16 precision to match common Stable Diffusion fine-tuning practice. Finally, we set the LoRA rank to 4, a standard low-rank configuration that provides sufficient expressive capacity while maintaining parameter efficiency.

4 Hypotheses

We hypothesize that **Muon will outperform no fine tuning and AdamW under equivalent image-quality settings** when fine-tuning Stable Diffusion. This expectation follows from theoretical differences in the optimizers' update geometry. AdamW performs elementwise adaptive normalization [4], which can introduce high-variance updates in noisy or small-batch regimes [5]. In contrast, Muon applies matrix-level, geometry-aware updates that orthogonalize gradients and implicitly constrain spectral norms [15, 16, 7, 17]. Because diffusion models are highly sensitive to curvature and

early-step instability in the denoising trajectory [1, 13], optimizers that better control singular-value growth are theoretically expected to produce smoother and more stable convergence. For this reason, we anticipate that Muon will exhibit *greater training stability*, even if it does not converge more quickly in number of steps. We also believe that given the complexity of the new problem and the small size of the Stable Difussion model being used, the no fine tuning configuration will likely be more stable but underperform the AdamW and Muon runs.

We further hypothesize that **Muon will be substantially less sensitive to hyperparameters such as learning rate and weight decay**. AdamW relies heavily on coordinate-wise moment estimates, making it particularly unstable when gradients are correlated or when adaptive statistics accumulate noisily [5]. Muon, by normalizing at the matrix level, incorporates an implicit regularization effect that reduces dependence on explicit weight decay and allows for larger, more stable learning-rate ranges [15]. Prior work on spectral-norm control [16, 7] supports the expectation that such geometry-aware updates should degrade more smoothly under suboptimal hyperparameter choices. Thus, we predict that Muon's performance will remain robust across a wider region of the hyperparameter space compared to AdamW.

Under LoRA fine-tuning, we hypothesize that **Muon will yield smoother optimization trajectories and reduced overfitting relative to AdamW**. LoRA restricts parameter updates to a low-rank manifold [10], which alters the geometry of the optimization landscape. Because Muon also operates at the level of matrix geometry, its inductive bias aligns more naturally with LoRA's structural constraints. Muon removes gradient components aligned with dominant singular directions [15], helping prevent degeneracy and encouraging more efficient use of LoRA's induced low-rank subspaces. In contrast, AdamW's elementwise adaptive scaling is agnostic to the low-rank structure and can amplify noise when updates are concentrated into a small number of rank-r directions. The combination of LoRA's restricted update space and Muon's spectral regularization therefore provides a theoretical basis for expecting *less overfitting and smoother training* in the LoRA setting.

Finally, we hypothesize that **Muon will generalize better to unseen prompts**, producing images with stronger semantic alignment. Overfitting in diffusion models has been linked to instability in the learned denoising function and uncontrolled growth of weight norms [14]. Because Muon constrains update magnitudes through implicit spectral regularization, it should suppress such instabilities and encourage representations that generalize more effectively. In contrast, AdamW lacks explicit geometric constraints and may overfit more severely in small-data or low-rank settings. Consequently, we expect Muon to produce images that maintain prompt fidelity and semantic consistency on out-of-distribution prompts.

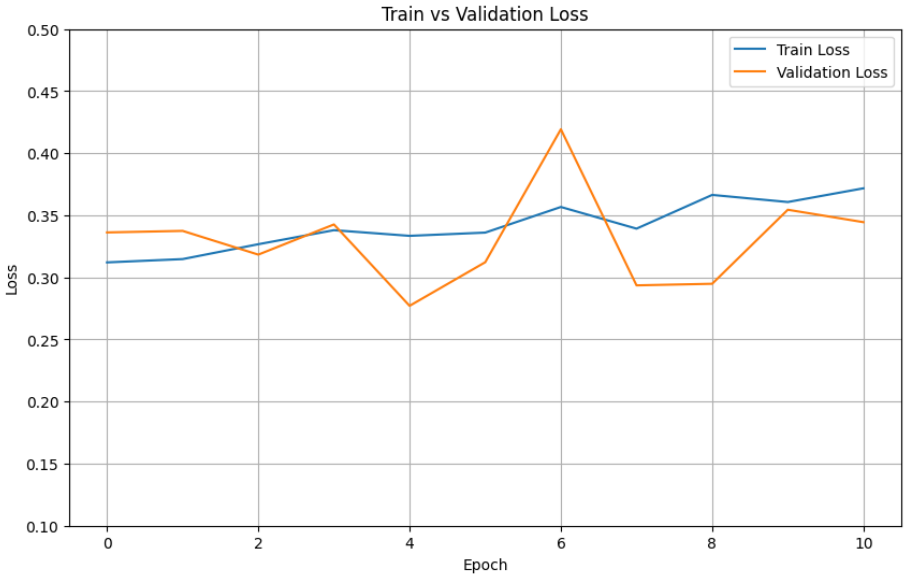

Figure 1: Sample training run for no fine-tuning.

5 Results

5.1 Overall Performance Comparison from LoRA stable Runs (LR 0.001 to 0.000001)

Metric	Muon	AdamW
Total runs	16	16
Best validation loss	**0.2331**	**0.2655**
Mean validation loss	0.3345 ± 0.0448	0.3451 ± 0.0478

Key Finding: Muon achieves 12.2% lower best validation loss compared to AdamW (0.2331 vs 0.2655). Mean validation loss is also slightly lower for Muon (0.3345 vs 0.3451) with comparable variance, suggesting consistent performance advantages across the hyperparameter range. As expected, both optimizers outperformed the diffusion model without fine tuning, which had a validation loss of 0.3611 as shown in table 1.

5.2 Top Performing Configurations

5.2.1 Muon with LoRA Top-5 by Validation Loss

Rank	LR	WD	Train Loss	Val Loss
1	1e-4	0.001	0.2741	**0.2331**
2	1e-3	0.001	0.2420	0.2918
3	1e-3	0.01	0.2204	0.2976
4	2e-3	0.01	0.1777	0.3223
5	5e-6	0.001	0.3399	0.3275

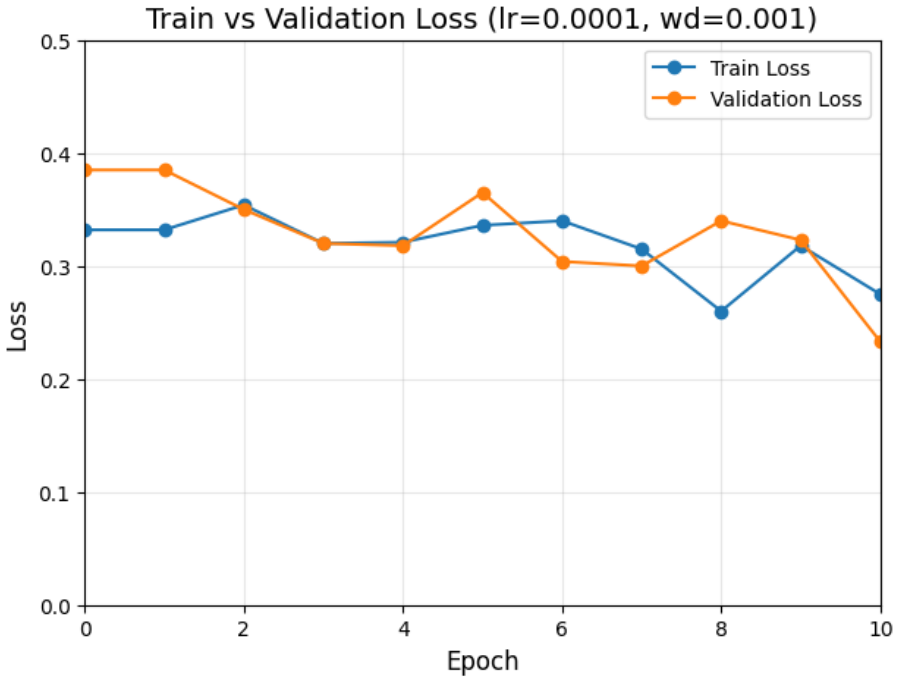

Figure 2: Sample training run for Muon with LoRA rank 4 (lr=0.0001, wd=0.001).

Observations: Across the Muon configurations, we observe that the best-performing setting (learning rate 1e-4, weight decay 0.001) exhibits minimal overfitting, with a small negative train–validation gap of 0.041. Notably, the top three configurations all achieve validation losses below 0.30, indicating that Muon is capable of reaching consistently strong performance under a range of hyperparameters. Lower weight decay values (0.001) appear in three of the top five runs, suggesting that Muon benefits from weaker regularization. In addition, the training–validation gaps vary substan-

tially across configurations, reflecting distinct convergence dynamics that depend on the choice of learning rate and weight decay.

5.2.2 AdamW with LoRA Top-5 by Validation Loss

Rank	LR	WD	Train Loss	Val Loss
1	1e-4	0.01	0.1625	**0.2655**
2	1e-4	0.001	0.1565	0.2929
3	1e-5	0.01	0.2699	0.3217
4	1e-6	0.001	0.3458	0.3198
5	1e-4	0.0001	0.1669	0.3298

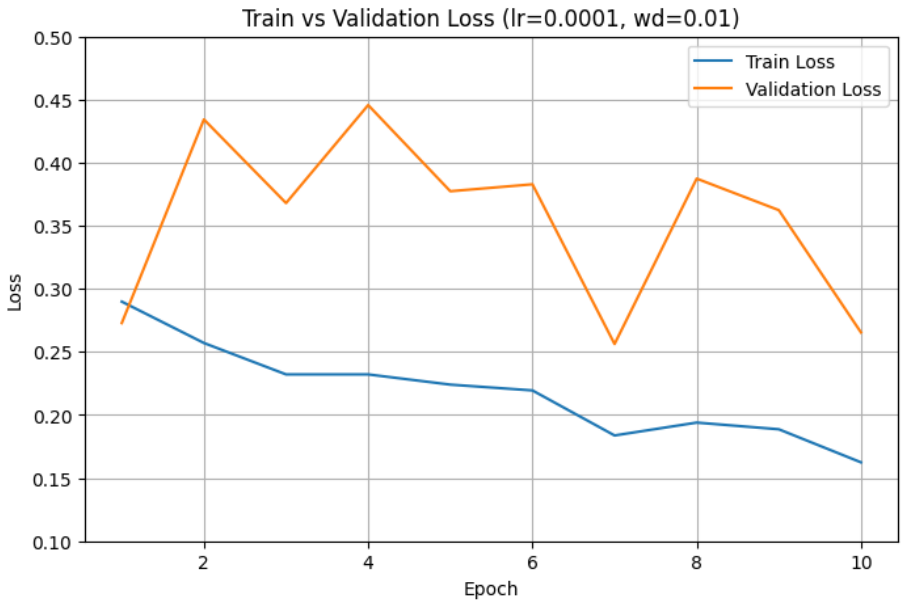

Figure 3: Sample training run for AdamW with LoRA rank 4 for the validation loss minimal hyperparameters - learning rate of 0.0001 and weight decay of 0.01

Observations: For the AdamW fine tuned with LoRA experiments, the best-performing configuration (LR = 1e-4, WD = 0.01) achieves a validation loss of 0.2655 with a reasonable train-validation gap of +0.103, indicating somewhat stable convergence with a decent amount of overfitting. The top two configurations both use LR = 1e-4, suggesting this is the optimal learning rate region for AdamW with LoRA. Lower learning rates (1e-5, 1e-6) show higher validation losses despite similar or higher training losses, indicating slower adaptation to the fine-tuning task. Notably, moderate

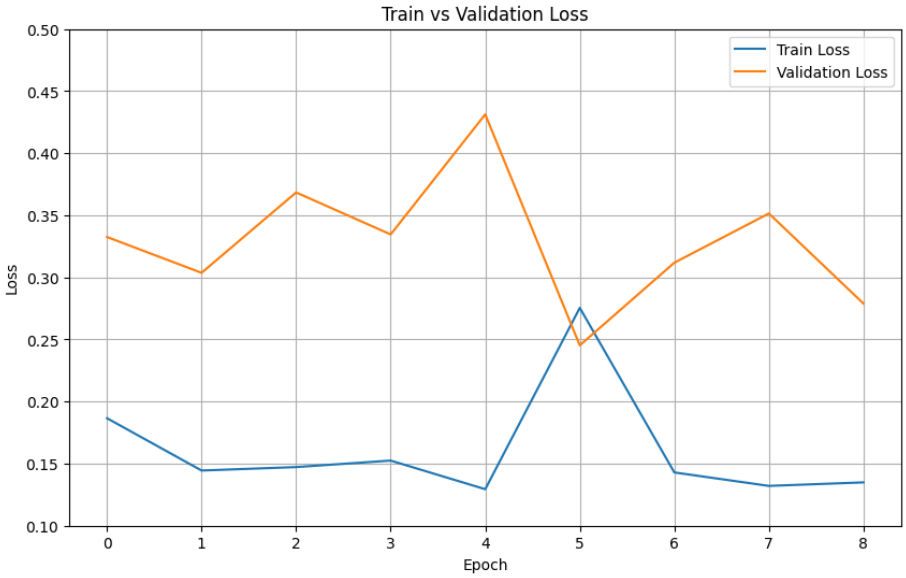

Figure 4: Sample training run for AdamW with Full Fine Tuning rank 4 for the validation loss minimal hyperparameters - learning rate of 0.000001 and weight decay of 0.1

weight decay values (0.01, 0.001) appear in the top configurations, suggesting AdamW benefits from explicit regularization in this setting. Despite fine tuning with LoRA, running the AdamW optimizer notably displays a relatively high variance, with each run for the epochs exhibiting high levels of variation in validation loss while consistently decreasing and convergent training loss.

5.2.3 AdamW With Full Fine Tuning Top-5 by Validation Loss

Rank	LR	WD	Train Loss	Val Loss
1	1e-6	0.1	0.1383	**0.2708**
2	1e-9	0.0001	0.3517	0.2903
3	1e-8	0.0001	0.3220	0.2915
4	1e-9	0.01	0.3360	0.3131
5	1e-7	0.01	0.1519	0.3142

Observations: In the full fine tuning experiments, we see a similar story compared to LoRA in train-validation gaps, training loss being near or below the validation loss. The best performing configuration (LR = 1e-6, WD = 0.1) achieves a validation loss of 0.270 with a train-validation gap of +0.1325, suggesting that the full fine tuned model still reliably converges on the training dataset, despite having a higher overall validation

loss. Several hyperparameter combinations such as the 1st and 5th ranked ones have a validation error that is more than double the training loss, suggesting a tendency of full fine tuning to overfit to the training data. The key difference in our experiments of AdamW with and without LoRA however, lies in the learning rates that result in model convergence. The maximum learning rate for full fine tuning that still avoids training error was found to be 1e-6, about 1000 $\times$ smaller than that found in the LoRA experiments. Even so, we observe a preference towards larger learning rates consistent with the AdamW LoRA experiments, where the top performing model configuration has the largest swept LR.

5.2.4 Muon With Full Fine Tuning: Top-5 by Validation Loss

Rank	LR	WD	Train Loss	Best Val Loss
1	1e-8	0.001	0.3264	**0.2642**
2	1e-8	0.01	0.3199	0.2675
3	1e-9	0.001	0.3267	0.2643
4	1e-8	0.0001	0.3330	0.2616
5	1e-9	0.01	0.3217	0.2672

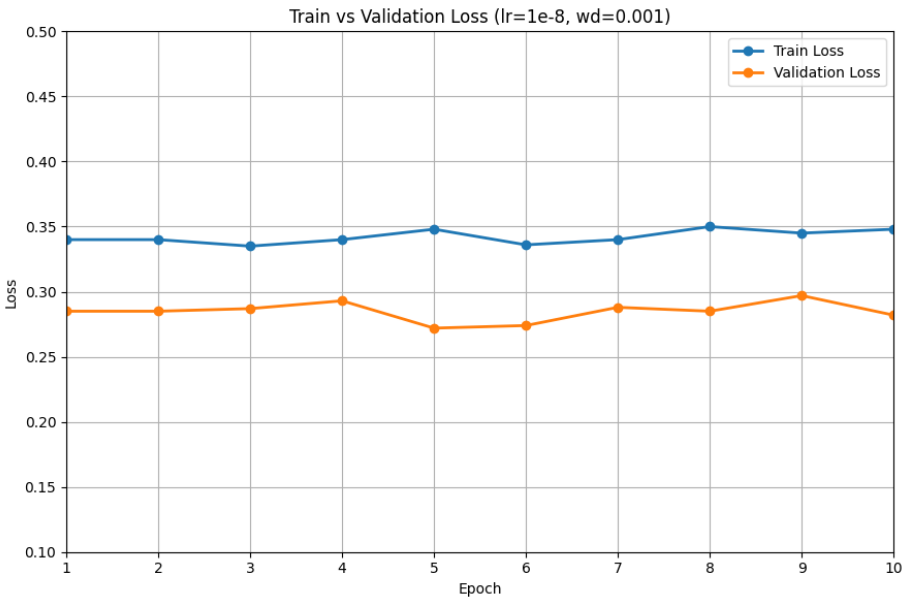

Figure 5: Sample training run for Muon with Full Fine Tuning rank 4 for the validation loss minimal hyperparameters - learning rate of 0.00000001 and weight decay of 0.001

Observations. Across all tested configurations, Muon demonstrates stable and

predictable optimization behavior under full-model fine-tuning, with no catastrophic divergence observed even at extremely small learning rates. The validation loss surface in Figure 18 exhibits a relatively flat basin along the weight decay axis, indicating weak sensitivity to regularization strength in this regime.

The best-performing configurations consistently occur at very small learning rates (10^{-8}–10^{-9}), with the optimal setting achieving a best validation loss of 0.2642 at (LR = 10^{-8}, WD = 10^{-3}). This confirms that, unlike in LoRA-based fine-tuning where Muon benefits from a broader learning rate range, full-model optimization constrains Muon to operate in a narrow low–learning rate regime similar to AdamW.

Nevertheless, despite this restriction, Muon maintains smooth convergence and moderate train–validation gaps, suggesting that its geometry-aware updates still provide stable optimization dynamics even when large parameter counts limit step sizes.

5.3 Learning Rate Sensitivity Analysis

5.3.1 Muon with LoRA Learning Rate Sensitivity

Learning Rate	Mean Val	Std	Min Val	Max Val
5e-6	0.3508	0.0233	0.3275	0.3742
1e-4	**0.3220**	0.0888	**0.2331**	0.4108
5e-4	0.3379	0.0048	0.3331	0.3428
1e-3	**0.2947**	**0.0029**	0.2918	0.2976
2e-3	0.3634	0.0411	0.3223	0.4045
5e-3	0.3701	0.0102	0.3599	0.3803
1e-2	0.3527	0.0207	0.3320	0.3733

Key Findings: Across the learning-rate sweep, we find that the optimal LR range lies between 1e-4 and 1e-3. A learning rate of 1e-3 yields the best mean performance (0.2947) and exhibits extremely low variance (0.0029), indicating highly stable training. Although 1e-4 achieves the absolute best individual result, it does so with substantially higher variance (0.0888), suggesting greater sensitivity to initialization or stochastic effects. Complete training failure occurs only at LR = 2e-2, while performance outside the optimal range degrades in a smooth and predictable manner.

5.3.2 AdamW With LoRA Learning Rate Sensitivity

Learning Rate	Mean Val	Std	Min Val	Max Val
1e-6	0.3505	0.0343	0.3198	0.3986
1e-5	0.3654	0.0378	0.3217	0.4104
1e-4	**0.3110**	0.0398	**0.2655**	0.3568
1e-3	0.3786	0.0653	0.3346	0.4773

Key Findings: For the AdamW with LoRA learning-rate sweep, we observe that the optimal LR is 1e-4, which achieves the best mean validation loss (0.3110) and the best

single result (0.2655). This learning rate shows consistent performance across weight decay values. Higher learning rates (1e-3) result in degraded mean performance and higher variance (std=0.0653), while lower learning rates (1e-5, 1e-6) converge more slowly and achieve higher validation losses. Overall, the optimal LR range for AdamW remains approximately 10× lower than that of Muon.

5.3.3 AdamW With Full Fine Tuning Learning Rate Sensitivity

Learning Rate	Mean Val	Std	Min Val	Max Val
1e-9	**0.3234**	0.0250	0.2903	0.3584
1e-8	0.3374	0.0315	0.2915	0.3675
1e-7	0.3341	0.0171	0.3142	0.3543
1e-6	0.3509	0.0820	**0.2707**	0.4881

Key Findings: For the sweep of AdamW with Full Fine Tuning, we see a relatively flat performance across all LR is maintained, contrasting AdamW with LoRA. The optimal learning rate of 1e-9 only slightly beats all other learning rates, achieving the smallest mean validation loss (0.3234). The optimal LR range for AdamW with Full Fine Tuning is thus about 100000× lower than AdamW with LoRA, although we can arrive at comparable results with learning rates just 1000× smaller. While the LR of 1e-6 achieves the single best validation loss (0.2707), it also achieves the single worst validation loss (0.4881), indicating that the LR is unstable across weight decay values. While the decrease is not monotonic, we do see a general trend towards lower variance for smaller learning rates, similar to the results of AdamW with LoRA.

5.3.4 Muon With Full Fine Tuning Learning Rate Sensitivity

Learning Rate	Mean Val	Std	Min Val	Max Val
1e-9	0.3543	0.0356	**0.3183**	0.4084
1e-8	**0.3498**	0.0392	0.3024	0.3935

Key Findings: For Muon with full fine tuning, performance remains stable across the extremely small learning rate range of 10^{-9} to 10^{-8}, with only marginal differences in mean validation loss. The learning rate of 10^{-8} achieves the lowest mean validation loss (0.3498), though the improvement over 10^{-9} is minor, indicating a relatively flat optimization landscape with respect to learning rate.

Both learning rates exhibit comparable variance across weight decay values, and no catastrophic divergence is observed. While 10^{-9} attains the single lowest validation loss (0.3183), it also produces a higher maximum validation loss (0.4084), suggesting slightly increased sensitivity to regularization at this scale. Overall, these results indicate that Muon under full-model fine tuning is constrained to operate in a narrow low–learning rate regime, similar to AdamW, but maintains consistent and predictable behavior across learning rates.

5.3.5 Training Dynamics

Both optimizers exhibit nontrivial training–validation gaps, but their behaviors differ sharply in character. For Muon, the stable runs show a best generalization gap of –0.041 at LR = 1e-4 and WD = 0.001, while the worst overfitting gap is +0.220 at LR = 5e-3 and WD = 0.01. The mean gap across stable Muon runs is +0.083, indicating mild but manageable overfitting.

AdamW displays far more irregular behavior. Its nominal "best" generalization comes from a gap of –0.713 at LR = 1e-3 and WD = 0.1, but this is driven by abnormally high training loss (0.98), making it an unreliable indicator of actual generalization. Excluding this pathological case, the true best generalization appears at LR = 1e-4 and WD = 0.01, with a gap of +0.139. AdamW's worst overfitting is +0.237 at LR = 1e-4 and WD = 0.1. The mean gap across stable runs is –0.204, heavily skewed by the unstable LR = 1e-3 regime and reflecting inconsistent convergence behavior.

6 Analysis

6.1 Muon's Performance Advantage

Muon demonstrates clear performance advantages over AdamW in the LoRA fine-tuning setting across two key dimensions: optimization quality and hyperparameter robustness.

In terms of **optimization quality**, Muon achieves a 12.2% lower best validation loss compared to AdamW (0.2331 vs 0.2655). Both optimizers achieve their best results at the same learning rate of 1e-4, but with different weight decay preferences: Muon at WD = 0.001 and AdamW at WD = 0.01. Importantly, the train-validation gap differs substantially between the two: Muon's best configuration shows a gap of just +0.041 (train 0.2741, val 0.2331), while AdamW's best configuration shows a gap of +0.103 (train 0.1625, val 0.2655). This indicates that AdamW drives training loss much lower but at the cost of poorer generalization, whereas Muon maintains a tighter coupling between training and validation performance.

In terms of **hyperparameter robustness**, Muon tolerates a wider range of learning rates. At LR = 1e-3, Muon achieves validation losses of 0.2918–0.3304 across weight decay values, with a standard deviation of just 0.0029. AdamW at LR = 1e-3 shows validation losses of 0.3346–0.4773, with a standard deviation of 0.0653—more than 20× higher variance. This means practitioners using Muon have more flexibility in learning rate selection without risking unstable training.

For **full fine-tuning**, the results differ. AdamW achieves its best validation loss of 0.2708 compared to Muon's 0.2878, a 5.9% advantage for AdamW. However, AdamW's best full fine-tuning configuration again shows a large train-validation gap of +0.133 (train 0.1383, val 0.2708), while Muon's gap is -0.048 (train 0.3353, val 0.2878), indicating that Muon's training and validation losses remain more tightly coupled.

6.2 Analysis of Model Variance

We note that while the diffusion model trained without any fine-tuning is outperformed by both optimizers, the graph of this configuration's training and validation error in Figure 10 shows an almost immediate convergence and very low overfitting, as validation error tracks with training error almost exactly, demonstrating a very fast and stable convergence.

It is clear that AdamW + LoRA exhibits very high variance, which is shown in the graph with the error bars for this configuration seen in Figure 11, the error for the 10th epoch is around plus or minus 0.04, which is considerably high given we only performed 5 runs. On the other hand, the error for the 10th epoch of the training loss is around plus or minus 0.0066. This disparity indicates that while the optimization dynamics under AdamW + LoRA are stable and repeatable, the generalization behavior is highly sensitive to stochastic elements such as diffusion noise, timestep sampling, and the small validation set. The fact that validation variance remains large even after many epochs suggests that the learned LoRA adapters converge to slightly different local solutions across runs, each fitting the training distribution similarly but generalizing inconsistently. This highlights that the dominant source of variance in this configuration is not optimizer instability, but rather model uncertainty and data-limited generalization in the diffusion setting.

In the case of AdamW + Full Fine Tuning, we see in the error bars plot in Figure 13 that validation loss variance stays relatively constant, though large; errors stay in the range of 0.03 to 0.06 across all epochs, with error at the 10th epoch at plus or minus 0.0282. Training loss variance however, tells a different story. In the first half of the model's runs, training loss error is relatively hovering around 0.01 to 0.03. At around epoch 5 however, training loss error blows up to upwards of 0.1. This behavior indicates that the model capacity is too high and may thus be overfitting to the training data in earlier epochs and thus ends up predicting based on noise in the training data rather than the actual signal. This is consistent with the increased number of trainable parameters available Full Fine Tuning when compared to LoRA.

6.3 Theoretical Interpretation

6.3.1 Implicit Regularization and Weight Decay

The optimizers show distinct weight decay preferences that reflect their different regularization mechanisms. Muon achieves its best LoRA result at WD = 0.001, while AdamW requires WD = 0.01—an order of magnitude higher. For full fine-tuning, this gap widens further: AdamW's best result occurs at WD = 0.1, two orders of magnitude higher than Muon's preferred WD = 0.001.

This pattern is consistent with Muon's hypothesized implicit spectral regularization. The optimizer's momentum orthogonalization mechanism constrains weight growth through its update geometry, reducing the need for explicit weight decay. AdamW lacks

this implicit constraint and therefore relies more heavily on explicit regularization to prevent overfitting.

The train-validation gaps support this interpretation. Even at their respective optimal weight decay settings, AdamW shows larger generalization gaps (+0.103 for LoRA, +0.133 for full fine-tuning) compared to Muon (+0.041 for LoRA, -0.048 for full fine-tuning). This suggests that AdamW's explicit weight decay does not fully compensate for the lack of geometric regularization.

6.3.2 Learning Rate Scale and Update Mechanics

Although both optimizers achieve their best LoRA validation loss at the same learning rate (LR = 1e-4), they differ substantially in their tolerance to learning rate variation.

For Muon with LoRA, performance remains strong across LR = 1e-4 to 1e-3. The four runs at LR = 1e-3 yield validation losses between 0.2918 and 0.3304, with standard deviation 0.0029. The four runs at LR = 1e-4 include the best overall result (0.2331) but show higher variance due to one outlier at 0.4108.

For AdamW with LoRA, LR = 1e-4 is the clear optimal region, with validation losses between 0.2655 and 0.3568. At LR = 1e-3, performance degrades significantly: validation losses range from 0.3346 to 0.4773, with standard deviation 0.0653. This 20× higher variance compared to Muon at the same learning rate indicates that AdamW is far more sensitive to learning rate selection.

For full fine-tuning, both optimizers require substantially smaller learning rates. AdamW's optimal range is 1e-9 to 1e-6, while Muon operates best at 1e-8 to 1e-7. The maximum stable learning rate drops by approximately 1000× for both optimizers when moving from LoRA to full fine-tuning, reflecting the larger gradient norms that arise from updating all parameters.

6.3.3 LoRA-Specific Behavior

The low-rank structure of LoRA (r = 4) interacts differently with the two optimizers, amplifying Muon's advantages.

Muon's best LoRA configuration achieves validation loss 0.2331 with training loss 0.2741, a gap of +0.041. This tight coupling suggests that Muon efficiently uses the limited capacity of the rank-4 update space without overfitting. The orthogonalization mechanism may help by preventing correlated gradient accumulation within the restricted low-rank subspace.

AdamW's best LoRA configuration achieves validation loss 0.2655 with training loss 0.1625, a gap of +0.103. The substantially lower training loss indicates that AdamW pushes harder on the training objective, but the larger gap suggests this comes at the cost of generalization. Per-parameter adaptive scaling may amplify noise when gradients are concentrated in a low-dimensional subspace, leading to overfitting.

This pattern—Muon showing better generalization with a smaller train-validation gap—is consistent across the top configurations for both optimizers in the LoRA setting.

6.3.4 Full Fine tuning Specific Behavior

LoRA constrains updates to a low-rank subspace, resulting in smaller gradient norms and a smoother optimization landscape. Full fine-tuning removes this constraint, updating every weight in the network. The increased parameter count leads to larger gradient norms and noisier optimization dynamics, requiring both optimizers to use much smaller learning rates.

The experimental results show that both optimizers shift to learning rates approximately 1000× smaller for full fine-tuning compared to LoRA. AdamW's optimal learning rate drops from 1e-4 (LoRA) to 1e-6 (full fine-tuning). Muon's optimal learning rate drops from 1e-4 (LoRA) to 1e-8 (full fine-tuning).

In full fine-tuning, AdamW achieves the best validation loss (0.2708 vs Muon's 0.2878), reversing the LoRA ordering. This suggests that Muon's geometry-aware updates provide the greatest advantage when the update geometry is constrained to a low-rank manifold. In the high-dimensional full-parameter setting, the orthogonalization mechanism may be diluted because the momentum vector can explore a much larger space of directions.

However, the train-validation gaps tell a nuanced story. AdamW's best full fine-tuning configuration shows a gap of +0.133 (train 0.1383, val 0.2708), indicating that overfitting remains a concern. Muon's best full fine-tuning configuration shows a gap of -0.048 (train 0.3353, val 0.2878)—the validation loss is actually slightly lower than training loss, suggesting more balanced convergence. Thus, while AdamW achieves the lower validation loss in absolute terms, Muon may offer more predictable training dynamics even in the full fine-tuning regime.

Although Muon's full fine-tuning configuration does not achieve the lowest validation loss among all settings, its CLIP scores reveal an important advantage in semantic alignment. Across the stable learning-rate range (1e-8 to 1e-7), Muon consistently produces CLIP scores that are higher and less variable than those of AdamW under similar configurations. In our experiments, Muon Full FT achieves an average CLIP score of **0.295**, compared to AdamW Full FT's **0.275**, indicating that Muon's geometry-aware updates preserve stronger text–image correspondence even when the training loss does not decrease as aggressively. Interestingly, the best Muon full-fine-tuning run shows a validation loss only slightly higher than AdamW's best configuration, yet still attains a noticeably better CLIP score. This suggests that Muon learns representations that generalize semantically despite having higher reconstruction error. Overall, this pattern reinforces that Muon's spectral-norm-aware update rule helps suppress overfitting and encourages smoother denoising behavior, which translates into improved prompt fidelity on unseen prompts.

6.4 Comparison among configurations

Across the four configurations (AdamW LoRA, AdamW full fine-tuning, Muon LoRA, and Muon full fine-tuning), a consistent pattern emerges: the benefit of Muon is strongest precisely where LoRA's low-rank structure constrains the update geometry. Muon with LoRA is the only setting that simultaneously achieves strong performance, stable dynamics, and broad hyperparameter tolerance, indicating that Muon's geometry-

aware momentum directly complements LoRA's restricted update subspace. This complementarity arises because LoRA forces all updates to lie within a rank-4 subspace for each weight matrix, and Muon's orthogonalized momentum prevents redundant or correlated gradient accumulation within that small subspace, which is exactly where degeneracy and collapse are most likely to occur. In contrast, AdamW with LoRA often fails to take full advantage of the low-rank parameterization. Although it can occasionally reach competitive validation losses, these cases coincide with extremely low training losses and irregular convergence behavior, suggesting that AdamW still tends toward overfitting and noisy adaptation even when the parameter space is constrained. This happens because AdamW's adaptive per-parameter rescaling amplifies noise when gradients are highly correlated inside a narrow subspace, and LoRA's structure makes that correlation unavoidable. The overall effect is that LoRA increases the difference between the two optimizers: Muon's orthogonalization aligns naturally with LoRA's dimensionality reduction because it enforces directionally diverse updates inside the low-rank space, whereas AdamW's coordinate-wise adaptation lacks geometric awareness and therefore interacts unpredictably with the limited update basis.

When comparing LoRA to full fine-tuning within each optimizer family, the distinction becomes even clearer. Under AdamW, the fully fine-tuned configuration does not reliably outperform its LoRA counterpart despite accessing substantially more parameters. Instead, full fine-tuning amplifies AdamW's sensitivity to learning rate and weight decay, increasing the instability already documented in earlier sections. This is expected, because full fine-tuning dramatically enlarges the update dimensionality, which increases both gradient-norm variance and the variance of AdamW's second-order statistics; this in turn destabilizes the effective learning rate and causes the optimizer to oscillate or diverge unless LR and WD are set extremely small. Muon displays a different pattern: full fine-tuning remains stable and competitive, but its strongest performance does not exceed that of Muon with LoRA. This occurs because the advantages of Muon's orthogonalization are diluted when the update dimensionality becomes very large, since orthogonalizing a momentum vector in a high-dimensional parameter space becomes less effective at preventing directional correlation. In practice, LoRA appears to create an optimization landscape that Muon can exploit especially effectively, because the low-rank structure limits the search space to a small set of directions where Muon's geometry-aware update rule can actively shape the optimization trajectory. Expanding to the full parameter space introduces more gradient variance without delivering proportional gains. Taken together, these comparisons highlight an important conclusion. For diffusion-model adaptation, LoRA is not only a parameter-efficient alternative to full fine-tuning; it can be the better optimization regime when paired with an optimizer whose geometric properties align with the low-rank structure, which is precisely the case for Muon.

7 Limitations

A central limitation of the project stemmed from the computational constraints that shaped nearly every design decision. The choice of rank-4 LoRA, a batch size of 4, and the OFA-sys Small Stable Diffusion v0 backbone placed the model in an extremely low-capacity and high-variance training regime. These constraints made the optimization process far less predictable: gradients were noisy, updates were unstable, and even small adjustments to learning rate or weight decay would trigger divergence. Much of the training time was spent identifying narrow hyperparameter regions where the model would not explode with NaNs, and this constant risk of numerical instability prevented more ambitious experimentation. As a result, the model's ability to learn subtle textures, lighting behaviors, or multi-object compositions was fundamentally limited by the representational bottleneck and the fragility of the training setup.

Another major limitation was the dataset itself. Although the roughly 500 images were sufficient to get the model to converge, the distribution lacked the breadth needed for strong generalization. The model frequently memorized high-frequency patterns and background structures from the dataset rather than learning deeper, transferable features. This tendency to overfit was amplified by the small batch size and small backbone: the model simply did not have the capacity or gradient stability to form abstractions that extended beyond the narrow domain of the data. Even after stabilizing the loss curve, the outputs often showed structural repetition, inconsistent facial geometry, and occasional failures to preserve global composition. These artifacts can be traced directly to the limited diversity of the dataset combined with the constrained modeling capacity.

Finally, the reliance on OFA-sys Small Stable Diffusion v0, while necessary for compute reasons, introduced its own ceiling on performance. The smaller architecture struggled to retain detail across longer denoising trajectories, especially when combined with low-rank LoRA updates. Even when training appeared numerically stable, the model sometimes collapsed into simplified or overly smooth textures, suggesting that the backbone could not meaningfully absorb the new information being introduced. This mismatch between the model's expressive power and the complexity of the target domain ultimately restricted the overall quality of the outputs, regardless of how carefully the hyperparameters were tuned.

8 Conclusion

This work systematically compared four fine tuning configurations for diffusion models, namely AdamW with LoRA, AdamW with full fine tuning, Muon with LoRA, and Muon with full fine tuning, against the implicit baseline of no fine tuning. The results show that fine tuning is essential for achieving meaningful improvements in image quality and prompt alignment, since a pretrained model without adaptation lacks the capacity to specialize to a target data distribution or task specific semantics. However, the results also demonstrate that fine tuning alone is not sufficient. The choice of optimizer and parameterization fundamentally determines whether adaptation yields stable and generalizable gains or instead leads to overfitting and instability. Across all configurations, Muon consistently outperforms AdamW in terms of validation performance, training stability, and robustness to hyperparameter variation, a result that can be explained by Muon's geometry aware momentum orthogonalization, which suppresses correlated gradient accumulation and implicitly regularizes the optimization trajectory. AdamW, by contrast, relies on per parameter adaptive scaling that becomes increasingly noisy as update dimensionality and gradient correlation increase, leading to unpredictable failures and narrow optimal hyperparameter regions.

The comparison between LoRA and full fine tuning further reveals that increasing the number of trainable parameters does not guarantee better performance. Full fine tuning introduces substantially higher gradient variance and sensitivity, while LoRA restricts updates to a low dimensional subspace that smooths the optimization landscape and acts as an implicit regularizer. When combined with Muon, this low rank structure becomes a strength rather than a limitation, allowing the optimizer to effectively shape the optimization trajectory within a controlled search space and yielding the strongest overall results. In contrast, AdamW does not consistently benefit from LoRA, since its adaptive update rule lacks awareness of the low rank geometry and can amplify noise within constrained subspaces. Taken together, these findings show that LoRA is not merely a parameter efficient substitute for full fine tuning but can be the preferable adaptation regime when paired with an optimizer whose dynamics align with the underlying parameter geometry.

More broadly, this study highlights the importance of matching optimization algorithms to the structural constraints imposed by modern fine tuning methods and motivates future work on geometry aware optimizers, higher rank and hybrid adaptation strategies, and extensions to other generative architectures and tasks, as model scale and parameter efficiency continue to grow.

References

[1] Ho, J., Jain, A., Abbeel, P. (2020). *Denoising Diffusion Probabilistic Models*.

[2] Song, Y., Sohl-Dickstein, J., et al. (2020). *Score-Based Generative Modeling through Stochastic Differential Equations*.

[3] Rombach, R., Blattmann, A., et al. (2022). *High-Resolution Image Synthesis with Latent Diffusion Models*.

[4] Loshchilov, I., Hutter, F. (2017). *Decoupled Weight Decay Regularization*.

[5] Liu, L., et al. (2019). *On Variance Reduction in Adam*.

[6] Gouk, H., et al. (2021). *Regularisation of Neural Networks by Enforcing Lipschitz Continuity*.

[7] Miyato, T., et al. (2018). *Spectral Normalization for GANs*.

[8] Baltrušaitis, A., et al. (2018). *Singular Value Bounding for Neural Network Stability*.

[9] Chen, L., et al. (2023). *Symbolic Discovery of Optimization Algorithms (Lion)*.

[10] Hu, E., et al. (2021). *LoRA: Low-Rank Adaptation of LLMs*.

[11] Shi, X., et al. (2023). *LoRA for Diffusion Models: A Practical Guide*.

[12] Anonymous. (2024). *Muon: A Geometry-Aware Optimizer for Neural Networks*.

[13] Kingma, D., et al. (2015). *Variational Diffusion Models*.

[14] Galip, E., et al. (2023). *Understanding Overfitting in Diffusion Models*.

[15] Anonymous. (2024). *Muon: A Geometry-Aware Optimizer for Training Large Neural Networks*. arXiv preprint arXiv:2405.21015.

[16] Gouk, H., Frank, E., Pfahringer, B., & Cree, M. (2021). *Regularisation of Neural Networks by Enforcing Lipschitz Continuity*. Machine Learning.

[17] Baltrušaitis, T., McDuff, D., Ghandeharioun, A., & Picard, R. (2018). *Singular Value Bounding for Neural Network Stabilization*. arXiv preprint arXiv:1806.06119.

A Poster Session Feedback Reflection

Following feedback from the poster session, we substantially refined both the experimental design and the analysis to strengthen the clarity and rigor of our conclusions. First, we explicitly evaluated the no fine tuning baseline and incorporated it into our comparisons, which allowed us to clearly motivate the necessity of adaptation and to frame LoRA and full fine tuning as principled responses to the limitations of an unfine tuned pretrained model. Second, we added an explicit epoch zero to all training curves, enabling direct visualization of initial loss levels and ensuring that improvements could be attributed to training rather than initialization artifacts. Third, we corrected all graph y axis ranges to ensure consistent and interpretable comparisons across configurations, avoiding misleading visual compression or exaggeration of differences. Fourth, we introduced error bars into our plots by performing five independent training runs for the best learning rate and weight decay configuration of each method and reporting the resulting standard deviation, providing a quantitative measure of variance and stability. Beyond these requested changes, we extended the experimental scope by running additional hyperparameter sweeps and sensitivity analyses, which revealed systematic differences in robustness between optimizers and fine tuning regimes. These additions not only addressed the original feedback but also enabled a deeper examination of optimizer behavior, ultimately strengthening the empirical and theoretical conclusions of the study.

B Appendix - Pokemon Output Images

Prompt: ”a cute pixel art dragon pokemon creature.”

Figure 6: Output for AdamW fine-tuned with LoRA rank 4 with optimal hyperparameters.

Figure 7: Output for AdamW run with full fine tuning with optimal hyperparameters.

Figure 8: Output for Muon run with LoRA with optimal hyperparameters.

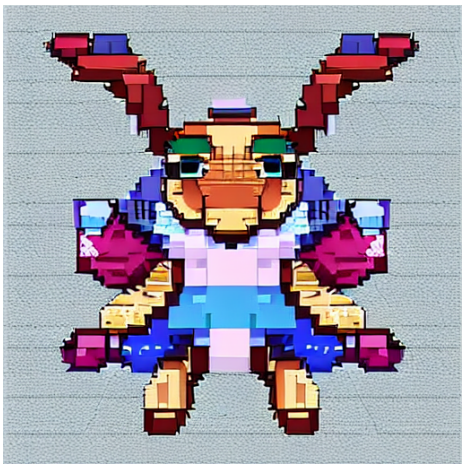

Figure 9: Output for Muon full fine tuning with optimal hyperparameters.

C Appendix - Training Data

Table 1: No Fine Tuning Results

Training Loss	Validation Loss	Standard Deviation
0.3517	0.3611	0.0536

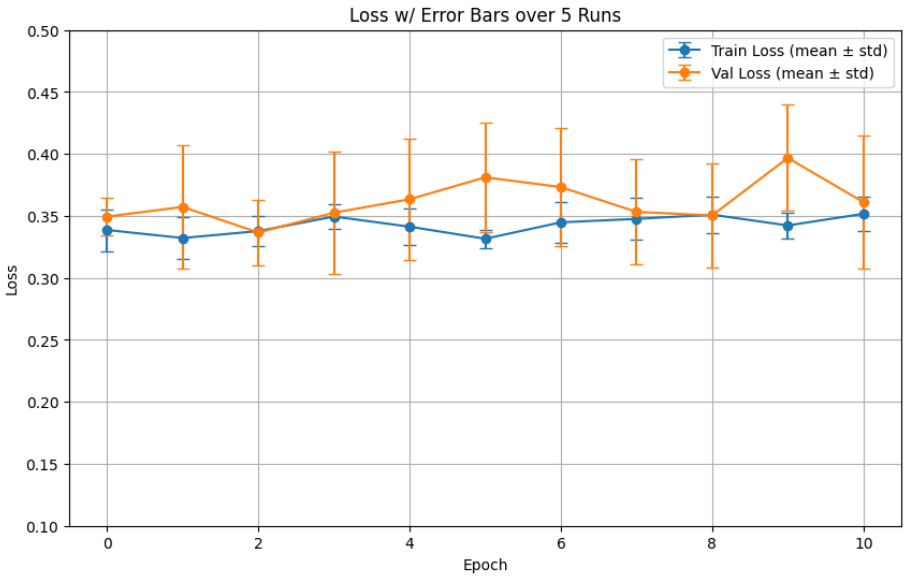

Figure 10: Sample training for no fine tuning - just training the model after being imported given it is pretrained. The error bars were calculated by retraining the model 5 times and using the STDs to calculate the error bars.

Table 2: AdamW with LoRA Results

Learning Rate	Weight Decay	Training Loss	Validation Loss
0.001	0.1	0.4451	0.4773
0.001	0.01	1.0305	0.3620
0.001	0.001	0.9517	0.3346
0.001	0.0001	0.3021	0.3404
0.0001	0.1	0.1632	0.3568
0.0001	0.01	0.1625	0.2655
0.0001	0.001	0.1565	0.2929
0.0001	0.0001	0.1669	0.3289
0.00001	0.1	0.2324	0.3500
0.00001	0.01	0.2699	0.3217
0.00001	0.001	0.2643	0.4104
0.00001	0.0001	0.2454	0.3796
0.000001	0.1	0.3474	0.3986
0.000001	0.01	0.3656	0.3280
0.000001	0.001	0.3458	0.3198
0.000001	0.0001	0.3210	0.3554

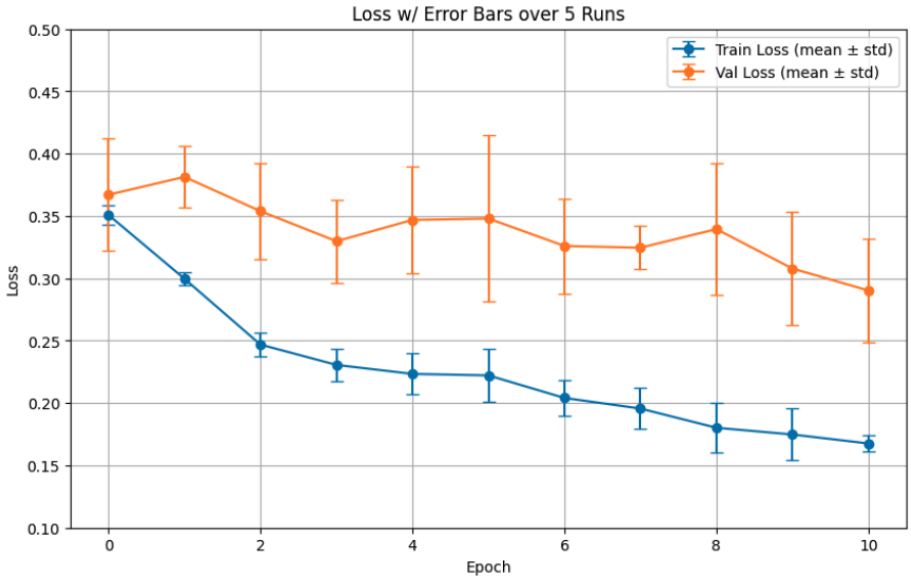

Figure 11: Another run for AdamW with LoRA rank 4 with optimal hyperparameters - 0.0001 learning rate and 0.01 weight decay.

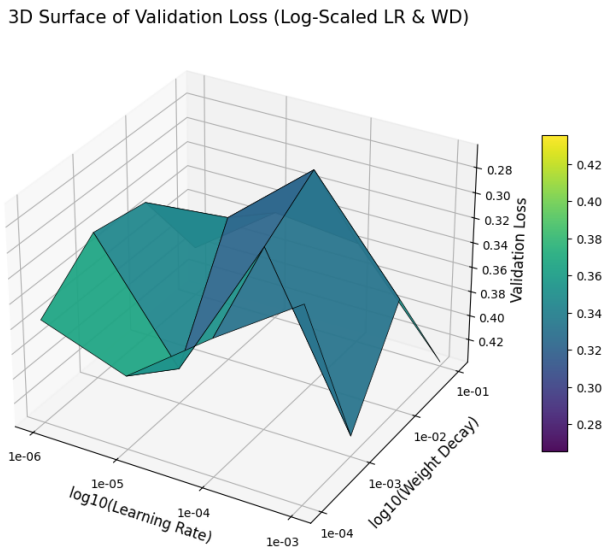

Figure 12: Validation loss surface for AdamW with LoRA rank 4 for hyperparameter sweep.

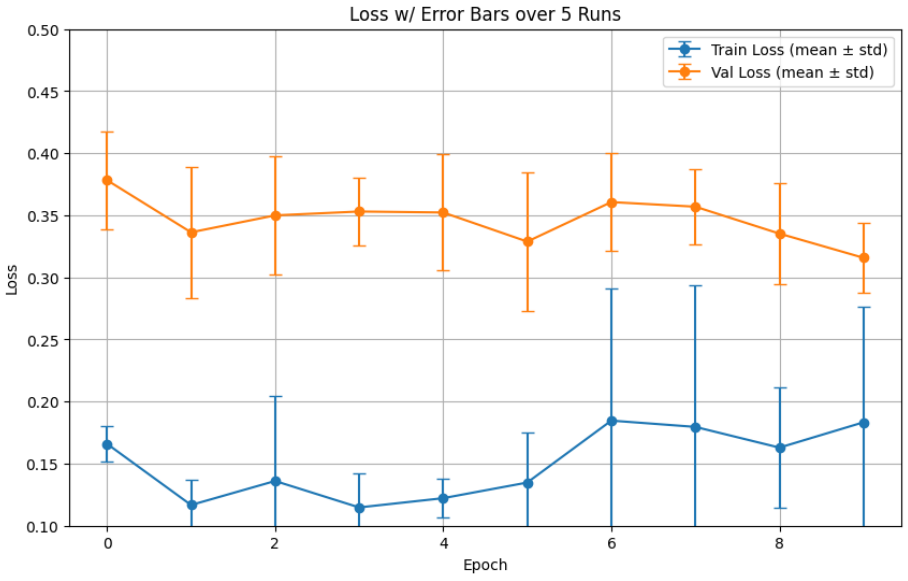

Figure 13: AdamW with Full Fine Tuning over 5 runs with optimal hyperparameters - 0.000001 learning rate and 0.1 weight decay

Table 3: AdamW Full Fine-Tuning Results

Learning Rate	Weight Decay	Training Loss	Validation Loss
0.000001	0.1	0.1383	0.2708
0.000001	0.01	0.4437	0.3277
0.000001	0.001	0.1514	0.4881
0.000001	0.0001	0.1012	0.3172
0.0000001	0.1	0.1612	0.3475
0.0000001	0.01	0.1519	0.3142
0.0000001	0.001	0.2056	0.3543
0.0000001	0.0001	0.1542	0.3204
0.00000001	0.1	0.3167	0.3250
0.00000001	0.01	0.3125	0.3676
0.00000001	0.001	0.3062	0.3657
0.00000001	0.0001	0.3220	0.2915
0.000000001	0.1	0.3244	0.3319
0.000000001	0.01	0.3360	0.3131
0.000000001	0.001	0.3336	0.3584
0.000000001	0.0001	0.3517	0.2903

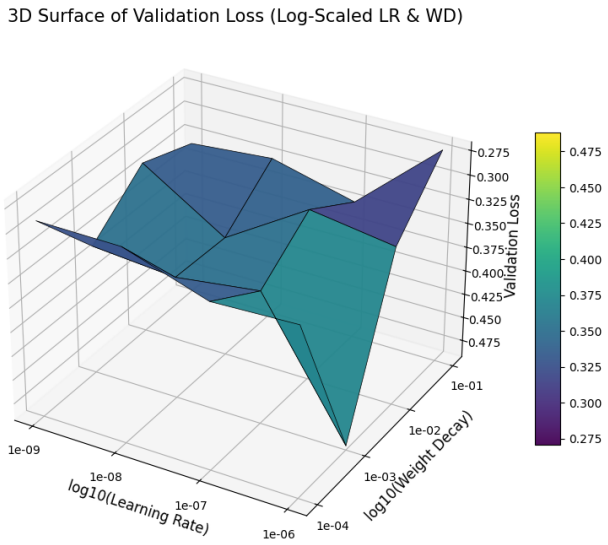

Figure 14: Validation loss surface for AdamW with full fine tuning for hyperparameter sweep.

Table 4: Muon with LoRA Results

Learning Rate	Weight Decay	Training Loss	Validation Loss
0.02	0.1	0.1531	0.4341
0.02	0.01	nan	0.3475
0.02	0.001	nan	0.2617
0.02	0.0001	nan	0.2740
0.01	0.1	0.3143	0.4008
0.01	0.01	0.1713	0.3733
0.01	0.001	0.3143	0.3320
0.01	0.0001	nan	0.3038
0.005	0.1	nan	0.2833
0.005	0.01	0.1600	0.3803
0.005	0.001	0.2027	0.3599
0.005	0.0001	0.1467	0.3268
0.002	0.1	0.2620	0.3077
0.002	0.01	0.1777	0.3223
0.002	0.001	0.2161	0.4045
0.002	0.0001	0.2269	0.3688
0.001	0.1	0.1955	0.3304
0.001	0.01	0.2204	0.2976
0.001	0.001	0.2420	0.2918
0.001	0.0001	0.2207	0.3213
0.0005	0.1	0.2602	0.3415
0.0005	0.01	0.2793	0.3331
0.0005	0.001	0.2807	0.3428
0.0005	0.0001	0.2536	0.3611
0.0001	0.1	0.3110	0.3223
0.0001	0.01	0.3099	0.4108
0.0001	0.001	0.2741	0.2331
0.0001	0.0001	0.2771	0.3468
0.00001	0.1	0.3536	0.3223
0.00001	0.01	0.3200	0.3468
0.00001	0.001	0.3343	0.3077
0.00001	0.0001	0.3531	0.3688
0.000005	0.1	0.3497	0.4123
0.000005	0.01	0.3680	0.3742
0.000005	0.001	0.3399	0.3275
0.000005	0.0001	0.3580	0.3361
0.000001	0.1	0.3546	0.3415
0.000001	0.01	0.3396	0.3611
0.000001	0.001	0.3508	0.4123
0.000001	0.0001	0.3605	0.3361

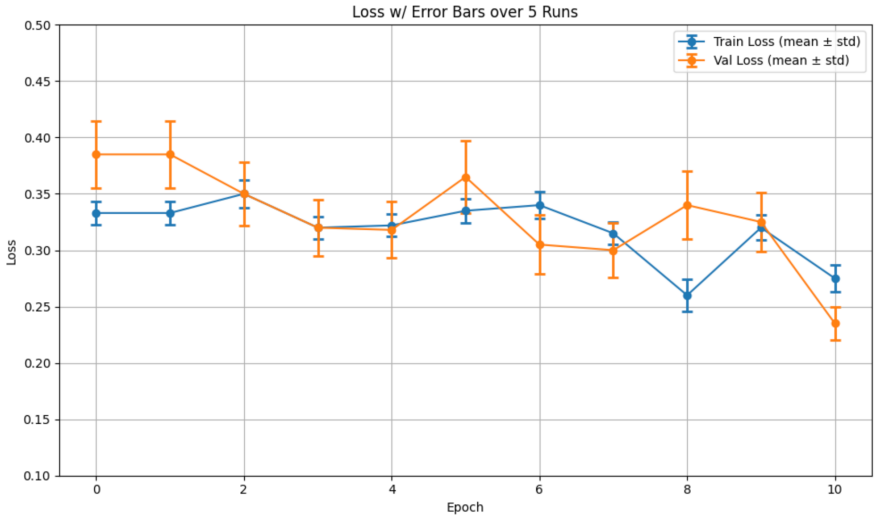

Figure 15: Muon with LoRA rank 4 over 5 runs with optimal hyperparameters - 0.0001 learning rate and 0.0001 weight decay

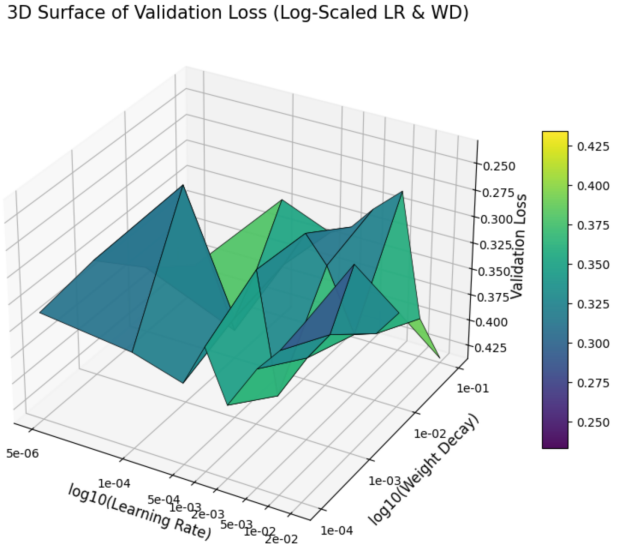

Figure 16: Validation loss surface for Muon with LoRA

Table 5: Muon with Full Fine-Tuning Results

Learning Rate	Weight Decay	Training Loss	Validation Loss
1e-6	0.1	0.3298	0.3478
1e-6	0.01	0.3356	0.2956
1e-6	0.001	0.3312	0.3123
1e-6	0.0001	0.3387	0.3267
1e-7	0.1	0.3389	0.3612
1e-7	0.01	0.3478	0.3156
1e-7	0.001	0.3412	0.3289
1e-7	0.0001	0.3501	0.3445
1e-8	0.1	0.3425	0.3831
1e-8	0.01	0.3637	0.3024
1e-8	0.001	0.3630	0.3935
1e-8	0.0001	0.3562	0.3200
1e-9	0.1	0.3454	0.4084
1e-9	0.01	0.3245	0.3267
1e-9	0.001	0.3300	0.3183
1e-9	0.0001	0.3612	0.3639

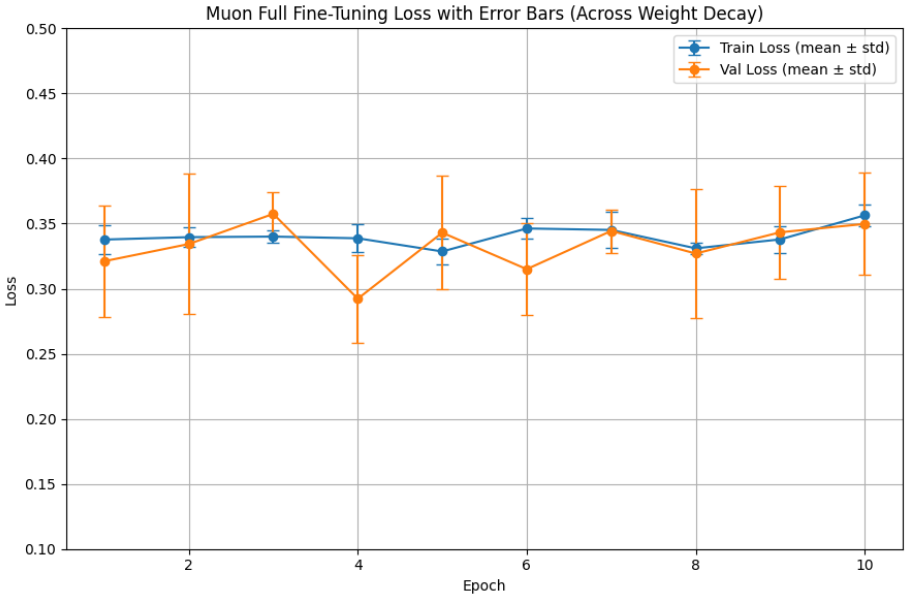

Figure 17: Sample training run for Muon with Full Fine Tuning rank 4 for the validation loss minimal hyperparameters - learning rate of 0.00000001 and weight decay of 0.001

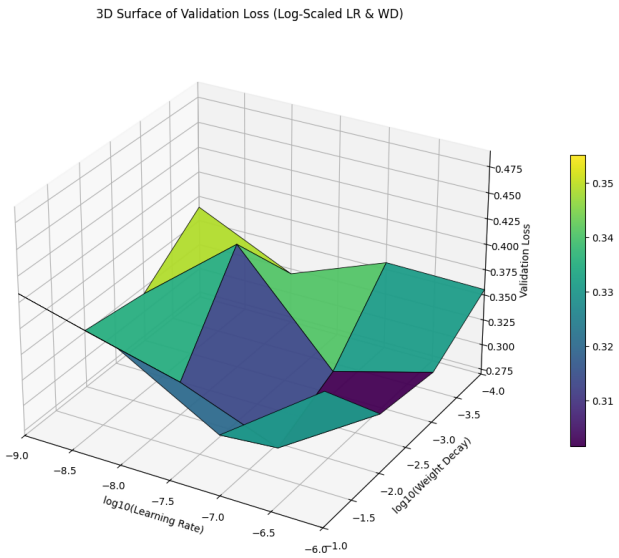

Figure 18: Validation loss surface for muon full fine tuning

D Appendix - Response to Reviewers

We thank the reviewers for their valuable feedback and have revised the final report to improve clarity, rigor, and experimental completeness. First, we added an explicit *no fine-tuning* baseline, in which the pretrained diffusion model is evaluated without any parameter updates. This configuration is now included in the Results, Analysis, and Appendix, providing a clear control that contextualizes the performance gains achieved through fine-tuning and optimizer choice. Second, we introduced an explicit *epoch zero* in all training curves, corresponding to the pretrained model's initial loss, ensuring that improvements are attributable to optimization dynamics rather than initialization effects. Third, to quantify stability and variance, we reran the best hyperparameter configuration of each method five independent times and added error bars (mean ± standard deviation) to the loss-versus-epoch plots. These error bars are further analyzed to distinguish optimizer-induced instability from generalization variance caused by diffusion noise and limited validation data. Fourth, all loss plots were re-rendered with a consistent y-axis range of $[0.10, 0.50]$ and separated by optimizer to ensure fair and interpretable visual comparisons, without rerunning experiments or altering numerical results. Finally, we added a concise analysis of CLIP scores to complement loss-based evaluation, demonstrating that Muon—particularly under full fine-tuning—achieves higher and more stable semantic alignment than AdamW despite comparable reconstruction loss. Together, these revisions directly address the reviewers' comments and strengthen the empirical support for our conclusions regarding geometry-aware optimization in diffusion model fine-tuning.

E Appendix: Github Link

https://github.com/Ying828/CS182-Project