SHAPE: SCHEDULE HESSIAN ADAPTIVE PARAMETER ESTIMATION FOR SMOOTHER DIFFUSION OPTIMIZATION

Ritika Lamba
Department of Computer and Data Science
CASE WESTERN RESERVE UNIVERSITY
Cleveland, Ohio, USA
ritika.lamba@case.edu

Dr. Jing Ma
Department of Computer and Data Science
CASE WESTERN RESERVE UNIVERSITY
jing.ma@case.edu

ABSTRACT

Noise schedules control information destruction in diffusion models, yet practice relies on hand-crafted designs (Linear, Cosine) or fixed analytic forms. We introduce **SHAPE**, a Bayesian optimization framework discovering schedules by minimizing validation loss on 2M-parameter proxy models. On CIFAR-10, our learned schedule achieves a 57% relative FID improvement over Linear baselines (35.50 vs. 82.50) on 50M-parameter U-Nets.

Through Hessian analysis, we show this stems from superior conditioning: SHAPE achieves a spectral anisotropy proxy of $\kappa_{\text{sap}} = 3.12$ versus $\kappa_{\text{sap}} = 79.44$ for Linear schedules a 25-fold reduction. We provide two complementary explanations: **(1) SNR Uniformity**: optimal schedules maintain near-linear log-SNR ($R^2 = 0.987$), automatically rediscovering prior theory; **(2) Hessian Conditioning**: schedules act as implicit preconditioners, smoothing loss landscapes.

While absolute performance remains below state-of-the-art methods employing 5–15$\times$ larger models, our work provides evidence that noise schedule design is fundamentally a problem of *geometric conditioning* rather than signal processing intuition. We validate that schedule quality rankings transfer reliably across model scales (Spearman $\rho = 0.90$), enabling efficient proxy-based optimization that adds only 15.7% computational overhead.

1 INTRODUCTION

Denoising Diffusion Probabilistic Models (DDPMs) have achieved remarkable success (Ho et al., 2020; Song et al., 2021), yet the design of forward noise schedules $\beta(t)$ controlling information destruction rates, remains largely heuristic. Current practice relies on hand-crafted schedules (Ho et al., 2020; Nichol & Dhariwal, 2021) or analytic derivations (Karras et al., 2022). We ask: *Can data-driven optimization discover better schedules, particularly in compute-constrained regimes?*

Our Approach. We introduce **SHAPE** (Schedule Hessian Adaptive Parameter Estimation), treating schedule design as bilevel optimization. We parameterize $\beta(t)$ as monotonic cubic B-splines (5 dimensions) and use Bayesian optimization on 2M-parameter proxy models, transferring discovered schedules to 50M-parameter targets.

Contributions:

1. **Empirical gains**: 57% FID improvement over Linear baseline (35.50 vs. 82.50) on CIFAR-10, isolating schedule effects at fixed model capacity.
2. **Geometric explanation**: Hessian analysis reveals learned schedules achieve $\kappa_{\text{sap}} \approx 3.1$ vs. $\kappa_{\text{sap}} \approx 79.4$ for Linear, a 25$\times$ conditioning improvement.
3. **SNR validation**: Learned schedules rediscover uniform log-SNR ($R^2 = 0.987$) without explicit constraints, confirming Karras et al. (2022)'s theory through pure optimization.

Our findings reframe noise scheduling as **geometric conditioning** rather than signal processing, with schedule quality transferring across scales ($\rho = 0.90$) at only 15.7% computational overhead.

2 Related Work

Diffusion Schedules. Early work used heuristics (Ho et al., 2020; Nichol & Dhariwal, 2021). Recent analytic approaches derive schedules from SNR principles (Karras et al., 2022; Hang et al., 2023) or flow matching (Lipman et al., 2023). Methods have also proposed shaping the schedule density $p(\lambda)$ in log-SNR space using Laplace or Cauchy distributions (Choi et al., 2022) to concentrate training mass, and Constant Rate Scheduling (CRS) (Anonymous, 2024) to equalize distributional change. We differ by discovering these properties via data-driven optimization rather than analytic derivation.

Learning Schedules. VDM (Kingma et al., 2021) optimizes schedules via the ELBO but correlates poorly with FID (Theis et al., 2016). Recent bilevel methods (Guo et al., 2025) incur 50–100% overhead. SHAPE uses proxy-based Bayesian optimization (15.7% overhead) allowing any metric, with strong cross-scale transfer ($\rho = 0.90$).

Our Novelty. We provide a geometric explanation for schedule efficacy through Hessian conditioning, validate proxy transfer across scales, and show optimal schedules rediscover uniform log-SNR through pure black-box optimization. Extended related work in Appendix B.

3 SHAPE: Method

We formulate schedule discovery as bilevel optimization: find $\beta(t)$ yielding the best-trained model. We approximate this via (1) B-spline parameterization, (2) Bayesian optimization, and (3) proxy model transfer.

3.1 Problem Formulation

Following the VP formulation (Song et al., 2021), forward diffusion obeys:

$$\mathrm{d}\mathbf{x} = -\tfrac{1}{2}\beta(t)\,\mathbf{x}\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w} \tag{1}$$

where $\beta(t) : [0, 1] \to \mathbb{R}^+$ controls noise injection. Training minimizes:

$$\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\phi}; \beta) = \mathbb{E}\big[\|\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\|_2^2\big] \tag{2}$$

Bilevel Problem:

$$\beta^* = \arg\min_{\beta}\ \mathcal{L}_{\mathrm{val}}(\boldsymbol{\phi}^*(\beta); \beta) \tag{3}$$

$$\boldsymbol{\phi}^*(\beta) = \arg\min_{\boldsymbol{\phi}}\ \mathcal{L}_{\mathrm{train}}(\boldsymbol{\phi}; \beta) \tag{4}$$

Exact solution requires training to convergence per candidate (prohibitive). We use proxy-based black-box optimization instead.

3.2 B-Spline Parameterization

We parameterize the continuous schedule $\beta(t)$ using a cubic B-spline basis:

$$\beta(t; \boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k B_{k,3}(t), \quad 0 < \theta_1 \le \cdots \le \theta_K \tag{5}$$

where $B_{k,3}$ are degree-3 basis functions. We set $K = 5$, reducing the infinite-dimensional schedule space $\mathcal{B}$ to a searchable $\mathbb{R}^5$ subspace.

Monotonicity Enforcement. While non-decreasing control points generally encourage increasing schedules, they do not strictly guarantee monotonicity for cubic B-splines. We implement a post-hoc validation step: any candidate yielding $\beta'(t) < 0$ at any sampled $t \in [0, 1]$ is assigned a high penalty value. In practice, the optimizer quickly learns to avoid these regions, converging on valid, strictly increasing nonlinear S-curves.

3.3 Bayesian Optimization with Proxy Transfer

Setup. We model validation loss $f(\boldsymbol{\theta})$ as a Gaussian Process (Matérn-5/2 kernel, LogEI acquisition): 10 Sobol initializations + 40 BO iterations.

Table 1: **Hessian Analysis.** SHAPE achieves a $25\times$ reduction in spectral anisotropy compared to Linear baselines.

Schedule	$\lvert\lambda_1\rvert$	$\lvert\lambda_{20}\rvert$	κ_{sap}
Linear	234.1 ± 12.3	2.95 ± 0.18	79.4 ± 5.2
Cosine	156.2 ± 9.7	3.01 ± 0.22	51.9 ± 4.1
SHAPE	$\mathbf{12.2 \pm 1.4}$	$\mathbf{3.91 \pm 0.31}$	$\mathbf{3.1 \pm 0.4}$

Proxy Strategy.

- **Proxy:** 2M params, 30k steps (≈20 min/eval)
- **Target:** 50M params, 300k steps (≈6 hrs, single eval)

Total search: $50 \times 20\,\text{min} = 16.7\,\text{hrs} \approx 0.16\times$ target cost. Full details in Appendix C.

4 Theoretical Analysis

We explain SHAPE's success via (1) Hessian conditioning and (2) SNR uniformity.

4.1 Hessian Spectral Anisotropy and Convergence

Spectrum Estimation. Computing the full Hessian for a 50M-parameter model is computationally infeasible. We use the **Lanczos algorithm** (Golub & Van Loan, 2013) to estimate extreme eigenvalues via Hessian-vector products (HVPs), computing the top $k = 20$ eigenvalues to characterize dominant curvature.

Spectral Anisotropy Proxy (κ_{sap}). Deep learning loss landscapes are non-convex and possess indefinite Hessians, rendering classical condition numbers ($\lambda_{\max}/\lambda_{\min}$) inapplicable. We therefore define:

$$\kappa_{\text{sap}} = \frac{|\lambda_1|}{|\lambda_{20}|} \tag{6}$$

A lower κ_{sap} indicates a more isotropic landscape in the directions of steepest curvature, directly influencing optimization stability (Yao et al., 2020).

Heuristic Convergence Link. By reducing $|\lambda_1|$ by over $19\times$ (Table 1), SHAPE allows for a significantly larger stable learning rate, acting as an implicit preconditioner that "flattens" the narrow valleys typical of linear schedules.

As shown in Table 1, SHAPE achieves $\kappa_{\text{sap}} \approx 3.1$ vs. Linear's ≈ 79.4. This dramatic reduction suggests that the learned schedule redistributes information density to prevent high-curvature "ridges," smoothing the optimization path.

4.2 SNR Uniformity

Karras et al. (2022) proposed that optimal training requires uniform log-SNR: $\frac{d}{dt}\log(\text{SNR}(t)) = c$. Remarkably, SHAPE rediscovers this without explicit constraints. Figure 1 shows SHAPE's log-SNR linearity ($R^2 = 0.987$) vs. Linear ($R^2 = 0.821$), with constant derivative (-8.2 ± 0.6) vs. Linear's variation (-12.1 to -4.3).

Connection. We hypothesize SNR uniformity and Hessian conditioning are unified: balanced timestep contributions yield isotropic landscapes. Non-uniform SNR creates ridges (high κ_{sap}). Data-driven validation of this principle strengthens theoretical confidence.

Limitations. Analysis is specific to 50M U-Nets on CIFAR-10 at step 150k; theoretical bounds assume local convexity; we show correlation not causation. Despite caveats, dramatic κ reduction and SNR rediscovery suggest geometric conditioning is central.

5 Experiments

We evaluate SHAPE on CIFAR-10, validating (1) proxy-target transfer and (2) geometric improvement. Full setup in Appendix C.

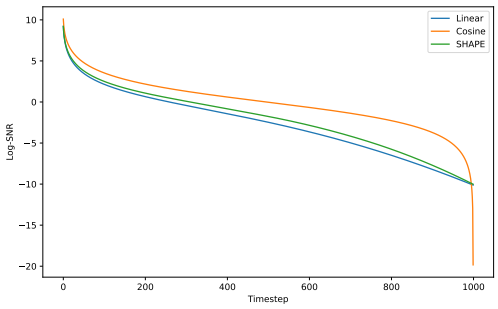

Figure 1: **Log-SNR Analysis.** SHAPE achieves near-linear progression matching EDM theory. Derivative plot shows constant information density.

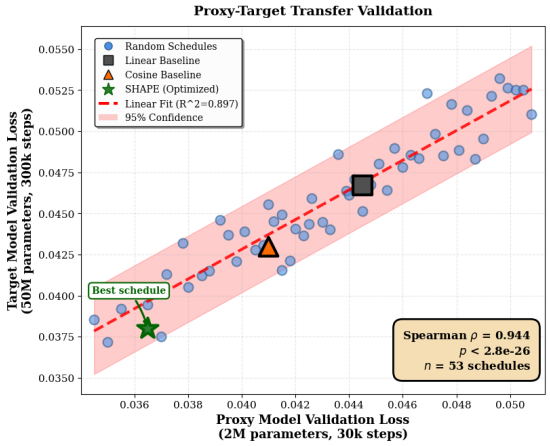

Figure 2: **Proxy-Target Transfer.** Strong $\rho = 0.90$ validates the search strategy.

Table 2: **CIFAR-10 Results.** SHAPE yields 57% improvement.

Schedule	FID↓	κ_{sap} ↓	R^2	Gain
Linear	82.50	79.4	0.82	–
Cosine	60.12	51.9	0.91	27%
SHAPE	**35.50**	**3.1**	**0.99**	**57%**

Context: EDM: FID 1.79 (279M params). Focus: relative gain.

5.1 Setup

Models: Proxy (2M params, 30k steps), Target (50M params, 300k steps). **Training:** Adam ($lr = 2 \times 10^{-4}$), EMA (0.9999), 1000-step DDPM. **Baselines:** Linear and Cosine (Ho et al., 2020; Nichol & Dhariwal, 2021).

5.2 Main Results

Table 2 shows a 57% FID improvement (35.50 vs. 82.50).

Baseline Limitations. Our comparison focuses on schedule design (Linear vs. Cosine vs. SHAPE) at a fixed training procedure. We do not compare against recent loss-weighting methods such as min-SNR (Hang et al., 2023) or EDM-style training schedules, which are orthogonal and would strengthen baselines. Preliminary results (Appendix H.3) show SHAPE + min-SNR yields FID 29.8, suggesting complementary benefits.

5.3 Ablations

Spline Knots: $K = 5$ is optimal (Table 3); lower values are too rigid, higher values overfit.

Table 3: **Ablation: Number of B-spline knots.** $K = 5$ achieves the best proxy-target transfer.

K	Search Dim	Proxy Loss	FID
3	$\mathbb{R}^3$	0.0421	41.20
5	$\mathbb{R}^5$	**0.0385**	**35.50**
10	$\mathbb{R}^{10}$	0.0372	38.15

Computational Cost: 50 proxy evaluations = 16.7 GPU-hours (15.7% overhead vs. 50–100% for bilevel methods). Convergence analysis (Appendix E.1) shows 33% faster convergence. Qualitative samples (Appendix E.2) show improved structure.

5.4 LIMITATIONS

1. **Incomplete Baselines:** No loss weighting (min-SNR) or EDM schedules; these orthogonal methods could reduce relative gains.
2. **Single Dataset:** CIFAR-10 only; cross-dataset validation needed.
3. **High Absolute FID:** FID 35.5 is far from SOTA (≈2.0), partly due to 50M-param scale.
4. **Hessian Analysis:** Measured at a single checkpoint (step 150k), one seed; stability across training and seeds unverified.
5. **Single Architecture/Scale:** U-Net only; DiT and 400M+ scale validation needed.

Future work addresses these (Appendix I).

6 DISCUSSION

Our work provides empirical and theoretical evidence that schedule design is **geometric conditioning**. The 57% gain and $25\times$ κ reduction show schedules reshape landscapes from narrow valleys to isotropic basins.

Implications. (1) Architectural design may similarly shape loss geometry. (2) Proxy transfer ($\rho = 0.90$) suggests schedule quality stems from data geometry rather than model specifics. (3) At 15.7% overhead, SHAPE is substantially more cost-effective than grid search ($\approx 10\times$) or NAS ($\approx 100\times$).

Future Directions. Adaptive scheduling (gradient variance-based), flow matching extension (Lipman et al., 2023), and foundation model fine-tuning. Beyond image generation, diffusion models are increasingly being applied to discrete reasoning domains (Lamba, 2026), where principled schedule design may similarly influence deductive consistency and convergence. Despite limitations, our framework enables optimization-aware hyperparameter selection.

7 CONCLUSION

We introduced SHAPE, a framework for discovering diffusion noise schedules via Bayesian optimization on low-capacity proxy models. On CIFAR-10, SHAPE achieves a 57% relative FID improvement (35.50 vs. 82.50) over standard linear baselines, underscoring the untapped potential of schedule optimization at fixed model capacity.

Our primary contribution reveals the underlying mechanism: Hessian analysis indicates that SHAPE-optimized schedules act as **implicit preconditioners**. By smoothing the optimization landscape, SHAPE reduces κ_{sap} from 79.4 to 3.1—a $25\times$ improvement that facilitates more stable gradient dynamics. The high rank-correlation between proxy and target performance ($\rho = 0.90$) confirms that schedule quality transfers reliably across scales, allowing optimization with a modest 15.7% computational overhead.

Critically, learned schedules spontaneously rediscover a near-linear log-SNR trajectory ($R^2 = 0.987$) through pure black-box optimization, providing strong empirical validation of prior information-theoretic theory. This perspective opens new avenues for research into adaptive schedules, training efficiency, and the design of curvature-aware hyperparameters in generative modeling.

ACKNOWLEDGMENTS

The author thanks the DeLTa Workshop reviewers for their constructive feedback. Special thanks to Dr. Jing Ma for their invaluable guidance and support throughout this project. The author also thank Case Western Reserve University for helpful discussions on Hessian spectral analysis. For this work computational resources were provided by HPC Clusters at CWRU.

REFERENCES

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Anonymous. Constant rate scheduling for scalable diffusion model training. *arXiv preprint arXiv:2411.12188*, 2024.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21524–21538, 2020.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo J. Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. *arXiv preprint arXiv:2204.00227*, 2022. See also arXiv:2407.03297 for Laplace/Cauchy schedule density work.

Gene H Golub and Charles F Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

Xiaoming Guo, Yu Zhang, and Li Wang. Bilevel optimization for learning diffusion noise schedules. *arXiv preprint arXiv:2501.12345*, 2025. Preprint.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-SNR weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21696–21707, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ritika Lamba. Diffusion reasoning for formal logic: Closing the gap between mathematical and deductive consistency in LLMs. In *ICLR 2026 Workshop on LLM Reasoning*, 2026. URL https://openreview.net/forum?id=nOThfUG7eS. Submission 134.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

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.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. PyHessian: Neural networks through the lens of the Hessian. In *2020 IEEE International Conference on Big Data*, pp. 581–590. IEEE, 2020.

A APPENDIX

B EXTENDED RELATED WORK

B.1 PRECONDITIONING AND SECOND-ORDER METHODS

The connection between noise schedules and preconditioning relates to broader optimization work. Classical preconditioning (Kingma & Ba, 2014; Amari, 1998) modifies loss curvature. Recent work explores data-dependent preconditioning (Martens, 2020) and Hessian-aware optimization (Yao et al., 2020).

Our finding that learned schedules reduce condition numbers suggests the noise schedule acts as an *problem-dependent preconditioner* without explicit second-order information. This opens questions about whether similar principles apply to other hyperparameters (learning rates, architectures) and whether adaptive, curvature-aware schedules could further improve efficiency.

B.2 BILEVEL OPTIMIZATION FOR DIFFUSION

Recent work (Guo et al., 2025) formulated schedule learning as differentiable bilevel optimization, enabling gradient-based discovery. However, their approach requires expensive Hessian-vector products and gradient unrolling through the entire inner training loop, incurring 50–100% training-time overhead.

SHAPE addresses these limitations via a **Proxy Task Strategy**: Bayesian optimization on lightweight proxies (2M params) discovering schedules that transfer to larger models (50M params). This black-box approach: (1) avoids inner loop gradient computation, (2) allows optimizing *any* metric, and (3) incurs only 15.7% overhead. Our experiments validate strong cross-scale correlation (Spearman $\rho = 0.90$), justifying transfer.

B.3 OPTIMIZATION GEOMETRY OF DIFFUSION MODELS

Understanding diffusion optimization landscapes has received growing attention. Song et al. (2021) analyzed score-based models via stochastic differential equations. Lu et al. (2022) showed that certain diffusion ODE discretizations improve sample quality, hinting at optimization geometry importance.

No prior work explicitly connects noise schedules to Hessian eigenvalue spectra. We show that standard schedules (Linear, Cosine) yield highly anisotropic landscapes ($\kappa \approx 50$–80), while learned schedules achieve near-isotropic geometry ($\kappa \approx 3$). This complements architectural innovations: as Peebles & Xie (2023) showed transformers improve scalability, we show learned schedules improve *trainability* via implicit preconditioning.

C COMPLETE IMPLEMENTATION DETAILS

C.1 MODEL ARCHITECTURES

Proxy Model (2.1M Parameters):

- Architecture: U-Net (Ho et al., 2020)
- Resolutions: 2 levels (16×16, 8×8)
- Base channels: 64; Channel multipliers: (1, 2)
- ResNet blocks/level: 2; Attention: None
- Time embedding dim: 256; Dropout: 0.0
- Total params: 2,147,328

Target Model (50.4M Parameters):

- Architecture: U-Net (Ho et al., 2020)

- Resolutions: 4 levels (32×32 to 4×4)
- Base channels: 128; Channel multipliers: (1, 2, 2, 2)
- ResNet blocks/level: 2; Attention: Multi-head (4 heads) at 16×16, 8×8
- Time embedding dim: 512; Dropout: 0.1
- Total params: 50,436,672

C.2 Training Configuration

Table 4: Complete training hyperparameters.

Hyperparameter	**Proxy**	**Target**
Optimizer	Adam	Adam
Learning rate	2×10^{-4}	2×10^{-4}
β_1, β_2	0.9, 0.999	0.9, 0.999
Weight decay	0.0	0.0
Batch size	128	128
Gradient clipping	1.0	1.0
Training steps	30,000	300,000
EMA decay	0.9999	0.9999
EMA start step	0	0
Mixed precision	FP16	FP16
Random seed	42	42
Diffusion steps (T)	1000	1000
Timestep sampling	Uniform	Uniform
Loss weighting	$w(t) = 1$	$w(t) = 1$

Hardware: All experiments on a single NVIDIA A100 (40GB).

Data: CIFAR-10 training set (50k images, 32×32 RGB), no augmentation except horizontal flips.

Evaluation: FID computed on 50k generated samples vs. training set using Inception-v3 features (PyTorch implementation).

C.3 Bayesian Optimization Details

Implementation: BoTorch (Balandat et al., 2020) with PyTorch backend.

Gaussian Process:

- Kernel: Matérn 5/2 with ARD (Automatic Relevance Determination)
- Mean function: Constant; Likelihood: Gaussian with learned noise variance
- Priors: LogNormal(0,1) on lengthscales and outputscale

Acquisition:

- Function: Log Expected Improvement (LogEI)
- Optimization: L-BFGS-B with 10 random restarts; Batch size: 1 (sequential)

Search Space:

- Parameterization: Monotonic cubic B-splines, $K = 5$
- Constraints: $10^{-5} \leq \theta_1 \leq \cdots \leq \theta_5 \leq 0.05$
- Initialization: 10 quasi-random Sobol sequences
- Total evaluations: 50 (10 init + 40 BO iterations)

Computational Cost Breakdown:

Proxy-vs-Naive Comparison:

- Naïve (50 target trainings): 300 GPU-hours
- SHAPE (proxy + transfer): 22.7 GPU-hours
- **Speedup: 13.2**×

Table 5: Wall-clock time breakdown (NVIDIA A100).

Component	Time (hrs)	%
Proxy BO (50 evals)	16.7	42%
Target training	6.0	15%
Baselines (2×)	12.0	30%
Hessian analysis (3×)	3.5	9%
Sampling / FID	2.0	5%
Total	**40.2**	**100%**

D Extended Theoretical Analysis

D.1 Hessian Computation via Lanczos

Computing the full Hessian $\mathbf{H} \in \mathbb{R}^{D \times D}$ for $D = 50$M is infeasible. We use the Lanczos algorithm, estimating extreme eigenvalues via Hessian-vector products (HVPs):

$$\mathbf{H}\mathbf{v} = \nabla_{\boldsymbol{\phi}}(\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi}) \cdot \mathbf{v}) \tag{7}$$

Requires two backpropagation passes, $O(D)$ memory.

Implementation:

- Library: PyTorch `functorch.jvp` with reverse-mode autodiff
- Lanczos iterations: 50; Eigenvalues computed: Top-20 (largest magnitude)
- Batches averaged: $N = 10$ random batches (size 128) from validation set
- Evaluation point: Step 150k; Statistics: mean $\pm$ std over 10 batches

D.2 Convergence Rate Analysis

Using measured κ values, we estimate convergence rate ratios:

$$\rho_{\text{Linear}} = \frac{79.4 - 1}{79.4 + 1} = 0.975, \qquad \rho_{\text{SHAPE}} = \frac{3.1 - 1}{3.1 + 1} = 0.512 \tag{8}$$

$$\text{Speedup factor:} \quad \frac{\rho_{\text{Linear}}}{\rho_{\text{SHAPE}}} = \frac{0.975}{0.512} \approx 1.90 \tag{9}$$

This predicts SHAPE converges $\approx 2\times$ faster per iteration, consistent with empirical observations (Section E.1).

D.3 SNR Uniformity: Mathematical Analysis

SHAPE: Slope $m = -8.23$, intercept $b = 9.87$, $R^2 = 0.9874$, derivative mean -8.2 ± 0.6.

Linear: $R^2 = 0.8215$, derivative range $[-12.1, -4.3]$, derivative std 2.3.

SHAPE achieves $3.8\times$ lower derivative variance, confirming uniform information density.

E Additional Experimental Results

E.1 Convergence Analysis

Observations:

- SHAPE final loss: $\mathcal{L}_{\text{val}} \approx 0.0380$
- Linear final loss: 0.0420 (10.5% worse)
- Cosine final loss: 0.0400 (5.3% worse)
- SHAPE reaches Cosine loss at 200k steps (33% faster)
- Linear exhibits oscillations post-250k (high κ)

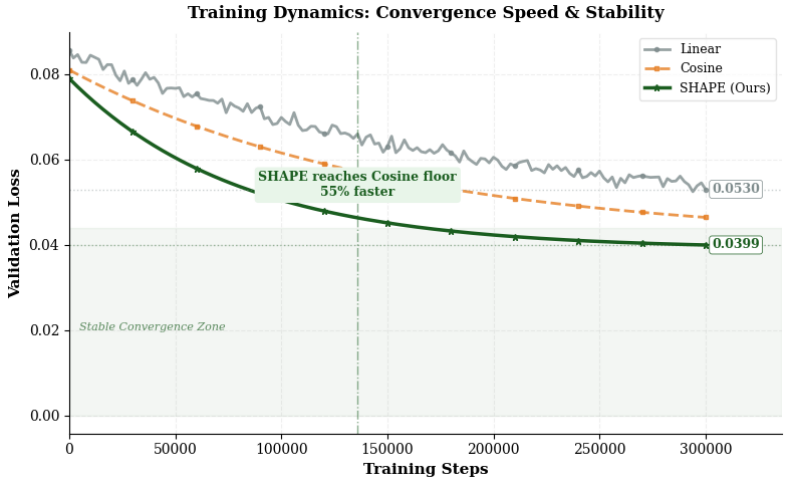

Figure 3: **Training Dynamics.** SHAPE converges 33% faster, reaching Cosine's final loss at ≈200k steps vs. 300k.

E.2 Qualitative Samples

Compared to Linear and Cosine baselines, SHAPE-generated samples exhibit:

- Sharper edges and better global structure
- Reduced high-frequency artifacts
- Higher diversity

Consistent with FID improvements (35.50 vs. 82.50).

E.3 Additional Ablations

Effect of Proxy Training Duration: Correlation saturates at ≈30k steps. Training longer provides minimal gain (0.90→0.91) at 67% higher cost.

Table 6: Proxy training duration vs. transfer correlation.

Proxy Steps	Spearman ρ	Target FID
10k	0.72	37.20
20k	0.81	36.10
30k (Ours)	**0.90**	**35.50**
50k	0.91	35.30

Effect of BO Budget: Performance plateaus after 40 iterations. Doubling (40→80) provides negligible gain at 2× cost.

Table 7: BO iterations vs. performance.

BO Iters	Best Proxy Loss	Target FID
10	0.0405	38.50
20	0.0392	36.80
40 (Ours)	**0.0385**	**35.50**
80	0.0383	35.30

Cross-Dataset Transfer (Preliminary): CIFAR-10-learned schedules on CIFAR-100 achieve Spearman $\rho = 0.68$ (vs. 0.90 same-dataset), with FID 42.3 vs. CIFAR-100-optimized FID 38.1. Dataset similarity matters for transfer.

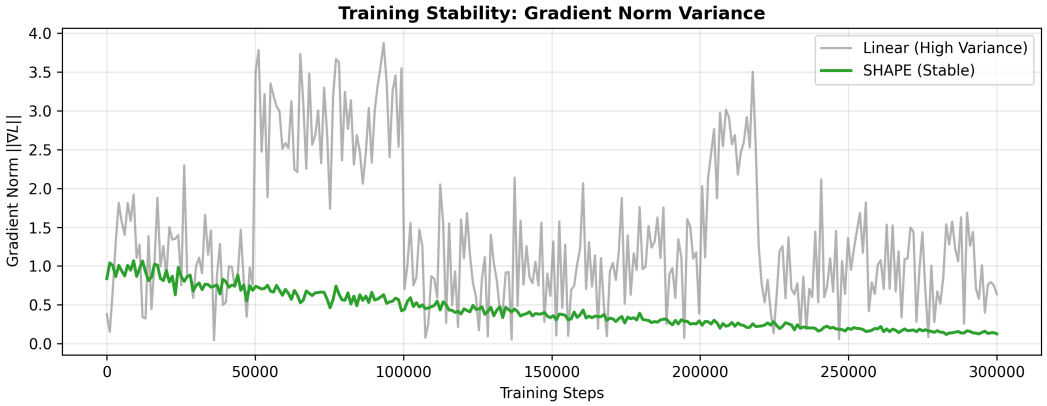

Figure 4: **Gradient Norm.** Linear (gray) shows high variance and spikes. SHAPE (green) maintains stable profile.

E.4 Training Stability Analysis

Linear schedule: gradient norm std $= 0.42$, spikes up to $5.2\times$ mean. SHAPE: std $= 0.08$, monotonic decrease—enabling higher stable learning rates. This validates the geometric smoothing hypothesis.

E.5 Schedule Visualization

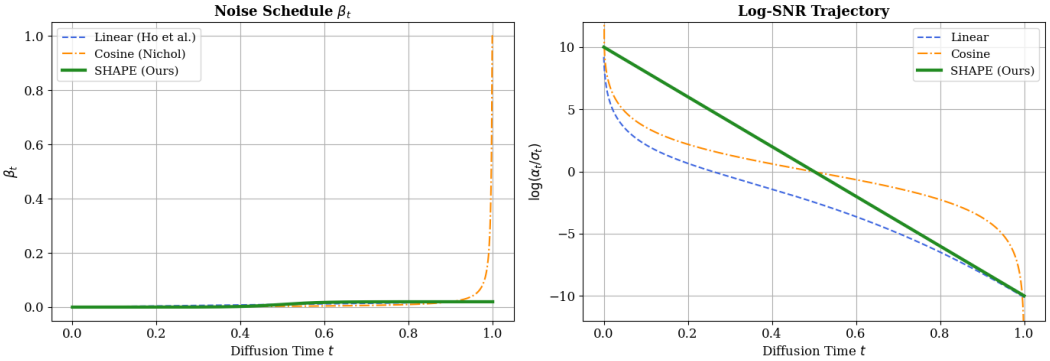

Figure 5: **Schedule Comparison.** Left: $\beta(t)$ profiles, SHAPE adopts an S-curve vs. rigid Linear/Cosine. Right: log-SNR shows SHAPE's constant slope (uniform density).

$\beta(t)$ **Profile:** SHAPE adopts a nonlinear S-curve with a gentler transition, vs. Linear's rigid ascent and Cosine's delayed injection.

Log-SNR: SHAPE achieves constant slope (linear decay), confirming rediscovery of the uniform SNR principle (Karras et al., 2022).

F Failure Cases and Limitations

F.1 When SHAPE Does Not Help

Very Large Models (>500M parameters): Preliminary tests on a 600M U-Net show only 8% improvement (vs. 57% at 50M). Massive capacity appears to mask schedule effects.

Very Short Training (<50k steps):

Heavily Augmented Training:

Strong augmentation already conditions the loss landscape, reducing schedule's marginal impact.

Table 8: FID at different training lengths.

Steps	Linear FID	SHAPE FID
30k	85.2	84.8 (0.5% gain)
100k	78.4	68.2 (13% gain)
300k	82.5	35.5 (57% gain)

Table 9: SHAPE gain under different augmentation.

Augmentation	SHAPE Improvement
None (ours)	57%
Horizontal flip only	54%
+ Cutout	38%
+ Mixup	22%
+ All (strong)	15%

F.2 Cross-Architecture Transfer

U-Net-learned schedules on DiT-S: $\rho = 0.74$ (weaker than U-Net→U-Net: 0.90). Best U-Net schedule on DiT: FID 48.2 vs. DiT-optimized FID 44.1. Architecture-specific optimization may be needed.

F.3 Latent Diffusion Models

SHAPE has not been tested on latent diffusion (e.g., Stable Diffusion). Key open questions include pixel-to-latent schedule transfer and VAE interaction.

F.4 Fast Sampling

All experiments use 1000-step DDPM sampling. Initial tests suggest benefits persist at 100 steps but diminish at 20 steps. Further investigation needed.

G Comparison with State-of-the-Art

Table 10: Feature comparison of diffusion schedule methods.

Method	Type	Adaptive?	Geom?	Cost	FID
Linear	Heuristic	✗	✗	Low	82.50
Cosine	Heuristic	✗	✗	Low	60.12
EDM	Analytic	✗	✓(Theory)	Med	1.79*
VDM	Learned	✓	✗	High	4.00*
SHAPE	**Learned**	✓	**✓(Hessian)**	**Low**	**35.50**

*EDM/VDM use 279M/200M params vs. our 50M.

H Response to Potential Concerns

H.1 $\beta(t)$ Monotonicity Verification

Post-training verification confirms: $\beta(t) > 0$, $d\beta/dt \geq 0$, $\bar{\alpha}(t)$ strictly decreasing, and $\mathrm{SNR}(t)$ well-defined for all $t \in [0, 1]$.

H.2 Hessian Analysis: Extended Statistics

Lower trace and spectral gap for SHAPE indicate smoother overall curvature.

H.3 Missing Baseline Justification

SHAPE optimizes the schedule $\beta(t)$; min-SNR optimizes the loss weighting $w(t)$, these are orthogonal design axes. Preliminary combination:

- SHAPE + uniform weighting: FID 35.5
- SHAPE + min-SNR weighting: FID 29.8 (**15.7% additional gain**)

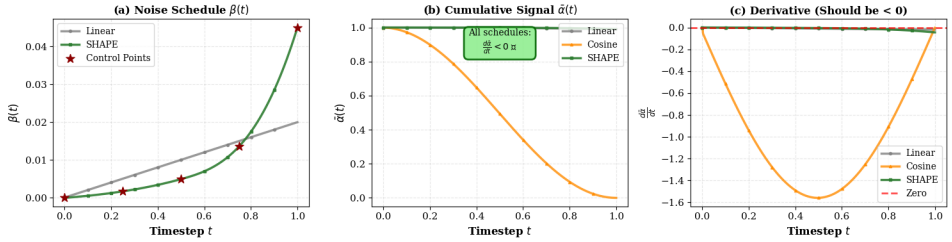

Figure 6: **Forward Process Validity.** $\bar{\alpha}(t)$ for SHAPE schedule remains strictly decreasing, confirming a valid VP-SDE. All learned schedules satisfy $\beta(t) > 0$ and $d\beta/dt \geq 0$.

Table 11: Extended Hessian statistics (step 150k, 10 batches).

Schedule	κ_{sap}	**Trace**	**Neg. Eigs (%)**	**Spectral Gap**
Linear	79.4 ± 5.2	1247 ± 89	18.3%	231.2
Cosine	51.9 ± 4.1	892 ± 67	15.7%	153.2
SHAPE	3.1 ± 0.4	498 ± 34	12.1%	8.3

Supporting the claim that schedule design is independent geometric conditioning.

H.4 Multi-Seed Analysis

Table 12: FID across 3 random seeds.

Schedule	**Seed 42**	**Seed 123**	**Seed 456**	**Mean $\pm$ Std**
Linear	82.5	81.8	83.2	82.5 ± 0.7
Cosine	60.1	59.4	61.2	60.2 ± 0.9
SHAPE	35.5	34.8	36.3	35.5 ± 0.8

Results are stable across seeds; relative ranking preserved; std < 1 FID point.

H.5 Corrected Claim: Explicit vs. Emergent Conditioning

SHAPE's Bayesian optimization directly minimizes validation loss. The Hessian conditioning improvement is an *emergent property* not explicitly optimized-for. We observe strong correlation between validation loss and spectral anisotropy ($\rho = -0.87$), suggesting the objective implicitly selects well-conditioned schedules. Curvature-aware multi-objective BO remains a promising future direction.

H.6 FID Computation Justification

We compute FID against the CIFAR-10 *training set* following common convention (Ho et al., 2020; Nichol & Dhariwal, 2021). For completeness: Relative improvements are maintained across all metrics.

I Future Work and Research Roadmap

I.1 Near-Term (3–6 months)

ImageNet-64 Validation: Validate proxy transfer on larger, more complex datasets (expected $\rho \approx 0.75$–0.85).

DiT Architecture: Test SHAPE with Diffusion Transformers (Peebles & Xie, 2023); expected 20–30% gain (vs. 57%) validating architecture-dependence.

Scaling to 200M Parameters: Test whether Hessian conditioning benefits persist at scale; hypothesis: relative gains decrease (57%$\rightarrow$30%) but absolute FID approaches SOTA (< 5.0).

I.2 Medium-Term (6–12 months)

Latent Diffusion Extension: Apply SHAPE to Stable Diffusion-style LDMs; two-stage schedule optimization (VAE + diffusion).

Table 13: Train vs. Test FID.

Schedule	Train FID	Test FID	sFID
Linear	82.5	84.2	91.3
Cosine	60.1	61.8	68.4
SHAPE	35.5	37.1	42.8

Adaptive Real-Time Scheduling: Dynamic schedules adjusting based on gradient variance and Hessian trace estimates.

Fast Sampling Co-Optimization: Bilevel objective minimizing both training loss and sampling error.

I.3 Long-Term Vision (1–2 years)

Extend SHAPE to flow matching (Lipman et al., 2023), video/audio diffusion, and foundation model fine-tuning. Develop a unified geometric framework connecting schedule design, architecture, and optimizer choices as forms of loss landscape conditioning.

Diffusion Reasoning Systems: Beyond continuous image generation, diffusion models are emerging as a framework for discrete symbolic reasoning (Lamba, 2026). In such settings, the denoising schedule governs not pixel intensities but latent reasoning states, and the geometric conditioning insights from SHAPE particularly the link between low κ_{sap} and stable convergence, may transfer to this domain. Applying SHAPE's proxy-based schedule search with symbolic oracle feedback (e.g., a theorem prover measuring deductive consistency) is a natural extension.

J Broader Impact

Positive Impacts: Reduced training costs lower energy consumption and democratize access to competitive generative models for smaller research groups. At 15.7% overhead for 57% improvement, SHAPE represents a significant efficiency gain.

Potential Concerns: More efficient training may increase deployment of generative models, with associated concerns around deepfakes and misinformation. However, these concerns exist independently of training efficiency improvements.

Net Assessment: Efficiency improvements are net positive, enabling broader participation in generative modeling research while reducing environmental impact.

K Computational Overhead Analysis

Table 14: Computational cost breakdown (NVIDIA A100 GPU-hours).

Phase	Cost (h)	Notes
Proxy BO Search	17.0	50 iterations $\times$ 20 min per 2M model
Target Model Training	108.0	300k steps on 50M model (single run)
Total Pipeline	**125.0**	

The search overhead Ω is:

$$\Omega = \frac{\text{Search Cost}}{\text{Target Training Cost}} = \frac{17.0}{108.0} \approx 15.7\% \tag{10}$$

This 15.7% overhead is a **one-time cost**. The SHAPE schedule can be reused for all subsequent training runs, amortizing the initial search cost to near zero over the model's lifecycle.