SWIFT: SELF-GUIDED WEAK-TO-STRONG DIFFUSION MODELS VIA TARGET EXTRAPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are typically trained to predict clean samples, while autoguidance improves generation at inference by extrapolating a trained denoiser away from a weaker one. We introduce SWIFT, which moves this weak-to-strong extrapolation into training: each clean-sample regression target is extrapolated away from the prediction of a frozen weak denoiser, which is discarded after training. Although this deliberately shifts the target away from the posterior mean, a spectral analysis of finite-time population gradient flow shows that a nonempty range of extrapolation strengths can yield a denoiser strictly closer to the original posterior mean than matched standard training. Experiments on CIFAR-10 and ImageNet-256 show consistent improvements across model scales and latent representations. On ImageNet-256, SWIFT improves unguided FID across every evaluated configuration, and the advantage persists with longer training. SWIFT also remains complementary to inference-time autoguidance and classifier free guidance: their combination gives the best FID across all tested configurations. On the DINOv3-L RAEv2 configuration, SWIFT reaches 1.15 gFID without inference-time guidance and **0.99** with retuned guidance.

1 INTRODUCTION

Diffusion models have achieved strong generative performance across images, video, and audio (Ho et al., 2020; 2022; Kong et al., 2021). They are typically trained with a simple denoising regression objective, yet many of the strongest generation results rely on guidance applied only at inference (Ho & Salimans, 2022; Karras et al., 2024). In particular, autoguidance (Karras et al., 2024) shows that a deliberately weaker denoiser can improve a stronger one by defining an extrapolation direction during sampling. This suggests a broader role for weak models: rather than serving as teachers to imitate, they can provide directions along which a stronger predictor should move.

Autoguidance realizes this idea by extrapolating the strong model's prediction away from that of a smaller or less-trained counterpart (Karras et al., 2024). Related training-time approaches instead use evolving model predictions, intermediate supervision, or guidance-conditioned objectives (Tang et al., 2025; Zhou et al., 2026; Chen et al., 2025; Yuan et al., 2026). These developments motivate a simple question:

> *Can weak-to-strong extrapolation be moved from inference into training, so that the final denoiser itself benefits from the correction and the weak model can be discarded afterward?*

We introduce SWIFT, a self-guided training method that uses a frozen weak denoiser to construct extrapolated regression targets. For each noisy input, instead of regressing directly toward the clean sample, SWIFT moves the target farther in the direction from the weak prediction toward and beyond the clean sample. The weak denoiser therefore acts as a *directional reference*, rather than a teacher whose output should be imitated. It can be obtained from an early checkpoint of ordinary denoising training, remains fixed during the main training run, and requires no backpropagation. After training, the guide is discarded, leaving a single denoiser and the original sampling procedure. The same construction applies to both conditional and unconditional generation.

Our motivation is finite-time underfitting: because a weak denoiser has learned less of the target function, extrapolating away from its prediction can amplify directions that remain underfit. We show

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

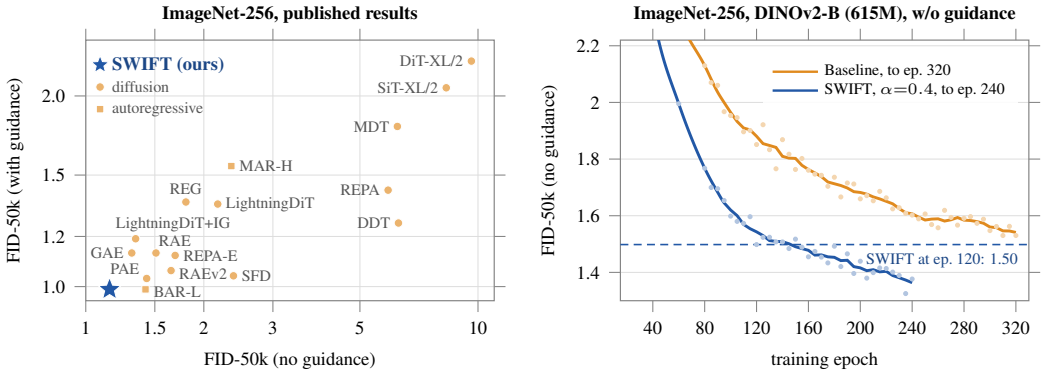

Figure 1: SWIFT **reaches sub-**1.0 **gFID on ImageNet-256.** *Left:* FID-50K without and with inference-time guidance for representative published ImageNet-256 models and SWIFT (Table 9). Our RAEv2/DINOv3-L model reaches 1.15 gFID without inference-time guidance and 0.99 with retuned internal guidance. *Right:* on the matched RAE/DINOv2-B configuration (Wang et al., 2025; Zheng et al., 2026; Oquab et al., 2024), the advantage of SWIFT persists throughout extended training. Dots denote raw evaluations and lines show five-point moving averages.

theoretically that, for a nonempty range of positive extrapolation strengths, deliberately shifting the training target in this way can produce a predictor strictly closer to the original posterior mean than matched standard training (Section 4).

Experiments on CIFAR-10 and ImageNet-256 show that the effect is not tied to a particular representation or model family. On ImageNet-256, SWIFT improves diffusion models built on conventional VAE latents, representation-autoencoder latents (Zheng et al., 2026; Singh et al., 2026), end-to-end learned latent representations, and direct pixel-space generation. Across the evaluated DiffusionBench configurations (Leng et al., 2026), the same fixed SWIFT recipe lowers unguided FID in every configuration, and the improvement persists under longer training.

SWIFT also remains complementary to inference-time guidance. When guidance is independently retuned for the baseline and SWIFT, the SWIFT model achieves lower FID in every tested configuration. Moreover, on RAEv2 the preferred internal-guidance scale decreases from 1.78 for the baseline to 1.20 after SWIFT training, while the best guided gFID improves from 1.04 to 0.99. This is consistent with SWIFT internalizing part of the weak-to-strong correction during training while leaving additional room for inference-time guidance.

Our **contributions** are:

- **Weak-to-strong learning through target extrapolation.** We introduce SWIFT, which uses a frozen weak denoiser as a directional reference to extrapolate clean-sample regression targets. The guide is fixed during main-model training and discarded at inference (Section 2.2).
- **Broad improvements across diffusion representations.** We demonstrate consistent gains across CIFAR-10 and ImageNet-256, spanning VAE, RAE, end-to-end latent, and pixel-space diffusion models. The gains persist under longer training and remain after inference-time guidance is independently retuned (Sections 5.1 and 5.2).
- **A finite-training explanation.** We show theoretically that target extrapolation can strictly reduce posterior-mean error relative to matched standard training over a nonempty range of positive extrapolation strengths (Section 4).

2 SWIFT: SELF-GUIDED TRAINING OF DIFFUSION MODELS

2.1 PRELIMINARIES

Denoising diffusion models generate samples by reversing Gaussian corruption. Let $t \in [0, 1]$ index the noise level, not the training iteration, and let σ_t increase with t. We use the additive-noise parameterization

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{x}_0 \sim p_{\text{data}}, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I}_d).$$

The noise $\boldsymbol{\varepsilon}_t$ is independent of $\boldsymbol{x}_0$. We write p_t for the distribution of $\boldsymbol{x}_t$ at fixed t. When a condition $\boldsymbol{c}$ is present, it is an additional denoiser input; we suppress it below and discuss conditional and null-condition predictions in Section 3.

Training. Standard clean-sample prediction trains a denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \sigma_t)$ to predict $\boldsymbol{x}_0$ by minimizing

$$\mathcal{L}_{\text{base}}(\boldsymbol{\theta}) = \mathbb{E}\left[\lambda(t)\,\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \sigma_t) - \boldsymbol{x}_0\|_2^2\right], \tag{1}$$

where $t \sim p_{\text{noise}}$ is sampled independently of $\boldsymbol{x}_0 \sim p_{\text{data}}$, and p_{noise} is the distribution of noise levels used during training. The expectation also includes $\boldsymbol{\varepsilon}_t$ drawn as above, and $\lambda(t) > 0$ on the support of p_{noise}. With finite second moments, the unrestricted population minimizer of Eq. (1) is the posterior mean $\mathcal{D}^\star(\boldsymbol{x}_t, \sigma_t) = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t, t]$. For $\sigma_t > 0$, Tweedie's formula relates it to the score:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) = \frac{\mathcal{D}^\star(\boldsymbol{x}_t, \sigma_t) - \boldsymbol{x}_t}{\sigma_t^2}. \tag{2}$$

Inference. With a differentiable variance-exploding noise schedule, the probability-flow ODE is (Song et al., 2021)

$$d\boldsymbol{x}_t = -\dot{\sigma}_t \sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\, dt. \tag{3}$$

Sampling starts at a large noise level, where the noisy marginal can be approximated by a Gaussian, and integrates Eq. (3) toward smaller t. In practice, the score in Eq. (2) is formed using the trained denoiser $\mathcal{D}_{\boldsymbol{\theta}}$ in place of $\mathcal{D}^\star$. Thus the sampler depends on the denoiser reached after finite training.

2.2 Proposed Method: Learning with a Frozen Weak Guide

Building on the preliminaries, we introduce SWIFT, which trains a denoiser using extrapolated regression targets. Instead of regressing directly onto the clean sample $\boldsymbol{x}_0$ as in Eq. (1), SWIFT minimizes

$$\mathcal{L}_{\text{SWIFT}}(\boldsymbol{\theta}) = \mathbb{E}\left[\lambda(t)\,\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \sigma_t) - \boldsymbol{y}_\alpha\|_2^2\right], \tag{4}$$

with the same expectation and weighting as in Eq. (1), but using the extrapolated target:

$$\boldsymbol{y}_\alpha = \boldsymbol{x}_0 + \alpha\big(\boldsymbol{x}_0 - \mathcal{D}_{\text{weak}}(\boldsymbol{x}_t, \sigma_t)\big) \tag{5}$$

Here, $\mathcal{D}_{\text{weak}}$ is a pretrained, frozen weak denoiser, and $\alpha \geq 0$ controls the extrapolation strength. Setting $\alpha = 0$ recovers standard denoising training. For $\alpha > 0$, the target extends beyond the clean sample in the direction away from the weak denoiser's prediction. Velocity-predicting models apply the same extrapolation to the velocity, which is equivalent (Section D.1).

Constructing the frozen weak denoiser. We train $\mathcal{D}_{\text{weak}}$ in the ordinary way with Eq. (1), using fewer parameters and fewer training epochs than the main denoiser, and freeze it before the main run. In our default ImageNet configuration, it has approximately one quarter as many parameters and is trained for half as many epochs; the guide of every setting is specified in Section D.1. The method is *self-guided* because the correction comes from a weaker denoiser trained on the same task, without an external teacher, a separate guidance objective, or an auxiliary prediction head.

The intuition is to compensate for what ordinary training has not learned within a finite training budget. Extrapolating the clean target away from $\mathcal{D}_{\text{weak}}$ emphasizes directions in which the weak denoiser's prediction remains inaccurate. Whether this improves the main denoiser's approximation to the posterior mean depends on how those directions relate to its own finite-training error. For a frozen weak denoiser obtained by early stopping ordinary training, Section 4 and Theorem 1 establish conditions under which SWIFT achieves strictly lower posterior-mean error than baseline training (proofs in Section A; a Gaussian-mixture check is in Section C).

Training and inference. During training, only the main denoiser $\mathcal{D}_{\boldsymbol{\theta}}$ is optimized with Eq. (4). The weak denoiser $\mathcal{D}_{\text{weak}}$ is evaluated without gradients and remains fixed. Thus, for a fixed clean sample, noise realization, and noise level, the regression target does not change as the main denoiser is updated. Beyond preparing the weak denoiser, each training step requires one additional weak-denoiser forward pass (about 9% more FLOPs on DiffusionBench; Section D.3). After training,

it is discarded; sampling uses the trained denoiser and the original sampler, with no additional inference-time model evaluation.

3 From Autoguidance to Self-Guided Training

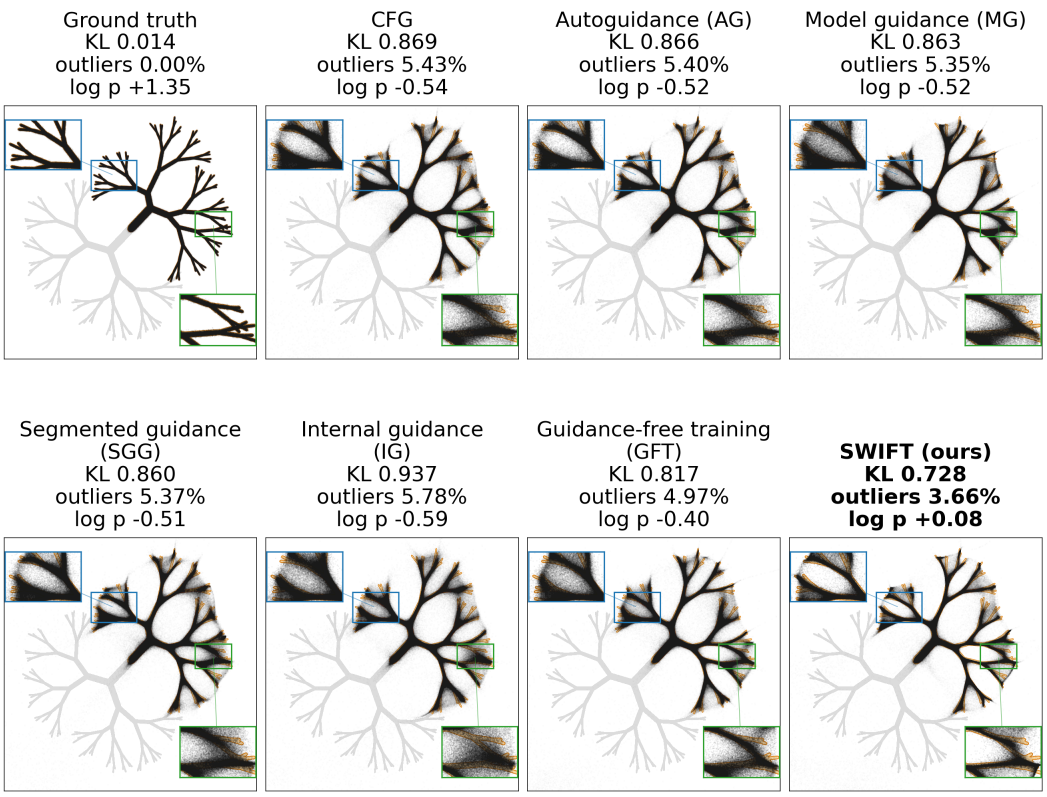

Figure 2: SWIFT **on a two-class, tree-shaped 2D toy distribution.** Ground truth and samples from CFG, AG, MG, SGG, IG, GFT, and SWIFT. Each compared method uses independently tuned settings. Blue boxes mark corresponding branching regions. Lower KL and outlier rates and higher $\log p$ are better.

A visual comparison. Following the fractal-like Gaussian-mixture example of Karras et al. (2024), we compare how different methods reproduce a distribution with thin, branching support. In Figure 2, SWIFT has the lowest reported KL divergence and outlier rate and the highest reported $\log p$ among the seven methods. Relative to GFT, the strongest baseline on these three reported metrics, its KL decreases from 0.817 to 0.728, its outlier rate from 4.97% to 3.66%, and its $\log p$ increases from -0.40 to $+0.08$. The highlighted regions show how the generated samples follow the narrow branches.

Inference-time guidance includes classifier-free guidance (CFG; Ho & Salimans, 2022) and autoguidance (AG; Karras et al., 2024). Training-based approaches include Model Guidance (MG; Tang et al., 2025), Segmented Guidance (SGG; Yuan et al., 2026), and Guidance-Free Training (GFT; Chen et al., 2025). Internal Guidance (IG; Zhou et al., 2026) trains an intermediate output for guidance at sampling time. Additional comparisons appear in Section B.

3.1 From Sampling-Time Guidance to Training Targets

CFG combines conditional and unconditional predictions during sampling. AG is closer to SWIFT in its use of a weak denoiser. Let $\mathcal{D}_{\text{base}}$ be a denoiser trained with the standard objective in Eq. (1). At

the same noisy input and noise level, AG evaluates

$$\mathcal{D}_{\mathrm{AG}} = (1+\alpha)\mathcal{D}_{\mathrm{base}} - \alpha\mathcal{D}_{\mathrm{weak}}$$

during sampling. This requires the weak prediction whenever AG is applied. In Eq. (5), SWIFT instead places the observed clean sample $\boldsymbol{x}_0$ in the position occupied by $\mathcal{D}_{\mathrm{base}}$ and uses the extrapolation as a *training target*. The weak denoiser is discarded after training.

Conditioned on $(\boldsymbol{x}_t, t)$, the expectation of the SWIFT target is

$$(1+\alpha)\mathcal{D}^\star(\boldsymbol{x}_t, \sigma_t) - \alpha\mathcal{D}_{\mathrm{weak}}(\boldsymbol{x}_t, \sigma_t),$$

the expression obtained by applying AG to the ideal posterior mean (Lemma A.1). AG extrapolates a baseline denoiser already trained for a finite time, while SWIFT changes the regression target before main-model training. Under the assumptions of Section 4, this change can improve approximation to the original posterior mean; it is not a guarantee for every weak denoiser.

3.2 Comparison with Training-Based Guidance

MG constructs a training correction using conditional and unconditional model predictions (Tang et al., 2025). SGG studies incorporating guidance into training using an updated weak network or an intermediate prediction branch in its training-time variants (Yuan et al., 2026). GFT trains across guidance strengths using an unconditional prediction (Chen et al., 2025). In these constructions, the auxiliary predictions can change as their models are trained. IG jointly supervises intermediate and final outputs for sampling-time guidance (Zhou et al., 2026); its first arXiv version additionally studies a training target involving their current prediction difference (Zhou et al., 2025, Sec. 4.3).

These designs use a changing prediction in the training correction or supervise an additional output. SWIFT obtains $\mathcal{D}_{\mathrm{weak}}$ by ordinary denoising training and freezes it before training the main denoiser. For a fixed training example and noise realization, its extrapolated target does not change as the main model is updated. This avoids joint optimization of the weak and main denoisers, which we found to end worse (Section F.4), and leaves a single denoiser at inference. Preparing and evaluating the weak denoiser still incur training compute, which our compute comparisons account for (Section D.3). Further methodological details appear in Section B.

4 Why Swift Can Help

The target in Eq. (5) shifts the population regression target away from the posterior mean. This may seem counterintuitive because the posterior mean is the population optimum of standard squared-loss regression. The key is finite optimization time: ordinary training has not yet learned the posterior mean exactly, while an earlier weak checkpoint has learned still less of the same learnable directions. Extrapolating the target away from this weaker predictor can therefore compensate for part of the finite-time training shortfall.

To make the data distribution explicit, we study a Gaussian mixture. Gaussian corruption preserves its mixture form, and its posterior mean can be nonlinear while remaining analytically tractable. Gaussian mixtures have also been used to study diffusion-model learning (Shah et al., 2023; Gatmiry et al., 2024). We do not require separated clusters or low-rank component means.

Assumption 1 (Mixture data and finite-time training)**.** Fix a diffusion noise level t with $\sigma_t > 0$, and use the corruption process of Section 2.1. The clean data follow

$$\boldsymbol{x}_0 \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \qquad \pi_k > 0, \quad \sum_{k=1}^{K} \pi_k = 1, \quad \boldsymbol{\Sigma} \succ \mathbf{0}, \tag{6}$$

where $K \geq 1$ is finite, $\boldsymbol{\mu}_k \in \mathbb{R}^d$ are arbitrary, and $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is a shared covariance matrix. Let $P_{\boldsymbol{x}_t}$ be the noisy-input distribution and $\mathcal{H}_2 := L^2(P_{\boldsymbol{x}_t}; \mathbb{R}^d)$. Since t is fixed, write $\mathcal{D}^\star$ for

$$\mathcal{D}^\star(\cdot, \sigma_t) = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t = \cdot, t],$$

as defined in Section 2.1.

Training setup and weak denoiser. Ordinary and SWIFT training use zero-initialized population gradient flow in the same coordinate-wise RKHS $\mathcal{H}_{\mathsf{K}}^d$ of a fixed, bounded, measurable, symmetric

positive-semidefinite scalar kernel K. The fixed kernel determines how predictions at different inputs are coupled during training. Optimization time is measured for the normalized objective $\frac{1}{2}\mathbb{E}\|\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{y}\|_2^2$, with the clean label $\boldsymbol{y} = \boldsymbol{x}_0$ for ordinary training and the target of Eq. (5) for SWIFT. The fixed positive loss weight $\lambda(t)$ is absorbed into the optimization times of both runs. The weak denoiser is an ordinary checkpoint at time T_g, frozen before the separate SWIFT run; the baseline and SWIFT runs both train until T, where $0 \le T_g < T < \infty$.

Theorem 1 (Finite-time improvement from a frozen weak denoiser)**.** *Under Assumption 1, define the population training operator and its time-s response by*

$$(\boldsymbol{\mathcal{L}}\boldsymbol{f})(\boldsymbol{x}) := \int \mathsf{K}(\boldsymbol{x}, \boldsymbol{x}')\boldsymbol{f}(\boldsymbol{x}')\, dP_{\boldsymbol{x}_t}(\boldsymbol{x}'), \qquad \mathbf{H}_s := \mathbf{I} - e^{-s\boldsymbol{\mathcal{L}}}.$$

Here $\mathbf{H}_s$ is the finite-time training response: starting from zero, a conditional-mean regression target $\boldsymbol{f}$ produces the predictor $\mathbf{H}_s\boldsymbol{f}$ after optimization time s. Assume that $\mathbf{H}_T\mathcal{D}^\star \neq \mathbf{0}$, i.e., the posterior mean contains at least one nonzero direction that the kernel can learn by time T. The frozen weak denoiser, matched baseline, and SWIFT *predictor at time T are respectively*

$$\mathcal{D}_{\mathrm{weak}} = \mathbf{H}_{T_g}\mathcal{D}^\star, \qquad \mathcal{D}_{\mathrm{base}} = \mathbf{H}_T\mathcal{D}^\star, \qquad \mathcal{D}^{(\alpha)}_{\mathrm{SWIFT}} = \mathbf{H}_T\big[(1+\alpha)\mathcal{D}^\star - \alpha\mathcal{D}_{\mathrm{weak}}\big].$$

For

$$R(\alpha) := \|\mathcal{D}^{(\alpha)}_{\mathrm{SWIFT}} - \mathcal{D}^\star\|^2_{\mathcal{H}_2},$$

the weak denoiser has strictly larger posterior-mean risk than the matched baseline:

$$\|\mathcal{D}_{\mathrm{weak}} - \mathcal{D}^\star\|^2_{\mathcal{H}_2} > R(0).$$

Moreover, there exists $\alpha^\star > 0$ such that

$$R(\alpha) < R(0) \qquad \textit{for every } 0 < \alpha < \alpha^\star. \tag{7}$$

Why this works. Ordinary training has not fully learned the posterior mean by time T. The earlier checkpoint has learned a smaller fraction of each learnable mode, so moving the target away from its prediction strengthens the signal that the guide still misses. The main denoiser can absorb part of this additional signal over the same optimization time T.

For a learnable kernel mode with eigenvalue $\lambda > 0$, the signed error is

$$\underbrace{-e^{-T\lambda}}_{\text{baseline shortfall}} + \underbrace{\alpha(1 - e^{-T\lambda})e^{-T_g\lambda}}_{\text{SWIFT correction}},$$

times the posterior-mean coefficient on that mode. The terms have opposite signs, so sufficiently small positive α partially cancels the baseline's finite-time shortfall and strictly reduces the total squared error whenever a learnable mode is present. If α is too large, the correction can overshoot, explaining why Theorem 1 guarantees only a beneficial interval. Unlearnable modes receive no correction.

The comparison is in population squared error to the posterior mean at one fixed noise level and equal main-model optimization time; guide preparation is excluded. It does not model finite-sample target noise or imply a particular sampling quality. The mixture posterior, proof, and extensions appear in Section A; the Gaussian/RBF condition is verified in Proposition A.6, and narrower guides in Section A.7.

5 EXPERIMENTS

We evaluate SWIFT on ImageNet-256 and on CIFAR-10. In every experiment SWIFT changes only the regression target; architectures, training schedules, samplers, and evaluation pipelines are kept the same as the baseline. On ImageNet, SWIFT uses one fixed recipe unless stated otherwise: α=0.4 with a half-width guide trained for half epoch number of the baseline and then frozen.

Evaluation protocol. Inspired by DiffusionBench (Leng et al., 2026), we evaluate SWIFT across 19 ImageNet-256 model configurations. For each configuration, we compare the baseline and SWIFT with and without inference-time guidance, tuning the guidance strength separately for the two models. We report FID, sFID, precision, recall, and feature-space Fréchet distance and MIND where

Table 1: **Representative ImageNet-256 results.** Each pair uses the same model configuration and sampler. Guidance strengths are tuned separately for the baseline and SWIFT. $\overline{\text{FD}}_r$ and $\overline{\text{MIND}}$ are arithmetic means across Inception, ConvNeXt, DINOv2, MAE, and SigLIP features.

	Without guidance						With guidance					
Family: representation	FID↓	sFID↓	$\overline{\text{FD}}_r$ ↓	$\overline{\text{MIND}}$ ↓	Prec.↑	Rec.↑	FID↓	sFID↓	$\overline{\text{FD}}_r$ ↓	$\overline{\text{MIND}}$ ↓	Prec.↑	Rec.↑
615M parameters, 80 epochs; guidance = CFG												
RAE: DINOv2-B	2.09	4.48	4.12	21.34	0.83	0.57	1.98	4.50	4.14	21.00	0.81	0.59
+ SWIFT	**1.75**	4.26	3.68	21.37	0.81	0.61	**1.59**	4.30	3.65	19.48	0.80	0.62
VAE: SD-VAE (EMA)+reg	5.49	5.52	11.08	87.61	0.71	0.66	2.32	4.70	7.33	32.44	0.80	0.60
+ SWIFT	**3.77**	5.06	8.45	58.11	0.73	0.66	**1.81**	4.42	5.88	26.60	0.80	0.61
E2E: Qwen-VAE	4.94	5.23	9.15	95.25	0.72	0.66	1.50	4.83	4.74	24.47	0.80	0.62
+ SWIFT	**3.23**	4.89	6.46	55.36	0.73	0.66	**1.40**	4.59	3.38	23.03	0.80	0.63
Pixel: PixNerd	20.37	6.14	24.50	384.10	0.61	0.63	4.17	5.50	9.42	42.89	0.65	0.63
+ SWIFT	**10.31**	5.38	15.29	197.31	0.70	0.61	**3.38**	4.95	7.72	35.15	0.68	0.62
875M parameters, 160 epochs; guidance = internal guidance												
RAE: DINOv3-L	1.56	4.10	3.38	19.87	0.82	0.65	1.04	4.00	3.31	15.58	0.82	0.61
+ SWIFT	**1.15**	3.60	2.94	17.95	0.81	0.67	**0.99**	3.80	3.08	14.45	0.81	0.63



Figure 3: **ImageNet-256 samples from SWIFT on RAEv2/DINOv3-L (gFID-50K 0.99).** Samples are generated with the SWIFT-trained model using retuned internal guidance and a 50-step ODE sampler.

available. Table 1 shows selected results; complete experimental settings and results are provided in the appendix.

5.1 GENERATION QUALITY AND TRANSFER ACROSS REPRESENTATIONS

Matched improvements. Table 1 isolates the target change within each model configuration. On RAE/DINOv2-B features (Wang et al., 2025; Zheng et al., 2026; Oquab et al., 2024), SWIFT lowers unguided FID from 2.09 to 1.75 and the five-feature mean FD_r from 4.12 to 3.68 after 80 epochs. The gain persists on the independent RAEv2/DINOv3-L configuration (Singh et al., 2026), where unguided FID improves from 1.56 to 1.15 after 160 epochs.

Transfer with one fixed recipe. Across 18 DiffusionBench configurations spanning latent and pixel representations, one fixed recipe lowers unguided FID in every case, by 16.3%–49.4% (median 31.3%). Even the three lowest-FID baselines improve by 16.3%–17.4%. FD_r and MIND each improve in at least 17 of 18 configurations in every feature space. Full results appear in Table 7.

(a) Autoguidance further improves SWIFT.

	Base	SWIFT	AG	SWIFT+AG
DINOv2-B	2.09	1.75	1.70	**1.63**
SD-VAE	5.49	3.77	2.16	**1.95**
Qwen-VAE	4.94	3.23	1.79	**1.62**
PixNerd	20.37	10.31	4.02	**3.77**
RAEv2	1.56	1.15	1.04	**0.99**

(b) SWIFT shifts the guidance optimum left.

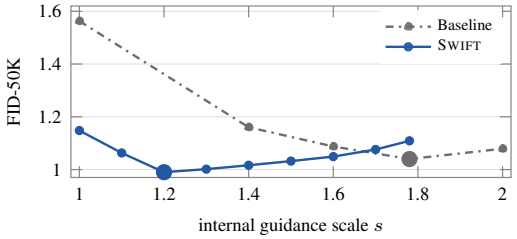

Figure 4: SWIFT **remains beneficial under autoguidance while shifting the preferred guidance scale from** 1.78 **to** 1.20**. (a)** Autoguidance further improves SWIFT across representative carriers. **(b)** On RAEv2, SWIFT shifts the optimum toward weaker inference-time guidance.

Compatibility with inference guidance. With guidance strength tuned separately for each model, SWIFT also lowers guided FID in all 18 DiffusionBench configurations (Table 8). On RAEv2, SWIFT improves internal-guidance FID from 1.04 to 0.99 (Table 1), while its unguided FID improves from 1.56 to 1.15. Representative samples from the 0.99-gFID model are shown in Figure 3.

5.2 COMBINING SWIFT WITH AUTOGUIDANCE

Autoguidance (Karras et al., 2024) applies a weak-to-strong extrapolation at sampling time, whereas SWIFT learns a related correction during training and then discards its guide. The two mechanisms therefore act at different stages, raising two questions: whether inference-time autoguidance still improves a SWIFT-trained model, and whether SWIFT changes how much additional extrapolation is preferred at sampling time.

Across all five configurations in Fig. 4(a), the same ordering appears. SWIFT improves the unguided baseline, autoguidance improves it further, and independently retuned autoguidance on top of SWIFT gives the best FID in every case. This shows that the improvement learned during training is not replaced by inference-time guidance: even after the baseline has been separately retuned for autoguidance, the SWIFT model retains an additional gain.

At the same time, the two corrections are not independent. On RAEv2, the optimal internal-guidance scale shifts from 1.78 for the control to 1.20 for SWIFT, while the best guided FID improves from 1.04 to 0.99 (Fig. 4(b)). The lower preferred scale is consistent with SWIFT having already absorbed part of the weak-to-strong extrapolation during training, so less additional correction is required at inference. The remaining gain from autoguidance shows that this correction is only partially internalized rather than exhausted.

5.3 LONGER TRAINING AND TRAINING COST

The gain persists with longer training. We continue the matched DINOv2-B pair to 240 epochs. SWIFT is ahead at all 36 evaluations from epochs 20 to 240, with the gap growing from 0.12 to 0.26 FID; the 50k re-evaluation at epoch 240 gives $1.64 \rightarrow 1.35$. A separate control continued to 320 epochs reaches at best 1.53, still above the 1.50 reached by SWIFT at epoch 120. The same persistent advantage holds on all four DiffusionBench configurations with recorded full trajectories.

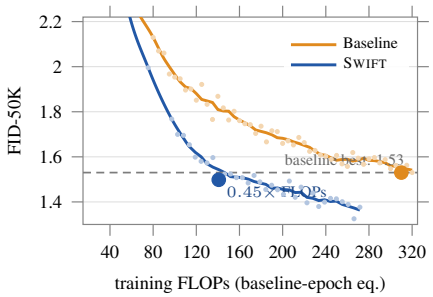

Figure 5: **Same-FLOPs comparison on DINOv2-B.** SWIFT surpasses the best 320-epoch baseline quality using 0.45× the training FLOPs.

Training cost. SWIFT adds 9% FLOPs per epoch for the frozen guide, plus roughly 10 baseline-equivalent epochs to pretrain it. This overhead is modest among training-time guidance methods: MG and SGG incur about 1.23× and 1.22× per-iteration cost in the unified SiT-B/2 comparison of Yuan et al., while GFT reports 10–20% additional computation per update; IG reports negligible overhead. Charging both of our costs, SWIFT at epoch 120 already surpasses the best DINOv2-B control result observed

through 320 epochs (1.53), using only $0.45\times$ its training FLOPs (Figure 5). On RAEv2, epoch-50 SWIFT similarly improves on the epoch-160 control (1.34 vs. 1.56) at $0.40\times$ the FLOPs.

These ratios are conservative time-to-quality comparisons. SWIFT continues to reach FID levels that the corresponding controls never attain within the observed training horizon; for such quality targets, no finite baseline time-to-quality is observed. A single finite speedup factor therefore understates the gain when SWIFT also improves the attainable quality frontier.

5.4 How to Choose α and the Weak Guide

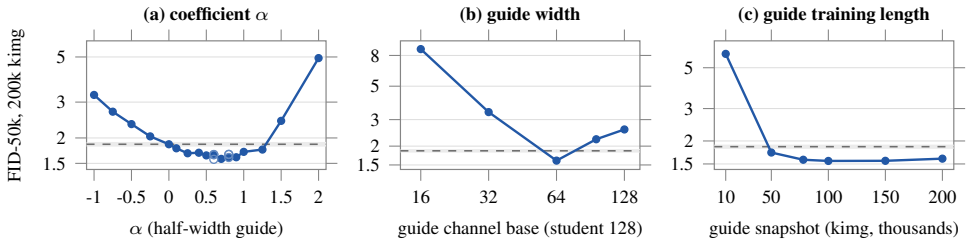

Figure 6: **Choosing α and the weak guide on CIFAR-10.** Moderate positive α, a half-width guide, and a sufficiently trained weak guide give the strongest results.

Choosing α. Performance is broadly U-shaped in α (Figure 6a). On CIFAR-10, moderate positive values form a broad favorable region around $\alpha = 0.7$–0.9, where differences are comparable to replicate noise. ImageNet shows the same qualitative behavior: on DINOv2-B, $\alpha \in [0.5, 1.0]$ substantially improves over the control . The precise optimum depends on the carrier, however, so our breadth experiments use the fixed default $\alpha = 0.4$ without tuning; in practice, a small sweep over moderate positive values is sufficient. Negative α can accelerate the initial decrease in FID, but this advantage disappears later and has little effect on final quality.

Choosing the weak guide. The guide should be weaker than the student, but still competent. At $\alpha = 0.8$ on CIFAR-10, reducing the channel base from the student's 128 to 64 improves FID from 2.58 to 1.60, while substantially smaller guides degrade rapidly (Figure 6b). The half-width guide is also the best width across the tested α values on CIFAR-10 and consistently outperforms wider alternatives on ImageNet. Guide training length shows a similar pattern: once the guide crosses a competence threshold, performance is relatively insensitive over a broad range (Figure 6c). On ImageNet, freezing the guide earlier can even help, with epoch 20 outperforming epoch 40 in our 50k re-evaluation.

A fixed guide is enough. We also tested more elaborate variants, including jointly training the weak guide with the student and changing α during training. Co-training the guide is less effective than keeping it frozen, suggesting that a stable weak reference is useful for defining the extrapolation target. Likewise, using negative α early can speed up early optimization but does not improve the eventual solution. Other adaptive schedules and online variants are similarly neutral or harmful (Section F.4). We therefore use a simple default throughout: a frozen half-width guide and a fixed, moderate positive α.

6 Conclusion

The regression target has received comparatively little attention as a design choice in diffusion training, yet it turns out to be a simple and effective place to intervene. By carrying the clean target past itself and away from a frozen weak model, SWIFT moves a weak-to-strong correction from inference into training, leaving a single denoiser and the original sampler at deployment with only about 9% additional main-training FLOPs. Crucially, the modified target is not itself a better estimator of the posterior mean: it is deliberately biased, yet finite training on it can produce a better final predictor and better samples. This suggests a broader view of the regression target as a design interface rather than a fixed quantity—and raises the question of what other useful corrections can be written into it.

REPRODUCIBILITY STATEMENT

SWIFT changes only the regression target: the objective is Eqs. (4) and (5), the frozen weak denoiser is constructed as described in Section 2.2, and everything else is the training loop of the respective carrier. Section D.1 lists every setting that differs from the released carriers (EDM, DiffusionBench, and RAEv2), including guide architectures, freezing points, coefficients, seeds, batch sizes, and hardware, and explains how the target is applied to velocity-predicting models. Section D.2 specifies the samplers, guidance settings, and metrics, and validates our evaluation harness against released checkpoints; Section D.3 reports the GPU hours and the FLOPs accounting behind Figures 1 and 5. Section E gives all 19 ImageNet-256 configurations with and without guidance on all 15 metrics, and Section F the CIFAR-10 experiments, including an account of all 156 launched (152 scored) arms (Section F.5). In Appendix C, we specifies both synthetic experiments, and all assumptions and proofs are in Section A. Code patches for the three carriers, the guide checkpoints, and the machine-readable campaign ledger (`DATA/`) will be released with the paper.

AI USE STATEMENT

We used large language models (LLMs) in preparing this paper. They edited the text for clarity; drafted parts of the text, including appendix sections, from our experiment logs and notes; generated LaTeX and TikZ code for tables and figures; computed summary statistics from our result files. The authors reviewed and revised all LLM-assisted content and take full responsibility for the paper.

REFERENCES

Huayu Chen, Kai Jiang, Kaiwen Zheng, Jianfei Chen, Hang Su, and Jun Zhu. Visual Generation Without Guidance. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8275–8295. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25ac.html.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked Diffusion Transformer is a Strong Image Synthesizer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. URL https://arxiv.org/abs/2303.14389.

Khashayar Gatmiry, Jonathan Kelner, and Holden Lee. Learning Mixtures of Gaussians Using Diffusion Models. *arXiv preprint arXiv:2404.18869*, 2024. URL https://arxiv.org/abs/2404.18869.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. *arXiv preprint arXiv:2204.03458*, 2022. URL https://arxiv.org/abs/2204.03458.

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 (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00364.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a Diffusion Model with a Bad Version of Itself. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.02507.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2009.09761.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for End-to-End Tuning with Latent Diffusion Transformers. *arXiv preprint arXiv:2504.10483*, 2025. URL https://arxiv.org/abs/2504.10483.

Xingjian Leng, Jaskirat Singh, Zhanhao Liang, Ethan Smith, Martin Bell, Aninda Saha, Yuhui Yuan, and Liang Zheng. DiffusionBench: On Holistic Evaluation of Diffusion Transformers. *arXiv preprint arXiv:2606.24888*, 2026. URL https://arxiv.org/abs/2606.24888.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive Image Generation without Vector Quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.11838.

Hangyu Liu, Jianyong Wang, and Yutao Sun. Geometric Autoencoder for Diffusion Models. *arXiv preprint arXiv:2603.10365*, 2026. URL https://arxiv.org/abs/2603.10365.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2401.08740.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2304.07193.

Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. Semantics Lead the Way: Harmonizing Semantic and Texture Modeling with Asynchronous Latent Diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2512.04926.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. URL https://arxiv.org/abs/2212.09748.

Kulin Shah, Sitan Chen, and Adam Klivans. Learning Mixtures of Gaussians Using the DDPM Objective. *arXiv preprint arXiv:2307.01178*, 2023. URL https://arxiv.org/abs/2307.01178.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved Baselines with Representation Autoencoders. *arXiv preprint arXiv:2605.18324*, 2026. URL https://arxiv.org/abs/2605.18324.

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 (ICLR)*, 2021. URL https://arxiv.org/abs/2011.13456.

Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion Models without Classifier-free Guidance. *arXiv preprint arXiv:2502.12154*, 2025. URL https://arxiv.org/abs/2502.12154.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled Diffusion Transformer. *arXiv preprint arXiv:2504.05741*, 2025. URL https://arxiv.org/abs/2504.05741.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation Entanglement for Generation: Training Diffusion Transformers Is Much Easier Than You Think. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2507.01467.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. Generation: Taming Optimization Dilemma in Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2501.01423.

Qihang Yu, Qihao Liu, Ju He, Xinyang Zhang, Yang Liu, Liang-Chieh Chen, and Xi Chen. Autoregressive Image Generation with Masked Bit Modeling. *arXiv preprint arXiv:2602.09024*, 2026. URL https://arxiv.org/abs/2602.09024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation Alignment for Generation: Training Diffusion Transformers Is Easier Than You Think. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.06940.

Liangyu Yuan, Yufei Huang, Mingkun Lei, Tong Zhao, Ruoyu Wang, Changxi Chi, Yiwei Wang, and Chi Zhang. Improving Diffusion Generalization with Weak-to-Strong Segmented Guidance. *arXiv preprint arXiv:2603.20584*, 2026. URL https://arxiv.org/abs/2603.20584.

Zhengrong Yue, Taihang Hu, Mengting Chen, Haiyu Zhang, Zihao Pan, Tao Liu, Zikang Wang, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, and Yali Wang. What Matters for Diffusion-Friendly Latent Manifold? Prior-Aligned Autoencoders for Latent Diffusion. *arXiv preprint arXiv:2605.07915*, 2026. URL https://arxiv.org/abs/2605.07915.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion Transformers with Representation Autoencoders. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.11690. arXiv:2510.11690.

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a Diffusion Transformer with the Internal Dynamics of Itself. *arXiv preprint arXiv:2512.24176v1*, 2025. URL https://arxiv.org/abs/2512.24176v1. Version 1; cite only for version-specific claims.

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a Diffusion Transformer with the Internal Dynamics of Itself. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11536–11545, 2026. URL https://arxiv.org/abs/2512.24176.

Overview. Section A proves the results of Section 4. Section B extends the comparison with guidance methods in Section 3. Section C describes the two synthetic experiments: the setup of the tree-shaped 2D example in Figure 2, and a Gaussian mixture whose optimal denoiser is known in closed form, on which we measure how close each training-time method places a capacity-limited network to it. Section D gives the three carriers, the evaluation protocol, and the compute we used. Section E reports every ImageNet-256 result, Section F the CIFAR-10 experiments, and Section G additional samples.

A Theory for Gaussian mixtures and finite-time kernel training

This appendix proves Theorem 1. At the fixed noise level, $\mathcal{D}^\star$ denotes the function $\mathcal{D}^\star(\cdot, \sigma_t)$ from Section 2.1. The core argument has two steps: we first prove a general improvement theorem for commuting training-response operators, which does not depend on the Gaussian-mixture assumption, and then specialize it to RKHS population gradient flow to prove Theorem 1. We begin with two preliminary calculations for the SWIFT target and the shared-covariance Gaussian-mixture posterior. The remaining subsections give extensions and robustness results covering Gaussian-kernel injectivity, reduced-rank guides, finite-step fixed-feature training, finite-sample target noise, smooth nonlinear optimizers, and perturbations away from the exact kernel regime.

Throughout,

$$\begin{aligned}\mathcal{H}_2 &= L^2(P_{\boldsymbol{x}_t}; \mathbb{R}^d),\\ \langle \boldsymbol{f}, \boldsymbol{g}\rangle_{\mathcal{H}_2} &= \mathbb{E}[\boldsymbol{f}(\boldsymbol{x}_t)^\top \boldsymbol{g}(\boldsymbol{x}_t)],\\ \|\boldsymbol{f}\|^2_{\mathcal{H}_2} &= \langle \boldsymbol{f}, \boldsymbol{f}\rangle_{\mathcal{H}_2}.\end{aligned}$$

Unsubscripted function inner products and norms in the population analysis are those of $\mathcal{H}_2$. The operator $\boldsymbol{\mathcal{L}}$ is the population kernel operator defined in Theorem 1. At the fixed noise level, we suppress the argument σ_t in $\mathcal{D}_{\text{weak}}$. For the main-theorem proof, write $\boldsymbol{d}_c := \mathcal{D}_{\text{base}}$ and $\boldsymbol{d}_\alpha := \mathcal{D}^{(\alpha)}_{\text{SWIFT}}$. All random variables and predictors below are square integrable. This is automatic for the Gaussian-mixture model and for bounded-kernel population flow applied to its regression targets.

Weak guide used in the main theorem. The guide in the main theorem is not an arbitrary external predictor. At each fixed noise level t, start from the zero predictor and run population gradient flow in the same vector-valued RKHS on the ordinary clean-target objective

$$\mathcal{L}_{\text{base}}(\boldsymbol{d}) = \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{x}_0\|_2^2\right].$$

At fixed t, the weighted loss in Eq. (1) is $2\lambda(t)$ times this normalized objective. If τ denotes optimization time for the weighted loss, then $s = 2\lambda(t)\tau$ gives the normalized gradient-flow trajectory. The same rescaling applies to the SWIFT loss, so the baseline and SWIFT are compared under the same time convention. If $\boldsymbol{d}_s^{\text{base}}$ denotes the resulting predictor after optimization time s, then

$$\boldsymbol{d}_s^{\text{base}} = \mathbf{H}_s \mathcal{D}^\star, \qquad \mathbf{H}_s := \mathbf{I} - e^{-s\boldsymbol{\mathcal{L}}}.$$

Choose $0 \le T_g < T < \infty$. Stop the standard trajectory at T_g, save that checkpoint as

$$\mathcal{D}_{\text{weak}} := \boldsymbol{d}_{T_g}^{\text{base}} = \mathbf{H}_{T_g}\mathcal{D}^\star,$$

and freeze it. Continue the same ordinary trajectory to time T to obtain the matched control

$$\boldsymbol{d}_c := \boldsymbol{d}_T^{\text{base}} = \mathbf{H}_T \mathcal{D}^\star.$$

Equivalently, one may train two zero-initialized copies with the same standard loss and stop them at T_g and T, respectively. In either implementation, the guide is learned only from the ordinary clean target $\boldsymbol{x}_0$; the SWIFT target is not used to construct it. Because $0 \le T_g < T$, spectral calculus gives

$$\|\mathcal{D}_{\text{weak}} - \mathcal{D}^\star\|^2_{\mathcal{H}_2} \ge \|\boldsymbol{d}_c - \mathcal{D}^\star\|^2_{\mathcal{H}_2},$$

with strict inequality under the nonzero-predictor condition of Theorem 1, as proved in Section A.5. In this precise sense, $\mathcal{D}_{\text{weak}}$ is the weak model. The SWIFT denoiser is then initialized from zero and trained for the same time T, but uses the extrapolated labels $(1+\alpha)\boldsymbol{x}_0 - \alpha\mathcal{D}_{\text{weak}}(\boldsymbol{x}_t)$. Consequently, the guide is weak only because it is less trained, not because it uses a different target or data distribution. The reduced-rank extension in Section A.7 treats a spectrally nested, narrower guide.

A.1 TARGET IDENTITY AND THE QUANTITY BEING COMPARED

The first calculation separates the mechanism from the particular training model. Write

$$\boldsymbol{\xi} = \boldsymbol{x}_0 - \mathcal{D}^\star(\boldsymbol{x}_t), \qquad \mathbb{E}[\boldsymbol{\xi} \mid \boldsymbol{x}_t] = \mathbf{0},$$
$$\boldsymbol{y}_\alpha = (1+\alpha)\boldsymbol{x}_0 - \alpha\mathcal{D}_{\mathrm{weak}}(\boldsymbol{x}_t).$$

Lemma A.1 (SWIFT target decomposition)**.** *For every fixed square-integrable guide $\mathcal{D}_{\mathrm{weak}}$,*

$$\boldsymbol{y}_\alpha = \mathcal{D}^\star + \alpha(\mathcal{D}^\star - \mathcal{D}_{\mathrm{weak}}) + (1+\alpha)\boldsymbol{\xi}, \tag{A.1}$$

$$\boldsymbol{g}_\alpha(\boldsymbol{x}_t) := \mathbb{E}[\boldsymbol{y}_\alpha \mid \boldsymbol{x}_t] = \mathcal{D}^\star(\boldsymbol{x}_t) + \alpha\big(\mathcal{D}^\star(\boldsymbol{x}_t) - \mathcal{D}_{\mathrm{weak}}(\boldsymbol{x}_t)\big). \tag{A.2}$$

Moreover,

$$\mathrm{Cov}(\boldsymbol{y}_\alpha \mid \boldsymbol{x}_t) = (1+\alpha)^2\,\mathrm{Cov}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t). \tag{A.3}$$

For every square-integrable predictor $\boldsymbol{d}$,

$$\mathbb{E}\|\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{y}_\alpha\|_2^2 = \|\boldsymbol{d} - \boldsymbol{g}_\alpha\|_{\mathcal{H}_2}^2 + (1+\alpha)^2\mathbb{E}\|\boldsymbol{x}_0 - \mathcal{D}^\star(\boldsymbol{x}_t)\|_2^2. \tag{A.4}$$

Consequently, the unrestricted population minimizer of the SWIFT *regression loss is $\boldsymbol{g}_\alpha$, and its posterior-mean risk is $\alpha^2\|\mathcal{D}_{\mathrm{weak}} - \mathcal{D}^\star\|_{\mathcal{H}_2}^2$.*

Proof. Substituting $\boldsymbol{x}_0 = \mathcal{D}^\star + \boldsymbol{\xi}$ gives Eq. equation A.1; conditioning on $\boldsymbol{x}_t$ gives Eq. equation A.2. Since $\mathcal{D}_{\mathrm{weak}}(\boldsymbol{x}_t)$ is fixed after conditioning on $\boldsymbol{x}_t$, the same decomposition gives Eq. equation A.3. Moreover, $\boldsymbol{y}_\alpha - \boldsymbol{g}_\alpha = (1+\alpha)\boldsymbol{\xi}$ and $\boldsymbol{d} - \boldsymbol{g}_\alpha$ is $\boldsymbol{x}_t$-measurable, so

$$\mathbb{E}\big[(\boldsymbol{d} - \boldsymbol{g}_\alpha)^\top\boldsymbol{\xi}\big] = \mathbb{E}\big[(\boldsymbol{d} - \boldsymbol{g}_\alpha)^\top\mathbb{E}[\boldsymbol{\xi} \mid \boldsymbol{x}_t]\big] = 0.$$

Expanding the square proves Eq. equation A.4; the remaining claims follow immediately. □

The posterior-mean risk used in Theorem 1 is also exactly the reducible part of clean-sample denoising error: for every $\boldsymbol{d}$,

$$\mathbb{E}\|\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{x}_0\|_2^2 = \|\boldsymbol{d} - \mathcal{D}^\star\|_{\mathcal{H}_2}^2 + \mathbb{E}\|\boldsymbol{x}_0 - \mathcal{D}^\star(\boldsymbol{x}_t)\|_2^2.$$

Thus comparing two predictors by distance to $\mathcal{D}^\star$ is equivalent to comparing their population clean-target MSE. Lemma A.1 also shows why finite training is essential: if the control reaches $\mathcal{D}^\star$ exactly, then its risk is zero, whereas the converged SWIFT target has positive risk for $\alpha \neq 0$ unless $\mathcal{D}_{\mathrm{weak}} = \mathcal{D}^\star$.

A.2 POSTERIOR MEAN OF THE SHARED-COVARIANCE GAUSSIAN MIXTURE

Let $I \in \{1, \dots, K\}$ denote the mixture component. For any component index $k \in \{1, \dots, K\}$, conditional on $I = k$,

$$\boldsymbol{x}_0 \mid I = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}),$$
$$\boldsymbol{x}_t \mid I = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\mathrm{obs}}),$$
$$\boldsymbol{\Sigma}_{\mathrm{obs}} = \boldsymbol{\Sigma} + \sigma_t^2\mathbf{I}_d.$$

Lemma A.2 (Gaussian-mixture posterior)**.** *Under Eq. equation 6, define*

$$\ell_k(\boldsymbol{x}_t) = \boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_{\mathrm{obs}}^{-1}\boldsymbol{x}_t - \tfrac{1}{2}\boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_{\mathrm{obs}}^{-1}\boldsymbol{\mu}_k,$$
$$w_k(\boldsymbol{x}_t) = \frac{\pi_k e^{\ell_k(\boldsymbol{x}_t)}}{\sum_{j=1}^K \pi_j e^{\ell_j(\boldsymbol{x}_t)}},$$
$$\overline{\boldsymbol{\mu}}(\boldsymbol{x}_t) = \sum_{k=1}^K w_k(\boldsymbol{x}_t)\boldsymbol{\mu}_k.$$

Then $w_k(\boldsymbol{x}_t) = \mathbb{P}(I = k \mid \boldsymbol{x}_t)$ and

$$\mathcal{D}^\star(\boldsymbol{x}_t) = \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \mathbf{B}\boldsymbol{x}_t + (\mathbf{I}_d - \mathbf{B})\overline{\boldsymbol{\mu}}(\boldsymbol{x}_t), \qquad \mathbf{B} = \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\mathrm{obs}}^{-1}. \tag{A.5}$$

Proof. For fixed k, $(\boldsymbol{x}_0, \boldsymbol{x}_t)$ is jointly Gaussian with $\mathrm{Cov}(\boldsymbol{x}_0, \boldsymbol{x}_t \mid I = k) = \boldsymbol{\Sigma}$ and $\mathrm{Cov}(\boldsymbol{x}_t \mid I = k) = \boldsymbol{\Sigma}_{\mathrm{obs}}$. Gaussian conditioning gives

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t, I = k] &= \boldsymbol{\mu}_k + \boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\mathrm{obs}}^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k) \\ &= \mathbf{B}\boldsymbol{x}_t + (\mathbf{I}_d - \mathbf{B})\boldsymbol{\mu}_k. \end{aligned} \tag{A.6}$$

Bayes' rule yields

$$\mathbb{P}(I = k \mid \boldsymbol{x}_t) \propto \pi_k \exp\left[-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_{\text{obs}}^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)\right]$$
$$\propto \pi_k \exp\left[\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_{\text{obs}}^{-1}\boldsymbol{x}_t - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_{\text{obs}}^{-1}\boldsymbol{\mu}_k\right],$$

where the common term $-\boldsymbol{x}_t^\top \boldsymbol{\Sigma}_{\text{obs}}^{-1}\boldsymbol{x}_t/2$ cancels. Averaging Eq. equation A.6 over these weights proves Eq. equation A.5.

□

A.3 A GENERAL COMMUTING-RESPONSE THEOREM

The key result is most transparent at the level of training-response operators. It does not require a finite feature map or even an RKHS.

Let $\mathbf{H}_c, \mathbf{H}_g : \mathcal{H}_2 \to \mathcal{H}_2$ denote the population responses of the main denoiser and the guide: training on a target with conditional mean $\boldsymbol{g}$ returns $\mathbf{H}_c\boldsymbol{g}$ under the main-denoiser training dynamics and $\mathbf{H}_g\boldsymbol{g}$ under the guide training dynamics. We require

$$\begin{aligned} \mathbf{H}_c = \mathbf{H}_c^*, \qquad & \mathbf{H}_g = \mathbf{H}_g^*, \\ \mathbf{0} \preceq \mathbf{H}_c \preceq \mathbf{I}, \qquad & \mathbf{0} \preceq \mathbf{H}_g \preceq \mathbf{I}, \\ \mathbf{H}_c\mathbf{H}_g = \mathbf{H}_g\mathbf{H}_c. & \end{aligned} \tag{A.7}$$

These assumptions say that the main denoiser and guide act as commuting spectral shrinkage operators. Their function classes may be infinite dimensional.

Theorem A.3 (Commuting-response improvement). *Suppose Eq. equation A.7 holds. Let*

$$\boldsymbol{d}_c = \mathbf{H}_c\mathcal{D}^\star, \qquad \mathcal{D}_{\text{weak}} = \mathbf{H}_g\mathcal{D}^\star,$$
$$\boldsymbol{d}_\alpha = \mathbf{H}_c\big((1+\alpha)\mathcal{D}^\star - \alpha\mathcal{D}_{\text{weak}}\big).$$

Define

$$\boldsymbol{e} = (\mathbf{I} - \mathbf{H}_c)\mathcal{D}^\star, \quad \boldsymbol{q} = \mathbf{H}_c(\mathbf{I} - \mathbf{H}_g)\mathcal{D}^\star,$$
$$A = \langle \boldsymbol{e}, \boldsymbol{q}\rangle, \qquad C = \|\boldsymbol{q}\|^2.$$

Then

$$\boldsymbol{d}_\alpha - \mathcal{D}^\star = -\boldsymbol{e} + \alpha\boldsymbol{q}, \qquad R(\alpha) = R(0) - 2A\alpha + C\alpha^2, \tag{A.8}$$

and

$$A = \langle \mathcal{D}^\star, (\mathbf{I} - \mathbf{H}_c)\mathbf{H}_c(\mathbf{I} - \mathbf{H}_g)\mathcal{D}^\star\rangle \geq 0. \tag{A.9}$$

If

$$\langle \mathcal{D}^\star, (\mathbf{I} - \mathbf{H}_c)\mathbf{H}_c(\mathbf{I} - \mathbf{H}_g)\mathcal{D}^\star\rangle > 0, \tag{A.10}$$

then $C > 0$, the unique minimizer is $\alpha_{\text{PM}}^\star = A/C > 0$, and

$$R(\alpha) < R(0) \quad \Longleftrightarrow \quad 0 < \alpha < 2\alpha_{\text{PM}}^\star.$$

Proof. Using $\mathcal{D}_{\text{weak}} = \mathbf{H}_g\mathcal{D}^\star$ and linearity of $\mathbf{H}_c$,

$$\begin{aligned} \boldsymbol{d}_\alpha - \mathcal{D}^\star &= \mathbf{H}_c\mathcal{D}^\star + \alpha\mathbf{H}_c(\mathbf{I} - \mathbf{H}_g)\mathcal{D}^\star - \mathcal{D}^\star \\ &= -(\mathbf{I} - \mathbf{H}_c)\mathcal{D}^\star + \alpha\mathbf{H}_c(\mathbf{I} - \mathbf{H}_g)\mathcal{D}^\star \\ &= -\boldsymbol{e} + \alpha\boldsymbol{q}. \end{aligned}$$

Taking the squared norm gives Eq. equation A.8. Because the three operators $\mathbf{I} - \mathbf{H}_c$, $\mathbf{H}_c$, and $\mathbf{I} - \mathbf{H}_g$ are commuting and positive semidefinite, their product is positive semidefinite. This proves Eq. equation A.9. Under Eq. equation A.10, $\boldsymbol{q} \neq \mathbf{0}$ and hence $C > 0$. The remaining claims follow by minimizing the strictly convex quadratic $R(0) - 2A\alpha + C\alpha^2$. □

The strict-overlap condition has a direct joint-spectral interpretation. If $h_c(\omega), h_g(\omega) \in [0,1]$ are the joint spectral multipliers and $\nu_{\mathcal{D}^\star}$ is the spectral measure induced by $\mathcal{D}^\star$, then

$$A = \int (1 - h_c(\omega))h_c(\omega)(1 - h_g(\omega))\, d\nu_{\mathcal{D}^\star}(\omega).$$

Thus strict improvement requires positive signal mass on modes that are visible to the main denoiser, not exactly fitted by the main denoiser, and not exactly fitted by the guide.

For the coordinatewise RKHS response used below, the same conclusion can be written mode by mode. Let $\boldsymbol{\mathcal{L}}_0$ denote the scalar integral operator on $L^2(P_{\boldsymbol{x}_t})$ induced by K. Since $\boldsymbol{\mathcal{L}}_0$ is compact and positive, let $(\lambda_j, \phi_j)_{j\geq 1}$ be an orthonormal list of its positive-eigenvalue modes, repeated with multiplicity. Expanding every output coordinate in this common scalar basis gives

$$\mathcal{D}^\star = \boldsymbol{d}_0^\star + \sum_{j\geq 1} \boldsymbol{d}_j^\star \phi_j, \qquad \boldsymbol{d}_0^\star \in \ker \boldsymbol{\mathcal{L}}, \qquad \boldsymbol{d}_j^\star \in \mathbb{R}^d.$$

For $\mathbf{H}_c = \mathbf{I} - e^{-T\boldsymbol{\mathcal{L}}}$ and $\mathbf{H}_g = \mathbf{I} - e^{-T_g\boldsymbol{\mathcal{L}}}$, put $b_j = (1 - e^{-T\lambda_j})e^{-T_g\lambda_j}$. Under the main theorem's nonzero-predictor condition, Parseval's identity and risk minimization give

$$R(\alpha) = \|\boldsymbol{d}_0^\star\|_{\mathcal{H}_2}^2 + \sum_{j\geq 1} \|\boldsymbol{d}_j^\star\|_2^2 \left(-e^{-T\lambda_j} + \alpha b_j\right)^2, \tag{A.11}$$

$$\alpha_{\mathrm{PM}}^\star(T_g) = \frac{\sum_{j\geq 1} \|\boldsymbol{d}_j^\star\|_2^2 e^{-T\lambda_j} b_j}{\sum_{j\geq 1} \|\boldsymbol{d}_j^\star\|_2^2 b_j^2}.$$

The sums are finite or convergent by Parseval and boundedness of all spectral multipliers. Equation equation A.11 also makes clear that the $\ker \boldsymbol{\mathcal{L}}$ component is an irreducible term shared by the control and every SWIFT predictor; strict improvement only requires at least one active positive-eigenvalue mode.

Why some training restriction is necessary. If the control is trained to the posterior mean, then $\mathbf{H}_c\mathcal{D}^\star = \mathcal{D}^\star$, $R(0) = 0$, and no method can become strictly closer. More generally, if $\mathbf{H}_c$ is an exact orthogonal projection onto a closed linear function class, then $(\mathbf{I} - \mathbf{H}_c)\mathcal{D}^\star \perp \operatorname{ran} \mathbf{H}_c$, so $A = 0$ for every guide and

$$R(\alpha) = R(0) + \alpha^2 \|\mathbf{H}_c(\mathbf{I} - \mathbf{H}_g)\mathcal{D}^\star\|^2.$$

Limited expressivity by itself is therefore not the mechanism. The positive result relies on finite-time partial fitting.

A.4 RKHS POPULATION GRADIENT FLOW

Let K be a bounded, measurable, symmetric positive-semidefinite scalar kernel and let $\mathcal{H}_{\mathsf{K}}^d$ be the RKHS associated with the matrix-valued kernel $\mathbf{K}_d = \mathsf{K}\mathbf{I}_d$. Applied coordinatewise, the integral operator

$$(\boldsymbol{\mathcal{L}}\boldsymbol{f})(\boldsymbol{x}_t) = \int \mathsf{K}(\boldsymbol{x}_t, \boldsymbol{x}_t')\boldsymbol{f}(\boldsymbol{x}_t')\, dP_{\boldsymbol{x}_t}(\boldsymbol{x}_t')$$

is bounded, compact, self-adjoint, and positive semidefinite on $\mathcal{H}_2$. All statements below also hold for a finite-rank kernel. For a finite-dimensional feature representation $\mathsf{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$, the scalar integral operator $\boldsymbol{\mathcal{L}}_0$ has the same nonzero eigenvalues, including multiplicities, as the uncentered feature second-moment matrix $\mathbb{E}[\boldsymbol{\phi}(\boldsymbol{x}_t)\boldsymbol{\phi}(\boldsymbol{x}_t)^\top]$.

Lemma A.4 (Prediction-space kernel flow)**.** *Let $\boldsymbol{y}$ be a square-integrable vector regression target with conditional mean $\boldsymbol{g}(\boldsymbol{x}_t) = \mathbb{E}[\boldsymbol{y} \mid \boldsymbol{x}_t]$. Population gradient flow for the objective $\frac{1}{2}\mathbb{E}\|\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{y}\|_2^2$ in $\mathcal{H}_{\mathsf{K}}^d$ from the zero predictor satisfies, in prediction space,*

$$\frac{d}{ds}\boldsymbol{d}_s = -\boldsymbol{\mathcal{L}}(\boldsymbol{d}_s - \boldsymbol{g}), \qquad \boldsymbol{d}_0 = \mathbf{0}, \tag{A.12}$$

and therefore

$$\boldsymbol{d}_s = \mathbf{H}_s\boldsymbol{g}, \qquad \mathbf{H}_s = \mathbf{I} - e^{-s\boldsymbol{\mathcal{L}}}.$$

The conditional variance of $\boldsymbol{y}$ affects the value of the loss but not this population trajectory.

Proof. Conditioning on $\boldsymbol{x}_t$ gives the orthogonal decomposition

$$\mathbb{E}\|\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{y}\|_2^2 = \|\boldsymbol{d} - \boldsymbol{g}\|_{\mathcal{H}_2}^2 + \mathbb{E}\|\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}_t)\|_2^2. \tag{A.13}$$

After multiplying Eq. equation A.13 by $1/2$, the second term is independent of $\boldsymbol{d}$. If $\mathbf{J} : \mathcal{H}_{\mathsf{K}}^d \to \mathcal{H}_2$ is the canonical inclusion, then $\mathbf{J}$ is bounded because $\|\mathbf{J}\boldsymbol{f}\|_{\mathcal{H}_2}^2 \leq \sup_{\boldsymbol{x}} \mathsf{K}(\boldsymbol{x}, \boldsymbol{x})\|\boldsymbol{f}\|_{\mathcal{H}_{\mathsf{K}}^d}^2$. For the

RKHS-valued trajectory $\boldsymbol{u}_s$, the gradient is $\mathbf{J}^*(\mathbf{J}\boldsymbol{u}_s - \boldsymbol{g})$. Hence

$$\frac{d}{ds}\boldsymbol{u}_s = -\mathbf{J}^*(\mathbf{J}\boldsymbol{u}_s - \boldsymbol{g}), \qquad \boldsymbol{u}_0 = \mathbf{0}.$$

Because $\mathbf{J}^*\mathbf{J}$ is bounded, this linear Hilbert-space ODE has a unique global solution. The reproducing property gives $\boldsymbol{\mathcal{L}} = \mathbf{J}\mathbf{J}^*$. Applying $\mathbf{J}$ to the ODE and writing $\boldsymbol{d}_s = \mathbf{J}\boldsymbol{u}_s$ proves Eq. equation A.12. This argument requires only $\boldsymbol{g} \in \mathcal{H}_2$, not membership in the RKHS. Since $\boldsymbol{\mathcal{L}}$ is bounded and self-adjoint, functional calculus gives

$$\boldsymbol{d}_s = e^{-s\boldsymbol{\mathcal{L}}}\boldsymbol{d}_0 + \int_0^s e^{-(s-u)\boldsymbol{\mathcal{L}}}\boldsymbol{\mathcal{L}}\boldsymbol{g}\, du = (\mathbf{I} - e^{-s\boldsymbol{\mathcal{L}}})\boldsymbol{g}.$$

□

For every $s \geq 0$, $\mathbf{H}_s$ is a self-adjoint positive contraction. Moreover, $\mathbf{H}_s$ and $\mathbf{H}_{s'}$ commute because they are functions of the same operator $\boldsymbol{\mathcal{L}}$. Lemma A.4 therefore places arbitrary RKHS gradient flow inside Theorem A.3.

A.5 Proof of Theorem 1

Proof of Theorem 1. By Lemma A.4, training on the clean target $\boldsymbol{x}_0$, whose conditional mean is $\mathcal{D}^\star$, gives

$$\boldsymbol{d}_c = \mathbf{H}_T\mathcal{D}^\star, \qquad \mathcal{D}_{\text{weak}} = \mathbf{H}_{T_g}\mathcal{D}^\star.$$

The SWIFT target satisfies

$$\mathbb{E}[(1+\alpha)\boldsymbol{x}_0 - \alpha\mathcal{D}_{\text{weak}}(\boldsymbol{x}_t) \mid \boldsymbol{x}_t] = (1+\alpha)\mathcal{D}^\star - \alpha\mathcal{D}_{\text{weak}}.$$

Applying the main-denoiser response $\mathbf{H}_T$ and using $\mathbf{I} - \mathbf{H}_T = e^{-T\boldsymbol{\mathcal{L}}}$ and $\mathbf{I} - \mathbf{H}_{T_g} = e^{-T_g\boldsymbol{\mathcal{L}}}$ yields

$$\begin{aligned}\boldsymbol{d}_\alpha - \mathcal{D}^\star &= -(\mathbf{I} - \mathbf{H}_T)\mathcal{D}^\star + \alpha\mathbf{H}_T(\mathbf{I} - \mathbf{H}_{T_g})\mathcal{D}^\star \\ &= -e^{-T\boldsymbol{\mathcal{L}}}\mathcal{D}^\star + \alpha(\mathbf{I} - e^{-T\boldsymbol{\mathcal{L}}})e^{-T_g\boldsymbol{\mathcal{L}}}\mathcal{D}^\star. \end{aligned} \tag{A.14}$$

For brevity, define

$$\begin{aligned}\boldsymbol{e}_T &= e^{-T\boldsymbol{\mathcal{L}}}\mathcal{D}^\star, \\ \boldsymbol{q}_{T,T_g} &= (\mathbf{I} - e^{-T\boldsymbol{\mathcal{L}}})e^{-T_g\boldsymbol{\mathcal{L}}}\mathcal{D}^\star, \\ A_{T,T_g} &= \langle \boldsymbol{e}_T, \boldsymbol{q}_{T,T_g}\rangle, \qquad\qquad C_{T,T_g} = \|\boldsymbol{q}_{T,T_g}\|^2.\end{aligned}$$

Then

$$R(\alpha) = R(0) - 2\alpha A_{T,T_g} + \alpha^2 C_{T,T_g}. \tag{A.15}$$

Let $\mathbf{E}_{\boldsymbol{\mathcal{L}}}$ be the spectral resolution of $\boldsymbol{\mathcal{L}}$ and define the finite positive measure

$$\nu_{\mathcal{D}^\star}(B) = \|\mathbf{E}_{\boldsymbol{\mathcal{L}}}(B)\mathcal{D}^\star\|_{\mathcal{H}_2}^2.$$

Functional calculus gives

$$\begin{aligned}A_{T,T_g} &= \int_{[0,\|\boldsymbol{\mathcal{L}}\|]} e^{-T\lambda}(1 - e^{-T\lambda})e^{-T_g\lambda}\, d\nu_{\mathcal{D}^\star}(\lambda), \\ C_{T,T_g} &= \int_{[0,\|\boldsymbol{\mathcal{L}}\|]} (1 - e^{-T\lambda})^2 e^{-2T_g\lambda}\, d\nu_{\mathcal{D}^\star}(\lambda).\end{aligned}$$

Both integrands vanish at $\lambda = 0$ and are strictly positive for every $\lambda > 0$. By the nonzero-predictor condition in Theorem 1, the ordinary-training predictor at time T is nonzero in $\mathcal{H}_2$. Lemma A.4 and spectral calculus therefore give

$$\begin{aligned}0 < \|\mathcal{D}_{\text{base}}\|_{\mathcal{H}_2}^2 &= \int_{[0,\|\boldsymbol{\mathcal{L}}\|]} (1 - e^{-T\lambda})^2\, d\nu_{\mathcal{D}^\star}(\lambda).\end{aligned}$$

Since $T > 0$, the integrand in this last display is positive exactly when $\lambda > 0$. Consequently, $\nu_{\mathcal{D}^\star}((0, \|\boldsymbol{\mathcal{L}}\|]) > 0$, and hence $A_{T,T_g} > 0$ and $C_{T,T_g} > 0$. The same spectral measure gives

$$\|\mathcal{D}_{\text{weak}} - \mathcal{D}^\star\|_{\mathcal{H}_2}^2 - R(0) = \int \left(e^{-2T_g\lambda} - e^{-2T\lambda}\right) d\nu_{\mathcal{D}^\star}(\lambda) > 0,$$

because $T_g < T$ and the integrand is positive on $\lambda > 0$. This proves the theorem's claim that the saved guide is strictly weaker. Expanding the squared norm in Eq. equation A.14 now gives the risk quadratic in Eq. equation A.15. Its unique minimizer is $\alpha^\star_{\mathrm{PM}} = A_{T,T_g}/C_{T,T_g} > 0$; completing the square gives $R(\alpha^\star_{\mathrm{PM}}) = R(0) - A^2_{T,T_g}/C_{T,T_g} < R(0)$. More explicitly,

$$R'(0) = -2A_{T,T_g} < 0, \qquad R(\alpha) < R(0) \iff 0 < \alpha < \frac{2A_{T,T_g}}{C_{T,T_g}}.$$

Define the improvement threshold used in Theorem 1 by

$$\alpha^\star := \frac{2A_{T,T_g}}{C_{T,T_g}} = 2\alpha^\star_{\mathrm{PM}} > 0.$$

Then every $0 < \alpha < \alpha^\star$ strictly improves the posterior-mean risk, proving Eq. equation 7. Here $\alpha^\star$ is the upper endpoint of the improvement interval; the risk-minimizing coefficient is $\alpha^\star_{\mathrm{PM}}$.

□

Meaning of the nonzero-predictor condition. For every $T > 0$, the spectral multiplier $1 - e^{-T\lambda}$ vanishes exactly at $\lambda = 0$. Therefore $\ker \mathbf{H}_T = \ker \boldsymbol{\mathcal{L}}$. With the inclusion $\mathbf{J}$ from Lemma A.4, $\boldsymbol{\mathcal{L}} = \mathbf{J}\mathbf{J}^*$ also gives

$$\langle \mathcal{D}^\star, \boldsymbol{\mathcal{L}}\mathcal{D}^\star \rangle_{\mathcal{H}_2} = \|\mathbf{J}^*\mathcal{D}^\star\|^2_{\mathcal{H}^d_{\mathsf{K}}}.$$

Thus the nonzero-predictor condition $\mathbf{H}_T\mathcal{D}^\star \neq \mathbf{0}$ in Theorem 1 is equivalent to the existence of an $\boldsymbol{f} \in \mathcal{H}^d_{\mathsf{K}}$ such that $\mathbb{E}[\boldsymbol{f}(\boldsymbol{x}_t)^\top \mathcal{D}^\star(\boldsymbol{x}_t)] \neq 0$. It only excludes the degenerate case in which the entire RKHS prediction space is orthogonal to the posterior mean. If this condition fails, both the ordinary and SWIFT population flows remain at zero, so strict improvement is impossible in this training model.

Proposition A.5 (Optimal coefficient and guide training time)**.** *Under the main theorem's assumptions, the posterior-mean-risk minimizer $\alpha^\star_{\mathrm{PM}}(T_g) = A_{T,T_g}/C_{T,T_g}$ is strictly increasing on $0 \le T_g < T$.*

Proof. For $\lambda > 0$, define

$$w_{T_g}(\lambda) = (1 - e^{-T\lambda})^2 e^{-2T_g\lambda},$$
$$a_{T_g}(\lambda) = \frac{e^{-T\lambda}}{(1 - e^{-T\lambda})e^{-T_g\lambda}}.$$

Then

$$\alpha^\star_{\mathrm{PM}}(T_g) = \frac{\int w_{T_g}(\lambda) a_{T_g}(\lambda)\, d\nu_{\mathcal{D}^\star}(\lambda)}{\int w_{T_g}(\lambda)\, d\nu_{\mathcal{D}^\star}(\lambda)} = \mathbb{E}_{\rho_{T_g}}[a_{T_g}(\lambda)],$$

where ρ_{T_g} is the probability measure proportional to $w_{T_g}\, d\nu_{\mathcal{D}^\star}$. Since $\boldsymbol{\mathcal{L}}$ is bounded, differentiation under the integral is valid. Using

$$\partial_{T_g} w_{T_g} = -2\lambda w_{T_g}, \qquad \partial_{T_g} a_{T_g} = \lambda a_{T_g},$$

we obtain

$$\begin{aligned}\frac{d}{dT_g}\alpha^\star_{\mathrm{PM}} &= 2\mathbb{E}_{\rho_{T_g}}[\lambda]\mathbb{E}_{\rho_{T_g}}[a_{T_g}] - \mathbb{E}_{\rho_{T_g}}[\lambda a_{T_g}] \\ &= \mathbb{E}_{\rho_{T_g}}[\lambda]\mathbb{E}_{\rho_{T_g}}[a_{T_g}] - \mathrm{Cov}_{\rho_{T_g}}(\lambda, a_{T_g}). \end{aligned} \tag{A.16}$$

For $T_g \le T$,

$$\frac{d}{d\lambda}\log a_{T_g}(\lambda) = -(T - T_g) - \frac{T}{e^{T\lambda} - 1} < 0.$$

Thus a_{T_g} is decreasing in λ. If λ' is an independent copy of λ under ρ_{T_g}, then

$$\mathrm{Cov}_{\rho_{T_g}}(\lambda, a_{T_g}) = \frac{1}{2}\mathbb{E}\big[(\lambda - \lambda')\big(a_{T_g}(\lambda) - a_{T_g}(\lambda')\big)\big] \le 0.$$

All expectations are finite: λ is bounded and $\mathbb{E}_{\rho_{T_g}}[a_{T_g}] = A_{T,T_g}/C_{T,T_g} < \infty$. The first term in the final line of Eq. equation A.16 is strictly positive because ρ_{T_g} is supported on $\lambda > 0$. Hence $\frac{d}{dT_g}\alpha^\star_{\mathrm{PM}} > 0$. □

A.6 GAUSSIAN KERNELS SATISFY THE NONZERO-PREDICTOR CONDITION

Consider the Gaussian/RBF kernel

$$\mathsf{K}_\ell(\boldsymbol{x}_t, \boldsymbol{x}'_t) = \exp\left(-\frac{\|\boldsymbol{x}_t - \boldsymbol{x}'_t\|_2^2}{2\ell^2}\right), \qquad \ell > 0. \tag{A.17}$$

Proposition A.6 (Injectivity of the Gaussian-kernel operator)**.** *For the mixture in Eq. equation 6, the marginal $P_{\boldsymbol{x}_t}$ has a smooth, strictly positive density on $\mathbb{R}^d$. The integral operator associated with Eq. equation A.17 is injective on $L^2(P_{\boldsymbol{x}_t}; \mathbb{R}^d)$. Moreover, $\mathcal{D}^\star \neq \mathbf{0}$. Therefore, for every $T > 0$, ordinary training produces a nonzero predictor in $\mathcal{H}_2$, as required by the nonzero-predictor condition of Theorem 1.*

Proof. Each conditional law $\boldsymbol{x}_t \mid I = k$ is Gaussian with covariance $\boldsymbol{\Sigma}_{\text{obs}} \succ \mathbf{0}$, so its density is smooth and strictly positive; the same is true of their positive mixture.

It suffices to prove injectivity for one scalar coordinate. Let $f \in L^2(P_{\boldsymbol{x}_t})$ and suppose $\boldsymbol{\mathcal{L}}_0 f = 0$. Then $\langle f, \boldsymbol{\mathcal{L}}_0 f\rangle_{L^2(P_{\boldsymbol{x}_t})} = 0$. The signed measure $d\eta(\boldsymbol{x}_t) = f(\boldsymbol{x}_t) dP_{\boldsymbol{x}_t}(\boldsymbol{x}_t)$ has finite total variation by Cauchy–Schwarz. The translation-invariant kernel energy admits the Fourier representation

$$\langle f, \boldsymbol{\mathcal{L}}_0 f\rangle = c_{d,\ell} \int_{\mathbb{R}^d} |\widehat{\eta}(\boldsymbol{\omega})|^2 e^{-\ell^2 \|\boldsymbol{\omega}\|_2^2/2}\, d\boldsymbol{\omega},$$

for a constant $c_{d,\ell} > 0$. The Gaussian Fourier multiplier is strictly positive everywhere, so zero energy implies that $\widehat{\eta} = 0$ Lebesgue-almost everywhere. The Fourier transform of a finite signed measure is continuous, hence it vanishes everywhere; uniqueness of Fourier transforms of finite measures gives $\eta = 0$. Since $d\eta = f\, dP_{\boldsymbol{x}_t}$, $f = 0$ $P_{\boldsymbol{x}_t}$-almost surely. Applying this coordinatewise proves injectivity.

Finally, the tower property and independence of the corruption noise give

$$\mathbb{E}[\mathcal{D}^\star(\boldsymbol{x}_t)\boldsymbol{x}_t^\top] = \mathbb{E}[\boldsymbol{x}_0 \boldsymbol{x}_t^\top] = \mathbb{E}[\boldsymbol{x}_0 \boldsymbol{x}_0^\top] \succeq \boldsymbol{\Sigma} \succ \mathbf{0}.$$

Thus $\mathcal{D}^\star$ cannot vanish in $\mathcal{H}_2$. Injectivity gives $\ker \boldsymbol{\mathcal{L}} = \{\mathbf{0}\}$. For every $T > 0$, spectral calculus implies

$$\ker \mathbf{H}_T = \ker(\mathbf{I} - e^{-T\boldsymbol{\mathcal{L}}}) = \ker \boldsymbol{\mathcal{L}} = \{\mathbf{0}\}.$$

Hence $\mathcal{D}_{\text{base}} = \mathbf{H}_T \mathcal{D}^\star \neq \mathbf{0}$ in $\mathcal{H}_2$, verifying the nonzero-predictor condition of Theorem 1. □

Proposition A.6 is useful because the RKHS can be universal: strict improvement does not rely on a deliberately underpowered function class. For every finite $T > 0$, exact fitting has not yet occurred in prediction space, leaving a nonempty beneficial interval of α.

A.7 REDUCED-RANK AND SPECTRALLY NESTED GUIDES

A narrower guide need not use exactly the same response as the main denoiser. The following construction gives an exact low-rank analogue.

Let $\mathbf{P}_g$ be an orthogonal projection on $\mathcal{H}_2$ satisfying

$$\mathbf{P}_g \boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}} \mathbf{P}_g.$$

For example, $\mathbf{P}_g$ may retain any subset of the eigenfunctions of $\boldsymbol{\mathcal{L}}$; if it retains finitely many, the guide response is finite rank. Define

$$\mathbf{H}_c = \mathbf{I} - e^{-T\boldsymbol{\mathcal{L}}}, \qquad \mathbf{H}_g = \mathbf{P}_g(\mathbf{I} - e^{-T_g \boldsymbol{\mathcal{L}}}). \tag{A.18}$$

Corollary A.7 (Reduced-rank guide)**.** *The operators in Eq. equation A.18 satisfy Eq. equation A.7. Therefore, whenever*

$$\left\langle \mathcal{D}^\star, e^{-T\boldsymbol{\mathcal{L}}}(\mathbf{I} - e^{-T\boldsymbol{\mathcal{L}}})\left[\mathbf{I} - \mathbf{P}_g(\mathbf{I} - e^{-T_g\boldsymbol{\mathcal{L}}})\right]\mathcal{D}^\star \right\rangle > 0,$$

there is a nonempty interval of positive coefficients for which the SWIFT *denoiser is strictly closer to $\mathcal{D}^\star$ than the control.*

Proof. Because $\mathbf{P}_g$ is an orthogonal projection commuting with $\boldsymbol{\mathcal{L}}$, $\mathbf{H}_g$ is self-adjoint, commutes with $\mathbf{H}_c$, and has eigenvalues either zero or $1 - e^{-T_g \lambda}$, all in $[0, 1]$. Apply Theorem A.3. □

This corollary is the rigorous counterpart of a reduced-width guide only when the narrower model corresponds to a nested spectral subspace. Arbitrary architectural width reduction need not preserve the eigenfunctions and is not covered without an additional approximation argument.

A.8 Exact label-affine response for finite-step training

We next separate a second exact statement from the population theorem. It applies to finite data and a finite number of optimization steps, provided the features are fixed. Let $\boldsymbol{x}_{t,1}, \ldots, \boldsymbol{x}_{t,n}$ be training inputs, let $\boldsymbol{\Phi} \in \mathbb{R}^{n \times p}$ be a fixed design matrix, and train the vector-valued linear predictor $\boldsymbol{\Phi}\boldsymbol{\Theta}$ with full-batch gradient descent:

$$\boldsymbol{\Theta}_{s+1} = \boldsymbol{\Theta}_s - \frac{\gamma_s}{n} \boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\boldsymbol{\Theta}_s - \mathbf{Y}), \qquad s = 0, \ldots, S-1,$$

where $\mathbf{Y} \in \mathbb{R}^{n \times d}$ is the label matrix. The initialization $\boldsymbol{\Theta}_0$ and step sizes γ_s are fixed independently of the labels and held the same across target choices. No convergence assumption is made.

Proposition A.8 (Finite-step label affinity)**.** *For every $S \geq 0$, there exist $\mathbf{U}_S \in \mathbb{R}^{p \times d}$ and a linear operator $\mathbf{A}_S : \mathbb{R}^{n \times d} \to \mathbb{R}^{p \times d}$, both independent of $\mathbf{Y}$, such that*

$$\boldsymbol{\Theta}_S = \mathbf{U}_S + \mathbf{A}_S(\mathbf{Y}). \tag{A.19}$$

Consequently, predictions under any fixed bounded linear evaluation map into a Hilbert prediction space are affine functions of the labels.

Proof. Set $\mathbf{Q}_s = \mathbf{I}_p - \gamma_s \boldsymbol{\Phi}^\top \boldsymbol{\Phi}/n$. Then

$$\boldsymbol{\Theta}_{s+1} = \mathbf{Q}_s \boldsymbol{\Theta}_s + \frac{\gamma_s}{n} \boldsymbol{\Phi}^\top \mathbf{Y}.$$

The assertion holds at $s = 0$ with $\mathbf{U}_0 = \boldsymbol{\Theta}_0$ and $\mathbf{A}_0 = \mathbf{0}$. If it holds at step s, it holds at step $s + 1$ with

$$\mathbf{U}_{s+1} = \mathbf{Q}_s \mathbf{U}_s, \qquad \mathbf{A}_{s+1}(\mathbf{Y}) = \mathbf{Q}_s \mathbf{A}_s(\mathbf{Y}) + \frac{\gamma_s}{n} \boldsymbol{\Phi}^\top \mathbf{Y}.$$

Induction proves Eq. equation A.19. Evaluation is linear in $\boldsymbol{\Theta}_S$, so it preserves affinity. □

The same proof covers mini-batch schedules fixed in advance, fixed linear preconditioners, weight decay, and momentum after augmenting the state. It does not cover label-dependent feature learning without further assumptions.

A.9 Finite-sample bias–variance extension

This extension is exact under the conditional-centering assumption below. That assumption is natural when the guide is trained independently, but it is not automatic when the guide is trained on the same clean labels.

Let $\mathcal{F}$ be a sigma-algebra with respect to which the noisy training inputs, the response map, and the guide are measurable. Let $\mathbf{M}$ be the matrix of stacked posterior means and write the clean labels as

$$\mathbf{M} = \big(\mathcal{D}^\star(\boldsymbol{x}_{t,1}), \ldots, \mathcal{D}^\star(\boldsymbol{x}_{t,n})\big)^\top \in \mathbb{R}^{n \times d}, \qquad \mathbf{X}_0 = \mathbf{M} + \mathbf{E}, \qquad \mathbb{E}[\mathbf{E} \mid \mathcal{F}] = \mathbf{0}.$$

For example, the displayed conditional-centering assumption holds for i.i.d. training pairs when the guide is trained on an independent dataset and the response map and initialization depend only on the noisy inputs and independent algorithmic randomness, with $\mathcal{F}$ generated by these noisy inputs, auxiliary data, and randomness. It is not automatic for a guide trained on the same clean labels. With $\mathbf{Y}_g$ the stacked guide predictions and $\boldsymbol{\Delta}_g = \mathbf{M} - \mathbf{Y}_g$, the SWIFT labels are

$$\mathbf{Y}_\alpha = (1 + \alpha)\mathbf{X}_0 - \alpha \mathbf{Y}_g = \mathbf{M} + \alpha \boldsymbol{\Delta}_g + (1 + \alpha)\mathbf{E}.$$

Let the trained predictor in a Hilbert space $\mathcal{Y}$ have the exact label-affine form $\boldsymbol{a} + \boldsymbol{\mathcal{G}}\mathbf{Y}$, where $\boldsymbol{a}$ and the bounded linear operator $\boldsymbol{\mathcal{G}}$ are $\mathcal{F}$-measurable. Let $\boldsymbol{d}^\star_{\mathrm{ev}} \in \mathcal{Y}$ be a fixed (or $\mathcal{F}$-measurable)

posterior-mean target on the evaluation distribution. Assume V defined below is finite, and set

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{d}^\star_{\mathrm{ev}} - (\boldsymbol{a} + \boldsymbol{\mathcal{G}}\mathbf{M}), & \boldsymbol{q} &= \boldsymbol{\mathcal{G}}\boldsymbol{\Delta}_g, \\ \boldsymbol{\eta} &= \boldsymbol{\mathcal{G}}\mathbf{E}, & A &= \langle \boldsymbol{e}, \boldsymbol{q} \rangle, \\ C &= \|\boldsymbol{q}\|^2, & V &= \mathbb{E}[\|\boldsymbol{\eta}\|^2 \mid \mathcal{F}]. \end{aligned}$$

Then $\mathbb{E}[\boldsymbol{\eta} \mid \mathcal{F}] = \mathbf{0}$ and, exactly,

$$\widehat{\boldsymbol{d}}_\alpha - \boldsymbol{d}^\star_{\mathrm{ev}} = -\boldsymbol{e} + \alpha \boldsymbol{q} + (1+\alpha)\boldsymbol{\eta}.$$

Proposition A.9 (Finite-sample variance tax)**.** *Under the preceding conditional label-affine model, define*

$$R_{\mathcal{F}}(\alpha) := \mathbb{E}[\|\widehat{\boldsymbol{d}}_\alpha - \boldsymbol{d}^\star_{\mathrm{ev}}\|^2 \mid \mathcal{F}].$$

Then

$$\mathbb{E}[\|\widehat{\boldsymbol{d}}_\alpha - \boldsymbol{d}^\star_{\mathrm{ev}}\|^2 \mid \mathcal{F}] = \|\boldsymbol{e} - \alpha \boldsymbol{q}\|^2 + (1+\alpha)^2 V, \tag{A.20}$$

$$R_{\mathcal{F}}(\alpha) - R_{\mathcal{F}}(0) = -2(A - V)\alpha + (C + V)\alpha^2. \tag{A.21}$$

If $C + V > 0$, the risk-minimizing coefficient over $\alpha \geq 0$ is

$$\alpha^\star_{\mathrm{FS}} = \left[\frac{A - V}{C + V}\right]_+, \qquad [u]_+ = \max\{u, 0\}. \tag{A.22}$$

There exists a strictly improving positive coefficient if and only if $A > V$; in that case the complete improvement interval is

$$0 < \alpha < \frac{2(A - V)}{C + V},$$

and the maximal conditional risk reduction is $(A - V)^2/(C + V)$. If $C + V = 0$, then $\boldsymbol{q} = \mathbf{0}$, $\boldsymbol{\eta} = \mathbf{0}$ almost surely, and the risk is constant in α.

Proof. Conditionally on $\mathcal{F}$, $\boldsymbol{e}$ and $\boldsymbol{q}$ are fixed and $\mathbb{E}[\boldsymbol{\eta} \mid \mathcal{F}] = \mathbf{0}$. Thus the cross term between $-\boldsymbol{e} + \alpha\boldsymbol{q}$ and $(1+\alpha)\boldsymbol{\eta}$ has conditional expectation zero, proving Eq. equation A.20. Expanding and subtracting the value at $\alpha = 0$ gives Eq. equation A.21. If $C + V > 0$, completing the square gives the unconstrained minimizer $(A - V)/(C + V)$ and hence Eq. equation A.22. The quadratic difference has roots 0 and $2(A - V)/(C + V)$ when $A > V$, yielding the interval and maximum reduction. Finally, $C + V = 0$ implies both nonnegative terms vanish. □

For a scale-free geometric interpretation, when $\boldsymbol{e}$ and $\boldsymbol{q}$ are nonzero set

$$\rho = \frac{\langle \boldsymbol{e}, \boldsymbol{q} \rangle}{\|\boldsymbol{e}\|\,\|\boldsymbol{q}\|}, \qquad \chi = \frac{\|\boldsymbol{q}\|}{\|\boldsymbol{e}\|}, \qquad \nu = \frac{V}{\|\boldsymbol{e}\|^2}.$$

Then improvement is possible exactly when $\rho\chi > \nu$, and

$$\alpha^\star_{\mathrm{FS}} = \left[\frac{\rho\chi - \nu}{\chi^2 + \nu}\right]_+.$$

Thus alignment must overcome the variance amplified by $1+\alpha$. Setting $V = 0$ recovers the population quadratic algebra; it recovers Theorem 1 only when $\boldsymbol{e}$ and $\boldsymbol{q}$ are additionally the population RKHS response quantities defined in its proof.

A.10 Local response of a smooth nonlinear optimizer

The finite-sample proposition is exact because the training response is label-affine. For a general optimizer one obtains only a local response unless the remainder is controlled. Let

$$\boldsymbol{\theta}_{s+1} = \mathbf{F}_s(\boldsymbol{\theta}_s, \mathbf{Y}_\alpha), \qquad \mathbf{Y}_\alpha = \mathbf{Y}_0 + \alpha\boldsymbol{\Delta}, \tag{A.23}$$

where each $\mathbf{F}_s$ and the vector-valued evaluation map $\boldsymbol{h}$ are continuously differentiable near the trajectory at $\alpha = 0$, and $\boldsymbol{\theta}_0$ is independent of α.

Proposition A.10 (First-order optimizer response). *Define* $\boldsymbol{j}_s = d\boldsymbol{\theta}_s/d\alpha|_{\alpha=0}$. *Then* $\boldsymbol{j}_0 = \mathbf{0}$ *and*

$$\boldsymbol{j}_{s+1} = \partial_{\boldsymbol{\theta}}\mathbf{F}_s(\boldsymbol{\theta}_s, \mathbf{Y}_0)\boldsymbol{j}_s + \partial_{\mathbf{Y}}\mathbf{F}_s(\boldsymbol{\theta}_s, \mathbf{Y}_0)[\boldsymbol{\Delta}], \tag{A.24}$$

where the right side is evaluated along the $\alpha = 0$ *trajectory. Moreover,*

$$\boldsymbol{h}(\boldsymbol{\theta}_S(\alpha)) = \boldsymbol{h}(\boldsymbol{\theta}_S(0)) + \alpha\, \partial_{\boldsymbol{\theta}}\boldsymbol{h}(\boldsymbol{\theta}_S(0))\boldsymbol{j}_S + o(\alpha). \tag{A.25}$$

If the relevant derivatives are locally Lipschitz, the remainder is $O(\alpha^2)$.

Proof. Differentiate Eq. equation A.23 at $\alpha = 0$ and apply the chain rule to obtain Eq. equation A.24. A first-order Taylor expansion of $\boldsymbol{h}$ gives Eq. equation A.25; locally Lipschitz derivatives give the quadratic remainder. □

The proposition does not imply that propagated label noise has conditional mean zero: the base trajectory and its Jacobians may depend on that noise. Therefore Proposition A.9 does not apply to general feature learning unless label affinity, suitable independence, or an explicit remainder-and-dependence bound is established.

A.11 Stability beyond the exact kernel regime

The next result converts an exact improvement into a margin condition.

Proposition A.11 (Risk-margin stability). *Let* $\boldsymbol{d}_c, \boldsymbol{d}_\alpha$ *be ideal predictors satisfying*

$$\Gamma(\alpha) := \|\boldsymbol{d}_c - \mathcal{D}^\star\|^2 - \|\boldsymbol{d}_\alpha - \mathcal{D}^\star\|^2 > 0.$$

Let $\widetilde{\boldsymbol{d}}_c, \widetilde{\boldsymbol{d}}_\alpha$ *satisfy*

$$\|\widetilde{\boldsymbol{d}}_c - \boldsymbol{d}_c\| \le \varepsilon_0, \qquad \|\widetilde{\boldsymbol{d}}_\alpha - \boldsymbol{d}_\alpha\| \le \varepsilon_\alpha.$$

Set $B_0 = \|\boldsymbol{d}_c - \mathcal{D}^\star\|$ *and* $B_\alpha = \|\boldsymbol{d}_\alpha - \mathcal{D}^\star\|$. *If*

$$\Gamma(\alpha) > 2B_0\varepsilon_0 + 2B_\alpha\varepsilon_\alpha + \varepsilon_\alpha^2, \tag{A.26}$$

then

$$\|\widetilde{\boldsymbol{d}}_\alpha - \mathcal{D}^\star\|^2 < \|\widetilde{\boldsymbol{d}}_c - \mathcal{D}^\star\|^2.$$

If both errors are at most ε *and* $B_0, B_\alpha \le B$, *it suffices that* $\Gamma(\alpha) > 4B\varepsilon + \varepsilon^2$.

Proof. The two perturbation bounds are

$$\|\widetilde{\boldsymbol{d}}_\alpha - \mathcal{D}^\star\|^2 \le \|\boldsymbol{d}_\alpha - \mathcal{D}^\star\|^2 + 2B_\alpha\varepsilon_\alpha + \varepsilon_\alpha^2,$$
$$\|\widetilde{\boldsymbol{d}}_c - \mathcal{D}^\star\|^2 \ge \|\boldsymbol{d}_c - \mathcal{D}^\star\|^2 - 2B_0\varepsilon_0.$$

Subtracting and using Eq. equation A.26 proves the claim. □

This result states what an approximation theorem must deliver; it does not itself establish kernel closeness for a feature-learning diffusion network.

A.12 Scope of the guarantee

What is exact. The shared-covariance mixture posterior, the commuting-response theorem, RKHS population gradient flow, Gaussian-kernel injectivity, and the spectrally nested reduced-rank guide are exact. The finite-step and finite-sample results are also exact under their stated fixed-feature, label-affinity, and conditional-centering assumptions. The RKHS may be infinite dimensional and universal; no assumption that $\mathcal{D}^\star$ lies outside the function class is needed.

Role of the data model and training budget. The Gaussian-mixture assumption gives an explicit posterior mean and finite second moments. The shared covariance is used in its displayed posterior formula; neither shared covariance nor Gaussianity is required for the RKHS improvement argument under the stated training dynamics, square integrability, and nonzero-predictor condition. No low-rank restriction is imposed on the data. The main theorem compares the baseline and SWIFT after

the same main-model optimization time T. Constructing the frozen guide requires the additional ordinary training time T_g; the theorem does not establish an advantage at equal total compute, nor does population flow account for guide forward-pass costs. Those costs are part of the experimental compute comparison.

What is not claimed. An arbitrary end-to-end neural network does not generally induce a fixed, self-adjoint, commuting response operator. Without a restriction on training dynamics, no distributional assumption can guarantee strict improvement: a control that reaches $\mathcal{D}^\star$ already has zero posterior-mean risk. Extension to feature learning therefore requires either an explicit response approximation bound, used through Proposition A.11, or a separate nonlinear-training theorem. Proposition A.10 is a local sensitivity identity, not such a global guarantee.

Posterior-mean risk versus sample quality. The theorem compares $L^2(P_{\boldsymbol{x}_t})$ distance to the one-step posterior mean. It does not identify this quantity with FID or with the error of an iterative diffusion sampler. Agreement between the coefficient predicted here and the coefficient minimizing FID is empirical evidence for the mechanism, not a consequence of the theorem.

B Additional Comparisons with Diffusion Guidance Methods

This section expands the comparison in Section 3. The main distinction is whether the auxiliary prediction used to construct the training correction is fixed or evolves together with the model being optimized. Several recent methods can be written in a form resembling target extrapolation at a single optimization step, but their effective weak prediction depends on the current model parameters. In SWIFT, the weak denoiser is instead trained beforehand and frozen throughout the entire main training run.

Sampling-time guidance. Autoguidance (AG) extrapolates a trained denoiser away from a weaker denoiser during sampling (Karras et al., 2024),

$$(1+s)\mathcal{D}_{\text{base}} - s\mathcal{D}_{\text{weak}},$$

where s is an inference-time guidance scale. AG provides the closest motivation for SWIFT: both use a weak model to define an extrapolation direction. The difference is when it is used. AG keeps the weak model at inference, whereas SWIFT uses it only to construct training targets and discards it afterward.

Fixed versus jointly evolving references. Training-time guidance methods such as MG, SGG, IG, and GFT use auxiliary predictions produced by the current model or by jointly optimized components (Tang et al., 2025; Yuan et al., 2026; Zhou et al., 2026; Chen et al., 2025). Their effective weak prediction can therefore be written schematically as

$$\mathcal{D}_{\text{aux}}^{(k)} = \mathcal{D}_{\text{aux}}(\theta_k),$$

and generally changes across optimization steps. This remains true when stop-gradient is used: detaching an auxiliary prediction within one update does not make it fixed across training.

In SWIFT, the guide is trained separately with the ordinary denoising objective and frozen before the main run. Thus $\mathcal{D}_{\text{weak}}(\boldsymbol{x}_t, t)$ is independent of the evolving main-model parameters. This distinction matters even when another method can be algebraically rewritten into an extrapolated-target form: a target containing $\mathcal{D}_{\text{aux}}(\theta_k)$ is still a moving target, whereas the SWIFT reference is fixed.

Algebraic relation to Guidance-Free Training. For example, GFT uses a trainable prediction $\boldsymbol{f}$, a detached unconditional prediction $\boldsymbol{u}$, and regression label $\boldsymbol{q}$ (Chen et al., 2025). For fixed $0 < \beta \leq 1$,

$$\|\beta\boldsymbol{f} + (1-\beta)\boldsymbol{u} - \boldsymbol{q}\|_2^2 = \beta^2 \left\| \boldsymbol{f} - \left[\boldsymbol{q} + \frac{1-\beta}{\beta}(\boldsymbol{q}-\boldsymbol{u})\right] \right\|_2^2.$$

The bracketed term therefore resembles an extrapolated target with $\alpha = (1-\beta)/\beta$. However, $\boldsymbol{u}$ is recomputed from the evolving model, so this is a pointwise algebraic correspondence rather than an equivalence of training dynamics.

Table 2: **Comparison with related training-time guidance methods.** "Joint" indicates that the auxiliary prediction is produced by the model being optimized, or by a jointly trained branch, and therefore changes during main training.

Method	Auxiliary prediction	Joint?	Fixed during training?	Needed at inference?
MG (Tang et al., 2025)	conditional / unconditional	✓	×	×
SGG (Yuan et al., 2026)	jointly generated inferior prediction	✓	×	×
IG (Zhou et al., 2026)	intermediate prediction	✓	×	✓†
GFT (Chen et al., 2025)	detached unconditional prediction	✓	×	×
SWIFT	pretrained weak denoiser	×	✓	×

†IG obtains its auxiliary prediction from an intermediate layer of the same backbone rather than from a separate network.

Why keep the guide frozen? Freezing the guide avoids jointly optimizing the model that produces the auxiliary prediction. More importantly, this choice is also supported empirically: on CIFAR-10, jointly training the weak guide does not improve performance and is worse than keeping the guide frozen; adaptive and online variants are similarly neutral or harmful (Section F.4). Thus the fixed reference is not merely an implementation simplification, but an important part of the SWIFT design.

B.1 Controlled Comparison on a Gaussian Mixture

We further compare SWIFT, MG, GFT, and IG in a controlled Gaussian-mixture setting where the optimal denoiser is known in closed form. All methods use the same model capacity and training length, and each learned output is mapped back to its implied estimate of the optimal denoiser before error is measured.

C Distance to the optimal denoiser on a Gaussian mixture

This experiment asks a question that image data cannot answer: at a fixed capacity and training budget, which training objective lets a network learn the most about the posterior mean $\mathcal{D}^\star$? On images $\mathcal{D}^\star$ is unknown; on a Gaussian mixture it is known in closed form, so the answer can be measured exactly. We train the same small network, on the same data and for the same number of steps, with the target of each method, convert each trained network into the estimate of $\mathcal{D}^\star$ that its objective implies, and measure the distance of that estimate to $\mathcal{D}^\star$.

Converting each network to an estimate of $\mathcal{D}^\star$. A method that changes the regression target does not train its network to approximate $\mathcal{D}^\star$. The SWIFT network, for example, is trained toward $\boldsymbol{g}_\alpha = \mathcal{D}^\star + \alpha(\mathcal{D}^\star - \mathcal{D}_{\text{weak}})$ (Lemma A.1), so comparing it with $\mathcal{D}^\star$ directly would penalize it for learning what it was asked to learn. Each objective, however, fixes how its network relates to $\mathcal{D}^\star$, and inverting that relation gives the estimate of $\mathcal{D}^\star$ the method implies. For SWIFT this is $\hat{\mathcal{D}} = (\mathcal{D}_\theta + \alpha\,\mathcal{D}_{\text{weak}})/(1+\alpha)$, which equals $\mathcal{D}^\star$ exactly when $\mathcal{D}_\theta = \boldsymbol{g}_\alpha$. The conversion uses only the trained networks, never $\mathcal{D}^\star$. Table 3 lists the target and the implied estimate of every method. Sampling-time methods such as CFG and autoguidance leave the trained network unchanged, so their implied estimate is the baseline's and they are not part of this comparison.

Table 3: **Compared objectives and the estimate of $\mathcal{D}^\star$ each one implies.** sg denotes stop-gradient. $\mathcal{D}_{\text{c}}$ is the baseline network on class A and $\mathcal{D}_{\text{u}}$ an ordinary denoiser on A+B, both fixed; MG uses these fixed references in place of its moving-average self-reference.

Arm	Training target of $\mathcal{D}_\theta$	Implied estimate of $\mathcal{D}^\star$	Swept
baseline	$\boldsymbol{x}_0$	$\mathcal{D}_\theta$	–
SWIFT (ours)	$(1+\alpha)\boldsymbol{x}_0 - \alpha\,\mathcal{D}_{\text{weak}}(\boldsymbol{x}_t)$	$(\mathcal{D}_\theta + \alpha\,\mathcal{D}_{\text{weak}})/(1+\alpha)$	$\alpha \in \{0.3, 0.5, 0.7, 1.0\}$
MG (Tang et al., 2025)	$\boldsymbol{x}_0 + w(\mathcal{D}_{\text{c}} - \mathcal{D}_{\text{u}})$	$\mathcal{D}_\theta - w(\mathcal{D}_{\text{c}} - \mathcal{D}_{\text{u}})$	$w \in \{0.25, 0.5, 1\}$
GFT (Chen et al., 2025)	minimize $\|\beta\mathcal{D}_\theta + (1-\beta)\,\text{sg}[\mathcal{D}_{\text{u}}] - \boldsymbol{x}_0\|^2$, $\beta = 1/w$	$\beta\mathcal{D}_\theta + (1-\beta)\mathcal{D}_{\text{u}}$	$w \in \{1.25, 1.5, 2\}$
IG (Zhou et al., 2026)	$\|\mathcal{D}_{\text{f}} - \boldsymbol{x}_0\|^2 + \gamma\|\mathcal{D}_{\text{i}} - \boldsymbol{x}_0\|^2$, $\mathcal{D}_{\text{i}}$ after the first hidden layer	$\mathcal{D}_{\text{f}}$ (target unchanged)	$\gamma = 0.5$

Setup. Data are drawn from a mixture of eight equally weighted Gaussians, $\boldsymbol{x}_0 \sim \sum_{j=1}^{8} w_j\,\mathcal{N}(\boldsymbol{\mu}_j, s^2\boldsymbol{I}_d)$, and noised as $\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma\boldsymbol{\epsilon}$, so the posterior mean is the shared-covariance

case of Lemma A.2,

$$\mathcal{D}^\star(\boldsymbol{x}_t,\sigma)=\sum_{j=1}^{8} r_j(\boldsymbol{x}_t)\,\frac{s^2\boldsymbol{x}_t+\sigma^2\boldsymbol{\mu}_j}{s^2+\sigma^2},\qquad r_j(\boldsymbol{x}_t)\propto w_j\,\mathcal{N}\big(\boldsymbol{x}_t;\boldsymbol{\mu}_j,(s^2+\sigma^2)\boldsymbol{I}_d\big). \tag{A.27}$$

As in the tree example, class A (the first four components) is the conditional distribution, on which every network is trained and evaluated, and the full mixture A+B is the unconditional distribution that MG and GFT need. The main network is an MLP of width 64 that cannot represent $\mathcal{D}^\star$ exactly; with unlimited capacity every objective would reach its own optimum and there would be nothing to compare. All networks share the initialization, the data stream, and the training budget, and differ only in the target (Table 4). The auxiliary models are matched in capacity: the SWIFT guide and the unconditional reference $\mathcal{D}_{\mathrm{u}}$ of MG and GFT are both width-16 networks trained for 4,000 steps, about 10% of the main network's parameters and a fifth of its steps. A full-capacity $\mathcal{D}_{\mathrm{u}}$ would make the GFT estimate an average of two full-size networks, which is closer to $\mathcal{D}^\star$ because of the extra capacity rather than the objective. The guide is a weak model: at $d{=}16$ its distance to $\mathcal{D}^\star$ is 3.3–7.1 times that of the ordinarily trained main network, so SWIFT does not gain from learning from a better model.

Table 4: **Gaussian-mixture setup.** All networks share everything except the training target.

components	8, equal weights; class A = components 1–4
means $\boldsymbol{\mu}_j$	$\mathcal{N}(\mathbf{0}, 4^2\boldsymbol{I}_d)$, redrawn for every seed
component standard deviation s	0.5
dimension d	16 (main results) and 8
training noise levels	$\log\sigma\sim\mathcal{U}(-3,1)$
main network	MLP, 3 hidden layers of width 64, input $[\boldsymbol{x}_t,\log\sigma]$
training	Adam, learning rate 2×10^{-3}, batch 512, 20,000 steps
$\mathcal{D}_{\mathrm{weak}}$ and $\mathcal{D}_{\mathrm{u}}$	MLP of width 16, 4,000 steps, target $\boldsymbol{x}_0$ (about 10% of the parameters)

Evaluation. For each noise level $\sigma\in\{0.05,0.1,0.2,0.5,1.0\}$ we draw 20,000 fresh samples of class A, noise them, and report the mean Euclidean distance $\frac{1}{N}\sum_i\|\hat{\mathcal{D}}(\boldsymbol{x}_{t,i})-\mathcal{D}^\star(\boldsymbol{x}_{t,i})\|_2$. Each seed redraws the mixture means and retrains every network, so seeds are independent replicates of the whole comparison: 40 at $d{=}16$ and 5 at $d{=}8$. At these dimensions the width-64 network is limited but adequate: the ordinarily trained baseline is within 1.94% ($d{=}16$) and 1.73% ($d{=}8$) of $\mathcal{D}^\star$ in relative error at $\sigma{=}0.05$.

Table 5: **Distance of each implied estimate to $\mathcal{D}^\star$** on the Gaussian mixture (capacity-matched, mean over seeds; lower is better, the best entry of each row in bold). Top: $d{=}16$, 40 seeds. Bottom: $d{=}8$, 5 seeds.

		SWIFT, α				MG, w			GFT, w			IG
σ	baseline	0.3	0.5	0.7	1.0	0.25	0.5	1	1.25	1.5	2	$\gamma{=}0.5$
$d=16$, 40 seeds												
0.05	0.3100	0.2855	0.2714	0.2611	**0.2566**	0.3179	0.3412	0.4011	0.2858	0.2699	0.2617	0.3418
0.1	0.2600	0.2408	0.2290	0.2209	**0.2175**	0.2685	0.2849	0.3300	0.2408	0.2282	0.2203	0.2885
0.2	0.2311	0.2145	0.2069	0.2004	**0.1952**	0.2388	0.2539	0.2936	0.2143	0.2045	0.1972	0.2522
0.5	0.2125	0.2008	0.1958	0.1914	**0.1865**	0.2216	0.2406	0.2888	0.2004	0.1959	0.1904	0.2258
1.0	0.2272	0.2136	0.2086	0.2038	**0.1988**	0.2358	0.2590	0.3240	0.2162	0.2128	0.2075	0.2312
$d=8$, 5 seeds												
0.05	0.1853	0.1676	0.1618	0.1538	**0.1450**	0.1913	0.2039	0.2269	0.1729	0.1659	0.1544	0.1976
0.1	0.1574	0.1455	0.1373	0.1302	**0.1202**	0.1659	0.1730	0.1826	0.1488	0.1411	0.1264	0.1657
0.2	0.1378	0.1310	0.1246	0.1192	**0.1120**	0.1491	0.1519	0.1667	0.1335	0.1285	0.1195	0.1468
0.5	0.1154	0.1144	0.1101	0.1055	**0.1018**	0.1246	0.1310	0.1568	0.1141	0.1129	0.1095	0.1301
1.0	0.1232	**0.1175**	0.1183	0.1184	0.1189	0.1356	0.1479	0.1906	0.1202	0.1223	0.1257	0.1330

Results. Table 5 reports the distances. At $d{=}16$, SWIFT with $\alpha{=}1$ gives the estimate closest to $\mathcal{D}^\star$ at every noise level: 12–17% closer than the baseline, and 0.9–3.8 percentage points ahead of the best alternative, GFT with $w{=}2$. The gain grows steadily with α over the scanned range $[0.3,1.0]$ at every noise level. MG and IG do not improve on the baseline at any setting. The result holds seed by seed: with each method at its best setting, SWIFT is closest to $\mathcal{D}^\star$ in 141 of the 200 (seed, noise level)

cells, against 58 for GFT, 1 for IG, and none for MG, and it improves on the baseline on all 40 seeds (mean 16.1%, range 7.8–28.6%). At d=8, SWIFT is again closest at every noise level and in 21 of the 25 cells, 3.9–6.6 percentage points ahead of GFT for $\sigma \leq 0.5$.

D EXPERIMENTAL SETUP

This section specifies the training configurations (Section D.1), sampling and evaluation protocol (Section D.2), and compute accounting (Section D.3) used throughout our experiments. Unless stated otherwise, SWIFT changes only the regression target; the carrier architecture, optimizer, training schedule, sampler, and evaluation pipeline are left unchanged.

D.1 TRAINING CONFIGURATIONS

CIFAR-10. We use the EDM configuration of Karras et al. (2022) without modification other than the regression target: a class-conditional DDPM++ model at 32×32 resolution with EDM preconditioning and loss weighting, batch size 512, fp32 training, and a 200k-kimg training horizon. The main denoiser uses channel base 128 (`cb128`). Guide capacity is specified by channel base (`cb16–cb128`), with `cb64` denoting the default half-width guide. NCSN++ and ADM-style guide variants use the corresponding channel bases; their full results are reported in Section F.3.

The guide is loaded from a frozen checkpoint, and α is implemented as a single scalar coefficient. No other training code path differs from the baseline. The $\alpha = 0$ arm (`D0`) is therefore the matched control used throughout the CIFAR-10 experiments, with the same seed, training schedule, dataset copy, hardware, and checkpoint lattice as the corresponding SWIFT runs.

ImageNet-256, DiffusionBench. We follow the released DiffusionBench configuration (Leng et al., 2026), using a representation-autoencoder tokenizer (Zheng et al., 2026) over frozen DINOv2-B features (Oquab et al., 2024) and a 615M-parameter decoupled diffusion transformer (Wang et al., 2025). The model is trained class-conditionally with global batch size 1024 for 80 epochs, corresponding to 1251 optimizer steps per epoch, and is evaluated every five epochs.

Unless stated otherwise, all ImageNet breadth experiments use a single SWIFT recipe: $\alpha = 0.4$ and a frozen half-width guide trained for 40 epochs. The guide has 155M parameters and approximately one quarter of the main denoiser's forward-pass FLOPs. The short-guide ablation uses the same guide architecture frozen at an earlier checkpoint. CFG is used only at sampling time and is tuned independently for the baseline and SWIFT.

The DiffusionBench carrier predicts velocity rather than $\boldsymbol{x}_0$. Because the velocity parameterization is affine in $\boldsymbol{x}_0$ for fixed $\boldsymbol{x}_t$, we apply SWIFT directly in the carrier parameterization:

$$\boldsymbol{v}_{\text{SWIFT}} = (1+\alpha)\boldsymbol{v} - \alpha\boldsymbol{v}_{\text{g}},$$

where $\boldsymbol{v}_{\text{g}}$ is the frozen guide prediction. This is equivalent to extrapolating the corresponding clean-sample target and then applying the carrier's parameterization transform.

The main DiffusionBench sweep is trained on $4\times$B200 GPUs. Longer-training experiments use the same optimization configuration and are run on $8\times$H100 GPUs where noted.

ImageNet-256, RAEv2. We use the released DINOv3-L configuration of Singh et al. (2026) (`dinov3l-k7`) together with its native internal-guidance implementation. The SWIFT guide is the carrier's `GUIDE-w05` weak-model configuration, loaded from its epoch-40 checkpoint and frozen before main-model training. Unless stated otherwise, we use $\alpha = 0.4$.

The RAEv2 guide uses hidden widths 720 and 1024, depths 28 and 2, and 10 and 8 attention heads, respectively, relative to the corresponding main-model dimensions. Here `w05` denotes approximately half the parameter count of the carrier, rather than exactly half of every hidden width. All other architectural and optimization settings follow the released RAEv2 configuration.

Table 6: **Sampling and evaluation settings.** Each experiment retains the carrier's original sampling configuration; SWIFT changes none of these settings.

	CIFAR-10 (EDM)	ImageNet-256 (DiffusionBench)	ImageNet-256 (RAEv2)
resolution / space	32×32, pixels	256×256, tokenizer latents	256×256, DINOv3-L latents
sampler	EDM deterministic (Heun)	Euler	ODE
NFE	35 (18 steps)	50	50
guidance	none	CFG, tuned independently	internal guidance, tuned independently
samples / evaluation	50k	50k	50k
reference statistics	CIFAR-10 training set	ImageNet-256 training set	ImageNet-256 training set
metrics	FID	FID, sFID, IS, precision, recall, FD_r, MIND	FID, sFID, IS, precision, recall, FD_r, MIND

D.2 SAMPLING AND EVALUATION PROTOCOL

Sampler configuration. Sampling follows each carrier's released evaluation configuration. CIFAR-10 uses the deterministic EDM Heun sampler. DiffusionBench uses a 50-step Euler sampler, while RAEv2 uses its released 50-step ODE sampler.

RAEv2 internal guidance constructs its auxiliary prediction using a truncated version of the same backbone, with the weak branch using the first 8 of 28 blocks (`base_model_depth=8`). This can be viewed as an autoguidance-style construction in which the weak prediction is obtained from a truncated shared backbone rather than from a separately trained weak network.

For unguided SWIFT evaluation, the training guide is removed entirely. Our evaluation script sets `self_guide.k=0` and clears the guide list before checkpoint loading, so the frozen weak denoiser used to construct training targets is not instantiated during sampling.

Guidance tuning. Whenever inference-time guidance is reported, the guidance strength is tuned independently for the baseline and SWIFT. The same sampling procedure, number of generated samples, and evaluation pipeline are used on both sides. This avoids fixing the guidance scale to the optimum of one model and applying it unchanged to the other.

Checkpoint protocol. Matched breadth comparisons use the same training endpoint for the baseline and SWIFT. Longer-training experiments evaluate both models throughout their training trajectories using the same checkpoint schedule. Whenever a result is selected over multiple checkpoints or guidance strengths, the corresponding table or figure explicitly identifies the selection procedure.

Metric computation. FID is computed from 50k generated samples against the standard reference statistics of the corresponding dataset (Heusel et al., 2017). All additional metrics are computed from the same generated sample set. On ImageNet-256, these include sFID, Inception Score, precision, recall, feature-space Fréchet distances, and MIND. The same implementations and reference features are used for matched baseline and SWIFT comparisons.

D.3 COMPUTE ACCOUNTING

We report compute separately for main-model training and guide preparation. The weak guide is pretrained once and then frozen, so its preparation cost can be shared across multiple SWIFT runs.

Wall-clock measurements. On the main DiffusionBench setup using $4\times$B200 GPUs, an 80-epoch baseline run takes approximately 44 hours, while an 80-epoch SWIFT main-model run takes approximately 49 hours once the guide is available. The default 40-epoch guide requires approximately 12 hours to pretrain and can subsequently be reused across different extrapolation coefficients or other runs using the same guide.

Wall-clock measurements are hardware- and implementation-dependent, so our primary accounting below uses FLOPs.

Per-step FLOPs. Let

$$\rho = \frac{\text{guide forward FLOPs}}{\text{main-denoiser forward FLOPs}}.$$

For the DiffusionBench configuration, $\rho \simeq 0.26$.

A baseline training step consists of one main-model forward pass and its backward pass. Approximating the backward pass as twice the forward cost, the baseline step costs three forward-pass equivalents. SWIFT adds one frozen-guide forward pass and no backward pass through the guide. Its relative main-training cost is therefore

$$\frac{C_{\text{SWIFT,step}}}{C_{\text{base,step}}} = 1 + \frac{\rho}{3} = 1.086 \simeq 1.09. \tag{A.28}$$

Thus the frozen guide adds approximately 9% FLOPs per main-training step.

Our analytical count gives 337.5 GFLOPs per sample for the DiffusionBench main-denoiser forward pass and 87.3 GFLOPs for the guide. The resulting full-step estimate agrees with a PyTorch FLOP counter to within 0.5%.

Guide preparation. Training the guide for T_g epochs costs

$$\rho T_g$$

baseline-equivalent training epochs under the same forward/backward approximation. For the default DiffusionBench guide with $T_g = 40$ and $\rho \simeq 0.26$, this corresponds to approximately 10.4 baseline-equivalent epochs.

Hence, if guide preparation is charged to a single e-epoch SWIFT run, its total training cost is

$$C_{\text{SWIFT}}(e) = \rho T_g + \left(1 + \frac{\rho}{3}\right) e \tag{A.29}$$

baseline-equivalent epochs. If a pretrained guide is reused across multiple runs, only the $1.09\times$ main-training factor in Eq. equation A.28 is incurred by each additional run.

The frozen representation encoder is used identically by the baseline and SWIFT and is excluded from this relative accounting. Including its cost changes the resulting ratios only slightly.

RAEv2 guide cost. The RAEv2 weak guide is relatively larger than the DiffusionBench guide. Using its released architecture, its forward-pass ratio is approximately $\rho \simeq 0.37$. The same accounting therefore gives a main-training factor

$$1 + \frac{0.37}{3} \simeq 1.12,$$

with guide preparation accounted for separately through Eq. equation A.29.

E IMAGENET-256: COMPLETE RESULTS

This appendix provides the complete ImageNet-256 results. We first report the full 19-configuration comparison with and without inference-time guidance (Section E.1), followed by the published results used in Figure 1 (Section E.2), a detailed analysis of the 875M-parameter RAEv2 carrier (Section E.3), training trajectories (Section E.4), and coefficient/guide ablations . Training configurations, evaluation protocols, and compute accounting are given in Sections D.1 to D.3.

E.1 ALL CONFIGURATIONS

The 19 configurations consist of 18 DiffusionBench configurations (615M parameters, 80 epochs; six representation autoencoders, five VAEs, four end-to-end-tuned autoencoders, and three pixel-space models) together with the larger RAEv2 carrier. Across all configurations, SWIFT uses the same default coefficient, α=0.4, with a pretrained half-width guide frozen before main-model training; no coefficient is tuned per configuration. Each baseline is either a matched control trained with the same schedule or the carrier's released checkpoint. DiffusionBench pairs are re-sampled with 50k samples and evaluated by the same pipeline, while RAEv2 uses its native evaluator.

Without guidance. SWIFT lowers FID in all 18 DiffusionBench configurations (Table 7), by 16.3–49.4% (median 31.3%), and improves RAEv2 from 1.56 to 1.15. Across all 19 configurations, 269 of the 285 reported metric comparisons favor SWIFT, 7 are ties at the displayed precision, and 9 favor the baseline. The exceptions are limited to four precision entries, two Inception Scores, one recall entry, and one ConvNeXt-based FD_r/MIND pair.

With guidance. Guidance strength is tuned independently for the baseline and SWIFT (Table 8). SWIFT has the lower guided FID in all 19 configurations: by 0.10–1.31 (6.7–33.0%, median 21.9%) across DiffusionBench with CFG, and from 1.04 to 0.99 on RAEv2 with its native internal guidance. Of the 270 DiffusionBench metric comparisons, 250 favor SWIFT, 16 favor the baseline, and 4 are ties at the reported precision. Figure 7 illustrates six representative DiffusionBench cases and also shows why the guidance scale must be tuned separately for the two models.

Table 7: **All 19 ImageNet-256 configurations without guidance.** SWIFT uses the same default coefficient (α=0.4) with a pretrained half-width guide frozen before main-model training, without per-configuration coefficient tuning. FD_r and MIND are reported in five feature spaces; the lower FID of each pair is bold.

						$FD_r \downarrow$					MIND ↓				
Configuration (baseline)	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	Inception	ConvNeXt	DINOv2	MAE	SigLIP	Inception	ConvNeXt	DINOv2	MAE	SigLIP
Representation autoencoders (RAE); 615M, 80 epochs															
DINOv2-B	2.09	4.48	210.6	0.83	0.57	1.30	2.34	3.12	6.24	7.58	2.44	70.80	26.70	0.44	6.34
+ SWIFT	**1.75**	4.26	202.8	0.81	0.61	1.10	2.44	2.82	5.20	6.86	1.55	75.40	23.80	0.35	5.77
DINOv2-B+reg	2.07	4.55	208.2	0.82	0.58	1.28	2.45	3.24	6.32	7.62	2.24	76.10	27.70	0.44	6.25
+ SWIFT	**1.71**	4.18	205.2	0.81	0.61	1.07	2.39	2.80	5.30	6.83	1.54	74.60	23.40	0.35	5.77
LangPE	2.73	4.43	183.2	0.80	0.60	1.63	3.42	6.57	8.98	10.20	2.52	135.00	63.90	0.69	8.89
+ SWIFT	**2.27**	4.01	184.6	0.79	0.61	1.37	3.30	5.72	7.59	9.36	1.76	129.00	55.30	0.56	8.46
PE	3.11	6.15	205.1	0.81	0.56	1.88	3.13	6.13	11.00	9.80	3.20	98.50	58.30	0.88	8.05
+ SWIFT	**2.34**	5.02	209.6	0.81	0.60	1.42	2.72	4.81	9.21	7.72	2.47	85.50	45.00	0.70	6.39
SigLIP2	3.47	7.02	180.5	0.79	0.58	2.10	4.16	8.59	11.00	11.80	2.45	158.00	86.70	0.90	10.20
+ SWIFT	**2.60**	5.39	188.3	0.79	0.61	1.58	3.54	6.99	9.10	9.59	1.87	133.00	70.20	0.72	8.34
SpatialPE	3.60	5.20	160.1	0.78	0.60	2.18	4.22	7.21	6.95	10.70	2.23	210.00	69.10	0.54	9.10
+ SWIFT	**2.92**	4.47	165.8	0.78	0.62	1.78	3.67	6.04	5.64	9.35	1.60	175.00	57.20	0.43	8.00
VAEs; 615M, 80 epochs															
SD-VAE (EMA)+reg	5.49	5.52	161.3	0.71	0.66	3.20	5.26	11.50	9.64	25.80	3.56	257.00	148.00	0.87	28.60
+ SWIFT	**3.77**	5.06	179.1	0.73	0.66	2.19	3.84	8.57	7.57	20.10	1.98	159.00	107.00	0.68	21.90
SD-VAE (MSE)	10.06	5.79	110.9	0.67	0.65	5.92	10.50	18.00	12.20	38.70	9.19	687.00	245.00	1.09	44.60
+ SWIFT	**6.51**	5.26	137.1	0.70	0.66	3.82	7.00	13.00	9.75	30.50	4.96	388.00	171.00	0.87	35.30
SD-VAE (EMA)	10.21	5.56	112.4	0.67	0.66	6.01	10.00	18.10	11.10	35.20	9.67	686.00	245.00	0.99	38.60
+ SWIFT	**6.73**	5.05	138.9	0.70	0.67	3.88	6.67	13.10	8.87	27.70	5.22	388.00	171.00	0.79	30.50
Qwen-VAE	10.69	8.46	109.1	0.66	0.64	6.42	13.30	19.60	13.40	41.30	9.09	806.00	266.00	1.31	43.80
+ SWIFT	**7.18**	7.68	134.8	0.69	0.65	4.14	8.87	14.20	10.70	32.60	4.91	455.00	186.00	1.05	34.70
SDXL-VAE	12.84	7.52	104.1	0.65	0.64	7.47	12.60	21.50	13.60	40.30	12.90	861.00	299.00	1.22	44.10
+ SWIFT	**8.38**	6.83	128.7	0.68	0.65	4.82	8.40	15.50	10.90	31.80	6.97	487.00	209.00	0.97	34.90
End-to-end tuned autoencoders (E2E); 615M, 80 epochs															
E2E VA-VAE	4.14	4.57	148.9	0.74	0.65	2.56	5.81	8.60	6.37	16.20	2.13	269.00	103.00	0.50	15.40
+ SWIFT	**3.13**	4.39	164.2	0.74	0.67	1.96	4.62	6.79	5.24	13.00	1.26	189.00	80.30	0.41	12.20
E2E Qwen-VAE	4.94	5.23	138.9	0.72	0.66	3.08	6.75	9.56	6.46	19.90	2.82	340.00	114.00	0.53	18.90
+ SWIFT	**3.23**	4.89	164.1	0.73	0.66	2.06	4.68	6.44	4.72	14.40	1.35	189.00	72.90	0.37	13.20
E2E SD3.5-VAE	5.26	5.18	141.1	0.71	0.65	3.33	4.99	9.14	7.13	18.50	3.17	260.00	107.00	0.55	17.50
+ SWIFT	**3.61**	4.88	160.9	0.72	0.67	2.36	3.66	6.64	5.50	14.00	1.70	161.00	75.10	0.41	12.90
E2E FLUX-VAE	6.20	5.13	134.9	0.70	0.66	3.77	6.84	10.90	6.63	21.20	4.09	372.00	128.00	0.53	20.30
+ SWIFT	**4.27**	4.78	153.4	0.71	0.67	2.62	4.88	7.81	5.03	15.60	2.21	223.00	88.10	0.39	14.50
Pixel space; 615M, 80 epochs															
PixelGen	12.10	6.15	104.2	0.66	0.61	7.01	13.90	17.90	19.00	34.60	9.03	772.00	222.00	1.68	32.40
+ SWIFT	**8.11**	5.46	126.3	0.68	0.62	4.73	9.83	13.70	15.90	27.90	5.03	482.00	167.00	1.39	26.30
PixNerd	20.37	6.14	65.9	0.61	0.63	12.00	20.70	24.40	18.70	46.70	22.30	1514.00	335.00	1.62	47.60
+ SWIFT	**10.31**	5.38	93.3	0.70	0.61	6.13	12.20	14.30	13.80	30.00	8.50	776.00	172.00	1.23	28.80
JiT	22.36	6.98	63.3	0.58	0.63	13.20	23.40	28.70	22.70	54.70	25.50	1690.00	422.00	2.04	56.40
+ SWIFT	**12.31**	6.15	98.4	0.64	0.63	7.27	13.50	18.50	18.20	38.40	11.10	804.00	254.00	1.65	39.30
RAEv2, 875M, 160 epochs															
DINOv3-L	1.56	4.10	233.3	0.82	0.65	1.00	2.20	2.60	4.80	6.30	1.40	70.00	22.00	0.34	5.60
+ SWIFT	**1.15**	3.60	242.2	0.81	0.67	0.82	2.05	2.25	4.15	5.45	1.10	65.00	18.50	0.29	4.85

Table 8: **All 19 ImageNet-256 configurations with guidance.** Inference-time guidance is tuned independently for the baseline and SWIFT. The training recipe is otherwise the same as in Table 7. FD_r and MIND are reported in five feature spaces; the lower FID of each pair is bold.

						$FD_r \downarrow$					MIND $\downarrow$				
Configuration (baseline)	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	Inception	ConvNeXt	DINOv2	MAE	SigLIP	Inception	ConvNeXt	DINOv2	MAE	SigLIP
Representation autoencoders (RAE); 615M, 80 epochs															
DINOv2-B	1.98	4.50	223.1	0.81	0.59	1.23	2.23	3.33	6.14	7.78	1.87	67.60	28.60	0.43	6.50
+ SWIFT	**1.59**	4.30	221.6	0.80	0.62	1.00	2.26	2.89	5.16	6.94	1.13	65.80	24.30	0.34	5.82
DINOv2-B+reg	1.84	4.55	236.2	0.82	0.58	1.15	2.15	3.21	6.43	7.71	1.90	64.30	27.40	0.44	6.43
+ SWIFT	**1.52**	4.18	232.9	0.81	0.61	0.96	2.10	2.77	5.39	6.91	1.31	63.01	23.15	0.35	5.94
LangPE	2.39	4.42	201.6	0.80	0.61	1.43	2.89	6.16	8.98	10.1	2.17	103.00	61.40	0.69	8.84
+ SWIFT	**2.06**	4.02	199.0	0.79	0.62	1.25	2.90	5.45	7.53	9.52	1.38	104.00	53.90	0.56	8.68
PE	2.84	5.80	221.5	0.81	0.57	1.72	2.80	5.91	10.90	9.88	2.86	90.07	56.80	0.88	8.19
+ SWIFT	**2.14**	4.73	226.4	0.80	0.62	1.31	2.43	4.64	9.12	7.78	2.21	78.18	43.85	0.71	6.51
SigLIP2	2.61	6.90	222.9	0.78	0.59	1.59	3.01	7.33	10.61	11.38	1.86	97.91	76.01	0.87	9.89
+ SWIFT	**2.15**	5.32	219.4	0.79	0.62	1.32	2.81	6.37	8.85	9.74	1.35	91.90	66.20	0.69	8.52
SpatialPE	1.86	4.50	247.1	0.79	0.61	1.16	1.82	4.67	6.62	8.57	1.35	55.46	45.79	0.51	7.23
+ SWIFT	**1.51**	3.87	255.9	0.79	0.63	0.95	1.58	3.91	5.37	7.49	0.97	46.21	37.91	0.41	6.36
VAEs; 615M, 80 epochs															
SD-VAE (EMA)+reg	2.32	4.70	271.5	0.80	0.60	1.30	1.76	7.37	7.61	18.6	1.88	53.40	86.00	0.70	20.20
+ SWIFT	**1.81**	4.42	272.9	0.80	0.61	1.01	1.48	5.71	6.11	15.1	1.63	48.30	66.20	0.55	16.30
SD-VAE (MSE)	2.56	5.50	259.7	0.70	0.65	1.45	2.15	7.61	7.82	21.82	3.10	55.66	86.97	0.70	25.01
+ SWIFT	**1.86**	4.99	268.4	0.73	0.66	0.94	1.43	5.49	6.25	17.19	1.67	31.45	60.71	0.56	19.78
SD-VAE (EMA)	2.43	5.40	259.6	0.69	0.65	1.38	1.92	7.71	6.94	19.15	2.92	48.52	87.44	0.62	20.39
+ SWIFT	**1.90**	5.08	260.9	0.69	0.66	1.07	1.61	5.98	5.57	15.55	2.53	43.86	67.33	0.49	16.45
Qwen-VAE	3.01	7.70	238.9	0.68	0.65	1.85	4.56	9.77	9.19	25.46	2.75	159.26	118.42	0.90	25.97
+ SWIFT	**2.32**	6.99	250.8	0.71	0.66	1.19	3.04	7.05	7.34	20.06	1.49	89.98	82.66	0.72	20.54
SDXL-VAE	3.07	6.90	256.0	0.67	0.65	1.69	2.74	9.26	8.64	21.55	3.38	70.85	109.22	0.75	22.66
+ SWIFT	**2.21**	6.27	268.8	0.70	0.66	1.09	1.83	6.69	6.91	16.98	1.83	40.03	76.24	0.60	17.92
End-to-end tuned autoencoders (E2E); 615M, 80 epochs															
E2E VA-VAE	1.65	4.50	275.4	0.74	0.65	1.08	1.99	4.51	4.71	9.83	2.13	62.17	50.75	0.36	9.17
+ SWIFT	**1.25**	4.32	289.2	0.73	0.67	0.83	1.58	3.56	3.88	7.88	1.26	43.71	39.59	0.30	7.26
E2E Qwen-VAE	1.50	4.83	263.3	0.80	0.62	1.04	2.25	4.59	4.61	11.2	1.48	61.40	49.00	0.37	10.10
+ SWIFT	**1.40**	4.59	296.5	0.80	0.63	0.97	1.76	3.04	3.31	7.82	1.86	76.50	29.80	0.25	6.76
E2E SD3.5-VAE	1.62	5.00	265.4	0.72	0.66	1.16	1.30	4.57	5.49	11.12	1.10	42.14	48.18	0.42	10.25
+ SWIFT	**1.31**	4.81	278.7	0.71	0.68	0.89	1.03	3.61	4.52	8.92	0.65	29.62	37.58	0.34	8.12
E2E FLUX-VAE	1.67	5.00	266.3	0.70	0.66	1.07	1.83	5.12	4.68	11.76	1.16	50.73	53.43	0.36	10.75
+ SWIFT	**1.25**	4.66	279.6	0.71	0.67	0.74	1.30	3.67	3.55	8.66	0.63	30.39	36.76	0.26	7.68
Pixel space; 615M, 80 epochs															
PixelGen	3.97	5.50	247.4	0.67	0.61	2.26	4.34	7.80	13.14	17.97	3.96	138.33	86.50	1.12	15.73
+ SWIFT	**2.66**	4.88	259.8	0.69	0.63	1.53	3.07	5.97	11.01	14.48	2.21	86.32	65.05	0.93	12.77
PixNerd	4.17	5.50	213.8	0.65	0.63	2.45	4.01	8.33	11.67	20.65	4.24	104.21	86.10	0.96	18.94
+ SWIFT	**3.38**	4.95	235.2	0.68	0.62	1.96	3.29	6.83	9.57	16.93	3.39	85.45	70.61	0.79	15.53
JiT	4.08	6.00	231.2	0.60	0.63	2.38	4.59	9.57	13.70	22.51	3.97	146.37	113.63	1.23	21.21
+ SWIFT	**2.86**	5.29	277.4	0.66	0.64	1.31	2.65	6.17	10.99	15.80	1.73	69.67	68.41	0.99	14.78
RAEv2, 875M, 160 epochs															
DINOv3-L	1.04	4.00	266.0	0.82	0.61	0.87	1.68	2.64	5.22	6.13	1.35	49.40	22.00	0.32	4.82
+ SWIFT	**0.99**	3.80	264.9	0.81	0.63	0.83	1.61	2.45	4.81	5.71	1.25	46.10	20.10	0.31	4.51

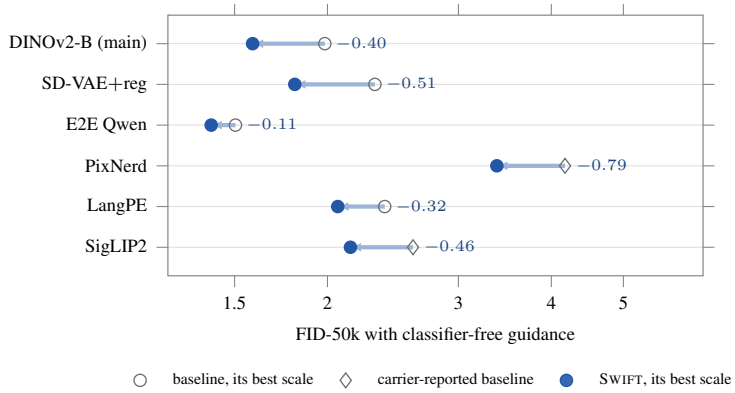

Figure 7: **Guided FID at each model's best available CFG scale** for six DiffusionBench configurations from Table 8. Open circles denote matched baselines whose guidance scale is swept independently; open diamonds denote carrier-reported guided baselines. Filled dots show SWIFT at its own best scale, and annotations give the FID change. The preferred scale can shift substantially (e.g., $2.2 \rightarrow 1.8$ for SD-VAE+reg and $2.4 \rightarrow 1.8$ for PixNerd), motivating separate tuning on the two sides.

E.2 PUBLISHED RESULTS IN FIGURE 1

Table 9 lists the published results plotted in the left panel of Figure 1. Every number is the one reported by the authors for their best configuration at 256×256; we did not re-evaluate these models.

Table 9: **Published ImageNet-256 results shown in Figure 1 (left).** FID-50K values are taken directly from the corresponding papers. Evaluation protocols differ between papers: some sample 50 images per class, others sample class labels at random, which can change FID by about 0.1 (Zheng et al., 2026, Table 14). Guidance: CFG, classifier-free guidance; int., with a guidance interval; AG, autoguidance; IG, internal guidance (Zhou et al., 2026); REPA guid., the REPA guidance of Singh et al. (2026). *Without CFG, but still with internal guidance. Our two RAEv2 rows use the settings of Section E.3.

Model	Type	Base FID	Guided FID	Guidance
DiT-XL/2 (Peebles & Xie, 2023)	latent diffusion	9.62	2.27	CFG
SiT-XL/2 (Ma et al., 2024)	latent diffusion	8.30	2.06	CFG
MDT-XL/2 (Gao et al., 2023)	latent diffusion	6.23	1.79	CFG
DDT-XL/2 (Wang et al., 2025)	latent diffusion	6.27	1.26	CFG, int.
REPA (Yu et al., 2025)	latent diffusion	5.90	1.42	CFG, int.
MAR-H (Li et al., 2024)	autoregressive	2.35	1.55	CFG
SFD-XXL (Pan et al., 2026)	latent diffusion	2.38	1.04	AG
LightningDiT (Yao et al., 2025)	latent diffusion	2.17	1.35	CFG, int.
REG (Wu et al., 2025)	latent diffusion	1.80	1.36	CFG, int.
REPA-E (Leng et al., 2025)	latent diffusion	1.69	1.12	CFG, int.
RAEv2 (Singh et al., 2026)	latent diffusion	1.65	1.06	REPA guid.
RAE (Zheng et al., 2026)	latent diffusion	1.51	1.13	AG
PAE (Yue et al., 2026)	latent diffusion	1.43	1.03	CFG
BAR-L (Yu et al., 2026)	autoregressive	1.42	0.99	CFG
LightningDiT+IG (Zhou et al., 2026)	latent diffusion	1.34*	1.19	IG, CFG, int.
GAE (Liu et al., 2026)	latent diffusion	1.31	1.13	CFG, int.
RAEv2, our control	latent diffusion	1.56	1.04	Section E.3
SWIFT on RAEv2 (ours)	latent diffusion	**1.15**	**0.99**	Section E.3

E.3 RAEv2

RAEv2 provides a second, stronger carrier: an 875M-parameter model on DINOv3-L latents with native internal guidance. SWIFT uses the default α=0.4 recipe and a half-width guide frozen at epoch 40 (Section D.1). The canonical RAEv2 results are those reported in Tables 7 and 8: $1.56 \rightarrow 1.15$ without guidance and $1.04 \rightarrow 0.99$ with independently tuned internal guidance.

Training trajectory. To examine when the gain appears, we additionally track SWIFT at 17 checkpoints and the α=0 control at epochs 20, 40, 80, 120, and 160 (Figure 8). The α=0.4 run is already below the released epoch-80 checkpoint by epoch 40 and remains on a substantially lower trajectory thereafter. Its best observed unguided checkpoint occurs at epoch 140 and yields the 1.15 value reported in Table 7. The denser checkpoint trace is used only to study training dynamics; the headline values remain those in the two complete-result tables.

Interaction with internal guidance. RAEv2 uses

$$D_s = D + (s-1)(D - D_{\text{REPA}}),$$

so $s = 1$ disables guidance. The preferred guidance strength shifts downward after SWIFT training, while the best guided result improves from 1.04 to 0.99 (Table 8). This is consistent with the main denoiser absorbing part of the weak-to-strong correction during training, while inference-time guidance remains complementary.

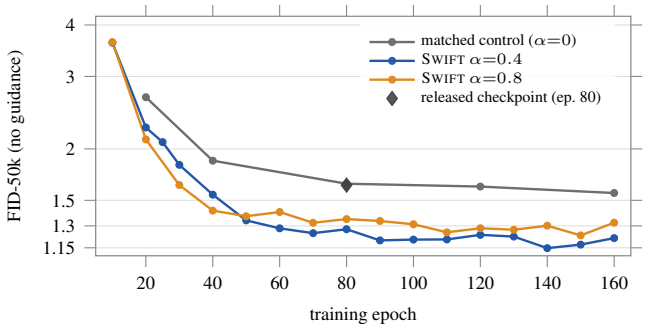

Figure 8: **RAEv2 unguided training trajectories.** The dense SWIFT traces are used to study training dynamics; the canonical RAEv2 results are reported in Tables 7 and 8. The diamond marks the released epoch-80 checkpoint.

E.4 TRAINING TRAJECTORIES AND THE LONG SCHEDULE

Across configurations. For the 12 DiffusionBench configurations whose baseline we also evaluated during training, SWIFT is ahead at every evaluation from epoch 40 on. Figure 9 shows four of them, with baselines spanning roughly an order of magnitude in FID.

The main configuration. On DINOv2-B we compared three arms with the matched control at every evaluation: α=0.4 and α=0.8 with the guide frozen at epoch 40, and α=0.4 with the guide frozen at epoch 20. The two arms with the epoch-40 guide are behind the control at epoch 20 (by 0.11 and 0.05 FID) and ahead from epoch 40 on (by 0.28–0.48); the arm with the earlier guide is ahead from epoch 20 (by 0.47–0.59) and lowest throughout.

The 240-epoch schedule. The matched DINOv2-B pair is continued to 240 epochs (Table 10 and Figure 10). SWIFT remains ahead at all 36 matched evaluations, showing that the gain is not an early-training effect that disappears with a longer schedule. Table 10 reports every 20th epoch, while Figure 10 shows the full evaluation grid. A separate α=0 hardware replicate is used only as an additional reference level, never to form a matched difference.

Table 10: **Matched long-schedule comparison on ImageNet-256** (DiffusionBench, DINOv2-B). Values are unguided FID from the training-time evaluation pipeline, shown every 20 epochs. The full matched evaluation grid is visualized in Figure 10.

Epoch	20	40	60	80	100	120	140	160	180	200	220	240
Control (α=0)	3.63	2.67	2.36	2.13	1.99	1.91	1.81	1.80	1.75	1.72	1.67	1.63
SWIFT	**3.52**	**2.32**	**2.00**	**1.77**	**1.60**	**1.50**	**1.51**	**1.46**	**1.49**	**1.44**	**1.41**	**1.38**
Δ FID	−0.12	−0.36	−0.36	−0.37	−0.39	−0.41	−0.30	−0.34	−0.26	−0.28	−0.26	−0.26

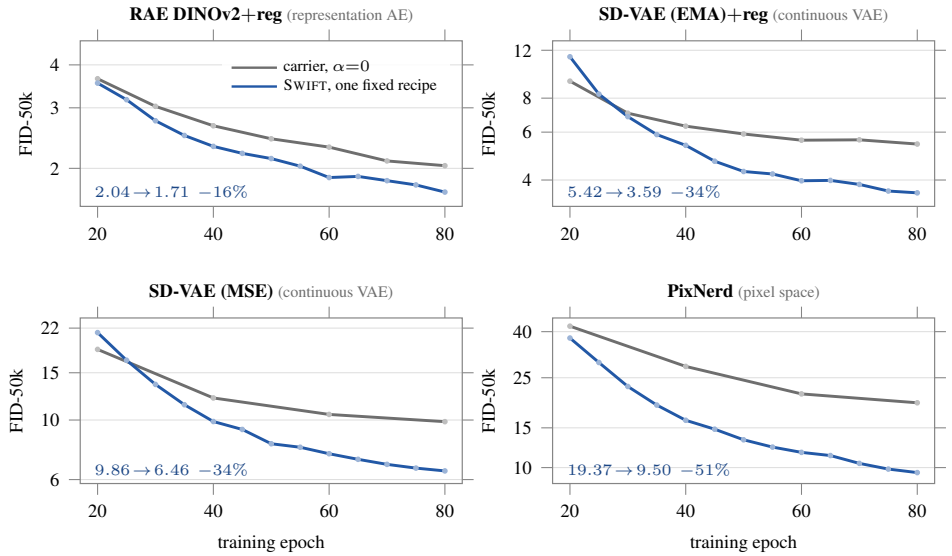

Figure 9: **The same fixed SWIFT recipe improves four different ImageNet-256 training trajectories.** Each panel shows a different DiffusionBench stage-2 model and representation. Grey is the matched α=0 control and blue is SWIFT with the same fixed recipe (α=0.4, half-width guide frozen at epoch 40), with no per-model tuning. SWIFT is ahead by epoch 40 in all four cases and remains ahead through the end of training, across baselines spanning roughly an order of magnitude in FID. The y-axis is logarithmic; faint markers denote individual FID evaluations. The trajectory controls reproduce the corresponding DiffusionBench endpoints within 0.6–6%; the comparisons in Table 7 instead re-evaluate both sides with 50k samples.

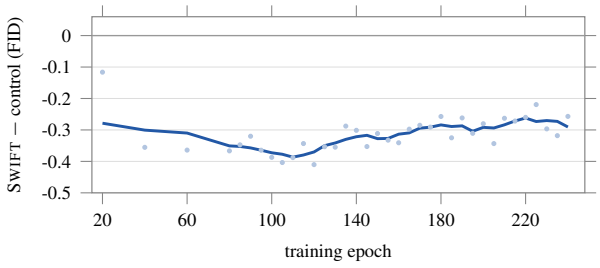

Figure 10: **The 240-epoch matched gap, evaluation by evaluation** (ImageNet-256, DiffusionBench DINOv2-B, unguided FID). All 36 matched differences are negative from the first evaluation at epoch 20 onward. Faint dots show individual evaluations and the solid line is a five-point centred moving average; Table 10 reports every 20th epoch from the same run.

F CIFAR-10 RESULTS

This appendix reports the CIFAR-10 experiments behind Section 5.4: full training trajectories (Section F.1), the campaign of arms (Section F.2), the interaction of guide width, coefficient, and architecture (Section F.3), schedules for α and a jointly trained guide (Section F.4), and the accounting of every launched arm (Section F.5). The configuration is in Section D.1.

F.1 FULL TRAINING TRAJECTORIES

Figure 11 shows the complete 200k-kimg trajectories of the matched control and three SWIFT arms on a logarithmic scale. SWIFT arms with a mature guide start behind the control because the $(1+\alpha)$ residual inflation dominates while the student's own error is still large, cross the control around 65k kimg, and separate from it in the final third of the budget.

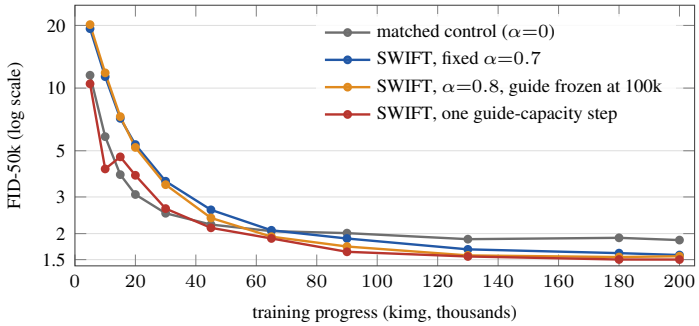

Figure 11: **CIFAR-10 FID over the full 200k-kimg budget** (log scale).

F.2 THE CAMPAIGN

Figure 12 places every arm whose endpoint this paper quotes on one axis, grouped by the factor it varies, against both reference arms: the matched α=0 control and the best plain constant-α arm. Read as a map of the search it says two things the three sweeps of Figure 6 do not. First, the region that beats doing nothing is broad along the coefficient and the guide's training length and is essentially a single point along guide width—which is why the recipe fixes the width and leaves the other two loose. Second, of the 43 arms reported here, 29 beat the control but only two beat the best plain constant-α arm by more than the replicate band, and those two switch once, in the first part of training, from a smaller guide to the half-width guide, about 0.10 better than keeping the half-width guide throughout at the same α=0.8 (1.603). The gain is small, it did not carry over to ImageNet beyond noise, and the recipe does not use it. The campaign's value is the map, not a better arm: the plain method is already close to the best form of itself.

Figure 13 then shows the campaign as it actually is: every scored arm's trajectory, and every endpoint grouped by the factor its family varies.

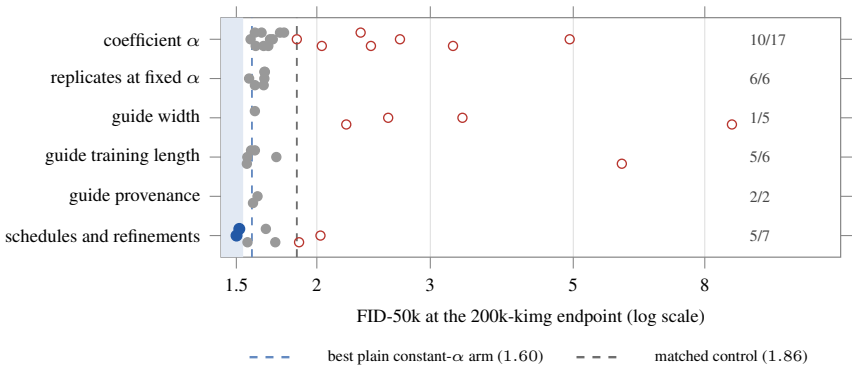

Figure 12: **The reported CIFAR-10 campaign, by the factor each arm varies** (FID-50k at the 200k-kimg endpoint). Filled markers beat the matched control by more than the measured replicate standard deviation; open markers do not. Blue markers beat the best plain constant-α arm by more than the measured replicate standard deviation, grey markers sit between the two references, and open red markers are no better than the control. The fraction at the right of each row counts arms that beat the control. Four arms fall outside the axis on the right: the two narrowest guides (8.82, 3.37), the earliest guide snapshot (5.95) and α=2 (4.94). These are the arms quoted in this paper, not the whole campaign; Section F.5 accounts for all 152 scored arms and the complete per-arm ledger is released as `DATA/`.

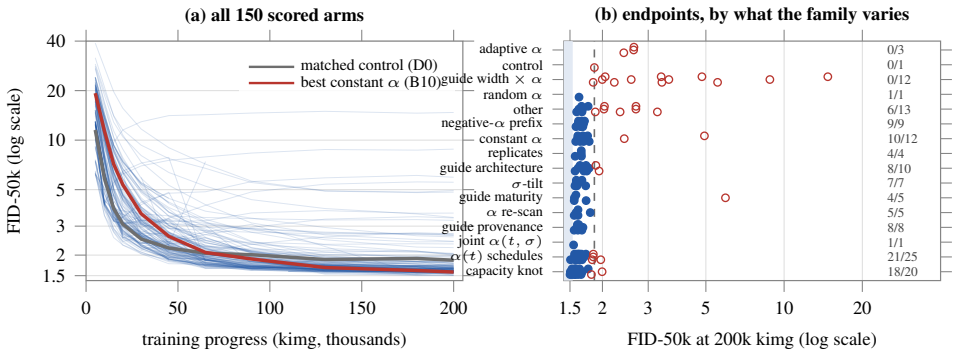

Figure 13: **The whole CIFAR-10 campaign.** (a) All 150 scored arms that reach the 200k-kimg grid (two divergent runs, excluded from the authors' own plots, are omitted), with the matched control and the best constant-α arm drawn on top. (b) Their endpoints, grouped by what the family varies; filled markers beat the control by more than the replicate standard deviation, the shaded region is better than the best plain constant-α arm, and the fraction at the right counts arms that beat the control. Whole families—adaptive α, random α, the σ-tilt—sit entirely on the wrong side of the control.

F.3 Guide width, coefficient, and architecture

Figure 14 crosses five guide widths with three coefficients. Guides from three architecture families (DDPM++, the student's own family; NCSN++; and ADM) all help when they are narrow, with endpoints between 1.57 and 1.78 over $\alpha \in [0.3, 1.0]$ and differences between families at the replicate floor. Both full-width guides are worse than using no guide at all (1.94 for DDPM++ and 1.89 for NCSN++, against the control's 1.86): what the method needs from the guide is that it be weak, not that it be different.

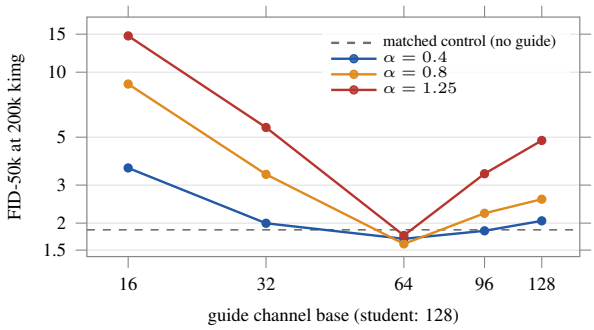

Figure 14: **Guide width and the coefficient are not independent.** The full 5×3 grid: five guide widths at three coefficients, guide mature in every cell. Half the student's width is the minimum at every coefficient, but the penalty for choosing wrongly is roughly proportional to α—at α=0.4 the curve is nearly flat from half to full width, and at α=1.25 the same choice costs an order of magnitude. Part of what reads as a sharp width optimum in Figure 6b is over-extrapolation at α=0.8 rather than width sensitivity as such.

F.4 Schedules for α and a jointly trained guide

The recipe keeps α fixed and the guide frozen. The two most natural alternatives both speed up the early phase of training, and neither improves the endpoint.

Changing α during training. A positive α applied only in the first half of training accelerates convergence, but the gain does not carry over to the end. Three seeds hold α at 0.1 until 30k kimg, raise it to 0.9 by 45k kimg, keep it there until 90k kimg, and lower it back to 0.1 by 130k kimg. At 90k kimg they reach FID 1.63–1.71, ahead of both the control (2.01) and constant α=0.7 (1.90). Once α is lowered the advantage disappears: they end at 1.83–1.85, level with the control (1.86) and well behind constant α (1.58–1.60). The same schedule with the late level held at 0.6 or 0.9 instead ends at 1.56 and 1.62, so the endpoint gain comes from α in the late phase. Pulling the target towards the guide at the start (α between -1 and -0.25 for the first 5–15k kimg, followed by the positive schedule) speeds up the first steps even more, to FID 6.2–9.2 at 5k kimg against 11.5 for the control, but these nine runs end at 1.58–1.71, again no better than a constant positive α. Choosing α online from a held-out loss (2.42–2.64, worse than the control), drawing it at random at every step (1.63), and tilting it with the noise level (1.57–1.72) did not help either.

Training the guide jointly with the student. The weak model can also be trained together with the student instead of beforehand: a half-width network is trained alongside the student, and the student's target uses its current EMA prediction. This converges much faster at the start (FID 6.3 and 7.3 at 5k kimg for α=0.8 and 0.6, against 11.5 for the control) and ends well below the control, at 1.618 and 1.667. It remains behind SWIFT with a frozen half-width guide at the same coefficients (1.557 and 1.583 with the guide frozen at 100k kimg, and 1.603 at α=0.8 with the guide frozen at 200k kimg), and each step costs 1.7× as much because the guide is trained as well.

F.5 Campaign accounting

The CIFAR-10 package contains 156 curve files: 152 arms with at least one scored FID point and four launched-but-unscored arms. Of the scored arms, 138 have a point exactly at the 200k-kimg endpoint; five continue to 400k and nine terminate earlier. No arm is ranked by an off-lattice

last point. The measured replicate floor is 0.049 FID, so differences inside roughly ± 0.05 are not stable rankings. Rather than freezing a 150-row leaderboard into the PDF, we ship the machine-readable campaign files with the code: `DATA/cifar10_fid_vs_kimg.csv` contains every scored trajectory, `DATA/CIFAR10.md` records the accounting, endpoint rules, extended arms, and unscored runs, and `DATA/in256_fid_vs_epoch.csv`, `DATA/guidance_sweeps.csv`, and `DATA/in256_published_baselines.csv` hold the ImageNet numbers. The paper reports only comparisons that answer a specific question; the ledger is what lets a reader audit that they were not selected from off-lattice checkpoints.

G ADDITIONAL SAMPLES

Figures 15 to 18 show samples from SWIFT on RAEv2/DINOv3-L, the model with guided FID 0.99 in Table 8, for 32 ImageNet classes. As in Figure 3, samples are generated with retuned internal guidance and a 50-step ODE sampler. Each block shows 28 samples of one class; the class index and name are given below the block.

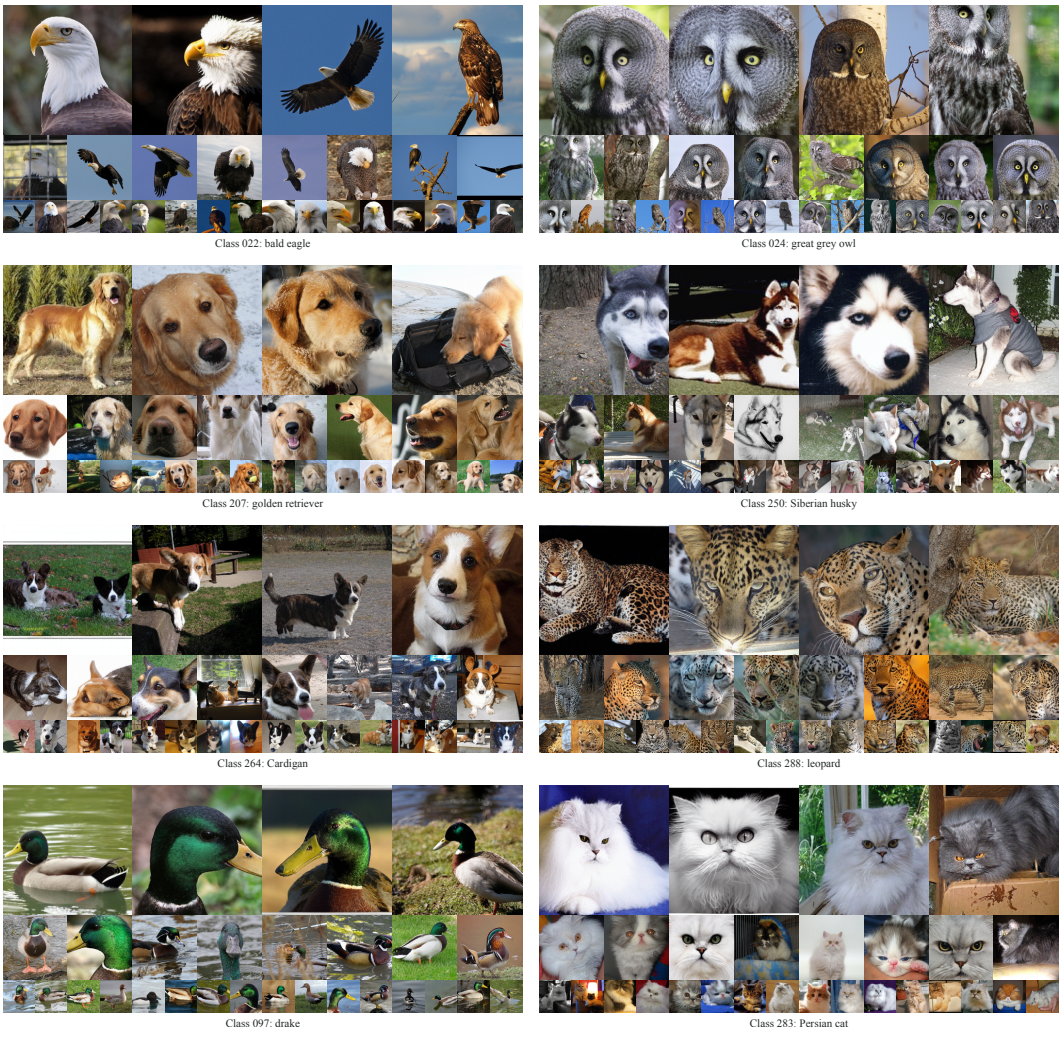

Figure 15: **Samples from SWIFT on RAEv2/DINOv3-L: birds and mammals.** Internal guidance, 50-step ODE sampler; 28 samples per class.

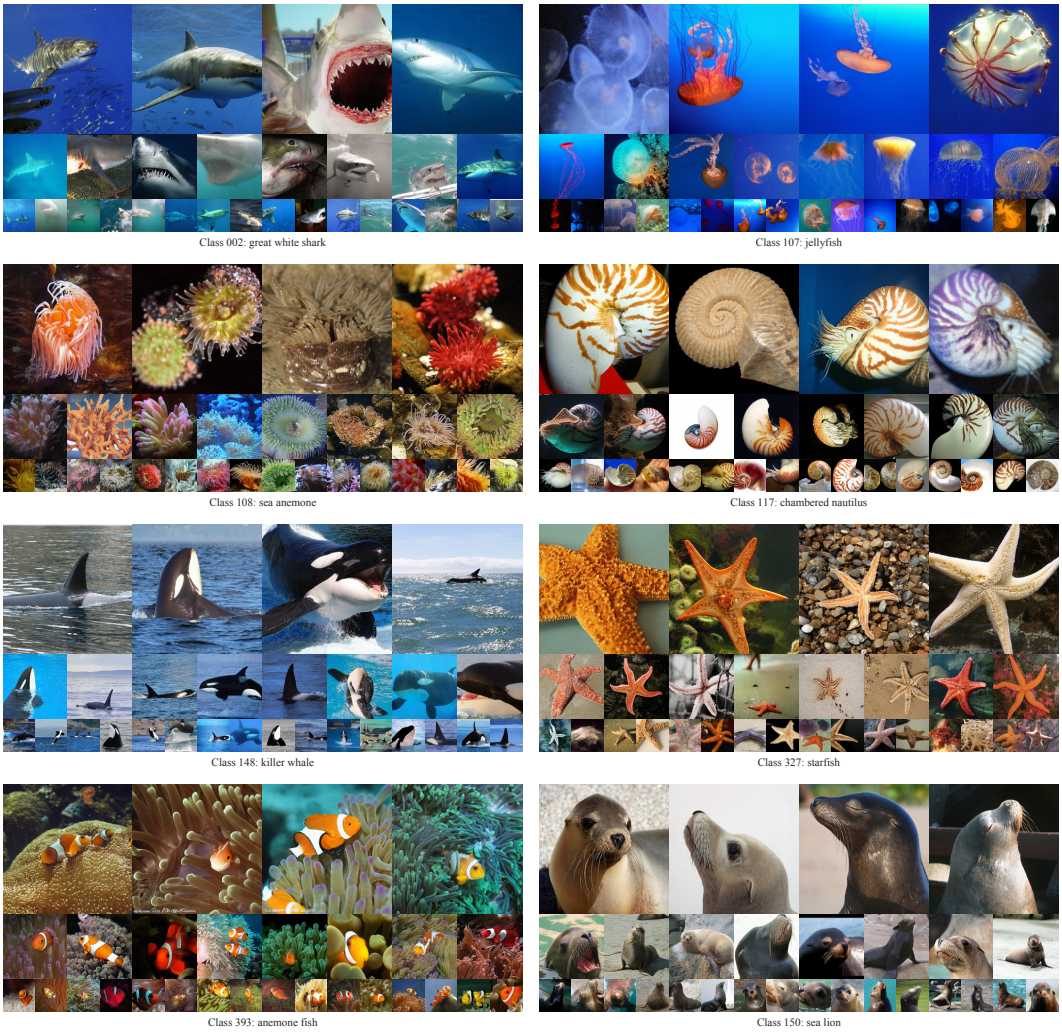

Figure 16: **Samples from SWIFT on RAEv2/DINOv3-L: marine life.** Internal guidance, 50-step ODE sampler; 28 samples per class.

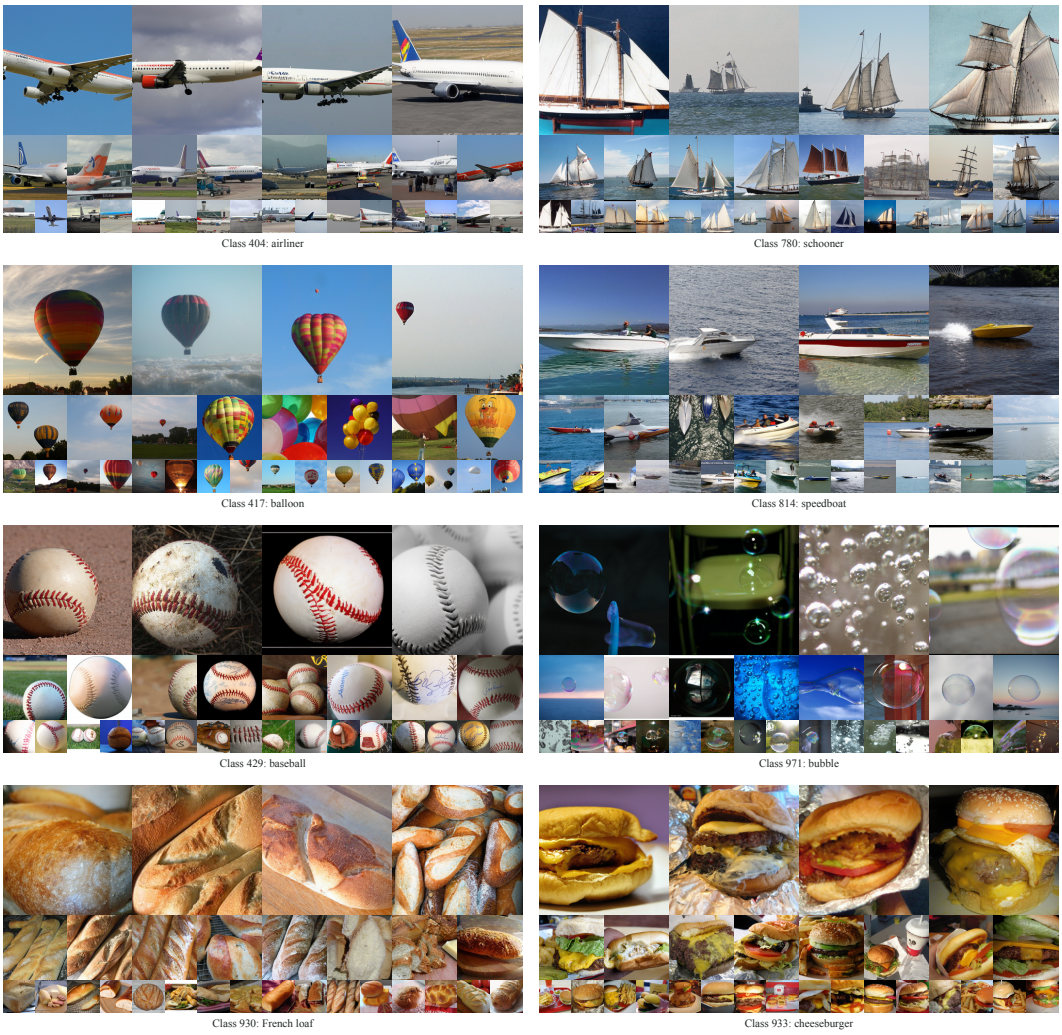

Figure 17: **Samples from SWIFT on RAEv2/DINOv3-L: vehicles, objects, and bread.** Internal guidance, 50-step ODE sampler; 28 samples per class.



Figure 18: **Samples from SWIFT on RAEv2/DINOv3-L: fruit, food, and plants.** Internal guidance, 50-step ODE sampler; 28 samples per class.