

Self-guided Weak-to-strong dIFfusion models via Target extrapolation

Yanjin He, Cuifeng Shen, Wanzhou Lei, Maohua Li, Tao Lan, Kan Liu, Hanlin Tang, Qing Qu

Presenter: Yanjin He

October 20, 2026

homepage: yanjinhe.github.io email: yanjinhe@umich.edu

1

Background

Preliminaries for Diffusion models

2

Method

Train toward a target extrapolated away from a small frozen model

3

Why it works

A target biased the other way cancels the training shortfall

4

Comparison with Other Methods

Compare with other inference/training methods

5

Experiments

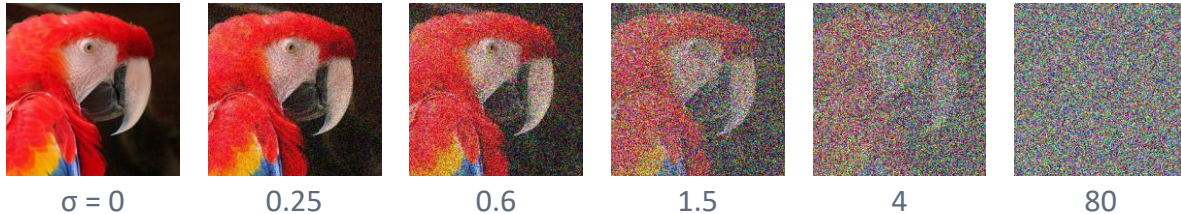
ImageNet-256 across 19 configurations; design choices on CIFAR-10

6

Conclusion

Takeaways and open directions

Forward process: add Gaussian noise



$$\mathbf{x}_t = \mathbf{x}_0 + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I})$$

Three equivalent dynamics

Forward SDE

$$d\mathbf{x}_t = \sqrt{2\dot{\sigma}_t \sigma_t} d\mathbf{w}_t$$

Reverse SDE

$$d\mathbf{x}_t = -2\dot{\sigma}_t \sigma_t \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) dt + \sqrt{2\dot{\sigma}_t \sigma_t} d\bar{\mathbf{w}}_t$$

Prob.-flow ODE

$$d\mathbf{x}_t = -\dot{\sigma}_t \sigma_t \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) dt$$

Sampling needs only the score

Reverse SDE and ODE share the noisy marginals p_t ; their only unknown is $\nabla \log p_t$, which is called the score function.

and we only need the score for sampling:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{\mathcal{D}^*(\mathbf{x}_t, \sigma_t) - \mathbf{x}_t}{\sigma_t^2}$$

$$\mathcal{D}^*(\mathbf{x}_t, \sigma_t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$$

$\mathcal{D}^*$ is the optimal denoiser: the posterior mean of the clean sample.

Learn $\mathcal{D}^*$, and you can sample.

One training step

- 1 Sample $x_0 \sim p_{\text{data}}$, a noise level σ_t , noise ε_t
- 2 Add noise to the sample: $x_t = x_0 + \varepsilon_t$
- 3 Predict the clean sample: $D_\theta(x_t, \sigma_t)$
- 4 Gradient step on $\lambda(t) \|D_\theta(x_t, \sigma_t) - x_0\|^2$

The training loss

$$\mathcal{L}_{\text{base}}(\theta) = \mathbb{E}[\lambda(t) \|D_\theta(x_t, \sigma_t) - x_0\|_2^2]$$

Its optimum: the posterior mean

$$\mathcal{D}^*(x_t, \sigma_t) = \mathbb{E}[x_0 | x_t] = \operatorname{argmin}_{\mathcal{D}} \mathbb{E}[\lambda(t) \| \mathcal{D}(x_t, \sigma_t) - x_0 \|_2^2]$$

Over all functions, the minimizer is exactly what the sampler needs.

SWIFT changes only the target x_0 .

SWIFT changes only the target: push $\mathbf{x}_0$ away from a frozen small model's prediction

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

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

$\alpha = 0$ recovers standard training.

How it is used

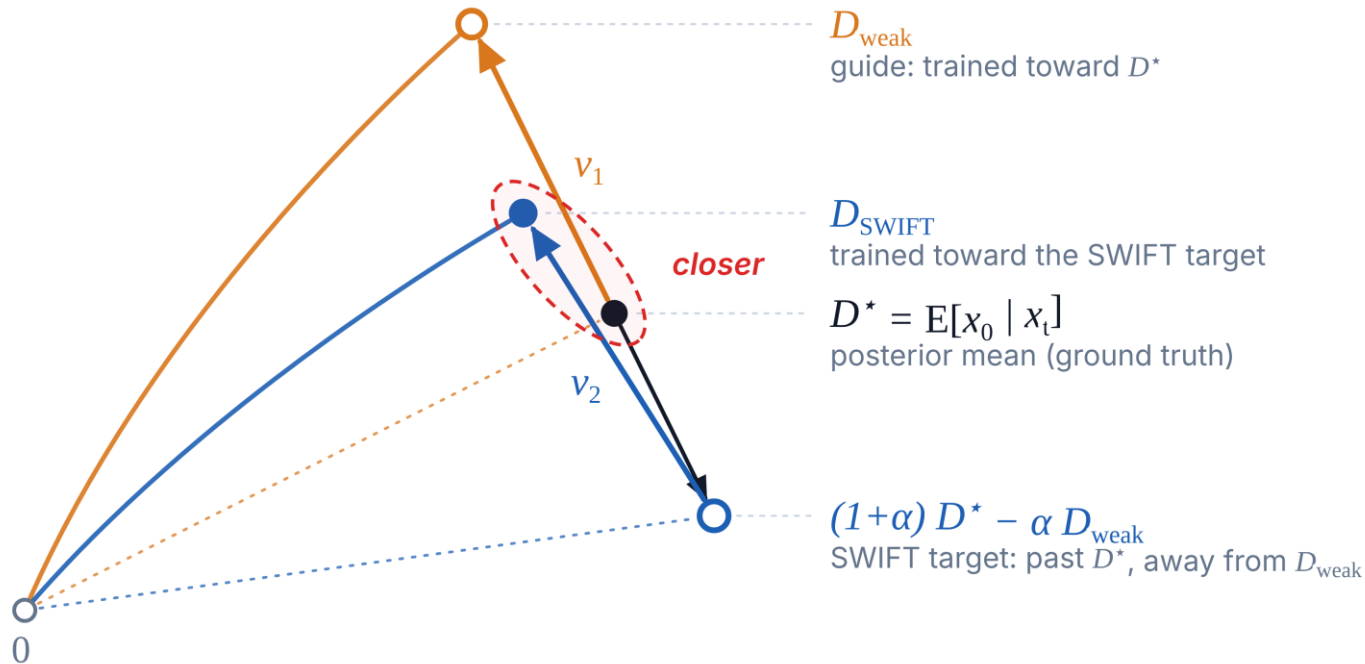
- 1 Train a small model $\mathcal{D}_{\text{weak}}$ with the ordinary loss; freeze it
- 2 Train the main model on $\mathbf{y}_\alpha$: one extra frozen forward pass per step
- 3 Sample with the main model alone; the small model is discarded

What changes, what stays

- **Changes:** only the regression target, $\mathbf{x}_0 \rightarrow \mathbf{y}_\alpha$
- **Unchanged:** architecture, noise schedule, weighting, optimizer, sampler
- **Inference:** one network, no extra NFE; guidance can still be added
- **Flow matching:** the same extrapolation on the velocity target

Why a biased target helps

Models of the same kind would have error in the same direction — so aim past the target and land closer



Since we use the same model and the same training procedure, v_1 and v_2 are highly similar.

- **Training stops short.** Aimed at D^* , the weak model lands at $D_{\text{weak}} = D^* + v_1$.
- **SWIFT aims past D^* :** its target is $D^* - \alpha v_1$.
- **Same family, same training $\Rightarrow v_2$ is similar to v_1 ,** so $D_{\text{SWIFT}} \approx D^* + (1 - \alpha)v_1$.

A finite-time guarantee

In a linear model of training, every α in a window $(0, \alpha^*)$ brings the denoiser strictly closer to $\mathcal{D}^*$

Theorem 1 (finite-time improvement from a frozen weak denoiser).

Kernel gradient flow at one noise level, same kernel and data for all: the guide $\mathcal{D}_{\text{weak}}$ runs to T_g , standard training and SWIFT to $T > T_g$. With $H_s = I - \exp(-s\mathcal{L})$,

$$\mathcal{D}_{\text{weak}} = \mathbf{H}_{T_g} \mathcal{D}^*, \quad \mathcal{D}_{\text{base}} = \mathbf{H}_T \mathcal{D}^*, \quad \mathcal{D}_{\text{SWIFT}}^{(\alpha)} = \mathbf{H}_T [(1 + \alpha) \mathcal{D}^* - \alpha \mathcal{D}_{\text{weak}}]$$

If $\mathbf{H}_T \mathcal{D}^* \neq 0$, then there exists $\alpha^* > 0$ such that

$$R(\alpha) := \|\mathcal{D}_{\text{SWIFT}}^{(\alpha)} - \mathcal{D}^*\|_{\mathcal{H}_2}^2 < R(0) = \|\mathcal{D}_{\text{base}} - \mathcal{D}^*\|_{\mathcal{H}_2}^2 \quad \text{for all } 0 < \alpha < \alpha^*$$

We use RKHS theory to analysis proves that: our method could learn a $\mathcal{D}_{\text{SWIFT}}^{(\alpha)}$ that approximate the posterior mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$ better.

Versus other training-time guidance

The others use a changing prediction or an extra output; SWIFT's reference is frozen before training

MG

Tang et al., 2025

$$\mathcal{L}_{\text{MG}} = \mathbb{E} \|\epsilon_{\theta}(\mathbf{x}_t, c) - [\epsilon + w \text{sg}(\tilde{\epsilon}_{\theta}(\mathbf{x}_t, c) - \tilde{\epsilon}_{\theta}(\mathbf{x}_t, \varnothing))]\|^2$$

EMA cond. / uncond. predictions · uncond. branch

GFT

Chen et al., 2025

$$\mathcal{L}_{\text{GFT}} = \mathbb{E}_{\beta \sim U(0,1)} \|\beta \epsilon_{\theta}(\mathbf{x}_t, c, \beta) + (1 - \beta) \text{sg}[\epsilon_{\theta}(\mathbf{x}_t, \varnothing)] - \epsilon\|^2$$

own detached uncond. output · uncond. branch

SGG

Yuan et al., 2026

$$\mathcal{L}_{\text{SGG}} = \mathbb{E} \|\mathbf{v}_{\theta}(\mathbf{x}_t, c) - [\mathbf{u} + w \text{sg}(\mathbf{v}_{\theta}(\mathbf{x}_t, c) - \mathbf{v}_{\text{aux}}(\mathbf{x}_t))]\|^2$$

uncond. output / intermediate head · uncond. branch

IG

Zhou et al., 2026

$$\mathcal{L}_{\text{IG}} = \mathbb{E} \|\mathcal{D}_f(\mathbf{x}_t) - \mathbf{x}_0\|^2 + \gamma \mathbb{E} \|\mathcal{D}_i(\mathbf{x}_t) - \mathbf{x}_0\|^2$$

extra output; sampling: $D_i + w (D_f - D_i)$

SWIFT

this work

$$\mathcal{L}_{\text{SWIFT}} = \mathbb{E} \|\mathcal{D}_{\theta}(\mathbf{x}_t) - [\mathbf{x}_0 + \alpha (\mathbf{x}_0 - \mathcal{D}_{\text{weak}}(\mathbf{x}_t))]\|^2$$

a separate small model, pretrained and frozen

sg = stop-gradient

What a frozen reference buys

- **A fixed target.** EMA or detached predictions change during training; SWIFT's target does not depend on the main model.
- **Nothing kept.** No unconditional branch, no extra head, no guidance scale at sampling.
- **GFT Method:** it's a special case of SWIFT, just joint training of weak model and base model.

Gaussian mixture with exact D^*

With D^* known in closed form, SWIFT lands closest to it at every noise level, at $d = 16$ and $d = 8$

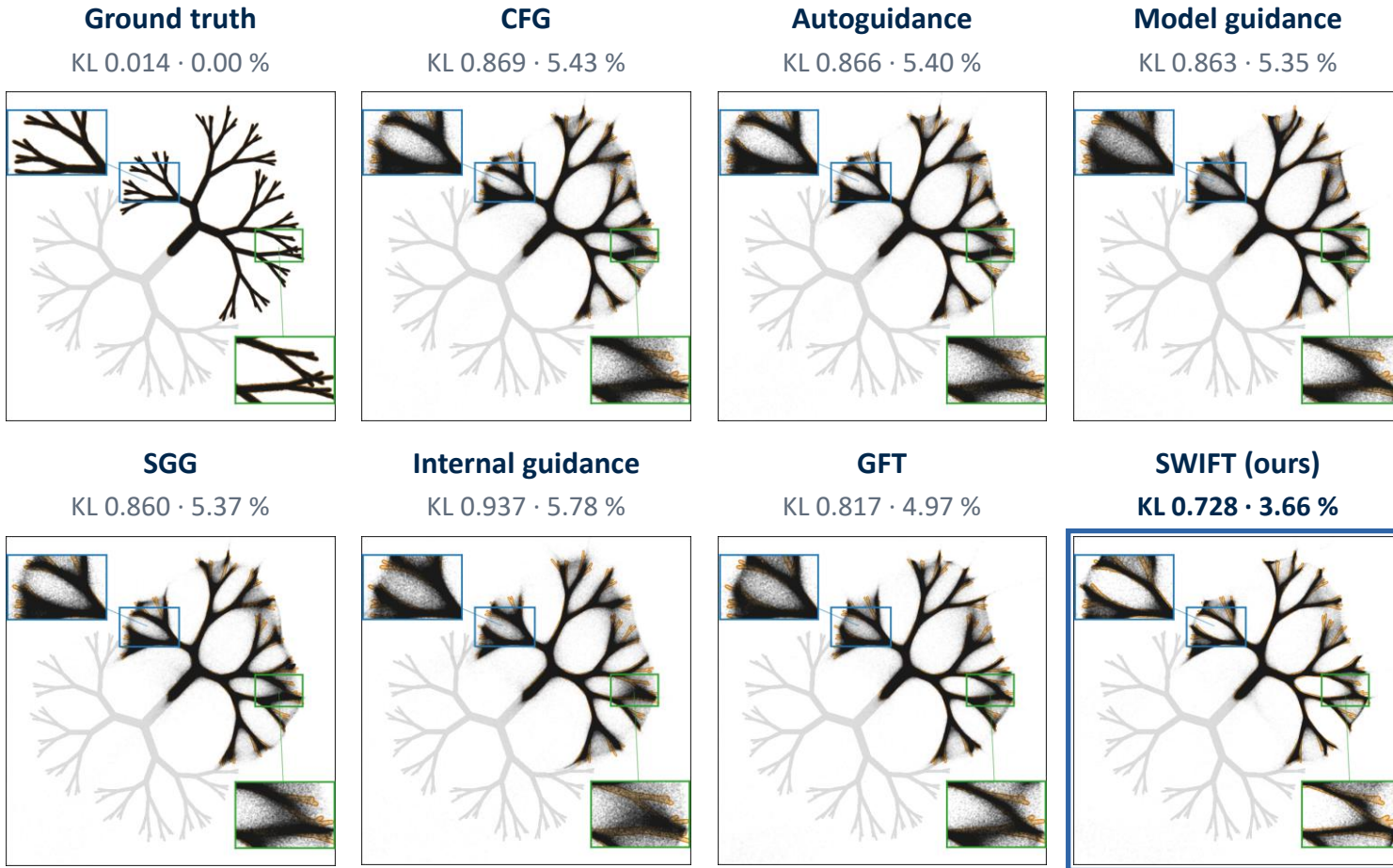
Method	d = 16 (40 seeds)					d = 8 (5 seeds)				
	$\sigma = 0.05$	0.1	0.2	0.5	1.0	$\sigma = 0.05$	0.1	0.2	0.5	1.0
baseline	0.3100	0.2600	0.2311	0.2125	0.2272	0.1853	0.1574	0.1378	0.1154	0.1232
SWIFT	0.2566	0.2175	0.1952	0.1865	0.1988	0.1450	0.1202	0.1120	0.1018	0.1189
MG	0.3179	0.2685	0.2388	0.2216	0.2358	0.1913	0.1659	0.1491	0.1246	0.1356
GFT	0.2617	0.2203	0.1972	0.1904	0.2075	0.1544	0.1264	0.1195	0.1095	0.1257
IG	0.3418	0.2885	0.2522	0.2258	0.2312	0.1976	0.1657	0.1468	0.1301	0.1330

Paper, Table 4: mean distance to the class-A posterior mean D^* (lower is better); each method at its best setting; bold = best per column.

- **SWIFT is closest at every noise level, in both dimensions:** 12–17 % below the baseline at $d = 16$
- **GFT is next;** MG and IG are worse than the baseline

2-D toy: branching support

On a tree-shaped 2-D mixture, SWIFT keeps samples on the branches — best of seven methods



- Tree-shaped mixture (Karras et al., 2024); each method tuned independently
- **SWIFT is best on all three metrics:**
KL 0.728, outliers 3.66 %, log p +0.08
- Next best, GFT: 0.817, 4.97 %, -0.40

19 ImageNet-256 configurations on two testbeds; in each comparison, only the target differs

DiffusionBench — 18 configurations

- **Varies:** Across the RAE, VAE, E2E, Pixel Diffusion.
- **Main model:** 615M DiT, 80 epochs
- **Small model:** half width ($\approx 155\text{M}$), 40 epochs, frozen

RAEv2 on DINOv3-L — 1 configuration

- **Tests:** a larger, state-of-the-art model
- **Main model:** 875M, up to 160 epochs
- **Small model:** $\approx \frac{1}{2}$ the parameters, pretrained checkpoint, frozen

How every comparison is made

- **Baseline:** the same setup with $\alpha = 0$ (or the released checkpoint)
- **One recipe:** $\alpha = 0.4$ everywhere, never tuned per configuration
- **Guidance:** scale tuned separately for each model
- **Metrics:** FID-50k, sFID, precision / recall, FD_r and MIND

Main result: sub-1.0 FID

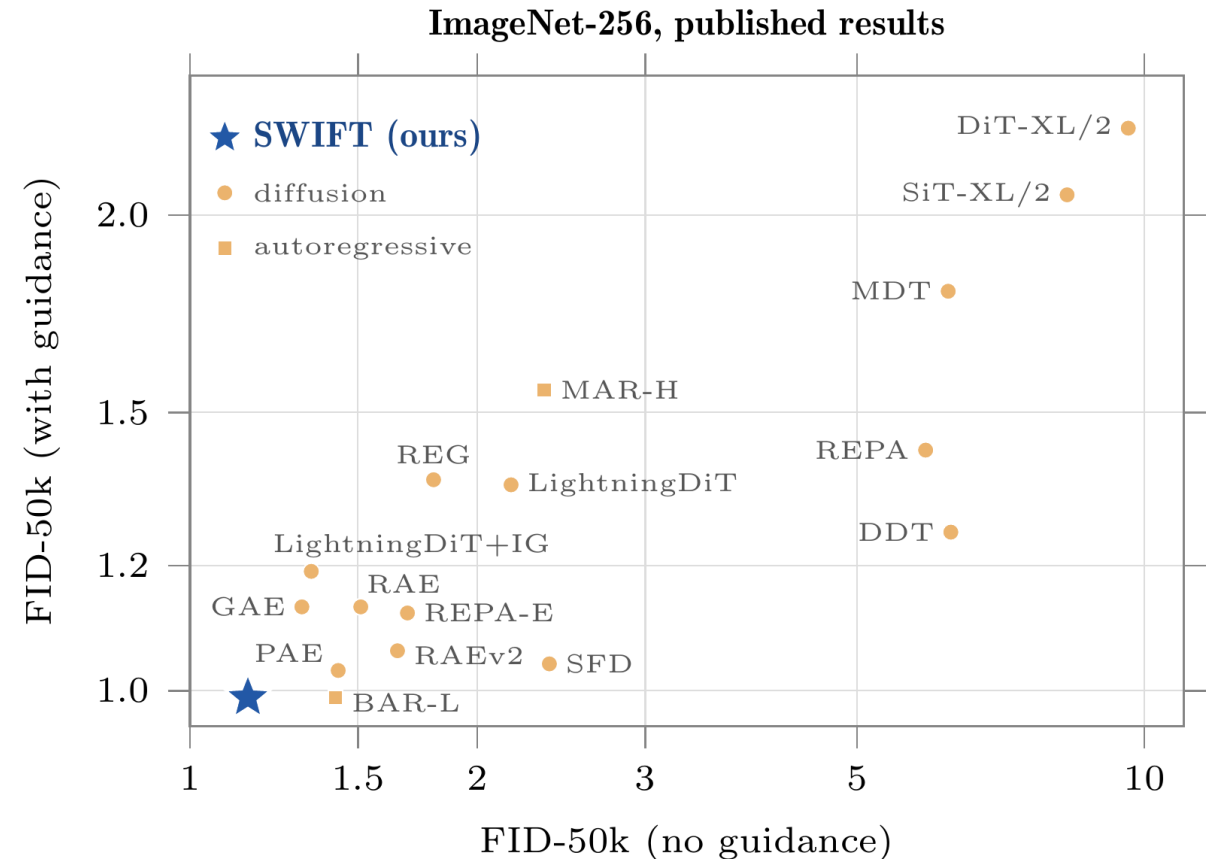
On RAEv2, SWIFT lowers FID with and without guidance — same network, same sampler

FID-50k (lower is better)

	Baseline	SWIFT
Without guidance	1.56	1.15
With guidance (retuned)	1.04	0.99

Baseline: epoch 160. SWIFT: best checkpoint (epoch 140) within the same 160-epoch budget.

- **Cost:** +12 % FLOPs per epoch
- **Inference:** unchanged — one network, same sampler



Paper, Fig. 1: published results ★ = SWIFT.

ImageNet-256 samples

Class-conditional samples from the SWIFT model with FID 0.99 (RAEv2 / DINOv3-L, 50-step ODE)



Selected experiment results

One configuration per family: SWIFT is better with and without guidance in every row

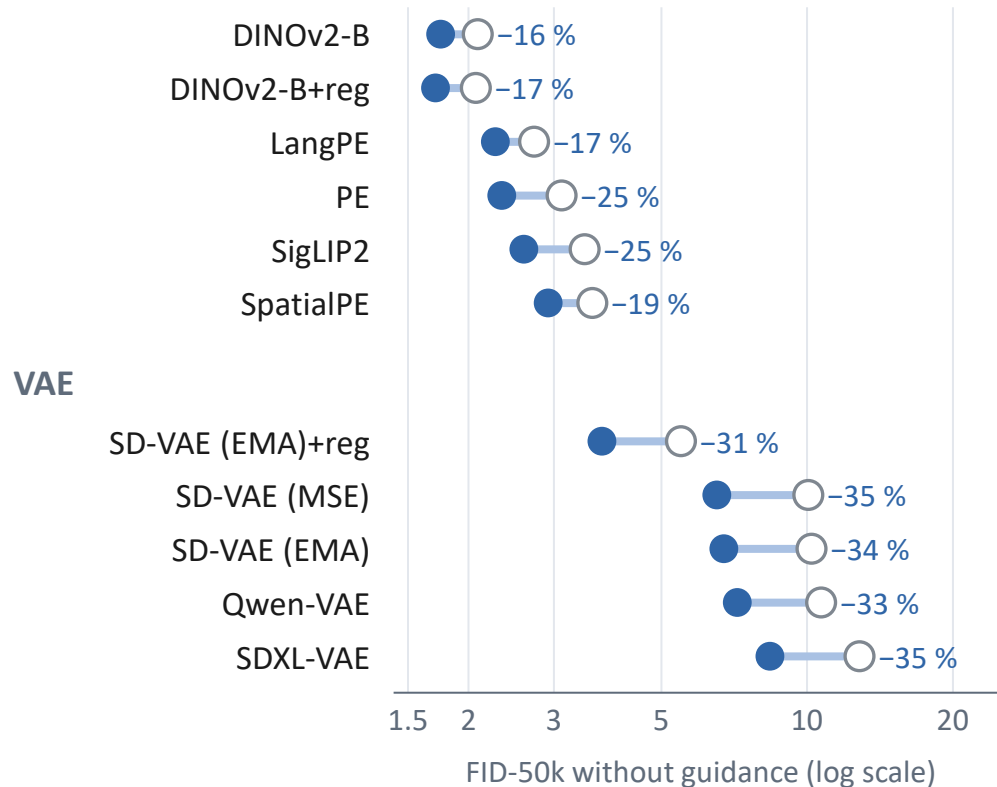
Family: representation	Without guidance						With guidance					
	FID↓	sFID↓	$\overline{FD}_r$ ↓	$\overline{MIND}$ ↓	Prec.↑	Rec.↑	FID↓	sFID↓	$\overline{FD}_r$ ↓	$\overline{MIND}$ ↓	Prec.↑	Rec.↑
<i>615M parameters, 80 epochs; guidance = CFG</i>												
RAE: DINOv2-B	2.09	4.48	4.13	21.34	0.83	0.57	1.98	4.50	4.17	20.71	0.81	0.59
+ SWIFT	1.75	4.26	3.69	21.54	0.81	0.61	1.59	4.30	3.57	19.40	0.80	0.62
VAE: SD-VAE (EMA)+reg	5.49	5.52	11.08	87.84	0.71	0.66	2.32	4.70	7.39	32.52	0.80	0.60
+ SWIFT	3.77	5.06	8.44	57.83	0.73	0.66	1.81	4.42	5.93	26.15	0.80	0.61
E2E: Qwen-VAE	4.94	5.23	9.17	95.20	0.72	0.66	1.50	4.83	4.73	24.40	0.80	0.62
+ SWIFT	3.23	4.89	6.47	55.63	0.73	0.66	1.40	4.59	3.33	23.40	0.80	0.63
Pixel: PixNerd	20.37	6.14	24.52	384.17	0.61	0.63	4.17	5.50	9.50	42.66	0.65	0.63
+ SWIFT	10.31	5.38	15.28	197.54	0.70	0.61	3.38	4.95	7.75	35.12	0.68	0.62
<i>875M parameters, up to 160 epochs; guidance = internal guidance</i>												
RAE: DINOv3-L	1.56	4.10	3.41	20.01	0.82	0.65	1.04	4.00	3.31	15.67	0.82	0.61
+ SWIFT	1.15	3.60	2.94	18.03	0.81	0.67	0.99	3.80	3.07	14.45	0.81	0.63

Paper, Table 1. Same configuration and sampler within each pair; guidance tuned separately where both were re-run.

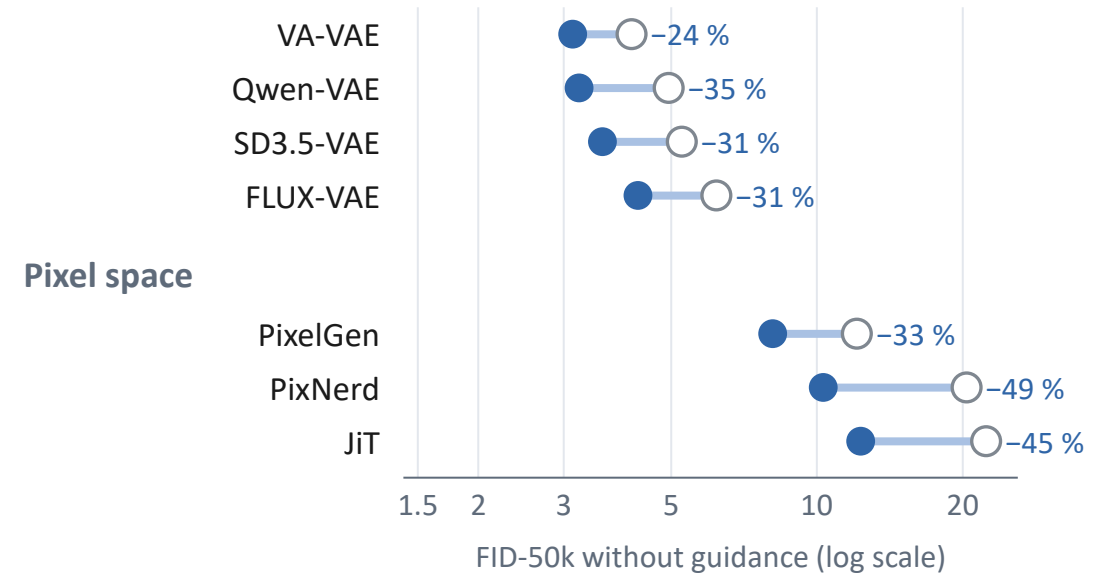
SWIFT helps in every 18 configurations

The same untuned recipe lowers unguided FID in all 18 configurations, by 16–49 %

Representation AE



End-to-end AE



○ baseline ($\alpha = 0$)

● SWIFT, one fixed recipe ($\alpha = 0.4$)

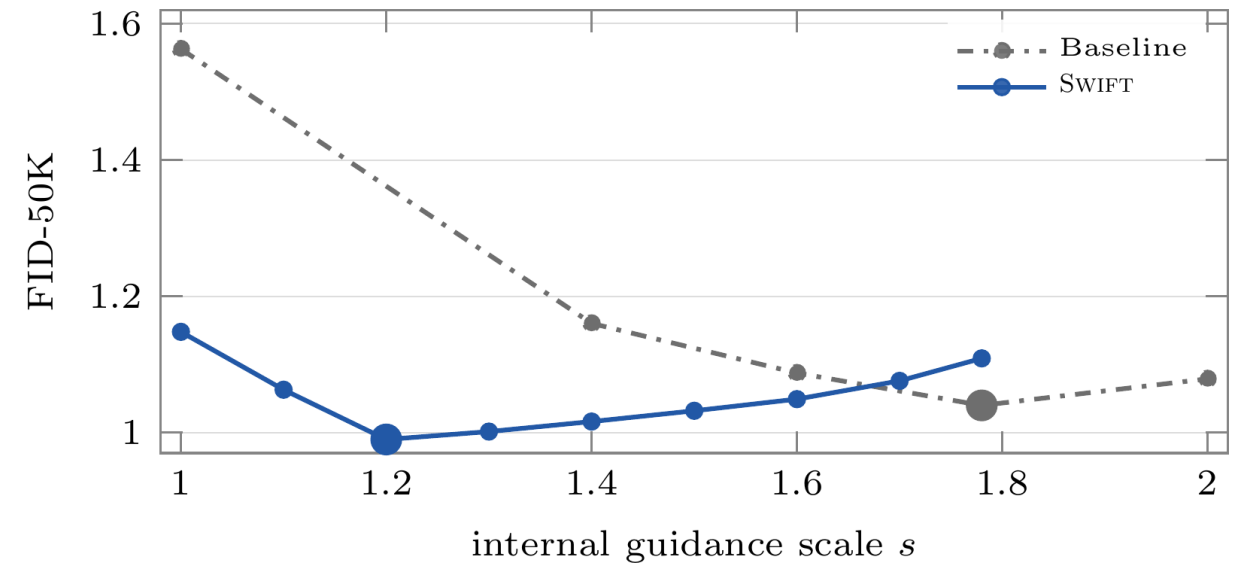
- Median **31 %**; the three best baselines improve by 16–17 %
- **269 of 285** metric comparisons favor SWIFT (7 ties)

Combining with autoguidance

Guidance still helps on top of SWIFT, at a smaller scale — the two corrections add up

	Base	SWIFT	Guided	SWIFT+ guidance
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

FID-50k (paper, Fig. 4a). Guided: autoguidance; RAEv2: internal guidance.



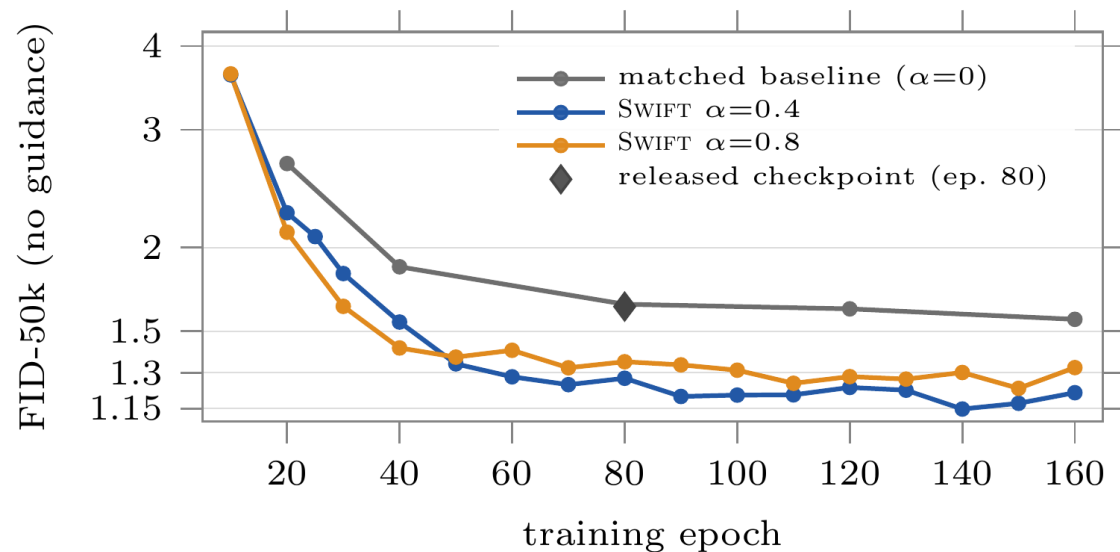
RAEv2 (Fig. 4b): FID vs. internal-guidance scale; large dots = optima.

- **Lower guided FID in all 19 configurations:** 7–33 % (median 22 %) with CFG; 1.04 → 0.99 on RAEv2
- **Same ordering in all five:** baseline → SWIFT → baseline + guidance → **SWIFT + guidance**
- **Less guidance needed:** the optimal scale drops from 1.78 to 1.20

Training curve

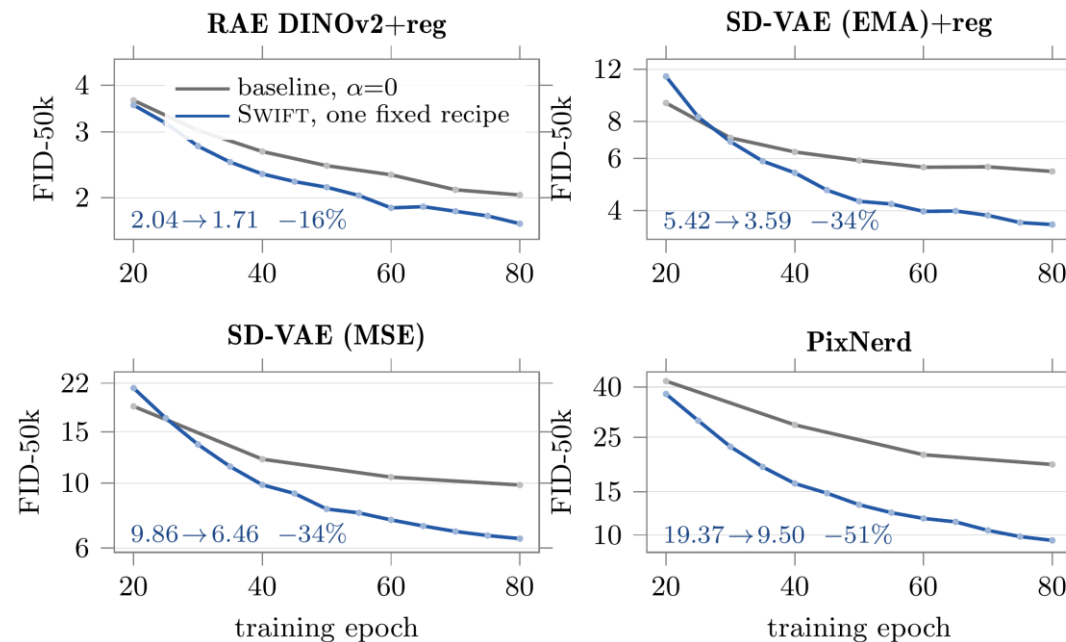
The baseline does not catch up: SWIFT stays ahead to 160 epochs on RAEv2 and 240 on DINOv2-B

RAEv2 / DINOv3-L — 160 epochs

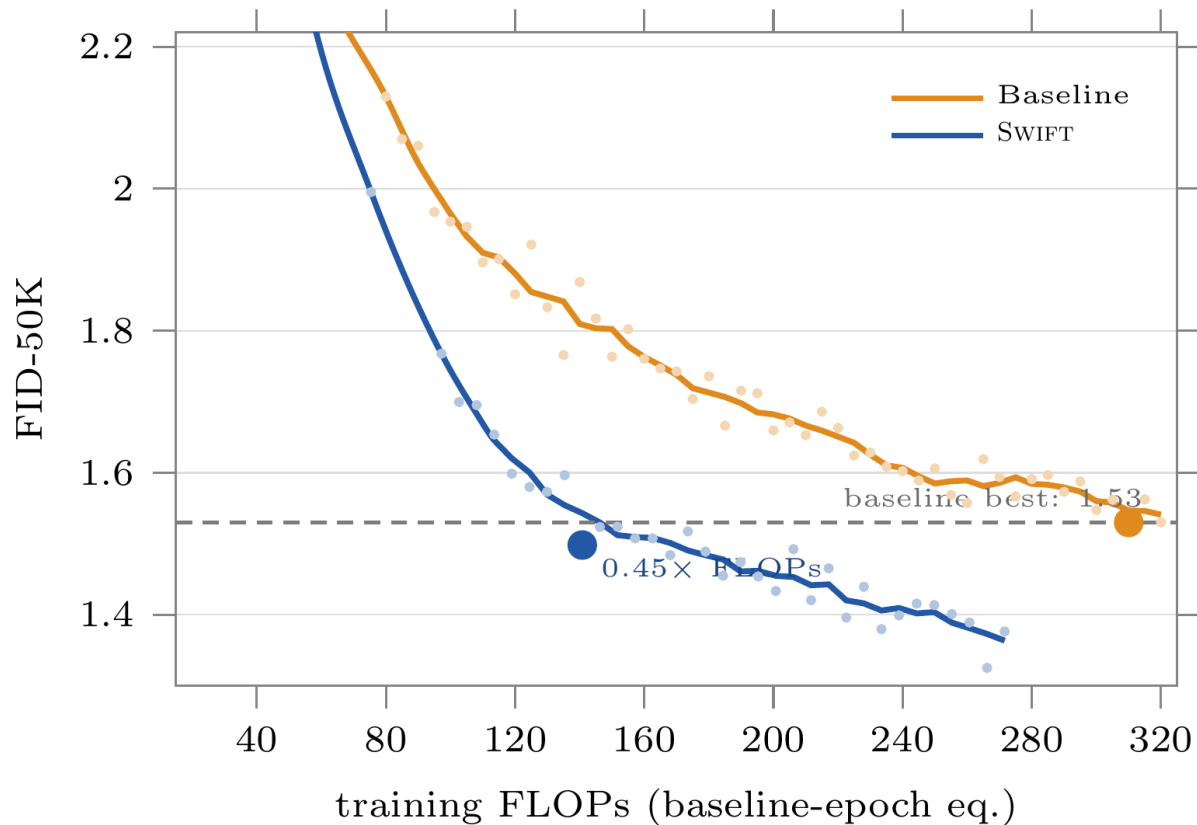


Unguided FID-50k; $\blacklozenge$ = released epoch-80 checkpoint.

Four more configurations — 80 epochs



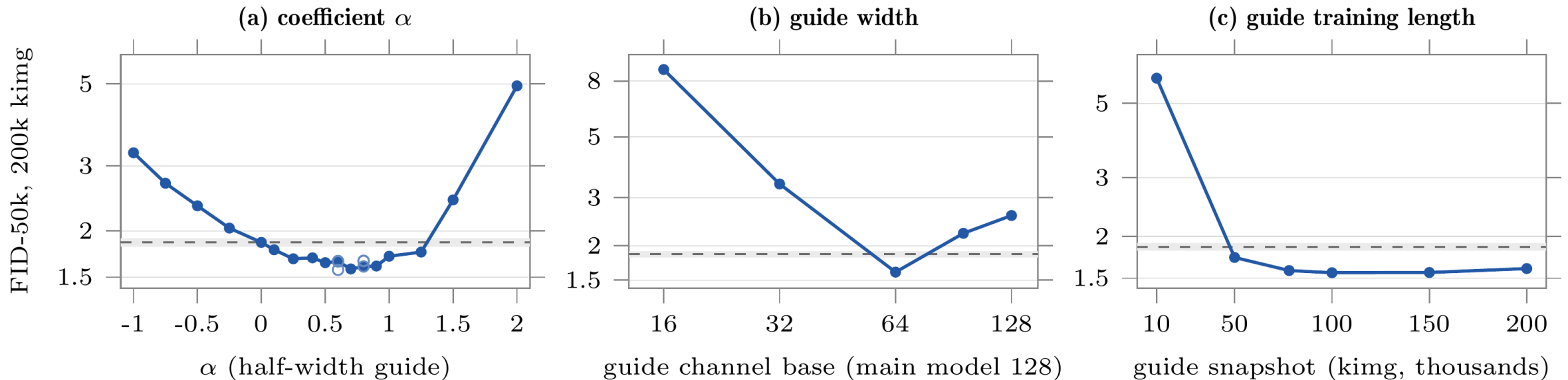
Charging the small model's cost, SWIFT beats the best 320-epoch baseline with 0.45× the FLOPs



Paper, Fig. 5 (DINOv2-B): small-model pretraining and the 9 % overhead charged to SWIFT.

- **Cost charged:** +9 % FLOPs per epoch, plus ≈ 10 baseline-epochs for the small model
- **DINOv2-B:** epoch-120 SWIFT (1.50) beats the best 320-epoch baseline (1.53) at **0.45×** the FLOPs
- **RAEv2:** epoch-50 SWIFT (1.34) beats the epoch-160 baseline (1.56) at **0.44×**
- **For reference:** MG 1.23×, SGG 1.22× per iteration; GFT +10–20 %

We have done a series of experiments on Cifar-10 to decide the choice of α and the Weak Model



Paper, Fig. 6 (CIFAR-10, FID-50k). Dashed line: standard training ($\alpha = 0$), 1.86.

- α : need to be 0.4-0.8; the ImageNet runs use **0.4, untuned**
- **Width**: half width is best (1.60 vs. 2.58 for a full-width small model)
- **Training length**: around half of the epoch number of the base model.

Thank you

Questions and comments are welcome!

SWIFT In one line:

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

Train on it, discard the small model, sample as usual.

Yanjin He · yanjinhe.github.io · yanjinhe@umich.edu



SWIFT samples (RAEv2, ImageNet-256)