SALD: SELF-REFERENCED ADVANTAGE LEARNING FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work on language-model adaptation has shown that single models can obtain informative training signals by evaluating their behavior in demonstration- or feedback-augmented contexts, with the help of a teacher network, which is driven by the students' learned parameters. Inspired by this internal-reference principle, we investigate how diffusion models can identify self-referenced training signals without external demonstrations or teacher networks. We introduce SALD, a self-referenced training framework that evaluates each image-caption pair at two noise levels using the same model. The easier, lower-noise path is evaluated without gradient tracking to provide a reference, while the harder, higher-noise path provides the training gradient. Rather than directly distilling the easy-path prediction, SALD uses the difference between two path errors to adapt the hard-path objective. The proposed Advantage-Guided Diffusion (AGD) converts this relative error into a differentiable sample-level weight. Temporal Advantage Memory (TAM) accumulates relative difficulty across training and adapts the future gap between the two noise levels. Spectral Advantage Decomposition (SAD) further compares the residual power spectra of the two paths and constructs a differentiable, frequency-derived latent-element weight. All components share a single set of model parameters, requiring neither an external teacher network nor additional trainable parameters during training or inference, and no modification to the inference procedure. Experiments across multiple architectures and datasets demonstrate consistent improvements in generation quality, while component-wise ablations quantify the contributions of the proposed components.



Figure 1: Images generated by SALD using the SANA 0.6B model at a resolution of 1024×1024.

1 INTRODUCTION

Diffusion Ho et al. (2020); Song et al. (2020) and flow-matching Lipman et al. (2022); Albergo et al. (2025) based generative models have become standard backbones for high-fidelity image synthesis, with Diffusion Transformers further improving scalability and generation quality Peebles & Xie (2023). In parallel, recent work on language-model adaptation has shown that a model can obtain richer online training signals by evaluating its behavior under additional information. Self-Distillation Fine-Tuning conditions a self-teacher on expert demonstrations, whereas Self-Distillation Policy

We are happy to release the code upon acceptance.

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.

Optimization conditions a self-teacher on feedback obtained from the model's own attempts Shenfeld et al. (2026); Hübotter et al. (2026). These approaches show that useful supervision can be constructed without relying on a stronger external teacher. Transferring this principle to diffusion training is not straightforward: diffusion models do not naturally receive textual demonstrations or retrospective feedback, and their training states are sampled from a prescribed corruption process rather than from autoregressive policy rollouts. Moreover, standard diffusion objectives still average gradients uniformly across training pairs and latent elements, although their difficulty varies substantially across images, captions, noise levels, and visual details Vincent (2011); Choi et al. (2022). Neural networks also exhibit spectral bias, learning coarse low-frequency structure before fine high-frequency detail Rahaman et al. (2019); Wang & Pehlevan (2025a). Consequently, uniform training can continue to emphasize already learned signals while underweighting examples and visual components that remain difficult.

Can a diffusion model construct its own easier reference and use it to adapt training across harder examples, prompts, noise levels, and visual frequencies, without teachers, demonstrations, or feedback datasets?

A unified answer remains missing because existing methods address only parts of this problem. Noise-level weighting Karras et al. (2022); Esser et al. (2024); Hang et al. (2025) adjusts training according to timestep or signal-to-noise ratio, but normally applies the same rule to all examples evaluated at a given noise level. Noise selection methods show that different noise instances provide unequal training signals Qi et al. (2024a), but they do not maintain a persistent estimate of how the difficulty of an individual image-caption pair changes throughout training. Curriculum learning Bengio et al. (2009) orders examples by difficulty, but typically depends on a fixed or externally defined curriculum and are not adaptive noise selection. Distributionally robust optimization emphasizes difficult subpopulations Sagawa et al. (2019), whereas frequency-aware objectives address spectral imbalance; neither directly combines the sample-level, temporal, and spectral difficulty in diffusion training. Likewise, demonstration- or feedback-conditioned language-model self-distillation cannot be transferred directly because diffusion training provides neither an in-context demonstration, nor a feedback-conditioned output distribution to distill Shenfeld et al. (2026); Hübotter et al. (2026). These limitations motivate a self-referenced diffusion objective in which an easier view of the same training pair provides a relative difficulty signal rather than a teacher prediction target.

We introduce SALD, which evaluates every image-caption pair at two noise levels using one shared model: an easier low-noise path and a harder high-noise path. The easy path is evaluated without gradient tracking and serves as an internal reference, while model parameters are updated through the hard path. SALD does not directly distill the easy-path prediction into the hard path. Instead, it measures the increase in weighted prediction error from the easy path to the hard path and uses this relative signal to adapt the hard-path objective. *Advantage-Guided Diffusion* (AGD) converts the scalar error difference into a normalized, differentiable sample weight, emphasizing training pairs whose error grows under the harder condition. *Temporal Advantage Memory* (TAM) maintains an exponential moving average of relative difficulty for every training pair and uses it to adapt future noise-level gaps, forming a continuous per-example curriculum Bengio et al. (2009). *Spectral Advantage Decomposition* (SAD) compares the residual power spectra of the two paths and constructs a differentiable, frequency-derived latent-element weight, providing a finer training signal related to spectral bias Rahaman et al. (2019); Wang & Pehlevan (2025a). All components reuse the same set of model parameters and require neither a teacher network nor additional trainable model parameters during training or inference. The method introduces no change to the inference procedure. Our contributions are summarized as follows:

- We introduce AGD, a self-referenced weighting mechanism that estimates sample importance from the relative error between hard and easy noise paths, without any teachers, demonstration datasets, or architectural changes.
- We introduce TAM and SAD to extend relative difficulty across training history and spectral structure, enabling an adaptive per-example noise curriculum and frequency-derived, latent-element weighting.
- We characterize the optimization behavior of the resulting differentiable weighting objective and evaluate SALD across multiple architectures and datasets. The experiments demonstrate consistent gains in generation quality, while component-wise ablations quantify the contributions of AGD, TAM, and SAD.

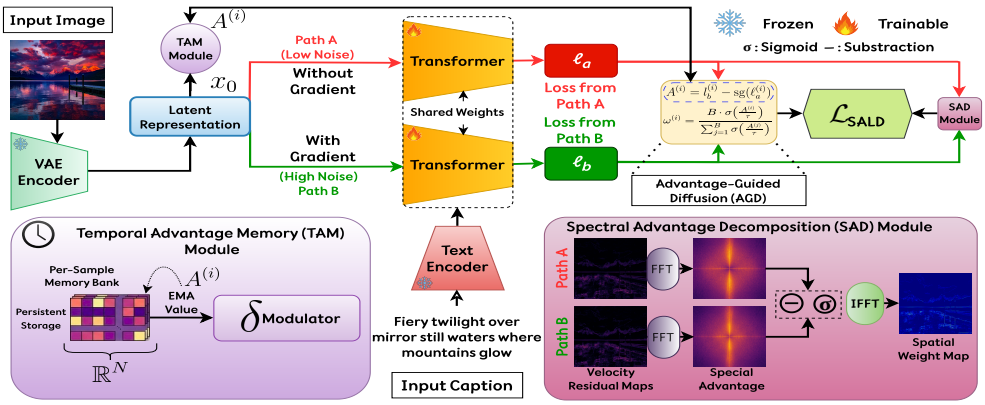


Figure 2: *SALD framework.* A frozen VAE encodes the input latent, processed through two paths: Low-Noise Path A (w/o gradient) and High-Noise Path B (with gradient). AGD computes an advantage for sample reweighting, while TAM stabilizes it via per-sample EMA, forming a persistent curriculum across timestep gaps. SAD compares FFT power spectra of velocity residual maps, applies a sigmoid gate, and reconstructs a spatial weight map via IFFT to focus gradients on high-frequency error regions. Together, these define $\mathcal{L}_{\text{SALD}}$ for adaptive gradient allocation across samples.

2 Proposed Approach

The same diffusion model evaluates each image-caption pair at two noise levels. The lower-noise evaluation forms an *easy path*, which is computed without gradient tracking and serves as an internal reference. The higher-noise evaluation forms a *hard path*, through which the model parameters are updated. By comparing these paths, SALD addresses three questions: *(i) Which examples are currently difficult? (ii) How large should the easy-to-hard noise gap be for each example? and (iii) At which frequencies does the hard-path residual remain large?* Advantage-Guided Diffusion (AGD), Temporal Advantage Memory (TAM), and Spectral Advantage Decomposition (SAD) address these questions, respectively. All components share the same model parameters; therefore, SALD requires neither an external teacher nor an additional trainable model during training or inference.

2.1 Flow-Matching Setup

Let $x_0 \in \mathbb{R}^{C\times H\times W}$ be the latent representation of an image, let c be its text condition, and let $\epsilon \sim \mathcal{N}(0, I)$ be Gaussian noise. At noise level $t \in [0, 1]$, the model input is

$$z_t = (1 - t)x_0 + t\epsilon. \tag{1}$$

The desired velocity is simply $u = \epsilon - x_0$. The model $v_\theta(z_t, t, c)$ predicts this velocity. To keep the remaining notation short, define the residual (velocity prediction error) and its scalar loss by

$$r(t, \epsilon) = v_\theta(z_t, t, c) - (\epsilon - x_0), \tag{2}$$

$$\ell(t, \epsilon) = \frac{w(t)}{D} \|r(t, \epsilon)\|_2^2, \qquad D = CHW, \tag{3}$$

where $w(t)$ is the usual time-dependent loss weight. SALD now chooses two noise levels for the same training samples: $t_{\text{e}} < t_{\text{h}}$, where t_{e} is the easy level and t_{h} is the hard level. The two paths use independent noise samples ϵ_{e} and ϵ_{h}. We write

$$r_{\text{e}} = r(t_{\text{e}}, \epsilon_{\text{e}}), \qquad \ell_{\text{e}} = \ell(t_{\text{e}}, \epsilon_{\text{e}}), \tag{4}$$

$$r_{\text{h}} = r(t_{\text{h}}, \epsilon_{\text{h}}), \qquad \ell_{\text{h}} = \ell(t_{\text{h}}, \epsilon_{\text{h}}). \tag{5}$$

The easy quantities are computed without gradient tracking, whereas the hard quantities remain differentiable. We choose the two noise levels such that: $t_{\text{e}}^{(i)} < t_{\text{h}}^{(i)}$, where $t_{\text{e}}^{(i)}$ and $t_{\text{h}}^{(i)}$ denote the easy and hard noise levels for training sample i, respectively. Their separation is controlled by a noise-level gap δ_i (defined in Eq. 10). A small δ_i produces two similar noise states, whereas a large δ_i creates a more difficult hard state. Temporal Advantage Memory (TAM) later adapts δ_i according to the historical difficulty of each training pair. In implementation, δ_i corresponds to the number of steps separating the two levels in the discretized noise schedule.

2.2 ADVANTAGE-GUIDED DIFFUSION (AGD)

For each sample i, AGD measures how much the loss increases from the easy path to the hard path:

$$A_i = \ell_{\mathrm{h},i} - \mathrm{sg}(\ell_{\mathrm{e},i})\,. \tag{6}$$

Here, $\mathrm{sg}(\cdot)$ means stop-gradient. Importantly, only the easy reference is stopped. *This is an approximation to obtain the demonstration or feedback sample, which is difficult to obtain for self-distillation Shenfeld et al. (2026); Hübotter et al. (2026) in image generations*. The advantage itself is *not* detached, because it backpropagates the gradient for the hard path as: $\nabla_\theta A_i = \nabla_\theta \ell_{\mathrm{h},i}$

A large positive A_i means that pair i becomes considerably more difficult at the higher noise level. Further, A_i is normalized into a positive mini-batch weight. For a batch of size B, define

$$q_i = \mathrm{sigmoid}(A_i/\tau), \qquad \omega_i = \frac{B q_i}{\sum_{j=1}^{B} q_j}, \tag{7}$$

where $\tau > 0$ is a temperature. These weights satisfy

$$\frac{1}{B}\sum_{i=1}^{B} \omega_i = 1. \tag{8}$$

Thus, AGD changes the relative importance of examples without altering the average batch weight. Since A_i is differentiable, ω_i is also differentiable through the hard path. The denominator in Eq. equation 7 further couples the weights of all examples in the batch. Harder samples therefore receive larger weights, and optimizing them encourages their losses to move closer to those of easier samples, enabling the model to generate outputs that progressively reduce the loss on difficult examples.

2.3 TEMPORAL ADVANTAGE MEMORY (TAM)

A single advantage value can be noisy because new Gaussian noise is sampled at every iteration. TAM therefore stores one running difficulty value m_i for each training sample. The memory vector $m \in \mathbb{R}^N$ is initialized to zero and is a non-trainable buffer.

Let Δ be the largest allowed gap s be the current training step, and $K > 1$ be the number of curriculum steps. A base curriculum that starts at one and gradually reaches Δ is

$$\delta_{\mathrm{base}}(s) = \begin{cases} 1 + \left\lfloor \dfrac{(\Delta - 1)s}{K - 1} \right\rfloor, & 0 \le s < K, \\ d, \quad d \sim \mathrm{Uniform}\{1, \ldots, \Delta_{\max}\}, & s \ge K. \end{cases} \tag{9}$$

Before the current pair is evaluated, its stored memory selects the actual gap:

$$\delta_i = \mathrm{clip}\big(\mathrm{round}\big[\delta_{\mathrm{base}}(s)\big(1 + \lambda \tanh(m_i)\big)\big]\,, 1, \Delta_{\max}\big)\,, \tag{10}$$

where $\lambda \ge 0$ controls the strength of memory modulation. After the current advantage has been computed, TAM updates the memory:

$$m_i \leftarrow \alpha m_i + (1 - \alpha) A_i, \qquad 0 < \alpha < 1. \tag{11}$$

This assignment stores the numerical value of A_i; it does not create a gradient path through the memory buffer. The updated value is used the next time the same pair is visited. Persistent positive memory increases future noise gaps, while negative memory decreases them. The $\tanh$ and clipping operations keep every gap within the valid range $(1 - \lambda, 1 + \lambda)$.

2.4 SPECTRAL ADVANTAGE DECOMPOSITION (SAD)

The proposed AGD assigns a single scalar weight to an entire example at each frequency. SAD adds frequency information by comparing the residual maps $r_{\mathrm{e},i}$ and $r_{\mathrm{h},i}$ in the frequency domain for the easy and hard respectively. Let $\mathcal{F}_r$ be an orthonormal two-dimensional real FFT applied independently to every latent channel. The easy and hard residual power spectra are

$$P_{\mathrm{e},i} = |\mathcal{F}_r(r_{\mathrm{e},i})|^{\odot 2}\,, \qquad P_{\mathrm{h},i} = |\mathcal{F}_r(r_{\mathrm{h},i})|^{\odot 2}\,. \tag{12}$$

Their spectral advantage is

$$S_i = P_{\mathrm{h},i} - \mathrm{sg}(P_{\mathrm{e},i})\,. \tag{13}$$

Again, only the easy reference is stopped. Therefore, S_i remains differentiable through the hard residual. A positive value of S_i at a frequency means that the hard path contains more residual energy than the easy reference at that frequency.

The spectral advantage is mapped to a bounded gate; the gate is then back-projected to the latent domain and normalized:

$$G_i = \mathrm{sigmoid}(S_i/\tau_s) \quad \& \quad \widetilde{M}_i = \left|\mathcal{F}_r^{-1}(G_i)\right|, \quad \tau_s > 0 \tag{14}$$

$$M_i = \frac{D\,\widetilde{M}_i}{\sum_{p=1}^{D} \widetilde{M}_{i,p} + \varepsilon_m}, \qquad \varepsilon_m = 10^{-8}. \tag{15}$$

The normalization keeps the mean weight approximately equal to one. Crucially, there is no stop-gradient in Eq. 15; M_i is differentiable through $P_{\mathrm{h},i}$.

M_i is a latent-domain weight obtained from the frequency-wise comparison of the easy and hard residuals. Because SAD uses residual power spectra, it discards the Fourier phase information that represents spatial location. Therefore, M_i indicates which frequency patterns remain difficult, but it does not identify where the corresponding errors occur in the latent representation. We consequently describe M_i as a frequency-derived latent weight rather than a localized pixel-level advantage map.

2.5 Unified SALD Objective

For pair i, SAD first weights the squared hard-path residual element by element:

$$\ell_i^{\mathrm{SAD}} = \frac{w(t_{\mathrm{h},i})}{D} \sum_{p=1}^{D} M_{i,p}\, r_{h,i,p}^2 \tag{16}$$

AGD then weights the examples in the mini-batch:

$$\mathcal{L}_{\mathrm{SALD}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \omega_i \ell_i^{\mathrm{SAD}} \tag{17}$$

The gradient follows the displayed equations exactly. Since neither ω_i nor M_i is detached and $\nabla_\theta \ell_i^{\mathrm{SAD}}$ includes the gradient through M_i. The only stopped branches are the easy-path references. TAM is stored as a non-trainable state and is not backpropagated through time. SALD uses two forward evaluations, but only the hard evaluation stores a backward graph. It introduces an N-value TAM buffer and FFT operations, but no additional trainable or non-trainable (teacher) model parameters are required during training or inference, and the inference procedure is unchanged.

Finally, Eq. 17 contains no term that directly matches the hard prediction to the easy prediction. The easy path supplies a reference for constructing adaptive weights; it is not a teacher output target. For this reason, the method is more precisely described as *self-referenced advantage learning* than as teacher-student self-distillation.

2.6 Theoretical Interpretation

Theorem 1 (Conditional convergence of the differentiable objective)**.** *Assume that F_m is lower bounded by $F_m^\star$ and is L-smooth. Let $g_s = \nabla_\theta \widehat{\mathcal{L}}_{\mathrm{SALD}}(\theta_s; \xi_s, m)$ include all gradient paths through ω_i and M_i. Assume*

$$\mathbb{E}[g_s \mid \theta_s] = \nabla F_m(\theta_s), \qquad \mathbb{E}\Big[\|g_s - \nabla F_m(\theta_s)\|_2^2\Big] \le \nu^2. \tag{18}$$

For the update $\theta_{s+1} = \theta_s - \eta g_s$ with $0 < \eta \le 1/L$,

$$\frac{1}{J}\sum_{s=0}^{J-1} \mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 \le \frac{2\big(F_m(\theta_0) - F_m^\star\big)}{\eta J} + L\eta\nu^2. \tag{19}$$

This standard result guarantees convergence toward a stationary point only under the stated assumptions. Exact statement supported by the spectral gate: for every frequency coordinate u,

$$G_i(u) > \frac{1}{2} \quad \Longleftrightarrow \quad P_{\mathrm{h},i}(u) > \mathrm{sg}(P_{\mathrm{e},i}(u))\,. \tag{20}$$

Thus, the gate is larger where the hard residual has more spectral energy than the easy reference. However, the final loss multiplies by M_i in the latent domain rather than multiplying Fourier coefficients directly by G_i. Table 7 in the Supplementary empirically support the theoretical claim and shows the faster and better convergence.

Table 1: Quantitative results on (512×512) resolution datasets. Best values per metric are in Blue.

Model	Method	COCO (2017)			MultiGen-20M			Flickr8k			CUB-200			Oxford 102 Flowers		
		FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓
SANA (1.6B)	Base	26.49	26.44	0.7411	26.44	26.25	0.7541	27.22	28.49	0.7336	21.63	26.01	0.7748	33.11	26.35	0.7896
	SpeeD	25.38	26.83	0.7409	26.11	26.26	0.7558	26.87	28.52	0.7324	21.53	26.28	0.7753	32.04	26.51	0.7782
	Temporal Diff	26.36	26.87	0.7405	26.31	26.29	0.7539	26.85	28.85	0.7325	20.72	26.25	0.7766	33.10	26.54	0.7801
	SRA	26.89	26.32	0.7461	26.05	25.89	0.7554	26.87	28.78	0.7341	20.78	26.18	0.7722	32.75	26.40	**0.7781**
	SALD	**22.76**	**26.88**	**0.7391**	**23.86**	**26.29**	**0.7533**	**26.75**	**28.88**	**0.7316**	**18.16**	**26.29**	**0.7537**	**27.37**	**26.57**	**0.7781**
SANA (0.6B)	Base	31.59	26.80	0.7408	31.93	26.68	0.7495	28.83	28.53	0.7351	26.06	26.40	0.7807	30.64	26.43	0.7805
	SpeeD	31.47	**27.06**	0.7411	31.57	26.69	0.7485	28.42	28.49	0.7313	25.98	**26.61**	0.7611	32.04	26.54	0.7753
	Temporal Diff	31.28	27.03	0.7424	31.67	26.66	0.7498	28.66	28.53	0.7330	26.18	26.53	0.7775	33.10	26.61	0.7736
	SRA	31.26	26.92	0.7397	31.63	26.56	0.7473	28.86	28.21	0.7314	24.94	26.56	0.7730	29.72	26.54	0.7748
	SALD	**29.31**	**27.06**	**0.7374**	**29.15**	**26.69**	**0.7472**	**27.91**	**28.59**	**0.7275**	**23.70**	**26.61**	**0.7686**	**25.83**	**26.65**	**0.7728**
PixArt-Σ	Base	29.26	25.82	0.7653	31.62	25.83	0.7306	30.83	28.44	0.7175	22.24	25.48	0.7491	37.88	25.37	0.7687
	SpeeD	29.25	25.80	0.7533	31.42	25.83	**0.7292**	30.76	28.44	0.7153	22.15	25.42	0.7459	37.16	25.39	0.7699
	Temporal Diff	29.21	25.85	0.7594	31.31	25.41	0.7306	30.71	28.36	0.7170	22.14	25.40	**0.7452**	37.49	25.39	0.7681
	SRA	29.21	25.88	0.7575	31.28	25.88	0.7297	30.72	28.42	0.7159	22.15	25.00	0.7566	37.69	25.38	0.7706
	SALD	**28.14**	**26.00**	**0.7516**	**30.47**	**25.93**	**0.7292**	**30.68**	**28.45**	**0.7141**	**21.71**	**25.48**	**0.7452**	**36.48**	**25.42**	**0.7682**

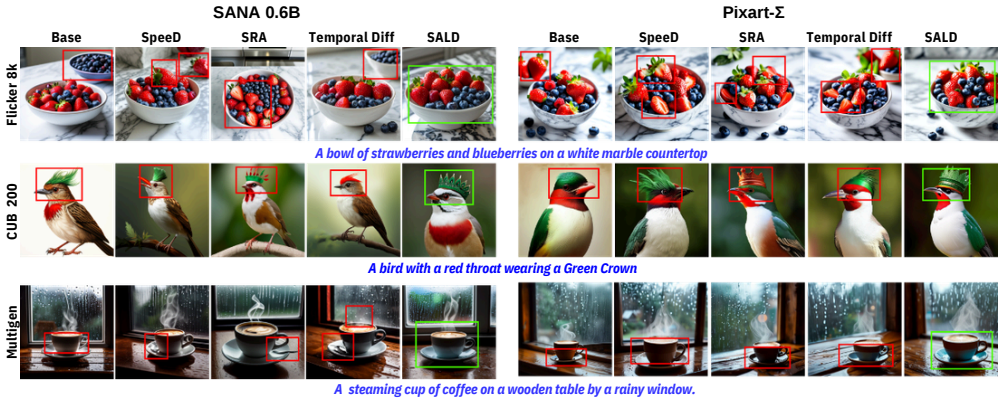


Figure 3: Visual comparison SALD with baselines. **Red boxes** indicate missing details or artifacts in latest baseline models. **Green boxes** shows our method generates better high-quality details.

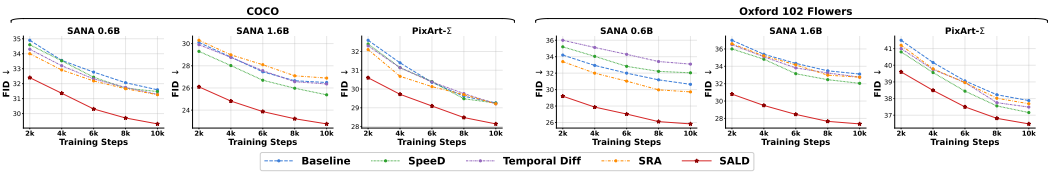


Figure 4: FID with training steps showing better and smoother convergence of SALD w.r.t. baselines.

3 Experiments

We conduct extensive experiments across 4 architectures, 7 benchmarks, and 2 resolutions to comprehensively evaluate the generality and effectiveness of our method.

Datasets. We evaluate scalability and precision on seven image–text datasets at 512×512 and 1024×1024 resolutions. At 512×512, MultiGen-20M Qin et al. (2023) and COCO (2017) Lin et al. (2014) assess large-scale performance: Fréchet Inception Distance (FID) Heusel et al. (2017) is computed on the COCO validation set; MultiGen-trained models are tested on COCO validation for cross-dataset generalization, while COCO-trained models use the same split. Fine-grained evaluation uses Flickr8k Hodosh et al. (2013), CUB-200-2011 Welinder et al. (2010); Xu et al. (2018), and Oxford Flowers Nilsback & Zisserman (2008); Reed et al. (2016) datasets. At 1024×1024 resolution we use MAGICK Burgert et al. (2024) and DALL·E 3 1M Egan et al. (2024). These 5 datasets use an 80/20 train–test split.

Implementation Details. All experiments are conducted on NVIDIA RTX A6000 GPUs under a unified setup using the AdamW optimizer Loshchilov & Hutter (2019) and mixed-precision (`bf16`) training. We evaluate SALD on four architectures: SANA (0.6B and 1.6B) Xie et al. (2024), PixArt-Σ Chen et al. (2024), and Flux.2 Klein Base (4B) Labs (2025). SANA and PixArt-Σ models are used

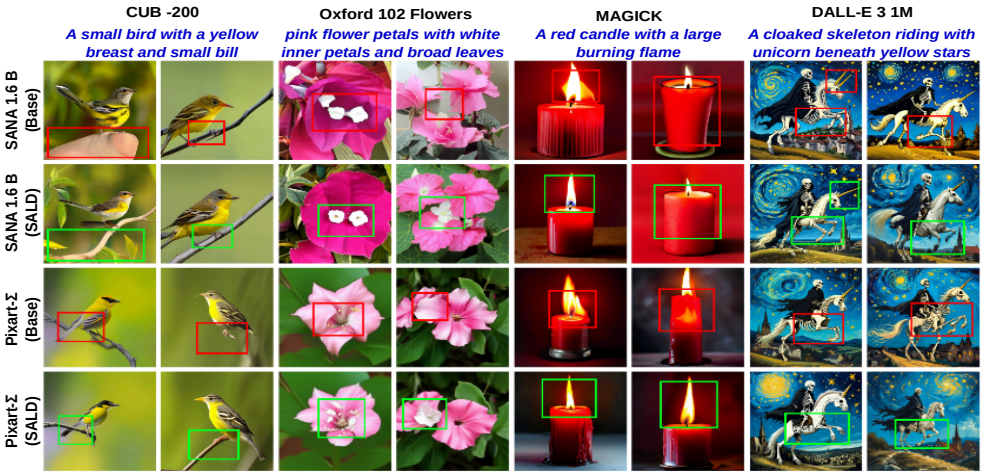


Figure 5: Visual comparison of generated samples. **Red boxes** indicate missing details or artifacts in latest baseline models. **Green boxes** highlight that our method generates better high-quality details. We use the same seed for all comparisons to ensure fairness. Captions are in ***Blue Color***.

for 512 resolution experiments, while the high-resolution variants of SANA (0.6B and 1.6B), Flux.2 Klein Base (4B) and PixArt-Σ are used for 1024 resolution experiments. All models are trained for 10k steps unless otherwise specified. Additional training details are mentioned in Appendix D.1.

Evaluation Protocol. We evaluate generated images using Fréchet Inception Distance (FID) Heusel et al. (2018), CLIP score Radford et al. (2021), and Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018). FID measures distributional similarity to real images (quality and diversity). CLIP score captures alignment with text prompts, while LPIPS assesses perceptual similarity, together evaluating fidelity, semantics, and realism. FID is computed using the Inception-V3 network with 2048-dimensional pool features. CLIP Score is computed using the CLIP ViT-L/14 model from OpenAI. LPIPS is computed using a VGG backbone. Further details are mentioned in Appendix D.1.

Table 2: Quantitative results on 1024×1024 resolution images. Best values per metric within each model are in Blue.

Model	Method	MAGICK			DALL·E 3		
		FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓
SANA 0.6B	Base	38.59	26.68	0.7470	14.68	29.96	0.7882
	SpeeD	38.27	26.65	0.7303	14.18	30.05	0.7878
	Temporal Diff	37.95	26.69	0.7294	13.84	30.38	0.7854
	SRA	38.15	26.70	0.7297	13.96	30.73	0.7839
	SALD	**36.74**	**26.85**	**0.7230**	**11.98**	**31.06**	**0.7826**
SANA 1.6B	Base	39.46	27.11	0.7947	16.19	30.95	0.8237
	SpeeD	39.34	27.28	0.7920	16.05	30.91	**0.8206**
	Temporal Diff	39.00	27.26	0.7929	16.04	30.97	0.8283
	SRA	39.26	27.11	0.7879	16.15	30.94	0.8246
	SALD	**37.39**	**27.54**	**0.7877**	**15.82**	**30.99**	**0.8206**
PixArt-Σ	Base	36.72	26.31	0.7311	12.03	30.23	0.8056
	SpeeD	36.56	**26.47**	0.7197	11.76	30.29	0.7844
	Temporal Diff	36.49	26.28	0.7295	11.65	30.66	0.7966
	SRA	36.52	26.44	0.7399	11.73	31.29	0.7864
	SALD	**35.38**	**26.47**	**0.7191**	**11.06**	**31.35**	**0.7700**
Flux.2 Klein	Base	35.93	24.93	0.7804	14.11	31.79	0.7666
	SpeeD	35.22	24.82	0.7904	14.00	31.79	0.7659
	Temporal Diff	35.92	24.93	0.7803	13.62	31.82	0.7618
	SRA	35.05	24.72	0.7914	13.44	31.76	0.7622
	SALD	**31.63**	**25.49**	**0.7591**	**10.85**	**31.84**	**0.7585**

3.1 Results and Discussion

Quantitative Analysis: Tables 1, and 2 report quantitative comparisons across seven benchmarks at 512 and 1024 resolutions. SALD consistently outperforms the competing acceleration and distillation baselines, including SpeeD Wang et al. (2025), Temporal Diff Ying et al. (2026), and SRA Jiang et al. (2025), across both SANA, PixArt and Flux.2 Klein architectures. In particular, SALD achieves the best FID, CLIP, and LPIPS scores across the majority of evaluated settings, demonstrating its ability to simultaneously improve generation fidelity, text-image alignment, and perceptual quality. The improvements are consistent across different model scales and datasets, with particularly substantial FID reductions observed at both 512 and 1024 resolutions. Compared with the existing baselines, these results demonstrate that SALD provides a more effective and consistent strategy for improving diffusion model generation quality while preserving semantic alignment.

Qualitative Analysis: Figures 3 and 5 present qualitative comparisons of SALD across diverse architectures, datasets, and training methods. Figure 3 compares SALD with Base, SpeeD, SRA, and Temporal Diff on SANA 0.6B and PixArt-Σ, where SALD consistently preserves finer details, clearer structures, and prompt-specific semantic attributes, while competing methods often exhibit missing details and visual artifacts. Figure 5 further compares Base and SALD on CUB-200, Oxford-102 Flowers, MAGICK, and DALL·E 3 1M, demonstrating improved structural coherence, fine-grained textures, and semantic fidelity, particularly in challenging high-resolution examples. Overall, these results highlight the consistent visual improvements achieved by SALD across models and datasets.

Convergence Analysis: Figure 4 shows FID convergence across training steps on COCO and Oxford 102 Flowers. Across all settings, SALD converges faster and achieves consistently lower FID than the other baselines, indicating improved optimization and fidelity. Further experiments on this are present in Appendix D.4

3.2 Ablation Study

We conduct an ablation study using SANA 0.6B on COCO 2017 (512×512) and MAGICK (1024×1024), using the uniform flow-matching baseline as the reference across all cases.

Case 1: **Individual Contributions.** AGD contributes most strongly among individual components, as direct difficulty-aware reweighting provides an immediate training signal. SAD offers complementary gains through spectral routing, improving structural fidelity at the cost of modest CLIP degradation. TAM alone yields the weakest improvement, since its EMA-based memory requires a sustained difficulty signal from AGD to stabilise, a notable limitation when used in isolation. Together, all three components are necessary to close the full performance gap.

Table 3: Ablation study of *SALD* on SANA 0.6B across COCO 2017 (512) and MAGICK (1024).

Cases	Setting	COCO (512)			MAGICK (1024)		
		FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓
Case 1	AGD only	30.95	26.30	0.7462	37.75	26.73	0.7354
	SAD only	31.02	26.49	0.7427	37.86	26.70	0.7314
	TAM only	30.98	26.50	0.7474	37.35	26.74	0.7323
Case 2	AGD + TAM	30.78	26.60	0.7529	37.34	26.72	0.7396
	AGD + SAD	30.56	26.66	0.7375	37.78	26.79	0.7400
	TAM + SAD	30.36	26.86	0.7391	37.23	26.78	0.7389
Case 3	$\tau = 0.1$	31.10	26.09	0.7459	38.22	26.66	0.7423
	$\tau = 0.5$ *(ours)*	29.31	27.06	0.7374	36.74	26.85	0.7230
	$\tau = 1.0$	31.22	26.08	0.7455	38.34	26.63	0.7445
	$\tau = 2.0$	31.25	26.13	0.7457	38.42	26.62	0.7431
Case 4	$\alpha = 0.50$	30.89	26.14	0.7454	38.96	26.64	0.7443
	$\alpha = 0.90$ *(ours)*	29.31	27.06	0.7474	36.74	26.85	0.7230
	$\alpha = 0.99$	29.52	26.11	0.7461	38.23	26.64	0.7443
	$\alpha = 0.999$	29.30	26.12	0.7463	38.76	26.68	0.7455
Case 5	$\Delta_{\max} = 100$	30.17	26.75	0.7393	36.78	26.62	0.7467
	$\Delta_{\max} = 250$ *(ours)*	29.31	27.06	0.7374	36.74	26.85	0.7230
	$\Delta_{\max} = 500$	31.96	26.64	0.7398	36.81	26.61	0.7405
Case 6	Abs. loss (no contrast)	29.64	26.98	0.7385	36.95	26.54	0.7305
	Fixed δ (no ramp)	29.79	26.77	0.7383	37.05	26.48	0.7475
	Detach disabled	30.05	26.85	0.7390	37.43	26.67	0.7445
	12 Steps	30.63	26.68	0.7472	37.85	26.62	0.7476
	Cross-Batch Advantage	30.18	26.92	0.7386	37.57	26.75	0.7434
Base	Uniform	31.59	26.80	0.7408	38.59	26.68	0.7470
SALD	**AGD-TAM-SAD**	29.31	27.06	0.7374	36.74	26.85	0.7230

Case 2: **Pairwise Combinations.** AGD+SAD achieves the best pairwise result, as the two components operate on complementary axes, sample difficulty, and spectral structure, without cross-dependency. TAM+SAD performs weakest among pairs, confirming TAM's dependence on AGD's advantage signal for effective curriculum updates. The SALD surpasses all pairs, demonstrating non-redundant synergy across components.

Case 3: **AGD Temperature τ.** Moderate sharpness ($\tau = 0.5$) best balances gradient focus and perceptual stability. Overly low τ over-concentrates on extreme samples, slightly harming LPIPS, while high τ degrades toward a near-uniform loss, weakening the difficulty signal. SALD exhibits a clear U-shaped sensitivity, with gains degrading at both extremes.

Case 4: **TAM EMA Decay α.** The default $\alpha = 0.90$ strikes the best balance between curriculum responsiveness and stability. Aggressive decay ($\alpha = 0.50$) introduces noisy difficulty estimates, while near-frozen memory ($\alpha = 0.999$) fails to track the evolving training landscape. The wider performance spread on MAGICK confirms that longitudinal difficulty tracking is more critical at higher resolution, where sample complexity varies more sharply.

Case 5: **Maximum Noise Gap $\Delta_{\max}$.** Both extremes degrade performance: a narrow gap ($\Delta_{\max} = 100$) weakens contrast between hard and easy paths, while an overly wide gap ($\Delta_{\max} = 500$) pushes the easy path into a near-noiseless regime, overestimating difficulty and destabilising training. The

default $\Delta_{\max} = 250$ yields the best results across both benchmarks, with a consistent U-shaped FID response confirming the sensitivity of the advantage signal to gap magnitude.

Case 6: **Core Design Decisions.** All five choices prove critical. *Absolute loss (no contrast):* replacing $A^{(i)} = \ell_b^{(i)} - \mathrm{sg}(\ell_a^{(i)})$ with $|\ell_a^{(i)}|$ collapses gains toward the baseline, most severely at 1024^2, confirming relative difficulty, not raw loss magnitude, as the effective signal. *Fixed δ (no ramp):* removing the warm-up destabilises the advantage signal during early training, mildly degrading perceptual quality. *Detach disabled:* allowing $w^{(i)}$ to backpropagate causes the model to exploit the weighting mechanism rather than improve predictions, validating $\mathrm{sg}(\cdot)$ as non-negotiable. *Reduced steps (*12 *vs.* 20*):* SALD at 12 steps still outperforms the 20-step uniform baseline, indicating that training against the hard-to-easy trajectory yields a smoother learned flow with fewer steps required at inference. *Cross-batch advantage:* computing advantage relative to a second independently drawn batch rather than the same sample's easier noise path yields partial gains but falls short of SALD, confirming that intra-sample noise contrast is a strictly stronger signal than inter-sample difficulty, which conflates sample hardness with distributional variation.

4 Related Work

Denoising diffusion models Ho et al. (2020) and score-based generative models Song et al. (2020) established iterative denoising as a powerful paradigm for image synthesis. Flow matching Lipman et al. (2022); Albergo et al. (2025) provides a continuous-time alternative by learning velocity fields along interpolation paths. Recent transformer-based generators, including DiT Peebles & Xie (2023), rectified-flow transformers Esser et al. (2024), SANA Xie et al. (2024), PixArt-α Chen et al. (2023), and PixArt-Σ Chen et al. (2024), further improve scalability and high-resolution generation. However, most objectives still treat samples, timesteps, and spatial locations uniformly. Several works improve training by reallocating the gradient budget across noise levels. EDM Karras et al. (2022) studies principled noise schedules and σ-weighting, while P2 Choi et al. (2022) emphasizes perceptually important SNR regions. Improved noise schedules Hang et al. (2025) and noise-instance analysis Qi et al. (2024a) further show that different noise levels and noise samples provide unequal training signals. These approaches define global timestep-dependent weights, whereas AGD computes per-sample relative difficulty from the loss gap between low- and high-noise paths.

Our work is also related to curriculum learning Bengio et al. (2009), self-paced learning Kumar et al. (2010), dynamic instance hardness Chang et al. (2020), adaptive curricula Kong et al. (2021), diffusion curricula Liang et al. (2025), and adaptive point-weighting Li et al. (2025). Hard-example mining Felzenszwalb et al. (2010); Shrivastava et al. (2016) and DRO Sagawa et al. (2020); Feoktistov et al. (2025); Ma et al. (2024) similarly emphasize difficult examples, while meta-hard mining Mu et al. (2025) and coreset selection Killamsetty et al. (2021) select informative samples. Unlike these methods, TAM stores an EMA of relative advantage and uses it to adapt the timestep gap for each sample. Finally, SAD connects to spectral bias studies Rahaman et al. (2019); Xu et al. (2020); Wang & Pehlevan (2025b) and frequency-aware diffusion methods Benita et al. (2025); Si et al. (2024); Jiralerspong et al. (2025); Esteves & Makadia (2026); Liu et al. (2025). Instead of modifying the architecture, sampler, or noising process, SAD uses contrastive spectral residuals to construct spatial importance weights for loss reweighting. Recent work uses feedback-conditioned output distribution to distill Shenfeld et al. (2026); Hübotter et al. (2026) from teacher to student in the language domain but requires demonstration example and EMA based teacher. The proposed model can be seen as distillation from easy to hard sample, however it does not requires extra copy of the model parameters.

5 Conclusion

We presented SALD, a unified training framework for flow-based diffusion models that reallocates gradient budget across samples, noise levels, and frequency components. AGD estimates relative sample difficulty by contrasting losses between low and high-noise states, producing normalized within-batch importance weights. TAM extends this signal across the training set through persistent per-sample memory, enabling an adaptive curriculum over timestep gaps. SAD further refines optimization by converting contrastive spectral residuals into frequency-derived latent weights, emphasizing frequency patterns that remain under-learned. Since only the easy-path reference is detached, SALD preserves the original prediction target while improving where and how training effort is applied. In general, SALD offers a simple, architecture-agnostic approach for improving diffusion training without modifying the model, sampler, or inference procedure. Extensive experiments across diverse datasets and architectures demonstrate the effectiveness of the proposed framework, while theoretical analysis supports the empirical findings.

AI USE STATEMENT

We used generative AI tools solely for language refinement and grammar correction. All other aspects of this work, including the methodology, implementation, experiments, analysis, results, and manuscript content, were completed by the authors. The authors take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal information, or private or proprietary data. All experiments are conducted using publicly available image-text datasets, including MultiGen-20M, COCO, Flickr8k, CUB-200-2011, Oxford 102 Flowers, MAGICK, and the DALL·E 3 1M dataset, which are used solely for training and evaluating SALD and the corresponding baseline methods. Our approach does not require human annotations, human preference data, external teacher models, or additional human feedback during training. Instead, SALD constructs an internal reference from the same diffusion model using an easier noise level, thereby avoiding reliance on external supervision or preference-based feedback. We acknowledge that text-to-image generation models may be misused to produce misleading, inappropriate, or harmful visual content. We therefore encourage the responsible use of the proposed method and compliance with the licenses and usage policies of the underlying datasets and pretrained models, as well as adherence to the ICLR Code of Ethics and established responsible research practices.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the SALD framework, training objective, and optimization procedure in Sections 2 and 3, with additional implementation details in Appendix D. The appendix includes the backbone configurations, training and evaluation settings, wall-clock comparisons, inference-step analysis, extended training experiments, and qualitative results. We evaluate SALD across four backbone configurations, including SANA 0.6B, SANA 1.6B, PixArt-Σ, and FLUX.2-Klein base, using seven datasets at 512×512 and 1024×1024 resolutions. The training configuration specifies the optimizer, learning-rate schedule, mixed-precision setting, gradient clipping, number of training steps, and inference settings. We additionally report extended training results up to 100k steps to examine convergence beyond the main training budget. These details are provided to facilitate the reproduction of our reported experiments and comparisons.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

Roi Benita, Michael Elad, and Joseph Keshet. Spectral analysis of diffusion models with application to schedule design. In *Advances in Neural Information Processing Systems*, 2025.

Ryan D Burgert, Brian L Price, Jason Kuen, Yijun Li, and Michael S Ryoo. Magick: A large-scale captioned dataset from matting generated images using chroma keying. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22595–22604, 2024.

H. Chang, C. You, T. Wang, Z. Chen, Y. Wang, and Z. Wang. Curriculum learning by dynamic instance hardness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. URL https://arxiv.org/abs/2310.00426.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024. URL https://arxiv.org/abs/2403.04692.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11472–11481, 2022.

Ben Egan, Alex Redden, XWAVE, and SilentAntagonist. Dalle3 1 Million+ High Quality Captions, May 2024. URL https://huggingface.co/datasets/ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Carlos Esteves and Ameesh Makadia. Spectrally-guided diffusion noise schedules. *arXiv preprint arXiv:2603.19222*, 2026. doi: 10.48550/arXiv.2603.19222. URL https://arxiv.org/abs/2603.19222.

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

Dmitrii Feoktistov, Igor Ignashin, Andrey Veprikov, Nikita Borovko, Alexander Bogdanov, Savelii Chezhegov, and Aleksandr Beznosikov. Aligning distributionally robust optimization with practical deep learning needs. *arXiv preprint arXiv:2508.16734*, 2025. doi: 10.48550/arXiv.2508.16734. URL https://arxiv.org/abs/2508.16734.

Tiankai Hang, Shuyang Gu, Jianmin Bao, Fangyun Wei, Dong Chen, Xin Geng, and Baining Guo. Improved noise schedule for diffusion training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4796–4806, 2025.

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. *Advances in neural information processing systems*, 30, 2017.

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, 2018. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. doi: 10.48550/arXiv.2601.20802. URL https://arxiv.org/abs/2601.20802.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

Thomas Jiralerspong, Berton Earnshaw, Jason Hartford, Yoshua Bengio, and Luca Scimeca. Shaping inductive bias in diffusion models through frequency-based noise control. *arXiv preprint arXiv:2502.10236*, 2025. doi: 10.48550/arXiv.2502.10236. URL https://arxiv.org/abs/2502.10236.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 2021. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

T. Kong et al. Adaptive curriculum learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

M. P. Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Black Forest Labs. Flux.2 [klein] base 4b. https://huggingface.co/black-forest-labs/FLUX.2-klein-base-4B, 2025.

Wensheng Li, Hao Wang, Ruifeng Zhou, Hanting Guan, Chao Zhang, and Dacheng Tao. Adaptively point-weighting curriculum learning. *arXiv preprint arXiv:2505.01665*, 2025. doi: 10.48550/arXiv.2505.01665. URL https://arxiv.org/abs/2505.01665.

Yijun Liang, Shweta Bhardwaj, and Tianyi Zhou. Diffusion curriculum: Synthetic-to-real data curriculum via image-guided diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Chengxu Liu, Lu Qi, Jinshan Pan, Xueming Qian, and Ming-Hsuan Yang. Frequency domain-based diffusion model for unpaired image dehazing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Xutao Ma, Chao Ning, and Wenli Du. Differentiable distributionally robust optimization layers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33880–33901. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ma24j.html.

Y. Mu et al. Meta-learning dynamic center distance: Hard sample mining for learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Zipeng Qi, Lichen Bai, Haoyi Xiong, and Zeke Xie. Not all noises are created equally: Diffusion noise selection and optimization. *arXiv preprint arXiv:2407.14041*, 2024a.

Zipeng Qi, Lichen Bai, Haoyi Xiong, and Zeke Xie. Not all noises are created equally: Diffusion noise selection and optimization. *arXiv preprint arXiv:2407.14041*, 2024b.

Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. Learning deep representations of fine-grained visual descriptions, 2016. URL https://arxiv.org/abs/1605.05395.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. doi: 10.48550/arXiv.2601.19897. URL https://arxiv.org/abs/2601.19897.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4733–4743, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Binxu Wang and Cengiz Pehlevan. An analytical theory of spectral bias in the learning dynamics of diffusion models. *arXiv preprint arXiv:2503.03206*, 2025a.

Binxu Wang and Cengiz Pehlevan. An analytical theory of spectral bias in the learning dynamics of diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12934–12944. IEEE, 2025.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformers, 2024. URL https://arxiv.org/abs/2410.10629.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. 2018.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020. doi: 10.4208/cicp.OA-2020-0085.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

Qizhen Ying, Yangchen Pan, Victor Adrian Prisacariu, and Junfeng Wen. Temporal difference learning for diffusion models, 2026. URL https://arxiv.org/abs/2606.15048.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

APPENDIX

A WHY IS UNIFORM TRAINING SUBOPTIMAL?

The standard flow-matching loss used in Section 2.1 averages the losses of all training samples and all elements of their latent residuals. Apart from the time-dependent factor $w(t)$, it assigns the same explicit coefficient to every sample and latent element. This does not mean that all samples produce equal gradients; samples with larger residuals can naturally produce larger gradient magnitudes. Rather, the objective has no explicit mechanism for adapting a sample's importance according to its current difficulty or for distinguishing the frequency content of its remaining residual. This uniform treatment leads to two limitations that can become more important as training progresses, especially during domain-specific fine-tuning.

The first limitation is at the sample level. Not all images are equally difficult for the current model. A clean portrait against a plain background may be learned early, after which its loss becomes small and further updates provide limited new information. Meanwhile, a cluttered street scene with reflections, thin text, and overlapping objects may remain difficult for much longer. Under uniform weighting, both samples retain the same explicit coefficient in the mini-batch objective, even though reallocating more emphasis to the difficult sample may be more useful. Prior work on hard-example mining and diffusion-noise selection similarly shows that examples and noise realizations can provide unequal training value Shrivastava et al. (2016); Qi et al. (2024b). Consequently, uniform weighting can spend part of the finite optimization budget on already well-learned samples while under-emphasizing examples that remain difficult.

The second limitation concerns the frequency content of the residual within each image. Deep networks are known to learn low-frequency content faster than high-frequency content, a behavior referred to as spectral bias or the frequency principle Rahaman et al. (2019); Xu et al. (2020). Related frequency-dependent behavior has also been studied in diffusion models Yang et al. (2023); Si et al. (2024); Wang & Pehlevan (2025a). Smooth structures such as skies and walls may be reconstructed earlier, whereas fine details such as hair strands, feather textures, specular highlights, and small text can lag behind. However, the standard mean-squared error collapses the full latent residual into a single scalar and does not explicitly indicate which frequency bands retain comparatively large error. Importantly, frequency magnitude does not identify the spatial locations of those errors because the power spectrum discards Fourier phase. The relevant goal is therefore to identify difficult frequency components and use them to construct a frequency-derived latent weight, rather than to claim a localized pixel-level error map.

One natural idea is curriculum learning Bengio et al. (2009); Kumar et al. (2010), in which easy samples are shown first and harder samples are introduced later. Classical curricula, however, often rely on a fixed notion of difficulty. In diffusion training, a sample's difficulty changes as the model improves, so a static ordering can quickly become stale. What is needed instead is a strategy that (i) measures which samples become difficult relative to the current model when noise is increased, (ii) remembers this relative difficulty across training steps, and (iii) identifies which residual frequencies remain difficult. Unlike demonstration- or feedback-conditioned language-model self-distillation Shenfeld et al. (2026); Hübotter et al. (2026), SALD constructs its reference from two noise states of the same image–caption pair: a lower-noise easy path evaluated without gradient tracking and a higher-noise hard path that supplies the training gradient. Advantage-Guided Diffusion (AGD) performs differentiable sample-level reweighting, Temporal Advantage Memory (TAM) maintains persistent relative-difficulty estimates and adapts future noise gaps, and Spectral Advantage Decomposition (SAD) constructs a frequency-derived latent weight. These components share one set of model parameters and require no external teacher or additional model copy during training or inference.

B EXTENSION TO DDPM

Our framework can be extended to DDPM-style noise-prediction models, although we do not validate this extension experimentally (Appendix E). The core difference between flow matching and DDPM lies in the prediction target Lipman et al. (2022); Ho et al. (2020). In the flow-matching formulation used in the main paper, the network predicts the velocity $u = \epsilon - x_0$ along the linear interpolation path. In an ϵ-prediction DDPM, the network predicts the Gaussian noise added to the clean representation.

The forward process and residual are

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad r(t,\epsilon) = \epsilon_\theta(x_t, t, c) - \epsilon, \tag{21}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_t$ is the cumulative noise schedule. The corresponding scalar loss is

$$\ell(t,\epsilon) = \frac{w(t)}{D} \left\| r(t,\epsilon) \right\|_2^2, \qquad D = CHW. \tag{22}$$

Thus, extending SALD requires replacing the velocity residual in the main formulation with the DDPM noise-prediction residual in Eq. equation 21.

The two paths keep the same roles as in the main paper. Path A is the easy, lower-noise path and is evaluated without gradient tracking; Path B is the hard, higher-noise path and supplies the training gradient. Under the standard DDPM ordering, $t_\mathrm{e} < t_\mathrm{h}$ and $\bar{\alpha}_{t_\mathrm{e}} > \bar{\alpha}_{t_\mathrm{h}}$. Using independent noises for the two paths, we compute

$$\ell_\mathrm{e} = \ell(t_\mathrm{e}, \epsilon_\mathrm{e}), \qquad \ell_\mathrm{h} = \ell(t_\mathrm{h}, \epsilon_\mathrm{h}), \qquad A_i = \ell_{\mathrm{h},i} - \mathrm{sg}(\ell_{\mathrm{e},i})\,. \tag{23}$$

Nothing about AGD changes except the predicted quantity. Importantly, A_i is not fully detached: only the easy reference is stopped, so $\nabla_\theta A_i = \nabla_\theta \ell_{\mathrm{h},i}$.

TAM also keeps the same update rule. It stores a scalar exponential-moving average for each training pair and updates it using the numerical value of the relative advantage. The memory vector m, decay α, and modulation by $\tanh(m_i)$ remain unchanged. The stored memory is used to select the current easy-to-hard gap before the two paths are evaluated; the current advantage then updates the memory for the next visit. The memory is a non-trainable state buffer and is not backpropagated through time.

SAD also carries over by operating on the noise-prediction residuals $r_\mathrm{e}^{(i)}$ and $r_\mathrm{h}^{(i)}$ instead of velocity residuals. These tensors remain in $\mathbb{R}^{C\times H\times W}$, so the real FFT, power-spectrum comparison, spectral advantage, sigmoid gate, and inverse-FFT construction of M_i are unchanged. As in the flow-matching formulation, only the easy residual spectrum is stopped; M_i remains differentiable through the hard residual. Frequency-dependent learning behavior motivates this comparison Rahaman et al. (2019); Wang & Pehlevan (2025a), but its interpretation remains limited: because the power spectrum discards phase, M_i is a frequency-derived latent weight rather than a localized error map. Moreover, applying M_i in the latent domain does not independently rescale each Fourier coefficient.

The remaining practical consideration is loss weighting. The distribution used to sample t and the factor $w(t)$ used to weight the loss are distinct choices. The original DDPM simple objective uses uniform loss weighting, whereas perceptual or signal-to-noise-ratio-derived alternatives can also be used Ho et al. (2020); Choi et al. (2022). SALD is compatible with any nonnegative rule $w(t)$, provided that the same rule is evaluated at the corresponding easy and hard levels when computing Eq. equation 23. The final SAD-weighted loss uses $w(t_\mathrm{h})$ because gradients are taken only through the hard path.

C Proof and Scope of Theorem 1

For a fixed TAM memory state m, let

$$F_m(\theta) := \mathbb{E}_\xi \left[\widehat{\mathcal{L}}_{\mathrm{SALD}}(\theta; \xi, m) \right], \tag{24}$$

where ξ denotes all randomness used to construct a mini-batch, including the sampled training examples, noise levels, and Gaussian noises. Because the normalized AGD weights couple the examples through their batch denominator, F_m is the population objective induced by an entire random mini-batch; it is not, in general, an expectation of independent single-example objectives. The unbiasedness assumption below is satisfied by Eq. equation 24 whenever differentiation and expectation can be interchanged.

Theorem 2 (Conditional stationarity bound for the differentiable SALD objective)**.** *Fix m. Assume that F_m is lower bounded by $F_m^\star$ and has an L-Lipschitz gradient. At iteration s, let*

$$g_s = \nabla_\theta \widehat{\mathcal{L}}_{\mathrm{SALD}}(\theta_s; \xi_s, m) \tag{25}$$

be the stochastic gradient containing all gradient paths through the AGD weights ω_i and the SAD kernels M_i. Suppose

$$\mathbb{E}[g_s \mid \theta_s] = \nabla F_m(\theta_s), \tag{26}$$

$$\mathbb{E}\Big[\|g_s - \nabla F_m(\theta_s)\|_2^2\Big] \le \nu^2. \tag{27}$$

For the update

$$\theta_{s+1} = \theta_s - \eta g_s, \qquad 0 < \eta \le \frac{1}{L}, \tag{28}$$

and a deterministic initial point θ_0,

$$\frac{1}{J}\sum_{s=0}^{J-1} \mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 \le \frac{2\big(F_m(\theta_0) - F_m^\star\big)}{\eta J} + L\eta\nu^2. \tag{29}$$

Proof. Since F_m is L-smooth, the descent lemma gives

$$F_m(\theta_{s+1}) \le F_m(\theta_s) + \langle \nabla F_m(\theta_s), \theta_{s+1} - \theta_s\rangle + \frac{L}{2}\|\theta_{s+1} - \theta_s\|_2^2 \tag{30}$$

$$= F_m(\theta_s) - \eta\,\langle \nabla F_m(\theta_s), g_s\rangle + \frac{L\eta^2}{2}\|g_s\|_2^2. \tag{31}$$

By conditional unbiasedness,

$$\mathbb{E}[\langle \nabla F_m(\theta_s), g_s\rangle] = \mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2. \tag{32}$$

To control the second moment, define

$$\zeta_s := g_s - \nabla F_m(\theta_s). \tag{33}$$

Equation equation 26 implies $\mathbb{E}[\zeta_s \mid \theta_s] = 0$. Therefore, the cross term vanishes and Eq. equation 27 yields

$$\mathbb{E}\,\|g_s\|_2^2 = \mathbb{E}\,\|\nabla F_m(\theta_s) + \zeta_s\|_2^2 \tag{34}$$

$$= \mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 + \mathbb{E}\,\|\zeta_s\|_2^2 \tag{35}$$

$$\le \mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 + \nu^2. \tag{36}$$

Taking expectations in Eq. equation 31 and applying Eqs. equation 32–equation 36 gives

$$\mathbb{E}F_m(\theta_{s+1}) \le \mathbb{E}F_m(\theta_s) - \eta\left(1 - \frac{L\eta}{2}\right)\mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 + \frac{L\eta^2}{2}\nu^2. \tag{37}$$

Because $\eta \le 1/L$, $1 - L\eta/2 \ge 1/2$. Hence,

$$\frac{\eta}{2}\mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 \le \mathbb{E}F_m(\theta_s) - \mathbb{E}F_m(\theta_{s+1}) + \frac{L\eta^2}{2}\nu^2. \tag{38}$$

Summing Eq. equation 38 from $s = 0$ to $J - 1$ telescopes the objective values:

$$\frac{\eta}{2}\sum_{s=0}^{J-1}\mathbb{E}\,\|\nabla F_m(\theta_s)\|_2^2 \le F_m(\theta_0) - \mathbb{E}F_m(\theta_J) + \frac{JL\eta^2}{2}\nu^2. \tag{39}$$

Using $F_m(\theta_J) \ge F_m^\star$, dividing by $\eta J/2$, and rearranging gives Eq. equation 29. □

Remark 3 (Meaning of the bound)**.** *For a constant stepsize and nonzero stochastic variance, the right-hand side of Eq. equation 29 approaches the nonzero error floor $L\eta\nu^2$. Thus, the displayed inequality guarantees convergence to a stationarity neighborhood, not necessarily to an exact stationary point. A vanishing average gradient norm follows when $\nu = 0$, or when the stepsize is decreased appropriately. For example, choosing $\eta = \Theta(J^{-1/2})$, while respecting $\eta \le 1/L$, gives the standard $\mathcal{O}(J^{-1/2})$ stationarity rate.*

Remark 4 (Fixed-memory scope)**.** *The telescoping argument uses one fixed objective F_m. During the full SALD procedure, TAM changes the memory from m_s to m_{s+1}, so the corresponding objective may also change from F_{m_s} to $F_{m_{s+1}}$. Theorem 1 therefore provides a convergence result conditional on a fixed TAM state. Extending it to the complete online-memory dynamics would require an additional assumption controlling the objective drift caused by each memory update, or a separate analysis of the joint state (θ_s, m_s).*

Remark 5 (What is not implied)**.** *The theorem assumes smoothness, unbiasedness, and bounded variance for the actual differentiable SALD gradient, including the paths through ω_i and M_i. It neither proves these assumptions for a particular neural network nor shows that SALD converges faster than uniform flow matching. It also does not imply that the SALD and baseline objectives have the same stationary points.*

D Additional Results

In this section, we present additional qualitative and quantitative results to further analyze SALD's behavior across different datasets, architectures, and training settings. We include extended visual comparisons and supplementary experiments that highlight the contributions of AGD, TAM, and SAD to generation quality, optimization stability, and fine-grained detail preservation.

Algorithm 1 summarizes a single SALD training step. Given a mini-batch of image–caption pairs, we first determine a base noise-level gap using a simple curriculum: during warm-up, the gap increases from 1 to Δ, after which it is sampled uniformly from $\{1, \ldots, \Delta\}$. This base gap is then modulated for each sample using Temporal Advantage Memory (TAM), which scales the gap according to the sample's stored difficulty score and clips it to the valid range. For each example, we sample a lower-noise easy level and obtain the corresponding higher-noise hard level by adding the TAM-adjusted gap. Two noisy views of the same latent are then constructed using independent Gaussian perturbations. The same model performs two forward passes: the easy path is evaluated under `no_grad` and serves as an internal reference, whereas the hard path remains differentiable and supplies the training gradient. From the residuals of these two paths, Spectral Advantage Decomposition (SAD) compares their frequency-domain error magnitudes and maps a zero-phase spectral gate back to the latent coordinates to obtain a frequency-derived weighting kernel. Since this construction does not retain the residual phase or recover the locations of the original errors, the kernel should not be interpreted as a localized pixel-level advantage map. Instead, it reweights the hard-path residual loss using the relative spectral errors of the two paths. In parallel, Advantage-Guided Diffusion (AGD) defines the sample advantage as the difference between the differentiable hard-path loss and the stop-gradient easy-path reference loss. It converts this advantage into a normalized sample weight that emphasizes harder examples within the batch. Finally, TAM stores the numerical advantage for future visits to the sample; the memory is non-trainable, and no gradient is propagated through this state update. The SALD objective is then formed by averaging the sample-weighted SAD losses of the hard paths before backpropagation. In this way, the algorithm adapts the training signal across samples, noise levels, and spectral components through a dynamic, self-referenced curriculum.

D.1 Training and Evaluation Configuration

Table 4 summarizes the implementation details of all backbone architectures used in our experiments. We evaluate SALD on SANA, PixArt-Σ, and FLUX.2 variants across multiple parameter scales and image resolutions to demonstrate the generality of the proposed framework. The table reports the number of trainable parameters, computational complexity measured in FLOPs per forward pass, training resolution, the learning rate used for finetuning, as well as the inference-time guidance scale and number of sampling steps. Experiments are conducted at both 512×512 and 1024×1024 resolutions, where higher resolutions naturally require substantially higher computational cost due to increased latent spatial dimensions. All models are trained using bfloat16 (bf16) mixed-precision training with gradient clipping set to a maximum norm of 1.0. Following the original training configurations of each architecture, we employ a constant with warmup learning rate scheduler with 500 warmup steps, after which the learning rate is kept constant. We use relatively larger learning rates for SANA models and more conservative rates for PixArt variants to ensure stable optimization. These settings are kept consistent across baseline and SALD experiments for fair comparison. For

Algorithm 1 SALD Training Step

Require: Mini-batch $\{(x_0^{(i)}, c^{(i)}, \mathrm{idx}^{(i)})\}_{i=1}^{B}$
Require: Noise levels $0 \le \sigma_0 < \cdots < \sigma_{T-1} \le 1$; current step s; $\tau, \tau_s, \alpha, \lambda$; $K > 1$; $1 \le \Delta_{\max} < T$
Global: non-trainable TAM memory $m \in \mathbb{R}^N$ (initialized to zero); $D = CHW$
Notation: subscript a denotes the easy path ($t_{\mathrm{e}} = \sigma_{k_a}$) and b the hard path ($t_{\mathrm{h}} = \sigma_{k_b}$)

if $s < K$ **then**
  $\delta_{\mathrm{base}} \leftarrow 1 + \left\lfloor \dfrac{(\Delta_{\max} - 1)s}{K - 1} \right\rfloor$
else
  $\delta_{\mathrm{base}} \sim \mathrm{Uniform}\{1, \ldots, \Delta_{\max}\}$
end if
for $i = 1, \ldots, B$ **do**
  $\delta^{(i)} \leftarrow \mathrm{clip}\big(\mathrm{round}\big[\delta_{\mathrm{base}}\big(1 + \lambda \tanh(m_{\mathrm{idx}^{(i)}})\big)\big], 1, \Delta_{\max}\big)$ ▷ TAM gap
  $u^{(i)} \sim \mathrm{LogitNormal}, \quad k_a^{(i)} \leftarrow \left\lfloor u^{(i)}(T - \delta^{(i)}) \right\rfloor, \quad k_b^{(i)} \leftarrow k_a^{(i)} + \delta^{(i)}$
  $\epsilon_a^{(i)}, \epsilon_b^{(i)} \sim \mathcal{N}(0, I)$
  $z_{k_a}^{(i)} \leftarrow (1 - \sigma_{k_a^{(i)}}) x_0^{(i)} + \sigma_{k_a^{(i)}} \epsilon_a^{(i)}$
  $z_{k_b}^{(i)} \leftarrow (1 - \sigma_{k_b^{(i)}}) x_0^{(i)} + \sigma_{k_b^{(i)}} \epsilon_b^{(i)}$
  $\widehat{v}_a^{(i)} \leftarrow \mathrm{no_grad}\left[v_\theta(z_{k_a}^{(i)}, k_a^{(i)}, c^{(i)})\right]$ ▷ easy path; no gradient
  $\widehat{v}_b^{(i)} \leftarrow v_\theta(z_{k_b}^{(i)}, k_b^{(i)}, c^{(i)})$ ▷ hard path; gradient on
  $r_a^{(i)} \leftarrow \widehat{v}_a^{(i)} - (\epsilon_a^{(i)} - x_0^{(i)})$
  $r_b^{(i)} \leftarrow \widehat{v}_b^{(i)} - (\epsilon_b^{(i)} - x_0^{(i)})$
  $M^{(i)} \leftarrow \mathrm{SAD}(r_a^{(i)}, r_b^{(i)}, \tau_s)$ ▷ not detached; mean approximately one
  $\ell_{\mathrm{sad}}^{(i)} \leftarrow \dfrac{w(\sigma_{k_b^{(i)}})}{D} \sum_{p=1}^{D} M_p^{(i)} \big(r_{b,p}^{(i)}\big)^2$
  $\ell_a^{(i)} \leftarrow \dfrac{w(\sigma_{k_a^{(i)}})}{D} \|r_a^{(i)}\|_2^2$ ▷ easy loss; no gradient
  $\ell_b^{(i)} \leftarrow \dfrac{w(\sigma_{k_b^{(i)}})}{D} \|r_b^{(i)}\|_2^2$ ▷ hard loss; gradient on
  $A^{(i)} \leftarrow \ell_b^{(i)} - \mathrm{sg}\left(\ell_a^{(i)}\right)$ ▷ only the easy loss is detached
end for
for $i = 1, \ldots, B$ **do**
  $\omega^{(i)} \leftarrow \dfrac{B \ \mathrm{sigmoid}(A^{(i)}/\tau)}{\sum_{j=1}^{B} \mathrm{sigmoid}(A^{(j)}/\tau)}$ ▷ not detached
end for
for $i = 1, \ldots, B$ **do**
  $m_{\mathrm{idx}^{(i)}} \leftarrow \alpha m_{\mathrm{idx}^{(i)}} + (1 - \alpha)\ \mathrm{sg}\left(A^{(i)}\right)$ ▷ numerical value only; no graph
end for
$\mathcal{L}_{\mathrm{SALD}} \leftarrow \dfrac{1}{B} \sum_{i=1}^{B} \omega^{(i)} \ell_{\mathrm{sad}}^{(i)}$
Backpropagate $\nabla_\theta \mathcal{L}_{\mathrm{SALD}}$ and update θ

evaluation, we use the following implementations: FID is computed using TorchMetrics, CLIP-based similarity is evaluated using CLIP ViT-L/14, and LPIPS is computed using the Perceptual Similarity (LPIPS) implementation.

D.2 Wall-Clock Evaluation

We further evaluate the training efficiency of SALD under a matched wall-clock budget. As shown in Table 5, SALD incurs approximately 25% additional wall-clock time per training step compared with the corresponding baseline, measured on the same NVIDIA RTX A6000 setup. This overhead comes from the additional easy noise reference path: SALD performs gradient computation and backpropagation only for the hard noise path, while the easy noise path is evaluated without gradient computation, requiring only an additional forward pass. To account for this overhead, SALD is trained for 10k steps, while the baselines are trained for 12.5k steps. Thus, the total training budgets

Table 4: Detailed model configurations used in our experiments, including parameter count, per-forward-pass FLOPs, training resolution, learning rate, guidance scale, and inference steps. † denotes models additionally evaluated at 1024×1024 resolution.

Model	Params (B)	FLOPs (G)	Resolution	LR	Guidance	Infer. Steps
SANA (0.6B)	0.6	548	512×512	1e-4	5	20
SANA (1.6B)	1.6	567	512×512	1e-4	5	20
PixArt-Σ	0.6	567	512×512	2e-5	5	20
SANA (0.6B)†	0.6	1397	1024×1024	1e-4	5	20
SANA (1.6B)†	1.6	2240	1024×1024	1e-4	5	20
PixArt-Σ†	0.6	1399	1024×1024	2e-5	5	20
FLUX.2 klein base	4.0	6953	1024×1024	1e-5	4	50

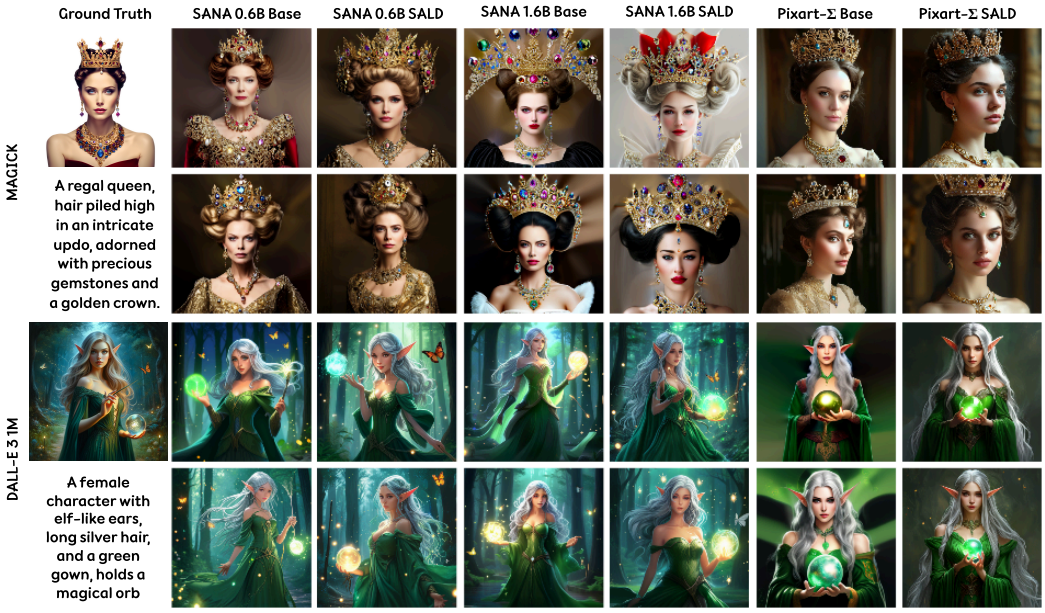


Figure 6: Qualitative comparison across baseline models and SALD on 1024 × 1024 datasets.

are approximately matched, since $10\text{k} \times 1.25 \approx 12.5\text{k}$. As reported in Table 5, under this comparable wall-clock budget, SALD consistently achieves better generation quality across both CUB-200 and MultiGen datasets. In particular, SALD improves FID, CLIP, and LPIPS over the corresponding baseline models across all evaluated backbone configurations. These results show that the limited additional cost introduced by the no-gradient reference path is compensated by SALD's faster convergence, allowing it to achieve superior generation quality within approximately the same overall training time as the baselines.

D.3 Impact of Inference Steps and Model Scalability

Table 6 reveals a consistent sampling-efficiency advantage of SALD across model scales, from SANA 0.6B/1.6B to PixArt-Σ 0.6B, on Oxford Flowers. As the number of inference steps is reduced, SALD preserves generation quality more effectively than the corresponding Base models, yielding lower FID and LPIPS while retaining comparable or higher CLIP scores. In particular, for both SANA models, SALD at 15 steps attains lower FID than the Base model at 20 steps (30.60 vs. 33.11 for SANA 1.6B and 25.54 vs. 30.64 for SANA 0.6B), using 25% fewer sampling steps, although LPIPS and CLIP at 15 steps remain slightly below the 20-step Base values.

The behavior is consistent with the relative error signal introduced in Eq. 6. Instead of treating prediction errors at different noise states independently, SALD contrasts the hard and easy states of the same sample and assigns greater importance when the harder state exhibits a larger error.

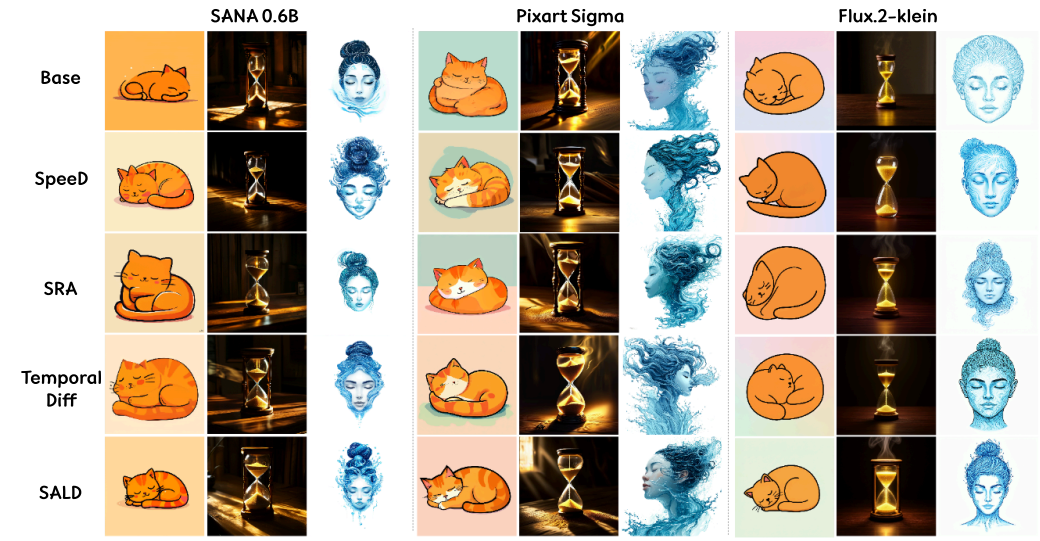


Figure 7: Qualitative comparisons: samples generated by SALD and other different Baselines on 1024 × 1024 Datasets. Captions are, Column-1: A cute orange cat sleeping curled up, flat vector illustration style. Column-2: An hourglass with golden sand, dramatic cinematic lighting on a wooden table. Column-3: A woman's face in profile with flowing blue hair made of water, digital art.



Figure 8: Qualitative comparison of inference-time generation across different sampling steps for SALD and the baseline using SANA 1.6B trained on DALL·E 3 1M.

Through the AGD weighting in Eq. 7, optimization therefore concentrates on trajectory regions where prediction remains comparatively difficult. This encourages more accurate learning of the velocity field along challenging portions of the noise trajectory, which in turn reduces the dependence on densely spaced inference updates. The cross-scale results in Table 6 suggest that this effect is not specific to a particular architecture size, but translates into consistently stronger quality–efficiency trade-offs across the evaluated models.

Furthermore, Figure 8 provides a qualitative view of this faster inference-time convergence using SANA 1.6B trained on DALL·E 3 1M. The distinction is most pronounced at low sampling budgets. SALD establishes the correct facial structure and overall appearance within the first few steps, with

Table 5: Wall-clock time and quantitative comparison on CUB-200 and MultiGen datasets. Best values in Blue

Dataset	Model	Variant	Params (B)	Steps	FID↓	CLIP↑	LPIPS↓
CUB-200	SANA	Base	1.6	12.5k	21.47	26.01	0.7709
		SpeeD	1.6	12.5k	20.83	26.08	0.7670
		SRA	1.6	12.5k	20.72	26.15	0.7632
		Temporal Diff	1.6	12.5k	20.61	26.22	0.7593
		SALD	1.6	**10k**	**18.16**	**26.29**	**0.7537**
	SANA	Base	0.6	12.5k	25.54	26.19	0.7802
		SpeeD	0.6	12.5k	24.77	26.26	0.7763
		SRA	0.6	12.5k	24.65	26.33	0.7724
		Temporal Diff	0.6	12.5k	24.52	26.40	0.7705
		SALD	0.6	**10k**	**23.70**	**26.61**	**0.7686**
	PixArt-Σ	Base	0.6	12.5k	21.57	29.27	0.7886
		SpeeD	0.6	12.5k	20.92	29.35	0.7847
		SRA	0.6	12.5k	20.81	29.42	0.7808
		Temporal Diff	0.6	12.5k	20.70	29.50	0.7768
		SALD	0.6	**10k**	**20.50**	**29.55**	**0.7452**
MultiGen	SANA	Base	1.6	12.5k	25.87	26.26	0.7593
		SpeeD	1.6	12.5k	25.09	26.33	0.7555
		SRA	1.6	12.5k	24.96	26.40	0.7517
		Temporal Diff	1.6	12.5k	24.84	26.47	0.7550
		SALD	1.6	**10k**	**23.86**	**26.29**	**0.7533**
	SANA	Base	0.6	12.5k	31.89	26.68	0.7648
		SpeeD	0.6	12.5k	30.93	26.75	0.7610
		SRA	0.6	12.5k	30.77	26.82	0.7572
		Temporal Diff	0.6	12.5k	30.61	26.89	0.7534
		SALD	0.6	**10k**	**29.15**	**26.69**	**0.7472**
	PixArt-Σ	Base	0.6	12.5k	31.57	25.88	0.7702
		SpeeD	0.6	12.5k	30.62	25.95	0.7664
		SRA	0.6	12.5k	30.46	26.02	0.7626
		Temporal Diff	0.6	12.5k	30.31	26.09	0.7588
		SALD	0.6	**10k**	**30.47**	**25.93**	**0.7292**

Table 6: Quantitative comparison across different inference steps on the Oxford Flowers dataset. Light blue highlights indicate the metric values for the **SALD** variant.

Model	Params (B)	Variant	Steps = 10			Steps = 15			Steps = 20		
			FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓	FID↓	CLIP↑	LPIPS↓
SANA	1.6	Base	55.06	25.35	0.8153	37.49	25.60	0.8055	33.11	26.35	0.7896
		SALD	41.56	25.48	0.8088	30.60	25.72	0.7962	27.37	26.57	0.7781
SANA	0.6	Base	44.05	24.31	0.8059	35.26	25.29	0.7971	30.64	26.43	0.7805
		SALD	33.68	24.48	0.7906	25.54	25.63	0.7865	25.83	26.65	0.7728
PixArt-Σ	0.6	Base	49.12	25.10	0.8120	43.25	25.25	0.7910	37.88	25.37	0.7687
		SALD	46.35	25.28	0.7985	40.80	25.38	0.7820	36.48	25.42	0.7682

the 10-step output already closely resembling its 15- and 20-step generations. In contrast, the baseline remains highly unstable at 5 and 10 steps, exhibiting substantial changes in geometry, illumination, and color before converging toward a consistent image at later steps. Thus, additional sampling primarily refines the SALD output rather than correcting its global structure, whereas the baseline requires more iterations to stabilize both structure and appearance. This visual progression further supports that SALD learns a more efficient generation trajectory, allowing coherent samples to emerge under substantially fewer inference steps.

D.4 What happens if we train 10× More?

To examine whether the gains reported at the main training budget are a consequence of insufficient training of the Baseline, we extend both the Baseline and SALD models to 100k training steps, corresponding to 10× the standard 10k-step training budget. Table 7 reports the results across SANA 1.6B and SANA 0.6B on the MultiGen and Oxford 102 Flowers datasets. SALD maintains a consistent

Table 7: Evaluation metrics across training steps for SANA 1.6 and SANA 0.6 on MultiGen and Oxford 102 Flowers datasets. Lower FID and LPIPS (↓) are better, while higher CLIP (↑) is better.

Steps	Variant	MultiGen						Oxford 102 Flowers					
		SANA 1.6			SANA 0.6			SANA 1.6			SANA 0.6		
		FID	CLIP	LPIPS	FID	CLIP	LPIPS	FID	CLIP	LPIPS	FID	CLIP	LPIPS
10k	Baseline	26.44	26.25	0.7541	31.93	26.68	0.7495	33.11	26.35	0.7896	30.64	26.43	0.7805
	SALD	23.86	26.29	0.7533	29.15	26.69	0.7472	27.37	26.57	0.7781	25.83	26.65	0.7728
20k	Baseline	26.18	26.31	0.7550	31.42	26.48	0.7605	32.45	26.30	0.7842	30.41	26.38	0.7761
	SALD	23.52	26.35	0.7542	28.78	26.52	0.7598	26.80	26.52	0.7745	25.70	26.60	0.7695
30k	Baseline	25.90	26.19	0.7548	31.10	26.47	0.7588	31.82	26.24	0.7791	30.19	26.32	0.7719
	SALD	23.21	26.38	0.7540	28.45	26.50	0.7580	26.35	26.46	0.7712	25.58	26.54	0.7665
40k	Baseline	25.65	26.15	0.7539	30.87	26.43	0.7588	31.25	26.18	0.7743	29.98	26.26	0.7680
	SALD	22.95	26.40	0.7535	28.15	26.48	0.7575	25.98	26.40	0.7682	25.47	26.48	0.7638
50k	Baseline	25.47	26.08	0.7537	30.29	26.39	0.7596	30.75	26.12	0.7698	29.79	26.20	0.7643
	SALD	22.73	26.41	0.7532	28.63	26.41	0.7576	25.68	26.34	0.7655	25.38	26.41	0.7613
60k	Baseline	25.24	26.11	0.7518	30.75	26.44	0.7584	30.30	26.06	0.7656	29.61	26.14	0.7609
	SALD	22.55	26.42	0.7520	28.35	26.45	0.7570	25.42	26.27	0.7630	25.30	26.34	0.7590
70k	Baseline	25.08	26.11	0.7520	30.86	26.44	0.7586	29.90	26.00	0.7617	29.45	26.08	0.7577
	SALD	22.40	26.42	0.7518	28.18	26.46	0.7568	25.20	26.20	0.7607	25.23	26.27	0.7569
80k	Baseline	24.94	26.11	0.7522	30.02	26.42	0.7587	29.55	25.94	0.7581	29.30	26.02	0.7547
	SALD	22.28	26.42	0.7517	28.05	26.45	0.7565	25.02	26.12	0.7586	25.18	26.19	0.7550
90k	Baseline	24.82	26.11	0.7521	30.02	26.43	0.7586	29.25	25.89	0.7548	29.16	25.96	0.7519
	SALD	22.18	26.42	0.7516	27.95	26.44	0.7562	24.95	26.04	0.7567	25.15	26.11	0.7533
100k	Baseline	24.70	26.11	0.7522	30.00	26.43	0.7586	28.61	25.84	0.7378	28.71	25.84	0.7373
	SALD	22.12	26.21	0.7531	28.04	26.39	0.7539	24.90	25.75	0.7393	25.14	25.93	0.7335

FID advantage over the corresponding Baseline throughout the extended training horizon, while CLIP and LPIPS remain comparable. For example, at 10k steps, SALD already achieves substantially lower FID than the Baseline, reducing it from 26.44 to 23.86 on MultiGen and from 33.11 to 27.37 on Oxford 102 Flowers with SANA 1.6B. Importantly, this advantage does not disappear as training continues, demonstrating that the improvements observed at the main 10k-step setting are not simply a consequence of under-training the Baseline.

The extended training results also reveal diminishing returns at later stages of optimization. After approximately 80k steps, both models improve more slowly. SALD maintains a clear FID advantage over the Baseline in this later regime: the gap remains stable on MultiGen (2.58 at both 10k and 100k for SANA 1.6B) and narrows but persists on Oxford 102 Flowers (from 5.74 to 3.71 for SANA 1.6B and from 4.81 to 3.57 for SANA 0.6B). Thus, extending training to 100k steps does not eliminate the advantage of SALD. Instead, the results demonstrate that the gain introduced by SALD persists throughout the extended training process, while further optimization beyond 80k steps provides only marginal improvements for either method. These observations support the use of 10k steps as a practical training budget for the main experiments and confirm that the reported improvements are not due to premature termination of Baseline training.

D.5 Additional Visual Samples

We provide additional visual samples generated by SALD across multiple datasets, architectures, and resolutions to analyze its ability to preserve semantic consistency, structural fidelity, and fine-grained visual details under diverse generation settings.

Figure 6 presents qualitative comparisons between baseline finetuning and SALD across multiple architectures on the MAGICK and DALL·E 3 1M datasets at 1024×1024 resolution. Across all settings, SALD produces sharper and more structurally coherent generations with improved preservation of fine-grained semantic details. On MAGICK, baseline models often generate unstable crown geometry, distorted facial structure, inconsistent jewelry placement, and blurred ornamental details, whereas SALD maintains cleaner symmetry, sharper textures, and more realistic facial composition. Similarly, on DALL·E 3 1M, baseline generations exhibit inconsistent hand–object interactions, oversmoothed hair textures, and weaker illumination consistency around the magical orb.

In contrast, SALD generates more stable anatomy, finer hair strands, improved lighting coherence, and stronger object-boundary preservation. These improvements are consistent across SANA and PixArt variants, demonstrating that SALD effectively enhances high-frequency detail generation and semantic consistency at high resolutions.

Figure 7 further showcases diverse samples generated by SALD using PixArt-Σ at 1024×1024 resolution on the DALL·E 3 1M dataset. Additionally, Figure 10 further showcases diverse samples generated by SALD using Flux.2 klein at 1024×1024 resolution on the DALL·E 3 1M dataset. The generated images demonstrate strong semantic consistency, realistic illumination, and preservation of fine-grained textures across a wide variety of scenes and subjects. In portrait generations, SALD maintains detailed facial structure, natural skin textures, and coherent lighting interactions. For reflective and transparent objects, such as the crystal sphere, the model accurately captures highlights, reflections, and depth cues. Indoor and environmental scenes exhibit consistent global composition and soft lighting transitions, while animal generations preserve high-frequency fur textures and sharp eye details. These results highlight the ability of SALD to generate visually coherent and high-fidelity images across diverse semantic categories at high resolution.

Figure 11 presents qualitative generations produced by SALD across diverse datasets, resolutions, and backbone architectures, demonstrating both visual fidelity and semantic consistency. In the first row, SALD applied to SANA 1.6B trained on DALL·E 3 1M generates highly detailed urban neon scenes with realistic illumination effects, accurate text rendering, and strong spatial coherence at 1024×1024 resolution. The reflections on wet cobblestone surfaces and the natural bokeh distribution highlight the model's ability to preserve fine-grained lighting structure and texture realism. The second row showcases generations on the CUB-200 dataset, where SALD captures intricate feather structures, wing symmetry, and motion-aware details in a snowy owl generated from an out-of-dataset prompt, while maintaining sharp focus on salient regions such as the eyes and plumage patterns. Finally, the third row demonstrates PixArt-Σ generations on Oxford-102 Flowers, where SALD produces smooth petal geometry, subtle color gradients, and realistic depth-of-field effects with visually pleasing softness and fine texture retention. Across all settings, the generated samples exhibit strong compositional stability, high-frequency detail preservation, and improved prompt alignment, indicating that SALD generalizes effectively across both natural and artistic image domains.

E LIMITATIONS

SALD evaluates each training example through two noise levels using the same diffusion model: the lower-noise easy path provides an internal reference under `no_grad`, whereas the higher-noise hard path supplies the training gradient. Consequently, SALD requires neither an external teacher nor additional trainable model parameters during training or inference. Nevertheless, the additional easy-path evaluation increases the per-step training cost by approximately 25%. Although our wall-clock-matched experiments (Table 5) show that the resulting improvement in optimization efficiency can offset this overhead, reducing the per-step training cost remains an important direction for future work. One possibility is to cache easy-path reference predictions from earlier checkpoints. However, such cached predictions would become stale as the model changes and would depart from the formulation considered in this work, where the reference is always computed with the current model parameters.

TAM stores one non-trainable scalar for each training sample. This memory cost is negligible for the dataset sizes considered in our experiments but grows linearly with the number of training examples. Moreover, TAM updates a sample's memory only when that sample is revisited, so its curriculum effect is weaker when each example is seen only a few times during training. For extremely large web-scale datasets, group-level, clustered, or compressed difficulty statistics could reduce this cost and increase the update frequency, although such approximations may sacrifice the precision of the per-sample curriculum.

Finally, our empirical evaluation is limited to fine-tuning pretrained text-to-image flow-matching models; we do not evaluate SALD for training from scratch. Our convergence analysis also holds only for a fixed TAM memory state (Remark 4). While we describe how the same easy–hard self-referencing principle can be extended to DDPM-style noise-prediction objectives, we do not experimentally validate that extension in this work. Evaluating SALD with noise-prediction models, in from-scratch training, and in other generative modalities, including video and 3D generation, remain directions for future research.



A realistic close-up portrait of a young woman with heterochromia eyes, soft golden hour sunlight streaming through her curly auburn hair, skin pores visible, subtle freckles, shallow depth of field, bokeh background of a blurred autumn forest.



A stunning macro photograph of a single soap bubble resting on a dewy spider web at sunrise, the bubble perfectly reflecting a miniature world inside green meadows and a tiny golden sun surrounded by crystalline water droplets catching rainbow light.

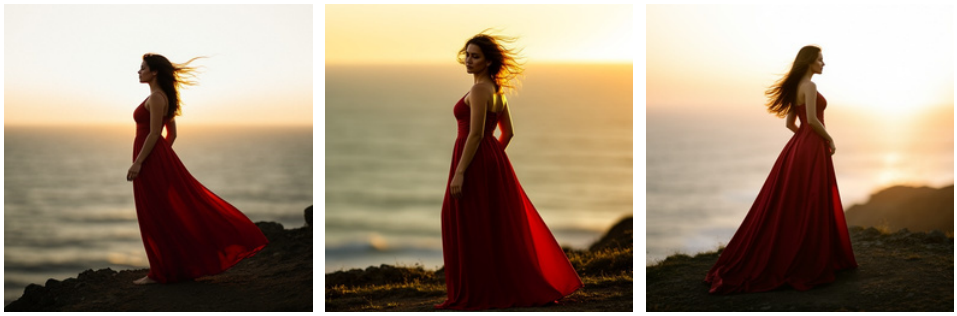


A serene Japanese tea house interior at dawn, tatami mats, paper shoji screens glowing soft warm amber, a ceramic matcha bowl with wisps of steam rising, single cherry blossom branch visible through an open window over a koi pond, morning mist outside, dust particles floating in light beams, wabi-sabi aesthetic.



Extreme close-up of a male lion's face during golden hour, individual fur strands visible, amber eyes with detailed iris texture, dust particles floating in warm backlight, shallow depth of field blurring the savanna behind.

Figure 9: Qualitative samples generated by SALD using PixArt-Σ at 1024×1024 resolution on the DALL·E 3 1M dataset.



A photograph of a woman in a owing red silk dress standing on a windswept cli at golden hour, hair caught mid-motion by the breeze, warm backlight rim-lighting her silhouette, soft blurred ocean behind.



A photograph of a couple dancing barefoot on a moonlit beach, waves gently lapping at their feet, warm string lights strung between palm trees, soft bokeh, romantic dusk sky in deep blues and pinks.



Close-up portrait of a woman crowned with fresh white owers in her braided hair, soft glowing skin, gentle smile, warm afternoon light ltering through green leaves behind her.



A photograph of a ballerina mid-leap in a sunlit studio, tulle skirt frozen in motion, dust particles glowing in the light beams, polished wood oor re ecting her silhouette, soft warm tones throughout.

Figure 10: Qualitative samples generated by SALD using Flux.2 klein Base at 1024×1024 resolution on the DALL·E 3 1M dataset.



A photograph of a glowing neon sign spelling "SALD" in bright electric blue light, mounted on a dark rainy night brick wall, neon light reflecting on wet cobblestone street below, bokeh background of blurred city lights.



A photograph of a snowy owl in mid-flight, pure white feathers with individual barb detail, golden piercing eyes focused forward, wings fully spread showing intricate feather layers, dramatic overcast sky background, motion blur on wingtips.



A close-up of a single white magnolia flower in full bloom, delicate silky petals with soft blush pink edges, fine petal surface texture, a dewdrop resting on petals, soft diffused morning light, pale cream and ivory tones, shallow depth of field, dreamy bokeh background in soft white and sage green.

Figure 11: Qualitative samples generated by SALD across multiple datasets and architectures. Row 1: SANA 1.6B at 1024×1024 resolution trained on DALL·E 3 1M. Row 2: SANA 1.6B generations on CUB-200. Row 3: PixArt-Σ generations on Oxford-102 Flowers.