SCALE: Improving Guidance-Free Visual Generation via Self-Contrastive Alignment for Likelihood Estimation

Jiwan Hur **Dong-Jae Lee** **Dongyeun Lee** **Gyojin Han** **Junmo Kim**
KAIST, South Korea
{jiwan.hur, jhtwosun, ledoye, hangj0820, junmo.kim}@kaist.ac.kr
Project page: https://jiwanhur.github.io/SCALE

Abstract

Classifier-free guidance (CFG) is essential for improving sample quality in visual generation, but it incurs substantial sampling overhead. Existing guidance-free approaches halve this cost by approximating the CFG with a single model inference, but their performance is bounded by the CFG and inherits its limitations. In this paper, we propose *self-contrastive guidance* (SCG), which contrasts the model's prediction on real and self-generated samples to obtain a direct, model-based signal that pushes generation toward the true data manifold. Integrating SCG into the standard likelihood objective via reparameterization yields *Self-Contrastive Alignment for Likelihood Estimation* (SCALE), a guidance-free training framework that produces SCG-enhanced predictions in a single forward pass. Across class-conditional and text-to-image generation on diffusion and autoregressive models, SCALE consistently outperforms CFG-mimicking guidance-free baselines and improves further when combined with DPO, demonstrating that the self-contrastive signal yields gains complementary to both CFG and preference-based fine-tuning.

1 Introduction

Likelihood-based generative models, including diffusion [19, 53] and autoregressive (AR) models [48, 54], have demonstrated remarkable scalability and performance. Yet, for complex visual datasets, trained models often produce samples whose clarity falls short of the training images. Classifier-free guidance (CFG) [18] has thus become the de facto remedy. By contrasting unconditional and conditional predictions, CFG effectively enhances fidelity, but at the cost of doubled sampling cost.

To eliminate this overhead, recent guidance-free approaches approximate the CFG distribution with a single inference via teacher distillation [41], contrastive objectives [6], or direct reparametrization of the CFG objective [7]. While effective, these methods are fundamentally constrained to CFG and are bounded by its performance. Furthermore, recent analyses have clarified that CFG is not flawless. A primary challenge is that condition alignment and sample fidelity are entangled along a single guidance. Therefore, amplifying the CFG for higher fidelity over-amplifies the conditional signal. This often leads to distorted sampling trajectories [8], greatly exaggerated truncation [26], and mode dropping that collapses samples onto a narrow set of per-class templates [30] (Fig. 1a). Whether a single-model approach can move beyond these limitations, therefore, remains an open question.

A complementary line of work, information-based sampling guidance [22, 24] generalizes CFG by amplifying any salient information while avoiding a degraded reference that lacks it. While effective at targeting specific quality dimensions, these methods still incur additional sampling-time forward passes (if combined with CFG, the total cost steeply rises to three model evaluations per step) and rely on heuristic, domain-specific definitions of the salient information and the degraded reference.

In this paper, we propose *self-contrastive guidance* (SCG), which contrasts the model's prediction on real and self-generated samples (Fig. 1b). This contrast yields a direct, model-based estimate of the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

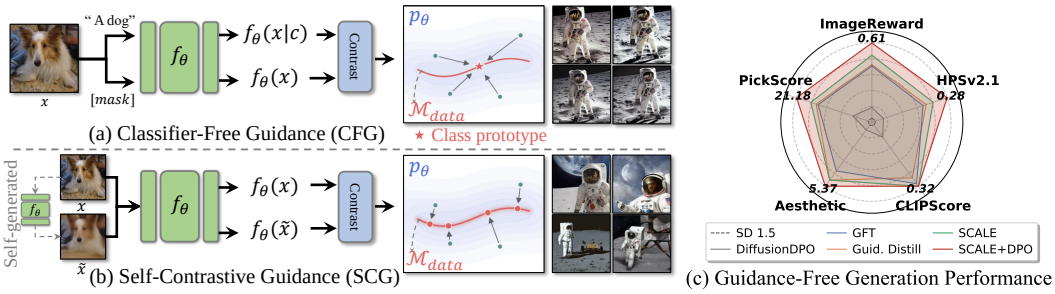


Figure 1: **Comparison between CFG and our self-contrastive guidance (SCG).** (a) CFG amplifies the conditional signal to improve sample quality, often reducing sample diversity by overemphasizing the condition. (b) SCG derives a model-based signal that measures the discrepancy between real and self-generated samples, pushing generation toward the true data manifold. (c) SCALE integrates CFG and SCG into a single model, achieving superior guidance-free generation performance.

model's own approximation error, and extrapolating along it pushes the generation process toward the true data manifold. Unlike prior information-based guidance, SCG is domain-agnostic and is directly applicable to both diffusion and AR models as it relies only on model predictions.

By reparametrizing the SCG signal as a learnable term within the standard likelihood objective, we obtain *Self-Contrastive Alignment for Likelihood Estimation* (SCALE), a guidance-free training framework in which the model learns to directly produce SCG-enhanced predictions in a single forward pass. For conditional generation, we further integrate CFG into SCALE so that the trained model directly refines the CFG distribution rather than simply mimicking it.

Experimental results show that SCALE consistently surpasses existing guidance-free methods that only approximate CFG across diffusion and AR architectures, on both class-conditional ImageNet (DiT [44], LlamaGen [54]) and text-to-image generation (Stable Diffusion v1.5 [49]). It matches DPO-finetuned guidance-free models alone, and applying DPO on top of SCALE brings further gains (Fig. 1c). SCALE introduces minimal additional training overhead and complexity without compromising the base model's capabilities. As a result, SCALE provides an orthogonal direction for improving guidance-free visual generation, widely applicable to both diffusion and AR models.

2 Background

2.1 Likelihood-based Generative Models in Vision

Likelihood-based generative models aim to explicitly model the data distribution $p(\boldsymbol{x})$. In this paper, we focus on two representative model types: diffusion [19] and autoregressive (AR) [54] models.

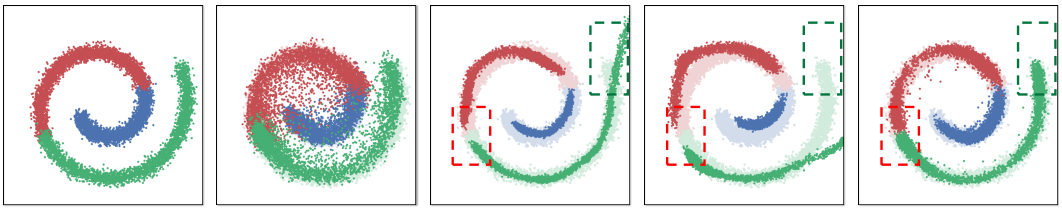
Diffusion models [19] aim to generate continuous data, where directly maximizing likelihood is intractable. These models learn to reverse a forward noising process that incrementally adds Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to a data $\boldsymbol{x}$. The reverse process is learned via a score-matching objective [53], where a neural network $\boldsymbol{\epsilon}_\theta^c(\boldsymbol{x}_t, t, c)$ with condition c is trained to approximate the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ through noise prediction from noisy sample $\boldsymbol{x}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}$ at timestep $t \in [0, T]$:

$$\mathcal{L}_\theta^{\text{DM}} = \mathbb{E}_{(\boldsymbol{x}_0, c) \sim p_{\text{data}}, \boldsymbol{\epsilon}, t} \left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta^c(\boldsymbol{x}_t, t, c)||^2 \right], \tag{1}$$

where α_t and σ_t are scheduling functions satisfying $\alpha_t^2 + \sigma_t^2 = 1$, and $\boldsymbol{x}_T \approx \boldsymbol{\epsilon}$, $\boldsymbol{x}_0 \approx \boldsymbol{x}$. After training, the optimal noise predictor satisfies $\boldsymbol{\epsilon}_\theta^*(\boldsymbol{x}_t) = -\sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, where p_t denotes the marginal distribution at step t, enabling data generation via progressive denoising from $t = T$ to 0.

Autoregressive models [58, 59] learn the joint distribution $p(\mathbf{x}, c) = \prod_{i=1}^N p^c(\boldsymbol{x}_i | \boldsymbol{x}_{<i}, c)$, where $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ is a discrete token sequence (e.g., vector-quantized image patches [48]) and $\boldsymbol{x}_{<i}$ denotes all tokens preceding $\boldsymbol{x}_i$. AR models are trained by minimizing the cross-entropy loss:

$$\mathcal{L}_\theta^{\text{AR}} = \mathbb{E}_{(\mathbf{x}, c) \sim p_{\text{data}}} \left[-\sum_{i=1}^N \log p_\theta^c(\boldsymbol{x}_i | \boldsymbol{x}_{<i}, c) \right]. \tag{2}$$



(a) Ground truth (b) Cond. Sample (c) Guid. Distill (d) GFT (e) SCALE (Ours)

Figure 2: **Toy experiment on guidance-free generation with diffusion models. (a)** Ground-truth samples from a three-class 2D Swiss-roll distribution. **(b)** Unguided conditional sampling scatters outliers across low-density regions. **(c, d)** Guidance-free methods that mimic CFG [41, 7] generate samples that drift away from class boundaries (red boxes) and off the data manifold (green boxes). **(e)** SCALE sharpens the distribution while preserving both the class boundary and the data manifold. The effective guidance scale is fixed at 5.0 across all guidance-free methods.

Classifier-free guidance. For complex visual datasets, finite-capacity networks often produce samples whose clarity falls short of the training images due to approximation errors. To mitigate this gap, likelihood-based visual generative models, including diffusion and AR models, commonly employ classifier-free guidance (CFG) [11, 18, 4], which amplifies the conditional signal by extrapolating between the unconditional prediction f_θ^u and the conditional prediction f_θ^c with scale s_{cfg}:

$$f_\theta^{\text{cfg}}(\boldsymbol{x}_t, c) = f_\theta^c(\boldsymbol{x}_t, c) + s_{\text{cfg}}\,(f_\theta^c(\boldsymbol{x}_t, c) - f_\theta^u(\boldsymbol{x}_t)). \tag{3}$$

For a unified formulation, we let $\boldsymbol{x}_t$ denote the model input at step t: the noised sample in diffusion models, or the preceding token sequence $\boldsymbol{x}_{<t}$ in AR models.[1] Accordingly, f_θ is a parameterized network that serves as the noise predictor or the next-token predictor, respectively. Likewise, p_t denotes the marginal at noise level t in diffusion and the next-token distribution given $\boldsymbol{x}_{<t}$ in AR.

2.2 Guidance-Free Visual Generation via Mimicking CFG

Several approaches approximate the CFG distribution p_t^{cfg} with a single inference, including guidance distillation [41] for diffusion models, Condition Contrastive Alignment (CCA) [6] for AR models, and Guidance-Free Training (GFT) [7] for both. GFT in particular integrates the CFG output directly into standard likelihood training by introducing a pseudo-temperature $\beta := 1/(1+s_{\text{cfg}})$ and reformulating the conditional output f_θ^c in Eq. (3) as an interpolation of CFG and unconditional predictions:

$$f_\theta^c(\boldsymbol{x}_t, c, \beta) = \beta f_\theta^{\text{cfg}}(\boldsymbol{x}_t, c, \beta) + (1-\beta)\text{sg}\left[f_\theta^u(\boldsymbol{x}_t|\varnothing, 1)\right], \tag{4}$$

where a stop-gradient $\mathbf{sg}[\cdot]$ is applied to keep the training cost minimal without affecting convergence.

Despite halving the inference cost of CFG, these methods remain constrained to the CFG-induced distribution p_t^{cfg} as their training target, which alone is insufficient for high-quality generation due to its inherent limitations. In the following section, we examine these limitations and motivate a complementary guidance signal that directly improves the guidance-free visual generation.

3 Revisiting Information-Based Guidance for Guidance-Free Training

CFG is not flawless. CFG enhances conditional generation by implicitly amplifying the conditional density to a power greater than one, sharpening the distribution toward its modes [2]. While effective, recent analyses have clarified that CFG does not merely trade compute for quality: its unconditional-to-conditional extrapolation fundamentally entangles prompt alignment and perceptual quality, making them difficult to control independently [26]. Consequently, scaling up CFG for higher fidelity disproportionately amplifies the conditional signal, driving samples off the data manifold and truncating the distribution toward a narrow set of per-class "templates" [30, 8]. Existing guidance-free generation methods halve the inference cost of CFG, but they do not address this underlying distortion. We illustrate this in Fig. 2. CFG-mimicking guidance-free methods over-amplify the conditioning signal, pulling samples away from class boundaries and off the data manifold (Fig. 2c and 2d).

[1] We slightly abuse the notation t to represent both inputs; t denotes the timestep in diffusion, or the token index in AR.

Generalized Information-Based Guidance. To address these limitations, we first revisit a complementary line of work, information-based guidance [22, 24]. It generalizes CFG by steering the sampling toward amplifying any salient information h_t while avoiding degraded input samples $\bar{\boldsymbol{x}}_t$ lacking it. For diffusion models, this is formulated as

$$\hat{\boldsymbol{\epsilon}}_\theta(\bar{\boldsymbol{x}}_t, h_t) = \boldsymbol{\epsilon}_\theta(\bar{\boldsymbol{x}}_t, h_t) + s\,(\boldsymbol{\epsilon}_\theta(\bar{\boldsymbol{x}}_t, h_t) - \boldsymbol{\epsilon}_\theta(\bar{\boldsymbol{x}}_t)), \tag{5}$$

and extends to discrete generative models from the optimization perspective[2]:

$$\arg\max_{\bar{\boldsymbol{x}}_t}[\log p_\theta(\bar{\boldsymbol{x}}_t, h_t) + s\,(\log p_\theta(\bar{\boldsymbol{x}}_t, h_t) - \log p_\theta(\bar{\boldsymbol{x}}_t))], \tag{6}$$

where s is a guidance scale. Both formulations play the same role of amplifying h_t throughout the sampling process: Eq. (5) minimizes the gradient of an implicit regressor $\nabla_{\bar{\boldsymbol{x}}_t} \log p_{\text{im}}(h_t|\bar{\boldsymbol{x}}_t)$ in the continuous score domain, whereas Eq. (6) directly amplifies the probability density $\log p_\theta(h_t|\bar{\boldsymbol{x}}_t)$. While these methods can target specific quality dimensions, they require additional sampling-time forward passes, and the total cost rises to three model evaluations per step if combined with CFG. Furthermore, they rely on heuristic and domain-specific choices for defining $\bar{\boldsymbol{x}}_t$ and h_t, such as applying a Gaussian blur with attention mask [22] or utilizing models trained on auxiliary tasks [24].

Our approach. Our key idea is to contrast the model's prediction on a real sample with its prediction on a self-generated sample. This contrast provides a direct model-based signal of the discrepancy between real and self-generated samples, and pushing generation along this direction encourages samples to move toward the data manifold. Because this signal depends on real samples, it is not available at sampling time, but it arises naturally during training. We therefore incorporate it into the guidance-free training objective so that the model learns to approximate the same guidance effect with a single forward pass at inference time (Fig. 2e). Furthermore, since this formulation relies only on comparing model outputs, it applies broadly across likelihood-based generative models. In this paper, we specifically focus on two representative model types: diffusion and AR architecture.

4 Self-Contrastive Alignment for Likelihood Estimation

In this section, we formalize self-contrastive guidance (SCG) and integrate it into a guidance-free training objective, named Self-Contrastive Alignment for Likelihood Estimation (SCALE).

4.1 Self-Contrastive Guidance

We formulate SCG by contrasting outputs from real and self-generated samples. Formally, we define this by substituting a pair of real and generated samples $(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t)$ into Eqs. (5) and (6), by setting $\bar{\boldsymbol{x}}_t = \tilde{\boldsymbol{x}}_t$ and $h_t = \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t$ and assuming $f_\theta(\tilde{\boldsymbol{x}}_t, \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t) = f_\theta(\boldsymbol{x}_t)$:

$$f_\theta^{\text{scg}}(\boldsymbol{x}_t) = f_\theta(\boldsymbol{x}_t) + s_{\text{scg}}\big(f_\theta(\boldsymbol{x}_t) - f_\theta(\tilde{\boldsymbol{x}}_t)\big), \tag{7}$$

where s_{scg} controls the strength of the SCG. This formulation applies to a broad class of generative models, including both continuous and discrete modalities. By the definition of information-based guidance above, Eq. (7) steers the sampling trajectory toward enhancing $h_t = \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t$, locally amplifying the model's residual contrast between real and self-generated supports. This residual encodes how the model's prediction on a real sample diverges from its prediction on the self-generated sample, and extrapolating along it pushes the generation toward the high-density region of the data.

4.2 Guidance-Free Training with Self-Contrastive Guidance

We integrate SCG into maximum likelihood training, so that the trained model directly approximates the self-contrastive signal and performs guidance-free sampling at sampling time. Inspired by recent guidance-free training methods [7], we reparameterize the model output $f_\theta(\boldsymbol{x}_t)$ as an interpolation between the target SCG output $f_\theta^{\text{scg}}(\boldsymbol{x}_t)$ and the prediction from the self-generated sample $f_\theta(\tilde{\boldsymbol{x}}_t)$:

$$f_\theta(\boldsymbol{x}_t) = \frac{1}{1 + s_{\text{scg}}} f_\theta^{\text{scg}}(\boldsymbol{x}_t) + \frac{s_{\text{scg}}}{1 + s_{\text{scg}}} f_\theta(\tilde{\boldsymbol{x}}_t). \tag{8}$$

[2]While Hur et al. [24] formulate this with $p_\theta(\tilde{\boldsymbol{x}}_t|h_t)$, we instead use the joint $p_\theta(\tilde{\boldsymbol{x}}_t, h_t)$. See Appendix A for details.

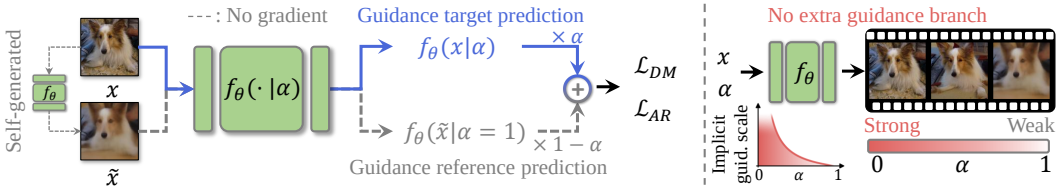


Figure 3: Illustration of SCALE algorithm without generation step t for brevity. **Left:** SCALE training derives a model-based signal that measures the discrepancy between real ($\boldsymbol{x}$) and self-generated ($\tilde{\boldsymbol{x}}$) samples, learning to approximate the self-contrastive guidance with condition α. **Right:** At sampling time, lowering α yields high-quality samples by amplifying the implicit guidance signal.

To enable the model to generalize across various guidance strengths, we train the model $f_\theta(\boldsymbol{x}_t|\alpha)$ conditioned on a pseudo-temperature $\alpha := 1/(1+s_{\text{scg}})$ from Eq. (8):

$$f_\theta(\boldsymbol{x}_t \mid \alpha) = \alpha f_\theta^{\text{scg}}(\boldsymbol{x}_t \mid \alpha) + (1-\alpha)\text{sg}[f_\theta(\tilde{\boldsymbol{x}}_t \mid 1)]. \tag{9}$$

Here, $\alpha = 1$ eliminates SCG ($s_{\text{scg}} = 0$), while decreasing α amplifies the self-contrastive signal. Substituting Eq. (9) into standard likelihood objectives (Eqs. (1) or (2)) yields SCALE training objective. To ensure the self-generated branch $f_\theta(\tilde{\boldsymbol{x}}_t \mid 1)$ serves as a static reference for guidance, we apply a stop-gradient operator sg$[\cdot]$ and fix $\alpha = 1$ on this branch.

We now characterize the optimal SCG prediction learned by SCALE training.

Theorem 4.1 (Optimal Solution of SCALE Training)**.** *At an ideal regression optimum with unlimited model capacity and training data, assume the base predictor $\boldsymbol{\epsilon}_{\theta^*}(\cdot \mid 1)$ matches the scaled data score at the real and reference inputs. For $\alpha \in (0,1]$, the SCG branch satisfies*

$$\boldsymbol{\epsilon}_{\theta^*}^{scg}(\boldsymbol{x}_t \mid \alpha) \;=\; -\sigma_t \left[\frac{1}{\alpha}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) - \left(\frac{1}{\alpha}-1\right)\mathbb{E}\left[\nabla_{\tilde{\boldsymbol{x}}_t}\log p_t(\tilde{\boldsymbol{x}}_t) \mid \boldsymbol{x}_t\right]\right], \tag{10}$$

where $\tilde{\boldsymbol{x}}_t = g_{\theta^*}(\boldsymbol{x}_t, \boldsymbol{\epsilon})$ is obtained by estimating the clean sample with the base predictor and re-noising it to level t using $\boldsymbol{\epsilon}$. The expectation is taken over $\boldsymbol{\epsilon}$.

We defer the proof and an analogous result for AR models to Appendix B. Since $\alpha = 1/(1+s_{\text{scg}})$, the SCALE-trained diffusion model outputs the conditional mean SCG prediction in a single forward pass at convergence, with guidance strength controlled by α. Fig. 3 illustrates the SCALE training and sampling without external condition c.

4.3 Practical Implementation

Regularizing SCALE via interpolation. The analysis above assumes that $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_t$ lie on nearby supports so that the prediction $f_\theta(\tilde{\boldsymbol{x}}_t)$ is well defined and the local density-ratio is informative. In practice, a self-generated sample can be highly degraded under limited model capacity or the stochasticity of early generation steps ($t \ll T$), making direct contrast unstable [26, 25].

To address this, we regularize SCALE training by interpolating the real and self-generated samples into an intermediate support, thereby providing a stable anchor for the guidance. Concretely, we replace a fraction of patches in $\boldsymbol{x}_t$ with those from $\tilde{\boldsymbol{x}}_t$. Given an interpolation ratio $\lambda \in [0,1]$ and a per-patch Bernoulli mask $M_\lambda \sim \text{Bernoulli}(\lambda)$, the mixed sample $\tilde{\boldsymbol{x}}_{t,\lambda}$ is defined as

$$\tilde{\boldsymbol{x}}_{t,\lambda} := \text{PatchMix}(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t; \lambda) = M_\lambda \odot \tilde{\boldsymbol{x}}_t + (1 - M_\lambda) \odot \boldsymbol{x}_t, \tag{11}$$

where $\odot$ denotes patch-wise multiplication. This preserves local semantic structures of dense visual data and applies uniformly across both continuous and discrete data representations, unlike linear interpolation, which is ill-defined for vector-quantized tokens (see Appendix G.2). With this interpolated output, the SCG and SCALE formulations in Eq. (7) and Eq. (9) can be redefined as

$$f_\theta^{\text{scg}}(\boldsymbol{x}_t) = f_\theta(\boldsymbol{x}_t) + s_{\text{scg}}(f_\theta(\boldsymbol{x}_t) - f_\theta(\tilde{\boldsymbol{x}}_{t,\lambda})). \tag{12}$$

$$f_\theta(\boldsymbol{x}_t \mid \alpha) = \alpha f_\theta^{\text{scg}}(\boldsymbol{x}_t \mid \alpha) + (1-\alpha)\text{sg}[f_\theta(\tilde{\boldsymbol{x}}_{t,\lambda} \mid 1)]. \tag{13}$$

Algorithm 1: SCALE Training (Diffusion)

Initialize denoiser $\boldsymbol{\epsilon}_\theta$ and ratio λ
for each gradient step **do**
  $\boldsymbol{x}, c \sim p(\boldsymbol{x}, c)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,
  $t, \alpha \sim U(\alpha_{\min}, 1)$, $\beta \sim U(0, 1)$
  $\boldsymbol{x}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}$
  $\boldsymbol{\epsilon}_\theta^{\text{scg-cfg}} = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t | c, \alpha, \beta)$ ▷ *Target*
  // Additional to Training (Evaluation Mode)
  $\boldsymbol{\epsilon}_\theta^u = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t | \varnothing, 1, 1)$
  Sample $\tilde{\boldsymbol{x}}_t$ from $\boldsymbol{\epsilon}_\theta^u$
  $\tilde{\boldsymbol{x}}_{t,\lambda} = \text{PatchMix}(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t; \lambda)$ ▷ *Eq. (11)*
  $\boldsymbol{\epsilon}_\theta^\lambda = \boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t,\lambda} | c, 1, \beta)$
  // Nested guidance formulation ▷ *Eq. (16)*
  $\boldsymbol{\epsilon}_\theta^{\text{cfg}} = \alpha\, \boldsymbol{\epsilon}_\theta^{\text{scg-cfg}} + (1-\alpha)\, \text{sg}[\boldsymbol{\epsilon}_\theta^\lambda]$
  $\boldsymbol{\epsilon}_\theta^c = \beta\, \boldsymbol{\epsilon}_\theta^{\text{cfg}} + (1-\beta)\, \text{sg}[\boldsymbol{\epsilon}_\theta^u]$
  $\theta \leftarrow \theta - \mu \nabla_\theta \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta^c\|_2^2$ ▷ *Eq. (1)*
end for

Table 1: **Comparison of guidance-free generation methods.** Computational costs are reported relative to CFG for both training and sampling. SCALE keeps the low training overhead while halving inference cost and improving the target distribution alignment.

Method	Distill	CCA	GFT	**SCALE**
Guidance target	$\frac{p_t(\boldsymbol{x}\|c)}{p_t(\boldsymbol{x})}$	$\frac{p_t(\boldsymbol{x}\|c)}{p_t(\boldsymbol{x})}$	$\frac{p_t(\boldsymbol{x}\|c)}{p_t(\boldsymbol{x})}$	$\frac{p_t^{\text{cfg}}(\boldsymbol{x}\|c)}{p_t^{\text{cfg}}(\tilde{\boldsymbol{x}}\|c)}$
Applicable to	Diffusion	Discrete	All	All
Train time / Step	$\times 1.12$	$\times 1.69$	$\times 1.06$	$\times 1.12$
Params	$\times 1.00$	$\times 1.00$	$\times 1.00$	$\times 1.00$
VRAM	$\times 1.21$	$\times 1.39$	$\times 1.00$	$\times 1.00$
Inference	$\times 0.50$	$\times 0.50$	$\times 0.50$	$\times 0.50$

Reference choice and training stability. In Eq. (13), we utilize the unguided reference from the $\alpha = 1$ slice with stop-gradient and PatchMix regularization. This stabilizes the SCALE training by keeping target dependencies acyclic and reducing the effects of reference displacement. We analyze these in the diffusion setting. *(i) Unguided reference.* Stop-gradient applies only to the reference branch; the same network is trained through the prediction branch with α sampled uniformly from $[\alpha_{\min}, 1]$. Assuming the predictor is smooth in α near 1, training at nearby values also constrains its $\alpha = 1$ slice, whose regression target is fixed by the data. The resulting chain of target definitions is acyclic within each update (data $\rightarrow \alpha = 1 \rightarrow \alpha < 1$). As the $\alpha = 1$ slice converges, the SCG target stabilizes, and the $\alpha < 1$ slices can converge toward it. *(ii) PatchMix.* Interpolation brings the reference closer to the real input on average. For finite-capacity models, PatchMix reduces the contributions from reference displacement to the bounds on SCG target variation and prediction error by a factor of $\sqrt{\lambda}$. These bounds support using mixed samples as more reliable guidance anchors when self-generated samples are degraded. Detailed proofs are provided in Appendix C.

SCALE for conditional generation. The basic SCALE objective sharpens sample distribution but does not enforce condition alignment, which becomes critical in large-scale visual generation. We therefore integrate CFG into SCALE training, while ensuring the inference fully guidance-free. Formally, we apply regularized SCG (Eq. (12)) on top of the CFG prediction f_θ^{cfg}, which means it directly refines CFG-generated outputs using the self-contrastive signal:

$$f_\theta^{\text{scg-cfg}}(\boldsymbol{x}_t \mid c) = f_\theta^{\text{cfg}}(\boldsymbol{x}_t \mid c) + s_{\text{scg}}(f_\theta^{\text{cfg}}(\boldsymbol{x}_t \mid c) - f_\theta^{\text{cfg}}(\tilde{\boldsymbol{x}}_{t,\lambda} \mid c)). \tag{14}$$

Following the derivation in Section 4.2, we reformulate this unified target into a learnable form:

$$f_\theta^{\text{cfg}}(\boldsymbol{x}_t | c, \alpha) = \alpha f_\theta^{\text{scg-cfg}}(\boldsymbol{x}_t | c, \alpha) + (1-\alpha)\ \text{sg}\left[f_\theta^{\text{cfg}}(\tilde{\boldsymbol{x}}_{t,\lambda} \mid c, 1)\right]. \tag{15}$$

Since f_θ^{cfg} on the left-hand side follows p_t^{cfg}, it is not directly learnable. We therefore integrate it into the guidance-free training framework for CFG (Eq. (4)), yielding a fully tractable training target:

$$\begin{aligned} &f_\theta^c(\boldsymbol{x}_t | c, \alpha, \beta) = \\ &\beta \left\{ \alpha f_\theta^{\text{scg-cfg}}(\boldsymbol{x}_t \mid c, \alpha, \beta) + (1-\alpha) \text{sg}\left[f_\theta^{\text{cfg}}(\tilde{\boldsymbol{x}}_{t,\lambda} \mid c, 1, \beta)\right] \right\} + (1-\beta)\ \text{sg}[f_\theta^u(\boldsymbol{x}_t | \varnothing, 1, 1)], \end{aligned} \tag{16}$$

where β is a hyperparameter controlling the CFG scale ($\beta = 1/(1 + s_{\text{cfg}})$). This formulation seamlessly integrates SCG with CFG, allowing the target trained model $f_\theta^{\text{scg-cfg}}$ to be directly optimized by replacing the model prediction in Eqs. 1 and 2 with $f_\theta^c(\boldsymbol{x}_t | c, \alpha, \beta)$. Setting $\beta = 1$ recovers basic SCALE, while setting $\alpha = 1$ reduces to the GFT [7]. Consequently, the trained model generates samples that incorporate both SCG and CFG in a single forward pass. In practice, we sample $\tilde{\boldsymbol{x}}_t$ from the model's unconditional prediction during training. This adds no extra cost and yields a stable reference, free from both CFG and SCG amplification. Algorithm 1 outlines the SCALE training pipeline for diffusion models. The SCALE training pipeline for AR models is in Appendix D.

4.4 Comparison with Other Methods

We compare SCALE against related methods; see Appendix F for more related works.

Computational overhead. As summarized in Table 1, the training-time overhead is minimal and VRAM is unchanged relative to CFG training as the extra forward passes introduced by SCALE are computed without gradients. Since SCALE introduces minimal additional parameters for embedding α and β, the parameter ratio is unchanged. We defer the detailed setup and analysis to Appendix G.1.

Versus CFG-mimicking methods. Prior guidance-free methods train the model to reproduce the CFG distribution p_t^{cfg}, so their quality is bounded by CFG itself. SCALE instead amplifies a model-based density ratio, $p_t(\boldsymbol{x}_t)/p_t(\tilde{\boldsymbol{x}}_t)$ for basic SCALE or $p_t^{\text{cfg}}(\boldsymbol{x}_t|c)/p_t^{\text{cfg}}(\tilde{\boldsymbol{x}}_t|c)$ for CFG-augmented SCALE, providing a stronger alignment with the data distribution and superior sample quality.

Versus RL post-training. RL post-training methods such as DPO [46] and GRPO [52] also use a contrastive signal but pursue a different goal. They align the model to *human preferences* via preference data or reward models and typically still rely on CFG in the visual domain [60, 38], whereas SCALE aligns to the *ground-truth data distribution* from the model's own predictions and is fully guidance-free. The two are therefore complementary, and Section 5.2 shows that DPO [60] on top of SCALE further improves quality while remaining guidance-free.

5 Results

5.1 Experimental Setups

Models, datasets, and metrics. We evaluate SCALE on class-conditional (C2I) ImageNet [9] at 256×256 by fine-tuning DiT-XL/2 [44] and LlamaGen-L [54], and additionally train DiT-B/2 for the analysis. We report FID-50k, IS, Precision, and Recall following Dhariwal and Nichol [11]. For text-to-image generation (T2I) at 512×512, we fine-tune Stable Diffusion v1.5 (SD1.5) [49] on LAION-Aesthetics 5+ [51] and report FID-30k, CLIP Score [45], PickScore [29], HPSv2.1 [62], Aesthetic Score [51], and Image Reward [63], complemented by a human preference study. Computational cost is measured by the number of function evaluations (NFE). See Appendix E for more details.

Baseline methods. We compare SCALE against three groups of baselines, which enhance the base model's performance in different ways. *Guidance-free methods*: We include guidance distillation [41] for diffusion models, CCA [6] for AR models, and GFT [7] for both. *Guided sampling methods*: We compare against CFG [18], SAG [22], and discriminator guidance (G++) [27]. For text-to-image, we further compare against RL post-training with DiffusionDPO [60]. We additionally report with representative diffusion models [11, 49, 20], and AR models [13, 66, 32, 57].

Implementation details. We train all models with the SCALE objective in Eq. (16), introducing only a few additional hyperparameters. For fine-tuning, we limit training to less than 5% of the pre-training budget and inherit the base model's hyperparameters. We fix the PatchMix ratio at $\lambda = 0.2$ across all experiments. The patch size is matched to each model's self-attention granularity: 2 for DiT, 1 for LlamaGen, and 2 for SD1.5 (the finest self-attention resolution in its UNet). For diffusion sampling, we use 50-step DPM-Solver++ [40]. See Appendix E for the full setup.

5.2 Enhancing Guidance-free Generation with SCALE

In this section, we demonstrate that incorporating self-contrastive signal effectively enhances guidance-free generation by comparing SCALE with various generative models without guidance.

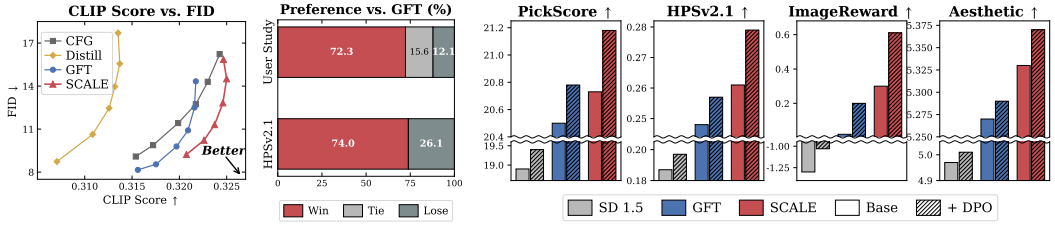
Fine-tuning from pretrained models (C2I). We compare SCALE against baseline guidance-free methods with various generative models on class-conditional generation on ImageNet 256×256. Among these, guidance distillation [41] applies only to diffusion models, and CCA [6] only to AR models. In contrast, SCALE and GFT apply to both. As shown in Table 2, SCALE outperforms all baselines on FID and IS across both diffusion (DiT-XL/2 [44]) and AR (LlamaGen-L [54]) models.

Table 2: Comparisons of guidance-free class-conditional generation on ImageNet 256×256. **Bold** for the best and underline for the second-best metrics for each model type.

Model Types	Methods	FID-50k ↓	IS ↑	Precision ↑	Recall ↑
Diffusion Models	ADM [11]	10.94	101.1	0.69	0.63
	LDM-4 [49]	10.56	103.4	0.72	0.62
	CDM [20]	4.88	158.7	–	–
	DiT-XL/2 [44]	9.62	121.5	0.67	**0.67**
	+Guid. Distill [41]	2.11	–	–	–
	+GFT [7]	1.99	266.6	**0.81**	0.60
	+SCALE (Ours)	**1.87**	**268.0**	**0.81**	0.60
Autoregressive Models	VQGAN [13]	18.65	80.4	0.78	0.26
	ViT-VQGAN [66]	4.17	175.1	–	–
	RQ-Transformer [32]	7.55	134.0	–	–
	VAR-d30 [57]	5.26	–	–	–
	LlamaGen-L [54]	19.0	61.7	0.60	**0.66**
	+CCA [6]	3.43	**288.2**	0.81	0.52
	+GFT [7]	2.88	238.4	–	–
	+SCALE (Ours)	**2.58**	253.0	**0.82**	0.55

Table 3: Guidance-free generation results on text-to-image generation using SD1.5.

Methods	PickScore ↑	CLIP ↑	HPSv2.1 ↑	FID-30k ↓	Aesthetic ↑	Img. Reward ↑
SD 1.5 [49]	18.90	0.252	0.187	22.47	4.97	-1.30
+Guid. Distill [41]	20.56	0.314	0.252	15.12	**5.33**	0.06
+GFT [7]	20.50	0.321	0.248	**12.50**	5.27	0.02
+SCALE ($\alpha = 0.8$)	20.58	**0.324**	0.253	12.83	5.30	0.18
+SCALE ($\alpha = 0.25$)	**20.73**	**0.324**	**0.261**	16.52	**5.33**	**0.30**



(a) Pareto curves. (b) Win rate vs. GFT. (c) Compatibility with DPO [60].

Figure 4: Performance comparison of SCALE and other guidance-free baselines using SD 1.5 [49].

Fine-tuning from pretrained models (T2I). Table 3 compares guidance-free generation results on SD1.5. The effective CFG scale for guidance-free methods is set to the standard CFG scale of 7.5. SCALE achieves superior performance while maintaining the competitive FID scores, and further improves the quality by lowering the α, confirming that the self-contrastive signal generalizes to large-scale zero-shot generation. Lowering α compensates FID but further improves the quality. The Pareto curves in Fig. 4a, obtained by sweeping the effective CFG scale from 4.0 to 12.5 at $\alpha = 0.8$, show that SCALE achieves a superior CLIPScore-FID trade-off. The head-to-head comparison in Fig. 4b further shows SCALE winning over GFT on both HPSv2.1 and PickScore by clear margins.

SCALE vs. DPO. As discussed in Section 4.4, SCALE and DPO operate along different perspectives. Experimental results in Fig. 4c demonstrate three key findings: *(1)* In the guidance-free setting, DPO barely improves over the SD 1.5 baseline, showing that DPO itself relies heavily on CFG for high-quality generation. *(2)* SCALE alone matches GFT+DPO, suggesting that self-contrastive signals are an effective alignment objective without preference data. *(3)* SCALE+DPO yields further gains across all metrics, indicating that the two signals are complementary rather than redundant.

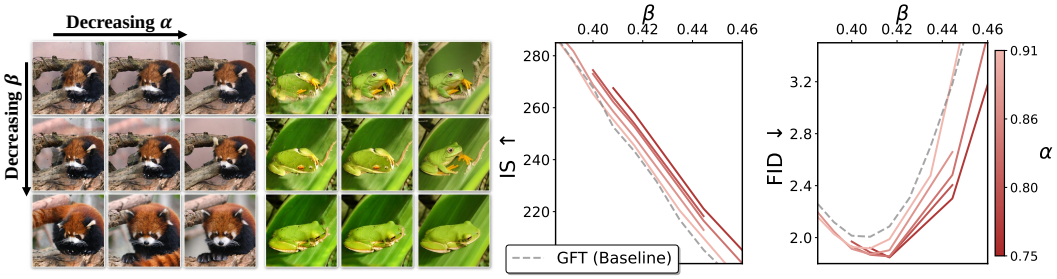
The qualitative comparisons in Fig. 5 further support these findings. Without sampling-time guidance, SCALE produces samples with finer details and higher visual quality than base SD 1.5 and GFT.



Figure 5: Guidance-free generation results of base SD 1.5 [49], GFT, and SCALE, with and without DPO post-training [60]. Prompts are sampled from the Pick-a-Pic test set [29] and PartiPrompts [67].

Table 4: Comparison with guided sampling methods on the ImageNet 256×256. † Re-evaluated using 50-step DPM-Solver++ [40] for a fair comparison. See Appendix G.3 for more details.

Models	Methods	Guidance	NFEs ↓	FID ↓	IS ↑	Precision ↑	Recall ↑
DiT-XL/2 [44]	CFG† [18]	Required	50×2	2.17	243.7	0.80	0.60
	SAG† [22]	Required	50×3	2.17	241.2	0.80	0.59
	G++† [27]	Required	50×3	2.14	240.0	0.76	0.63
	SCALE (Ours)	Free	50	**1.87**	**268.0**	**0.81**	0.60
LlamaGen-L [54]	CFG [18]	Required	256×2	3.07	**256.0**	**0.83**	0.52
	SCALE (Ours)	Free	256	**2.58**	253.0	0.82	0.55



(a) Generated samples.

(b) FID and IS curves.

Figure 6: Effect of varying α and β of SCALE using DiT-XL/2 [44] at sampling time.

Applying DPO to SCALE further enhances these qualities, confirming the complementary benefits of self-contrastive signal and preference-based post-training. More qualitative comparisons of guidance-free generation with DPO post-trained models are in Figs. 11 and 12 of the Appendix.

Comparison with guided sampling. We compare SCALE against CFG [18], information-based guidance for diffusion models (SAG [22]), and guidance that utilizes an external discriminator (G++) [27]. As shown in Table 4, while these methods introduce additional sampling overhead, SCALE matches them without any guidance at sampling, and even surpasses CFG in FID.

5.3 Analysis on SCALE

Effect of varying α and β. As noted in Section 4.3, SCALE provides inference-time flexibility for controlling the sample quality by varying α and β. Fig. 6a visualizes their effects. Decreasing β improves fidelity by amplifying the conditional signal, but stronger CFG pushes samples toward canonical templates and can change their semantic structure. Decreasing α instead targets fidelity

Table 5: Guidance-free generation with trained from scratch.

Model	Methods	FID ↓	IS ↑
DiT-B/2 [44]	Baseline	44.8	30.7
	GFT [7]	9.04	166.6
	SCALE	**8.75**	**169.9**

Table 6: Ablation on interpolation type for SCALE.

Model	Interp.	FID ↓	IS ↑
DiT-B/2 [44]	Baseline	44.8	30.7
	Linear	8.67	171.8
	Patch	**8.60**	**172.4**

Table 7: Ablation on interpolation ratio λ.

Model	λ	FID ↓	IS ↑
LlamaGen-L [54]	1.0	2.59	245.9
	0.5	**2.58**	245.2
	0.2	**2.58**	**253.0**

directly while preserving semantic structure, though it may fall short when samples lie far from the data manifold (first row of Fig. 6a). More detailed visual analysis is provided in Appendix H.

Combination of SCG and CFG. Aforementioned analysis suggests that SCG and CFG are complementary to each other. Therefore, we combine SCG on top of the CFG prediction, which directly rectifies the CFG-generated samples toward the data manifold. This compensates for the limitation of each individual guidance: moderate CFG enhances prompt alignment and rectifies the underlying semantic structure, while SCG improves the sample quality on top of the CFG prediction. Utilizing CFG alone with a higher guidance scale may push samples away from the data manifold or incur reduced diversity as discussed in Section 3. Notably, SCALE reduces to GFT [7] at $\alpha = 1$, which only approximates the CFG distribution. Fig. 6b shows that lowering α consistently improves FID and IS over this baseline. SCALE thus serves as a plug-and-play framework for improving guidance-free generation that maintains baseline performance rather than just adding complexity.

Training SCALE from scratch. Unlike most guidance-free methods that require a CFG-trained model [41, 6], SCALE can be trained from scratch as it does not require a pretrained teacher model. Due to resource constraints, we verify this on DiT-B/2 [44], where SCALE significantly improves guidance-free generation (Table 5). Scaling to larger models is left as future work.

Interpolation type. As discussed in Section 4.3, we employ patch-wise mix-up to maintain compatibility with diverse generative architectures. Table 6 also confirms that patch-wise mix-up provides robust improvements over linear interpolation by preserving local semantics of dense visual data.

Interpolation ratio. Table 7 reports the impact of the interpolation ratio λ. For each λ, we report the best FID with its corresponding IS. Lower λ achieves better FID and IS, indicating that a small but non-zero λ is sufficient to provide stable and effective guidance.

6 Conclusion

We introduce *SCALE*, a guidance-free training framework built on *self-contrastive guidance* (SCG), a signal derived from the model's own predictions on real and self-generated samples. Grounded in the information-based guidance theorem, SCG directly pushes the generation process toward the true data manifold. Extensive experiments confirm that SCALE outperforms existing guidance-free baselines on both diffusion and AR models, and show complementary performance gain when combined with human preference-based fine-tuning.

Limitations. SCALE's nested guidance formulation introduces additional training overhead. Extensions to broader visual alignment tasks, post-training beyond DPO, and larger, more recent generative architectures remain open directions. See Appendix I for further discussion.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-00439020, Developing Sustainable, Real-Time Generative AI for Multimodal Interaction, SW Starlab).

References

[1] D. Ahn, H. Cho, J. Min, W. Jang, J. Kim, S. Kim, H. H. Park, K. H. Jin, and S. Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *European Conference on Computer Vision*, pages 1–17. Springer, 2024. 21

[2] A. Bradley and P. Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024. 3

[3] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 20

[4] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 3, 20

[5] C. Chen, J. Zhu, X. Feng, N. Huang, C. Zhu, M. Wu, F. Mao, J. Wu, X. Chu, and X. Li. Stochastic self-guidance for training-free enhancement of diffusion models. In *International Conference on Learning Representations*, volume 2026, pages 134462–134490, 2026. 21

[6] H. Chen, H. Su, P. Sun, and J. Zhu. Toward guidance-free ar visual generation via condition contrastive alignment. *arXiv preprint arXiv:2410.09347*, 2024. 1, 3, 7, 8, 10, 21, 22

[7] H. Chen, K. Jiang, K. Zheng, J. Chen, H. Su, and J. Zhu. Visual generation without guidance. *arXiv preprint arXiv:2501.15420*, 2025. 1, 3, 4, 6, 7, 8, 10, 18, 21, 22

[8] H. Chung, J. Kim, G. Y. Park, H. Nam, and J. C. Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *arXiv preprint arXiv:2406.08070*, 2024. 1, 3

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7, 34

[10] Y. Deng and P. Mineiro. Flow-dpo: Improving llm mathematical reasoning through online multi-agent learning. *arXiv preprint arXiv:2410.22304*, 2024. 22

[11] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 3, 7, 8

[12] B. Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496): 1602–1614, 2011. 15

[13] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 7, 8, 20, 21

[14] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 20, 27

[15] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 15

[16] J. Han, J. Liu, Y. Jiang, B. Yan, Y. Zhang, Z. Yuan, B. Peng, and X. Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15733–15744, 2025. 20

[17] X. He, S. Fu, Y. Zhao, W. Li, J. Yang, D. Yin, F. Rao, and B. Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025. 22

[18] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 3, 7, 9, 21, 23

[19] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 20

[20] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 7, 8

[21] S. Hong. Smoothed energy guidance: Guiding diffusion models with reduced energy curvature of attention. *Advances in Neural Information Processing Systems*, 37:66743–66772, 2024. 21

[22] S. Hong, G. Lee, W. Jang, and S. Kim. Improving sample quality of diffusion models using self-attention guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7462–7471, 2023. 1, 4, 7, 9, 21, 23

[23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 19

[24] J. Hur, D. Lee, G. Han, J. Choi, Y. Jeon, and J. Kim. Unlocking the capabilities of masked generative models for image synthesis via self-guidance. *Advances in Neural Information Processing Systems*, 37: 130977–130999, 2024. 1, 4, 15, 21

[25] J. Hyung, K. Kim, S. Hong, M.-J. Kim, and J. Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11006–11015, 2025. 5

[26] T. Karras, M. Aittala, T. Kynkäänniemi, J. Lehtinen, T. Aila, and S. Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024. 1, 3, 5, 21

[27] D. Kim, Y. Kim, S. J. Kwon, W. Kang, and I.-C. Moon. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022. 7, 9, 23

[28] K. Kim and J. C. Ye. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874, 2021. 15

[29] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36: 36652–36663, 2023. 7, 9, 19, 20, 26

[30] T. Kynkäänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024. 1, 3

[31] B. F. Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024. 20, 27

[32] D. Lee, C. Kim, S. Kim, M. Cho, and W. S. HAN. Draft-and-revise: Effective image generation with contextual rq-transformer. *Advances in Neural Information Processing Systems*, 35:30127–30138, 2022. 7, 8

[33] T. Li, H. Chang, S. Mishra, H. Zhang, D. Katabi, and D. Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2152, 2023. 20

[34] T. Li, Y. Tian, H. Li, M. Deng, and K. He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 20

[35] Z. Liang, Y. Yuan, S. Gu, B. Chen, T. Hang, J. Li, and L. Zheng. Step-aware preference optimization: Aligning preference with denoising performance at each step. *arXiv preprint arXiv:2406.04314*, 2(5):7, 2024. 22, 27

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 19

[37] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 20

[38] J. Liu, G. Liu, J. Liang, Y. Li, J. Liu, X. Wang, P. Wan, D. Zhang, and W. Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 7, 22, 27

[39] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 20

[40] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 7, 9, 19

[41] C. Meng, R. Rombach, R. Gao, D. Kingma, S. Ermon, J. Ho, and T. Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14297–14306, 2023. 1, 3, 7, 8, 10, 21, 22

[42] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 20

[43] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 20

[44] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 7, 8, 9, 10, 19, 20, 22, 23, 24, 25, 34

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7, 19

[46] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741, 2023. 7, 20, 22

[47] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 20

[48] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 1, 2, 21

[49] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 7, 8, 9, 18, 19, 20, 24, 26, 27, 34

[50] S. Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning, 2023. 19

[51] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 7, 19

[52] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 7, 22

[53] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2, 20

[54] P. Sun, Y. Jiang, S. Chen, S. Zhang, B. Peng, P. Luo, and Z. Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1, 2, 7, 8, 9, 10, 19, 20, 23, 24, 25

[55] H. Tang, Y. Wu, S. Yang, E. Xie, J. Chen, J. Chen, Z. Zhang, H. Cai, Y. Lu, and S. Han. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024. 20

[56] Z. Tang, J. Bao, D. Chen, and B. Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025. 21

[57] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 7, 8, 20, 22

[58] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 2

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[60] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024. 7, 8, 9, 20, 22, 26, 27

[61] M. Weber, L. Yu, Q. Yu, X. Deng, X. Shen, D. Cremers, and L.-C. Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv preprint arXiv:2409.16211*, 2024. 20

[62] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 7, 19

[63] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 15903–15935, 2023. 7, 19

[64] Z. Xue, J. Wu, Y. Gao, F. Kong, L. Zhu, M. Chen, Z. Liu, W. Liu, Q. Guo, W. Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025. 22, 27

[65] K. Yang, J. Tao, J. Lyu, C. Ge, J. Chen, W. Shen, X. Zhu, and X. Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024. 22, 27

[66] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 7, 8

[67] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 9, 26

[68] Q. Yu, J. He, X. Deng, X. Shen, and L.-C. Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18431–18441, 2025. 20

[69] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 18

[70] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 23

[71] H. Zhu, T. Xiao, and V. G. Honavar. Dspo: Direct score preference optimization for diffusion model alignment. International Conference on Learning Representations (ICLR 2025), 2025. 22, 27

A Derivation of Generalized Information-Based Guidance for Discrete Generative Models

In this section, we derive the optimization-perspective formulation of information-based guidance for discrete generative models in Eq. (6), following the framework of Tang et al. [15] and Hur et al. [24].

Let $\bar{\boldsymbol{x}}_t$ denote a degraded sample that lacks some salient information h_t present in the true sample $\boldsymbol{x}_t$, such that the data distribution decomposes as $p(\boldsymbol{x}_t) = p(\bar{\boldsymbol{x}}_t, h_t)$. Generalized information-based guidance recovers $\boldsymbol{x}_t$ from $\bar{\boldsymbol{x}}_t$ by jointly maximizing the joint likelihood and amplifying the correlation between $\bar{\boldsymbol{x}}_t$ and h_t, captured by $p(h_t \mid \bar{\boldsymbol{x}}_t)$:

$$\arg\max_{\bar{\boldsymbol{x}}_t} \big[\log p(\bar{\boldsymbol{x}}_t, h_t) + s \, \log p(h_t \mid \bar{\boldsymbol{x}}_t)\big], \tag{17}$$

where $s \geq 0$ is the guidance scale that controls the strength of information amplification.

Applying the chain rule of probability $p(h_t \mid \bar{\boldsymbol{x}}_t) = p(\bar{\boldsymbol{x}}_t, h_t)/p(\bar{\boldsymbol{x}}_t)$, the conditional log-density expands as $\log p(h_t \mid \bar{\boldsymbol{x}}_t) = \log p(\bar{\boldsymbol{x}}_t, h_t) - \log p(\bar{\boldsymbol{x}}_t)$. Substituting into Eq. (17) yields

$$\arg\max_{\bar{\boldsymbol{x}}_t} \big[\log p(\bar{\boldsymbol{x}}_t, h_t) + s\big(\log p(\bar{\boldsymbol{x}}_t, h_t) - \log p(\bar{\boldsymbol{x}}_t)\big)\big], \tag{18}$$

which recovers Eq. (6) in the main text. This formulation explicitly amplifies the pointwise mutual information $\log p(\bar{\boldsymbol{x}}_t, h_t) - \log p(\bar{\boldsymbol{x}}_t) = \log p(h_t \mid \bar{\boldsymbol{x}}_t)$ by a factor of s on top of the base log-density $\log p(\bar{\boldsymbol{x}}_t | h_t) = \log p(\boldsymbol{x}_t)$.

Connection to Hur et al. [24]. Hur et al. [24] formulate the same objective with the conditional $p(\bar{\boldsymbol{x}}_t \mid h_t)$ in place of our joint $p(\bar{\boldsymbol{x}}_t, h_t)$. Since $\log p(\bar{\boldsymbol{x}}_t \mid h_t) = \log p(\bar{\boldsymbol{x}}_t, h_t) - \log p(h_t)$ and $\log p(h_t)$ does not depend on $\bar{\boldsymbol{x}}_t$, the two formulations differ only by an additive constant inside the $\arg\max$ when h_t is held fixed, and therefore lead to identical optima. We adopt the joint form because it avoids treating h_t as a freely defined random variable, which the conditional $p(\bar{\boldsymbol{x}}_t \mid h_t)$ requires. Setting $p(\bar{\boldsymbol{x}}_t, h_t) = p(\boldsymbol{x}_t)$ instead lets us work directly with the data distribution at hand without specifying a separate distribution over h_t.

B Proof of Theorem 4.1 and Analogous Result for AR Models

In this section, we derive the conditional mean SCG prediction at the diffusion regression optimum stated in Theorem 4.1, followed by the analogous result for AR models.

Proof for diffusion models. Under the score matching objective, the optimal noise predictor $\boldsymbol{\epsilon}_{\theta^*}$ converges to the true scaled score from the noisy sample $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$:

$$\boldsymbol{\epsilon}_{\theta^*}(\boldsymbol{x}_t) = -\sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t), \tag{19}$$

where t is the diffusion timestep. Let $\tilde{\boldsymbol{x}}_t$ be the self-generated sample from the model's current prediction. In the context of diffusion models as a noise predictor, $\tilde{\boldsymbol{x}}_t$ can be obtained simply. Specifically, we define a mapping g_θ that first estimates the clean data using trained noise predictor $\boldsymbol{\epsilon}_\theta$ with Tweedie's formula [12, 28] and subsequently re-noises it back to the same noise level t:

$$\tilde{\boldsymbol{x}}_t = g_\theta(\boldsymbol{x}_t, \boldsymbol{\epsilon}) = \alpha_t \tilde{\boldsymbol{x}}_0 + \sigma_t \boldsymbol{\epsilon}, \text{ where} \tag{20}$$

$$\tilde{\boldsymbol{x}}_0 = (\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t))/\alpha_t, \tag{21}$$

By the SCALE training objective, the trained model output is parameterized as the interpolation between the SCG branch and the self-generated branch:

$$\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t \mid \alpha) = \alpha \, \boldsymbol{\epsilon}_\theta^{\text{scg}}(\boldsymbol{x}_t \mid \alpha) + (1-\alpha) \, \text{sg}\big[\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_t \mid 1)\big], \tag{22}$$

By construction, the $\alpha = 1$ branch is the unconditional predictor,i.e., $\boldsymbol{\epsilon}_\theta(\cdot \mid 1) \equiv \boldsymbol{\epsilon}_\theta(\cdot)$. Hence, we may rewrite Eq. (22) as

$$\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t \mid \alpha) = \alpha \, \boldsymbol{\epsilon}_\theta^{\text{scg}}(\boldsymbol{x}_t \mid \alpha) + (1-\alpha) \, \text{sg}\big[\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_t)\big]. \tag{23}$$

The SCALE training directly optimizes $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t \mid \alpha)$ to minimize the score matching loss

$$\min_\theta \mathbb{E}_{(\boldsymbol{x}_0, c) \sim p_{\text{data}}, \boldsymbol{\epsilon}, t} \left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t \mid \alpha)||^2 \right]. \tag{24}$$

The reference branch is held fixed by stop-gradient. At an ideal regression optimum ($\theta = \theta^*$), the SCG branch therefore predicts the conditional mean of $(\boldsymbol{\epsilon} - (1-\alpha)\boldsymbol{\epsilon}_{\theta^*}(\tilde{\boldsymbol{x}}_t))/\alpha$ given $\boldsymbol{x}_t$. For fixed t and $\alpha \in (0, 1]$, using $\mathbb{E}[\boldsymbol{\epsilon} \mid \boldsymbol{x}_t] = -\sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ in Eq. (23) gives

$$\alpha\, \boldsymbol{\epsilon}^{\text{scg}}_{\theta^*}(\boldsymbol{x}_t \mid \alpha) + (1-\alpha)\, \mathbb{E}\big[\boldsymbol{\epsilon}_{\theta^*}(\tilde{\boldsymbol{x}}_t) \mid \boldsymbol{x}_t\big] = -\sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t). \tag{25}$$

Here $\tilde{\boldsymbol{x}}_t = g_{\theta^*}(\boldsymbol{x}_t, \boldsymbol{\epsilon})$, and the expectation is over $\boldsymbol{\epsilon} \mid \boldsymbol{x}_t$. By the assumed score accuracy at the reference inputs, $\boldsymbol{\epsilon}_{\theta^*}(\tilde{\boldsymbol{x}}_t) = -\sigma_t \nabla_{\tilde{\boldsymbol{x}}_t} \log p_t(\tilde{\boldsymbol{x}}_t)$. Substituting this identity inside the expectation and solving for the SCG branch $f^{\text{scg}}_{\theta^*}(\boldsymbol{x}_t \mid \alpha)$ gives

$$f^{\text{scg}}_{\theta^*}(\boldsymbol{x}_t \mid \alpha) = \frac{1}{\alpha}\Big[-\sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) - (1-\alpha)\, \mathbb{E}\big[\boldsymbol{\epsilon}_{\theta^*}(\tilde{\boldsymbol{x}}_t) \mid \boldsymbol{x}_t\big]\Big] \tag{26}$$

$$= -\sigma_t \left[\frac{1}{\alpha} \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) - \left(\frac{1}{\alpha} - 1\right) \mathbb{E}\big[\nabla_{\tilde{\boldsymbol{x}}_t} \log p_t(\tilde{\boldsymbol{x}}_t) \mid \boldsymbol{x}_t\big]\right]. \tag{27}$$

This form coincides with Eq. (10), recovering the conditional mean SCG prediction. □

Proof for autoregressive models. We now derive the autoregressive (AR) counterpart, where the optimization directly works in the probability domain. For AR models, f_θ is a next-token predictor that outputs a categorical distribution over the token vocabulary at each generation step. To match the diffusion notation, we write $\boldsymbol{x}_t$ for the next-token target at position t given $\boldsymbol{x}_{<t}$.

Concretely, the next-token logit is $\ell_\theta(\boldsymbol{x}_t \mid \alpha)$ over the token vocabulary, and the predicted probability is $f_\theta(\boldsymbol{x}_t \mid \alpha) = \text{softmax}\big(\ell_\theta(\boldsymbol{x}_t \mid \alpha)\big)$. Under the cross-entropy objective in Eq. (2), the optimal next-token predictor converges to the true next-token conditional:

$$f_{\theta^*}(\boldsymbol{x}_t \mid \alpha) = p_t(\boldsymbol{x}_t), \qquad \text{i.e., } \ell_{\theta^*}(\boldsymbol{x}_t \mid \alpha) = \log p_t(\boldsymbol{x}_t) + Z(\alpha), \tag{28}$$

for all α in the training distribution, where $p_t(\boldsymbol{x}_t)$ refers to the true next-token distribution at position t given $\boldsymbol{x}_{<t}$ and $Z(\alpha)$ is a vocabulary-wide additive constant absorbed by the softmax normalization.

By the SCALE training objective, the trained logit is parameterized as the interpolation between the SCG branch and the self-generated branch:

$$\ell_\theta(\boldsymbol{x}_t \mid \alpha) = \alpha\, \ell^{\text{scg}}_\theta(\boldsymbol{x}_t \mid \alpha) + (1-\alpha)\, \text{sg}\big[\ell_\theta(\tilde{\boldsymbol{x}}_t \mid 1)\big], \tag{29}$$

where the self-generated token $\tilde{\boldsymbol{x}}_t$ is drawn from the unconditional next-token distribution $f_\theta(\cdot \mid 1)$ via standard token-wise sampling. At $\alpha = 1$, the SCG branch reduces to the unconditional next-token prediction, so $\ell_{\theta^*}(\tilde{\boldsymbol{x}}_t \mid 1) = \log p_t(\tilde{\boldsymbol{x}}_t) + Z(1)$ at convergence.

Substituting the convergence condition into the SCALE parameterization and solving for the SCG branch:

$$\alpha\, \ell^{\text{scg}}_{\theta^*}(\boldsymbol{x}_t \mid \alpha) + (1-\alpha)\big[\log p_t(\tilde{\boldsymbol{x}}_t) + Z(1)\big] = \log p_t(\boldsymbol{x}_t) + Z(\alpha), \tag{30}$$

$$\ell^{\text{scg}}_{\theta^*}(\boldsymbol{x}_t \mid \alpha) = \frac{1}{\alpha} \log p_t(\boldsymbol{x}_t) - \left(\frac{1}{\alpha} - 1\right) \log p_t(\tilde{\boldsymbol{x}}_t) + Z', \tag{31}$$

where Z' denotes the sum of additive constants. With $\alpha = 1/(1 + s_{\text{scg}})$, applying softmax to Eq. (31) yields

$$f^{\text{scg}}_{\theta^*}(\boldsymbol{x}_t \mid \alpha) \;\propto\; p_t(\boldsymbol{x}_t) \left[\frac{p_t(\boldsymbol{x}_t)}{p_t(\tilde{\boldsymbol{x}}_t)}\right]^{s_{\text{scg}}}, \tag{32}$$

which exactly matches the SCG-induced next-token distribution obtained by substituting $\bar{\boldsymbol{x}}_t = \tilde{\boldsymbol{x}}_t$ and $h_t = \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t$ into the discrete information-based guidance objective in Eq. (6). The SCALE-trained AR model therefore recovers the SCG distribution at convergence in the probability domain, mirroring the score-domain result for diffusion. □

C Analysis of the SCALE Training

The use of a self-generated reference in Eq. (9) makes the SCALE training target dependent on the model's current predictions. We examine how this target evolves under stop-gradient updates and how prediction errors propagate through the PatchMix reference. The analysis is presented for diffusion models, where f_θ denotes the noise predictor and ϵ the ground-truth noise. We fix t and $\alpha \in [\alpha_{\min}, 1]$ with $\alpha_{\min} > 0$; conditional expectations average over $\epsilon \mid x_t$ and an independent PatchMix mask.

C.1 Training with Stop-Gradient and PatchMix

Across training iterations. Within each update, the SCALE training target uses a frozen reference. Under stop-gradient in Eq. (9), each training step follows the target-network iteration

$$\theta_{k+1} \leftarrow \theta_k - \mu\nabla_\theta \|\epsilon - \alpha \underbrace{f_\theta^{\text{scg}}(x_t \mid \alpha)}_{\text{optimization target}} - (1-\alpha) \underbrace{f_{\theta_k}(\tilde{x}_{t,\lambda} \mid 1)}_{\substack{\text{frozen reference} \\ \text{(stop-gradient)}}} \|^2. \tag{33}$$

Each iteration applies a gradient step to a regression loss with a frozen reference, evaluated at $\theta = \theta_k$. Both $\tilde{x}_{t,\lambda}$ and the prediction at that input are obtained from the $\alpha{=}1$ slice. At $\alpha = 1$, the objective reduces to standard noise regression toward the fixed, model-independent target $f_{\theta^*}(x_t \mid 1) = -\sigma_t \nabla \log p_t(x_t)$. Within each update, the resulting reference dependency chain is acyclic (data $\rightarrow \alpha{=}1 \rightarrow \alpha{<}1$); the target depends on the model but is anchored to the data distribution through the base slice's fixed noise-regression target. Since the slices share parameters, guided-slice updates can still change the base slice across iterations.

Each gradient step. The reference evolves across successive model snapshots ($\theta_k \rightarrow \theta_{k+1}$), inducing a moving target for the $\alpha{<}1$ slices. The target variation is bounded by the base-slice change and vanishes as this change tends to zero. Rearranging the loss in Eq. (33) gives a sample target for the SCG branch. We denote its conditional mean given x_t by $T[\theta_k]$, where θ_k specifies the fixed model snapshot used to construct and evaluate the reference:

$$T[\theta_k](x_t, \alpha) = \frac{1}{\alpha}\big(-\sigma_t \nabla \log p_t(x_t)\big) - \big(\frac{1}{\alpha} - 1\big)\mathbb{E}\big[f_{\theta_k}(\tilde{x}_{t,\lambda} \mid 1) \mid x_t\big], \tag{34}$$

so a base-slice change $\Delta f := f_{\theta_{k+1}}(\cdot \mid 1) - f_{\theta_k}(\cdot \mid 1)$ enters only through the last term. With a uniform change bound $\|\Delta f\|$ and a common Lipschitz constant L, we bound its effect through two contributions. First, the prediction at the reference input $\tilde{x}_{t,\lambda}$ itself changes, contributing $\frac{1-\alpha}{\alpha}\|\Delta f\|$. Second, the base prediction also changes the generated sample $\tilde{x}_t = x_t + \sigma_t(\epsilon - f_{\theta_k}(x_t \mid 1))$, shifting the reference input $\tilde{x}_{t,\lambda}$. PatchMix reduces the displacement bound from $\sigma_t\|\Delta f\|$ to $\sqrt{\lambda}\,\sigma_t\|\Delta f\|$ in RMS. Feeding the shifted reference into the L-Lipschitz base predictor changes its output, giving a second contribution of at most $\frac{1-\alpha}{\alpha}\sqrt{\lambda}\,\sigma_t L\|\Delta f\|$. Together,

$$\|\Delta T\| \le \kappa\,\|\Delta f\|, \qquad \kappa = \frac{1-\alpha}{\alpha}(1 + \sqrt{\lambda}\,\sigma_t L). \tag{35}$$

If the base slice converges uniformly to $f_{\theta^*}(\cdot \mid 1)$, then $\|\Delta f\| \rightarrow 0$ and hence $\|\Delta T\| \rightarrow 0$. The bound also gives $T[\theta_k] \rightarrow T[\theta^*]$, so the moving target settles to a fixed limit. The $\alpha{<}1$ slices converge to it if their regression error relative to $T[\theta_k]$ also vanishes.

C.2 Optimum with PatchMix under Finite Capacity Bounds

At an ideal regression optimum ($\theta_k \rightarrow \theta^*$), the conditional mean of the mixed output matches the data score:

$$\alpha f_{\theta^*}^{\text{scg}}(x_t \mid \alpha) + (1-\alpha)\mathbb{E}\big[f_{\theta^*}(\tilde{x}_{t,\lambda} \mid 1) \mid x_t\big] = -\sigma_t \nabla \log p_t(x_t). \tag{36}$$

The regression argument in the proof of Theorem 4.1 does not depend on how the reference is constructed, so it also applies under PatchMix: substituting $f_{\theta^*}(\tilde{x}_{t,\lambda} \mid 1) = -\sigma_t \nabla \log p_t(\tilde{x}_{t,\lambda})$ assuming the base predictor matches the data score at the reference inputs and solving for the SCG branch gives

$$f_{\theta^*}^{\text{scg}}(x_t \mid \alpha) = -\sigma_t\big[\frac{1}{\alpha}\nabla \log p_t(x_t) - \big(\frac{1}{\alpha} - 1\big)\mathbb{E}\big[\nabla \log p_t(\tilde{x}_{t,\lambda}) \mid x_t\big]\big]. \tag{37}$$

Setting $\lambda{=}1$ recovers Eq. (10), while $\lambda{=}0$ disables guidance at this ideal optimum. For a fixed real/generated pair, Bernoulli PatchMix gives

$$\mathbb{E}_{M_\lambda}\|\tilde{x}_{t,\lambda} - x_t\|^2 = \lambda\|\tilde{x}_t - x_t\|^2, \tag{38}$$

where M_λ is the PatchMix mask in Eq. (11).

For finite capacity, let $\hat{\theta}$ be the trained model, $\delta_\alpha(x_t)$ the residual of its mixed output at conditioning α in Eq. (36), and $\delta(y) := f_{\hat{\theta}}(y \mid 1) - f_{\theta^*}(y \mid 1)$ the error of its $\alpha{=}1$ slice. Since the generated reference

depends on the model, write $\tilde{x}_{t,\lambda}$ and $\tilde{x}^*_{t,\lambda}$ for references constructed with $\hat{\theta}$ and θ^*, respectively, using the same ϵ and mask. Using the mixed output $f_\theta(x_t \mid \alpha)$ from Eq. (9), the deviation from Eq. (37) is

$$\begin{aligned}
&f^{\text{scg}}_{\hat{\theta}}(x_t \mid \alpha) - f^{\text{scg}}_{\theta^*}(x_t \mid \alpha) \\
&= \frac{1}{\alpha}\mathbb{E}\big[f_{\hat{\theta}}(x_t \mid \alpha) - f_{\theta^*}(x_t \mid \alpha) \mid x_t\big] - (\frac{1}{\alpha}-1)\mathbb{E}\big[f_{\hat{\theta}}(\tilde{x}_{t,\lambda} \mid 1) - f_{\theta^*}(\tilde{x}^*_{t,\lambda} \mid 1) \mid x_t\big] \\
&= \frac{1}{\alpha}\delta_\alpha(x_t) - (\frac{1}{\alpha}-1)\mathbb{E}\big[f_{\hat{\theta}}(\tilde{x}_{t,\lambda} \mid 1) - f_{\theta^*}(\tilde{x}_{t,\lambda} \mid 1) + f_{\theta^*}(\tilde{x}_{t,\lambda} \mid 1) - f_{\theta^*}(\tilde{x}^*_{t,\lambda} \mid 1) \mid x_t\big] \\
&= \frac{1}{\alpha}\delta_\alpha(x_t) - (\frac{1}{\alpha}-1)\mathbb{E}\big[\delta(\tilde{x}_{t,\lambda}) + f_{\theta^*}(\tilde{x}_{t,\lambda} \mid 1) - f_{\theta^*}(\tilde{x}^*_{t,\lambda} \mid 1) \mid x_t\big].
\end{aligned} \tag{39}$$

Assume the prediction error is bounded by $\|\delta_\alpha(x_t)\|, \|\delta(x_t)\| \le e$ at the real input and $\|\delta(y)\| \le e + L_e\|y - x_t\|$ at the reference inputs. This assumption allows for reduced prediction accuracy as the reference input deviates from the real training data. Also assume $f_{\theta^*}(\cdot \mid 1)$ is L-Lipschitz between $\tilde{x}_{t,\lambda}$ and $\tilde{x}^*_{t,\lambda}$. Applying these bounds and Eq. (38) to Eq. (39) gives

$$\begin{aligned}
&\|f^{\text{scg}}_{\hat{\theta}}(x_t \mid \alpha) - f^{\text{scg}}_{\theta^*}(x_t \mid \alpha)\| \\
&\le \frac{1}{\alpha}e + (\frac{1}{\alpha}-1)\big[e + \sqrt{\lambda}\big(L_e\mathbb{E}[\|\tilde{x}_t - x_t\| \mid x_t] + L\sigma_t e\big)\big] \\
&= (1 + 2s_{\text{scg}})\, e + s_{\text{scg}}\sqrt{\lambda}\big(L_e\mathbb{E}[\|\tilde{x}_t - x_t\| \mid x_t] + L\sigma_t e\big),
\end{aligned} \tag{40}$$

where $\tilde{x}_t$ is the unmixed reference generated by $\hat{\theta}$ and $\alpha = 1/(1 + s_{\text{scg}})$. The $L\sigma_t e$ term accounts for the change in the reference input. At λ=1, a weak model can generate $\tilde{x}_t$ far from x_t, and the error at this reference enters the target amplified by s_{scg}. For a given model, λ<1 reduces the displacement-dependent terms in the bound by a factor of $\sqrt{\lambda}$.

Thus PatchMix brings the reference query $\tilde{x}_{t,\lambda}$ closer to the real input in RMS, tightening the assumed error bound on the base prediction used to approximate the optimum in Eq. (37). Under the stated assumptions, Eq. (40) shows that mixing degraded self-generated samples with real inputs suppresses the displacement-dependent contribution to the bound on the deviation from the PatchMix-SCG optimum. This supports the role of interpolation in Section 4.3 as a way to obtain a more reliable guidance anchor under limited model capacity.

D SCALE Training Pipeline for Autoregressive Models

We provide the training pipeline of SCALE for autoregressive (AR) models in Algorithm 2. The procedure mirrors the diffusion case (Algorithm 1 in the main text), with the noise predictor $\boldsymbol{\epsilon}_\theta$ replaced by the next-token predictor f_θ, and the score-matching loss replaced by the standard cross-entropy loss on the softmax-normalized logits. PatchMix is applied at the token level along the spatial dimension following Eq. (11), and the nested guidance formulation in Eq. (16) is applied in the logit space before the softmax.

E Further Implementation Details

In this section, we provide more detailed implementation details for clarity. As denoted in the main manuscript, we explicitly follow the training recipe from the base generative models and hyperparameters [7]. For the fine-tuning, the training epoch is set to 1 5% of the pre-training. We elaborate on the detailed training configurations used for SCALE training.

Input of α, β. Following implementation in GFT [7], we use Fourier embedding with MLP layers [49] to add conditioning for the models. For the fine-tuning, we adopt zero initialization for the conditioning network [69]. With this formulation, the additional embeddings for the α and β provide a lightweight (0.0012%) parameter overhead, and negligible impact on inference latency.

PatchMix patch size. We set the PatchMix patch size in Eq. (11) to align with the model's self-attention granularity, so that each mixed patch respects the unit at which the model processes spatial

Algorithm 2: SCALE Training (AR)

Initialize next-token predictor p_θ and ratio λ
for each gradient step **do**
  $\boldsymbol{x}, c \sim p(\boldsymbol{x}, c)$, $\alpha, \beta \sim U(0,1)$
  $\ell_\theta^{\text{scg-cfg}} = p_\theta(\boldsymbol{x}|c,\alpha,\beta)$ ▷ *Target*
  // Additional to Training (Evaluation Mode)
  $\ell_\theta^u = p_\theta(\boldsymbol{x}|\varnothing,1,1)$
  Sample $\tilde{\boldsymbol{x}}$ from ℓ_θ^u
  $\tilde{\boldsymbol{x}}_\lambda = \text{PatchMix}(\boldsymbol{x},\tilde{\boldsymbol{x}};\lambda)$ ▷ *Eq.* (11)
  $\ell_\theta^\lambda = p_\theta(\tilde{\boldsymbol{x}}_\lambda|c,1,\beta)$
  // Nested guidance formulation ▷ *Eq.* (16)
  $\ell_\theta^{\text{cfg}} = \alpha\,\ell_\theta^{\text{scg-cfg}} + (1-\alpha)\,\text{sg}[\ell_\theta^\lambda]$
  $\ell_\theta^c = \beta\,\ell_\theta^{\text{cfg}} + (1-\beta)\,\text{sg}[\ell_\theta^u]$
  $p_\theta^c = \text{softmax}(\ell_\theta^c)$
  $\theta \leftarrow \theta - \mu\nabla_\theta\,[-\log p_\theta^c]$ ▷ *Eq.* (2)
end for

information. For Transformer-based models with an explicit patch tokenizer, this is the token/patch size of the input: 2 for DiT-XL/2 and DiT-B/2 [44] (latent patches of size 2), and 1 for LlamaGen-L [54] (one VQ token per spatial location). For UNet-based models such as SD 1.5 [49], which contain multiple self-attention resolutions, we use the finest self-attention resolution in the latent space, corresponding to a patch size of 2.

Training and sampling. For fine-tuning, we limit training to less than 5% of the pre-training budget and inherit all hyperparameters from the base model. For training from scratch, we follow the original training recipe of the base model. For diffusion sampling, we use 50-step DPM-Solver++ [40]. For text-to-image, we additionally use LoRA [50] due to compute constraints (full setup in the SD 1.5 paragraph below). In our experiments, we found that combining CFG with SCG yields the best performance, as it provides an orthogonal guidance direction (see 5.3). To ensure that the model focuses on the effective range of guidance strengths, we set a minimum α of 0.5 for class-conditional models and 0.25 for text-to-image models during training, which corresponds to setting the maximum SCG strength as 1.0 and 3.0, respectively.

E.1 Fine-tuning

LlamaGen We utilize LlamaGen-L for fine-tuning AR models with SCALE. We fine-tune these models with 15 epochs with a batch size of 256. The learning rate is set to 2e-4 with cosine decay. All pre-trained models are obtained from the official repository [54]. We obtain $\tilde{\boldsymbol{x}}_t$ by categorical sampling from the unconditional logits, without temperature scaling or top-k truncation, so that $\tilde{\boldsymbol{x}}_t$ is drawn directly from the model's distribution.

DiT For the main experiments and ablation studies, we fine-tune DiT-B/2 and DiT-XL/2 with SCALE using a batch size of 256 and a learning rate of 1e-4 with cosine decay. The EMA decay rate is set to 0.9999. We observed that initializing from a pretrained from GFT objective model yields slightly better performance. We posit that incorporating self-contrastive guidance subsequent to CFG training is advantageous. For sampling, we adopt 50-step DPM-Solver++ [40].

Stable Diffusion v1.5 (Text-to-Image) For text-to-image generation at 512×512 resolution, we fine-tune Stable Diffusion v1.5 (SD 1.5) [49] with SCALE on LAION-Aesthetics 5+ [51], which contains 18M text-image pairs. Due to compute constraints, we use LoRA [23] with rank $r = 128$ and scaling factor 128, trained for 50k iterations using a batch size of 256, a learning rate of $1\text{e}{-5}$, and 1000 warmup steps. We initialize from a GFT fine-tuned SD 1.5 checkpoint and apply SCALE on top, which compensates for the limited capacity of LoRA. For evaluation, zero-shot FID-30k, CLIP Score [45], and Aesthetic Score [51] are measured on 30k images generated from COCO 2014 [36] validation prompts, while the human-preference metrics including PickScore [29], HPSv2.1 [62], and Image Reward [63] are measured on the Pick-a-Pic test prompts [29]. For all quantitative and qualitative results, we utilize 50-step DPM-Solver++ [40] for sampling.

DiffusionDPO on top of SCALE. For the DPO experiment, we fine-tune the GFT and SCALE-trained SD 1.5 with DiffusionDPO [60] following the `diffusers` implementation[3]. We use LoRA with rank $r = 8$ and $\alpha = 8$, learning rate $1e{-5}$, gradient accumulation of 8 for an effective batch size of 128, and train for 10k iterations on a single RTX 4090 GPU using the Pick-a-Pic dataset [29].

During DPO training, we fix the SCALE conditioning at $\alpha = 1, \beta = 1$ (i.e., the unamplified branch) and only vary α, β at inference to control the SCG and CFG strengths. This mirrors the standard practice in language-model alignment, where DPO training is performed with sampling temperature 1.0 and the temperature is lowered only at inference [46, 60]. Training GFT with DPO follows the same practice, setting $\beta = 1$ for the DPO training and a lower β at inference, keeping other training hyperparameters unchanged, such as LoRA rank, learning rate, and effective batch size.

E.2 Training From Scratch

Due to the computational resource limitations within an academic research setting, pre-training with large-scale visual generative models was not feasible. In this paper, to demonstrate SCALE's applicability to pre-training, we primarily conduct experiments using DiT-B/2 models [44]. This model was selected as it requires moderate training resources, requiring 80 epochs (400k iterations with a batch size of 256) for training and taking $\sim$ 3 days with 8 RTX4090 GPUs. For the pre-training, we drop conditions for the generated sample branch of SCALE. In this case, $\bar{\boldsymbol{x}}_t = \tilde{\boldsymbol{x}}_t$ and the h_t is a combination of $\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t$ and the condition c for the information-based guidance formulation in Eqs. (5) and (6). By omitting the conditional signal, the objective initially emulates a CFG, which effectively stabilizes the model before the self-contrastive signal becomes reliable. This approach facilitates a natural curriculum, where the model first learns CFG signals and seamlessly transitions to incorporating self-contrastive signals as generation quality matures.

E.3 Human Preference Study

We complement the automated human preference metrics on SD 1.5 with a user study that directly compares SCALE against the strongest guidance-free baseline (GFT) in a side-by-side setting. The study is conducted as a **blind A/B test** with **20 participants**, each responding to **15 image pairs** for a total of 300 votes. For each trial, the participant is shown one text prompt and two generated images (A and B), one from GFT and one from SCALE, with model identities hidden and the left/right order randomized per trial. The participant is asked to select the image they prefer overall, or to choose "Tie" if the two are of comparable quality. Prompts are sampled from the Pick-a-Pic test set so that the comparison covers diverse content. An example trial is shown in Fig. 7.

Following common practice in text-to-image preference studies, participants are asked to consider three axes when making their choice:

- *Text alignment*: which image more accurately represents the content of the given prompt;
- *Visual quality*: which image exhibits better clarity, texture, and fine details;
- *Photorealism*: which image looks more realistic and aesthetically pleasing, without artificial artifacts.

F Related Works

F.1 Visual Generative Models

Likelihood-based visual generative models have been actively researched along several architectural lines. Diffusion models [19, 53, 43, 49, 44, 42] learn the data score and generate samples through iterative denoising. Autoregressive models [13, 54, 57, 47, 55, 34, 16, 68] factorize the joint distribution into a sequence of next-token predictions over discrete or continuous tokens. Masked generative models [3, 33, 4, 61] predict masked tokens in parallel and trade autoregressive ordering for faster sampling. Flow matching models [39, 37, 14, 31] learn a velocity field between noise and data and have recently scaled to large text-to-image systems such as Stable Diffusion 3 [14] and FLUX [31]. Despite their architectural differences, all of these families produce samples whose unconditional fidelity falls short of the training distribution and rely on additional guidance signals to close the gap.

[3] `github.com/huggingface/diffusers/research_projects/diffusion_dpo`

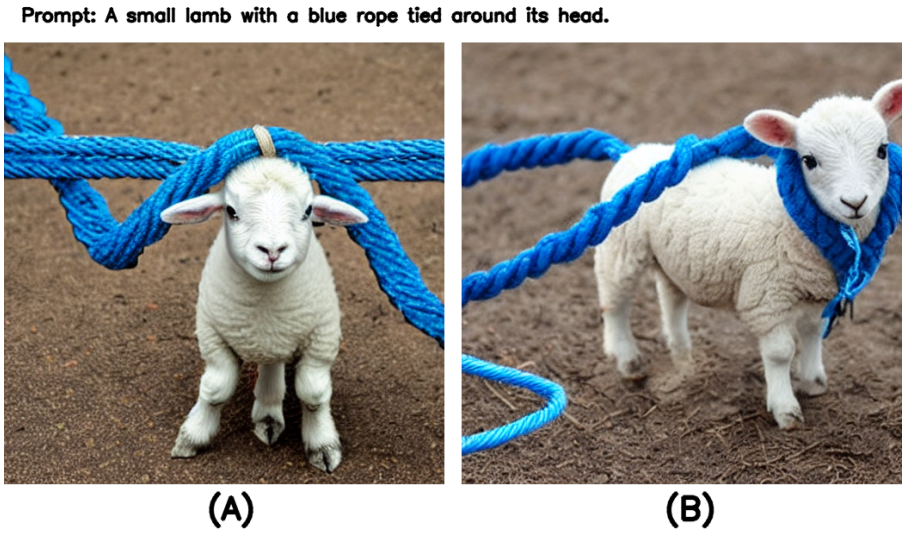


Figure 7: Example trial from the blind A/B human preference study. **Left (A):** GFT [7]. **Right (B):** SCALE. Participants selected the image without these labels.

F.2 Contrastive Signals for High-Fidelity Visual Generation

A broad range of methods enhances sample fidelity by leveraging a contrastive signal that distinguishes desired samples from undesired ones.

Information-based guidance. Classifier-free guidance (CFG) [18] contrasts a conditional and an unconditional prediction at sampling time, amplifying the conditional signal. Information-based guidance [22, 24] generalizes CFG in terms of model-based interpretation, constructing the undesired sample from a degraded version of the input itself. From this perspective, CFG is a special case where the degradation is dropping the condition. Formally, these methods amplify any salient information h_t relative to a degraded reference $\bar{\boldsymbol{x}}_t$. Self-Attention Guidance (SAG) [22] utilized a Gaussian blur to amplify high-frequency details of the input, with a self-attention mask to stabilize the guidance. Self-Guidance [24] utilizes an auxiliary model to produce a degraded reference input for the semantic vector-quantized tokens [48, 13]. While effective, these methods require additional forward passes and rely on heuristics for constructing degraded inputs, which are often domain-specific and hard to generalize to other domains.

Weak-model guidance. These methods leverage a weaker version of the model itself to produce undesired samples. These methods generalize CFG in terms of model-based interpretation by replacing the unconditional reference with a weaker version of the model itself: autoguidance [26] contrasts the main model with a smaller or earlier checkpoint, Perturbed Attention Guidance [1] and Smoothed Energy Guidance [21] construct weak references through attention perturbation, and S^2-guidance [5] contrasts with random subnetworks of the main model, constructed by stochastic block-dropping.

Guidance-free generation. Guidance-free methods incorporate CFG during training to enable sampling with a single model evaluation per step. Guidance distillation [41] trains a student to match the guided predictions of a pretrained diffusion teacher. Condition Contrastive Alignment (CCA) [6] fine-tunes AR models using matched and mismatched image–condition pairs and a frozen pretrained reference model, with a contrastive alignment objective whose optimum recovers the CFG sampling distribution. Guidance-Free Training (GFT) [7] retains the standard likelihood objective and reparameterizes the conditional model output as an interpolation between the sampling and unconditional predictions. This enables guidance-free training from scratch across diffusion, AR, and masked generative models. Model-guidance (MG) [56] similarly learns CFG during training, but adds a stop-gradient CFG residual directly to the regression target rather than reparameterizing the model output. These approaches derive their guidance signal from the conditional–unconditional contrast

of CFG. SCALE introduces a complementary contrast between real and self-generated samples and learns it alongside CFG, aiming to improve generation beyond reproducing CFG behavior.

Reinforcement learning-based methods. Developed from the language model alignment literature, reinforcement learning (RL) post-training methods often construct a desired and undesired contrast through reward or preference signals. Direct preference optimization (DPO) [46, 60, 71, 65, 35, 10] optimizes the policy directly on paired win–lose paired preference data. In this case, the model is to maximize the win probability over the lose probability, which implicitly encourages the model to generate samples that are preferred by the human. Recently, group relative policy optimization (GRPO) [52, 38, 64, 17] samples a group of responses for each prompt and uses their reward statistics within the group as a critic-free baseline, optimizing the model with the resulting relative advantages. While effective, these methods require an external network or curated preference data and typically still rely on CFG at inference for generating high-quality samples in the visual domain.

G Additional Results

G.1 Computational Overhead Analysis

This section details the experimental setup and per-row analysis of Table 1 in the main text.

Measurement setup. For diffusion models, we report the costs of fine-tuning from a pretrained checkpoint on $8\times$RTX 4090 GPUs. We use DiT-B/2 [44] as the base model rather than DiT-XL/2 because guidance distillation requires loading the teacher network alongside the student, and the resulting peak memory exceeds the 24 GB capacity of a single RTX 4090 at the DiT-XL/2 scale. DiT-B/2 fits comfortably for all methods and yields a fair comparison under identical hardware. For AR models, the numbers for CCA [6] and GFT are taken directly from the GFT paper [7], which measured costs on VAR-d30 [57].

Per-row analysis. We unpack each row of Table 1 below.

Train time / step. The relative cost is determined by the number of model evaluations per gradient step. Guidance distillation [41] evaluates the teacher's conditional and unconditional predictions in addition to the student's forward pass, yielding a $\times 1.12$ overhead. GFT performs only one extra unconditional forward as a stop-gradient reference, which gives the lightest overhead at $\times 1.06$. SCALE matches Distillation's $\times 1.12$ because, in addition to the conditional forward, it computes a stop-gradient unconditional reference and a stop-gradient SCG reference output, both without backward passes. CCA on AR models reports a higher $\times 1.69$ as it requires sampling negative tokens and an additional contrastive forward per step.

Params. SCALE introduces only the conditioning embeddings for α and β (Fourier features followed by a small MLP), accounting for roughly 0.0012% of the base model parameters. The added parameters are therefore negligible for both model size and inference latency, leaving the parameter ratio at $\times 1.00$.

VRAM. Guidance distillation must keep the teacher network resident in memory alongside the student, which inflates VRAM to $\times 1.21$ on DiT-B/2 (and prevents fitting DiT-XL/2 on a 24 GB GPU). CCA similarly requires storing reference statistics for the contrastive objective, leading to $\times 1.39$ on AR. GFT and SCALE, in contrast, share the same single network for all branches and only run extra forward passes without gradient retention, so VRAM remains at $\times 1.00$.

Inference. All single-network guidance-free methods (Distill, CCA, GFT, SCALE) perform a single forward pass per sampling step rather than CFG's two, halving the inference cost to $\times 0.50$.

Summary. SCALE matches GFT in memory and inference cost while keeping training-time overhead within the same order as guidance distillation, despite delivering a strictly richer training target ($p_t^{\text{cfg}}(\boldsymbol{x}_t|c)/p_t^{\text{cfg}}(\tilde{\boldsymbol{x}}_t|c)$ vs. $p_t^{\text{cfg}}(\boldsymbol{x}_t|c)$). The added cost is concentrated in two gradient-free reference passes during training and vanishes entirely at sampling time.

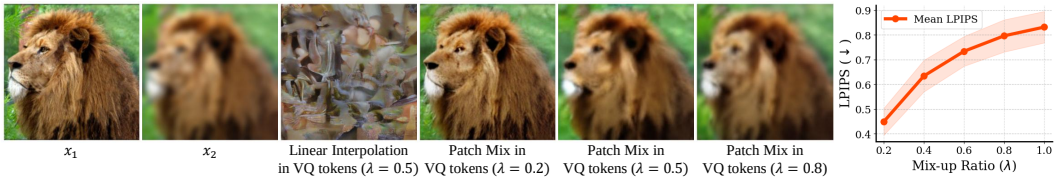


Figure 8: **Effect of interpolation choice. (Left)** Linear vs. patch-wise interpolation in the VQ token space (LlamaGen tokenizer [54]). Linear interpolation collapses to noise at intermediate λ, while PatchMix transitions smoothly between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. **(Right)** LPIPS distance between real samples and PatchMix-interpolated samples on DiT-XL/2 [44]: lower λ keeps the interpolated sample closer to the real-data manifold, providing a stable anchor for self-contrastive guidance.

Table 8: Comparison with guided sampling methods on the ImageNet generation at 256×256.

Methods	Guid.	NFEs ↓	FID ↓	IS ↑	Precision ↑	Recall ↑
DiT-XL/2 [44]	✗	250	9.62	121.5	0.67	**0.67**
+CFG [18]	✓	250 × 2	2.27	278.24	**0.83**	0.57
+SAG [22]	✓	250 × 3	2.11	–	–	–
+G++ [27]	✓	250 × 3	**1.83**	281.53	0.78	0.64
+SCALE (Ours)	✗	250	2.10	**288.08**	0.81	0.59

G.2 Visualization of PatchMix Interpolation

Fig. 8 provides additional support for the patch-wise design choice in Eq. (11). **(Left)** We compare linear and patch-wise interpolation in the vector-quantized (VQ) token space using the LlamaGen tokenizer [54]. For linear interpolation, we apply $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ on the token indices and round to the nearest integer. Despite the high structural similarity between the two source images, linear interpolation fails to preserve any semantic content at intermediate λ, since rounded VQ indices do not lie on a smooth manifold of the codebook. PatchMix, by replacing whole patches rather than interpolating coordinates, transitions smoothly from $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$ as λ increases from 0 to 1 and remains semantically interpretable throughout. **(Right)** We measure the LPIPS distance [70] between 50 real images and their PatchMix-interpolated counterparts $\tilde{\boldsymbol{x}}_{t,\lambda}$ on DiT-XL/2 [44], varying λ. The perceptual distance decreases monotonically as λ shrinks, confirming that a smaller λ keeps $\tilde{\boldsymbol{x}}_{t,\lambda}$ close to the real-data manifold and therefore provides a stable anchor for the self-contrastive guidance.

G.3 Comparison with Guided Sampling Methods

Table 8 reports an additional comparison under the original 250-step DDPM solver of DiT-XL/2 [44], where the reported numbers for SAG [22] and G++ [27] are taken from their original papers under their original solvers, ensuring a faithful baseline. Even without any guidance at sampling time, SCALE attains the best IS and a competitive FID, surpassing CFG and remaining on par with guided sampling methods that incur substantial extra inference cost.

While SAG and G++ further improve over CFG, they do so at the price of an additional forward pass per step and rely on settings that limit their generalizability. SAG operates at a tiny guidance scale of 0.005, providing only a marginal guidance signal that is unlikely to significantly improve quality over CFG. G++ adopts a timestep-aware guidance schedule tightly coupled to its solver: when transferred from the 250-step DDPM solver to the 50-step DPM-Solver++, we observe a significant degradation in IS (compared to Table 4 in the main text), indicating that its gains are sensitive to the sampling configuration. Moreover, both methods are specifically designed for score-based diffusion models and cannot be readily applied to discrete generative models such as AR models.

In contrast, SCALE delivers consistent improvements without any guidance at sampling time, achieving the highest IS in this comparison. Crucially, SCALE is agnostic to the choice of diffusion solver, applies uniformly to both diffusion and AR models, and requires no model-specific tuning, making it a broadly applicable framework for high-fidelity guidance-free generation.

H Additional Visualizations and Detailed Analysis

Fig. 9 and Fig. 10 show the samples of SCALE by varying α and β using DiT-XL/2 [44] and LlamaGen-L [54], respectively. We note that *all samples are generated with a single model without guidance at sampling time*, and varying α and β at inference time does not incur additional computational cost.

Effect of varying β (implicit CFG). By decreasing β, the sample quality is improved by amplifying the conditional signal. However, it often pushes samples toward canonical templates and the semantic structure of samples can be steeply changed as CFG becomes stronger, as shown in each column of the image grid in Fig. 9 and Fig. 10. Consequently, strong CFG (i.e., lower β) produces samples with reduced diversity as shown in the bottom row of each image grid.

Effect of varying α (implicit SCG). By decreasing α, the sample quality is improved by amplifying the self-contrastive signal. It also improves the sample quality, but preserves the semantic structure as shown in each row of the image grid in Fig. 9 and Fig. 10. As a result, strong SCG (i.e., lower α) produces more diverse high-quality samples as shown in the rightmost column of each image grid. However, limitations exist. Since SCG improves the sample quality while preserving the semantic structure, it may not be able to produce high-quality samples if the unguided sample is far from the data manifold, as shown in the top row of each image grid.

Combination of SCG and CFG. Aforementioned analysis suggests that SCG and CFG are complementary to each other. Therefore, we combine SCG on top of the CFG prediction, which directly rectifies the CFG-generated samples to the data manifold. This compensates for the limitation of each individual guidance: moderate CFG enhances prompt alignment and rectifies the underlying semantic structure, while SCG improves the sample quality on top of the CFG prediction. Utilizing CFG alone with a higher guidance scale may push samples away from the data manifold or incur reduced diversity as discussed in Section 3.

Visualization with DPO post-training. Fig. 11 provides additional visualization of guidance-free generation with DPO post-training using stable diffusion v1.5 (SD 1.5) [49]. We note that *all samples are generated with a single model without guidance at sampling time*, and varying α and β does not incur additional computational cost. DPO post-training without CFG at sampling time fails to produce high-quality samples. While GFT and its DPO fine-tuned model improve the sample quality, SCALE further enhances the sample quality by directly refining the samples towards the data manifold, implicitly. Fig. 12 also confirms that amplifying the SCG signal by decreasing α gradually improves the sample quality of the DPO fine-tuned SCALE model.

I Limitations and Future Work

While SCALE provides consistent gains across both class-conditional and text-to-image generation on diffusion and AR architectures, several questions remain open.

Unified guidance formulation. The current SCALE framework is built upon the hierarchical guidance formulation, which combines the CFG and the SCG. This two-step formulation is effective by combining the benefits of each individual guidance as discussed in Appendix H, and provides an interpretable guidance signal. However, this approach inevitably introduces additional training overhead ($+6\%$p per step compared to CFG). A unified, single-step guidance formulation to further improve the guidance-free training efficiency could be a promising future direction.

Extension to visual alignment tasks. The SCALE framework currently focuses on improving the sample quality of visual generative models without guidance. However, self-contrastive signals are applicable to broader domains, such as visual alignment tasks. For instance, large visual generative models can generate unsafe, NSFW samples, which can be harmful to society. Self-contrastive signals can be used to pull the models toward the distribution of safe data while pushing it away from unsafe, NSFW distributions, without any guidance at sampling time and without loss of sample quality.



Figure 9: Visualization of SCALE by varying α and β using DiT-XL/2 [44].

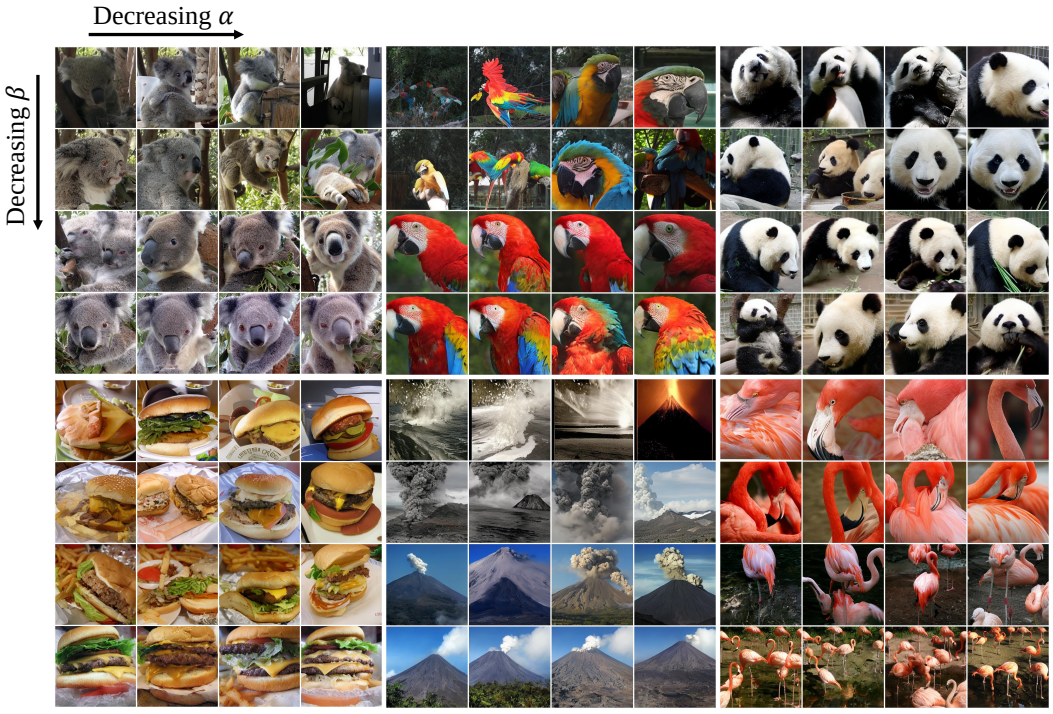


Figure 10: Visualization of SCALE by varying α and β using LlamaGen-L [54].

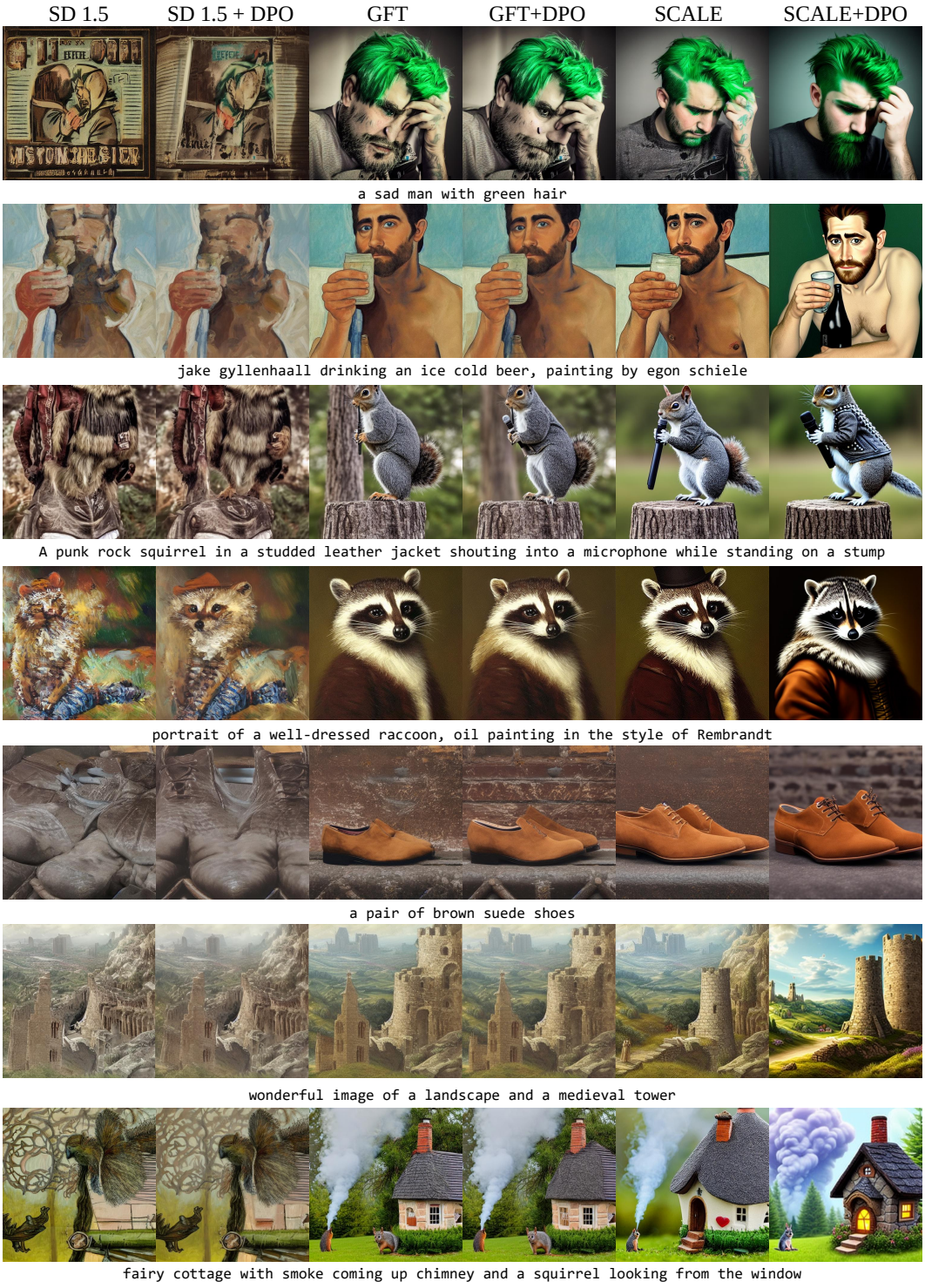


Figure 11: Guidance-free generation results of base SD 1.5 [49], GFT, and SCALE (ours), with and without DPO post-training [60]. Prompts are sampled from the Pick-a-Pic test set [29] and PartiPrompts [67].

Beyond DPO post-training. We empirically verify that SCALE is complementary to DiffusionDPO [60] by post-training a SCALE-trained model with DPO. However, our investigation



Figure 12: Visualization of SCALE by varying α using SD 1.5 [49] with DPO post-training [60].

is currently limited to a preliminary combination with DPO, and a systematic study of how SCALE interacts with RL-based post-training remains an open question. In particular, validating the compatibility of SCALE with more recent RL-based post-training methods, such as DSPO [71], D3PO [65], SPO [35], FlowGRPO [38], or DanceGRPO [64], would be a valuable direction for future work.

Scaling to larger and more recent generative architectures. Our experiments cover diffusion models (DiT, SD 1.5) and AR models (LlamaGen) on academic-scale compute. Recent text-to-image models adopt rectified-flow transformers and a billion-scale parameter regime, including Stable Diffusion 3 [14] and FLUX [31]. Validating that the self-contrastive signal scales to these flow-based, large-scale models would be a promising direction.

J Ethics Statement

SCALE is a technical framework for improving the visual fidelity of guidance-free generative models. While SCALE can enhance sample quality and align outputs more closely with learned data distributions without relying on external classifier guidance, the same capabilities could be misused by unauthorized actors. Potential risks include the generation of more convincing, deceptive, or misleading visual media, with downstream harms to privacy, reputation, and public trust. Because SCALE often operates as a post-training method on pretrained backbones while sharing the training datasets, it inherits both the capabilities and the limitations of the underlying models, including any biases, memorization effects, or harmful content patterns present in their training data.

Our experiments are conducted exclusively on publicly available, well-established generative models that ship with safety filters and community-vetted usage policies. SCALE itself is not a safety mechanism; it modifies the sampling trajectory to improve fidelity but does not detect, suppress, or mitigate unsafe content. It should therefore be deployed only in conjunction with robust prompt- and output-level moderation, provenance signals (e.g., watermarking or content credentials) where appropriate, and human oversight for sensitive applications.

This work is intended for academic research and constructive applications such as creative tools, scientific visualization, and the study of guidance mechanisms in diffusion and autoregressive generative models. We explicitly prohibit malicious or unethical use, including the generation of non-consensual imagery, impersonation, disinformation, or any content intended to cause harm to individuals or groups. We encourage practitioners adopting SCALE to carefully document their assumptions and design choices, adhere to the licenses and safety settings of the underlying models, and contribute to the development of clear ethical guidelines, so as to support the responsible advancement of guidance-free generation methods and the broader generative modeling community.

K Declaration of LLM Usage

We used LLMs solely for editorial assistance. Their use was limited to polishing grammar, converting handwritten mathematical expressions into LaTeX (including the formatting of proofs in the appendix), and assisting with the visualization of figures and tables (e.g., generating boilerplate plotting code and refining layouts). The model did not contribute to ideation, research direction, or the main claims of this work, and it was not used for data analysis or core algorithmic design. All technical statements, equations, experimental results, and scientific contributions were authored and verified by the authors.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claim in the abstract and introduction is comprehensively dealt with in the paper, including the background, method, results, and additional analysis in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitations are discussed in the appendix, which is also clearly stated in the conclusion of the main manuscript

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide a theoretical justification of the proposed method in the main manuscript and the detailed proof in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed experimental setup for reproducing the results in the main manuscript.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not include code for the experiments. However, the dataset and baseline code for our implementation is clearly described in the paper, which is fully publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide a detailed experimental setup for training and sampling in the main manuscript. The hyperparameters for training is clearly ablated and the effect of hyperparameters for sampling is analyzed in detail.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not include the error bars for the experiments. However, most of the reported metrics are computed over a large number of samples (50,000 for class-conditional generation and 30,000 for text-to-image generation), following the standard evaluation protocol, which is known to be statistically stable accross different runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: In appendix, we report the training environment, as well as required training iterations for each experiment. We also compare the computational analysis of the proposed method with the baseline methods in the main manuscript.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We wrote a paper in compliance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts of the proposed method in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our work utilizes publicly released large pretrained visual generative model, Stable Diffusion 1.5 [49]. We use the model through the official `diffusers` library, which provides a built-in safety checker that filters NSFW and otherwise unsafe outputs by default. We do not release any large pretrained model that could be potentially misused. We only release the fine-tuning code and model weights that build upon existing checkpoints and publicly available datasets, and inherit the license and safety settings of the underlying models. While a pretrained model is involved (DiT-B/2 [44]), it poses low risk of misuse as it is relatively small and trained on publicly available, non-sensitive data (ImageNet [9]).

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use open source implementation for our baseline properly denoted in the paper. Otherwise, we do not release any new models pretrained from sensitive, private datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

Justification: We do not release any new assets in the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: We conduct a blind A/B study with voluntary adult participants. Participation was voluntary and uncompensated, and participants were informed in advance about the purpose of the study and that the displayed images were generated by AI.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: We conduct a blind A/B study with voluntary adult participants. The participants were properly anonymized and detailed description for the study is provided in the appendix. The prompts used in the study were sampled from the public dataset, and we manually filtered out any prompts that could be unsafe or sensitive.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We declare the usage of LLMs for writing, editing, and formatting purposes in the appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.