Diffusion Models without Classifier-free Guidance

Zhicong Tang[1] **Dong Chen**[2] **Jianmin Bao**[2] **Baining Guo**[2]
[1]Tsinghua University [2]Microsoft Research Asia

Abstract

Classifier-free Guidance (CFG) is central to modern diffusion models, but operates as an inference-time correction where the training objective learns the unbiased conditional score, while guided sampling targets a posterior-tilted score that never explicitly learned. Rather than another inference-time patch or distillation, Model-guidance (MG) directly integrates the posterior tilt into the training objective and eliminates CFG during inference. MG serves as a plug-and-play module compatible with existing methods, yet accelerates convergence by online self-bootstrapping and doubles inference speed. Unlike distillation surrogates, a reparameterisation property reveals that MG implicitly learns the vanilla score without mode collapse. Extensive experiments demonstrate that MG matches and even surpasses CFG baselines, and achieves a state-of-the-art FID of 1.34 on ImageNet 256 benchmark. Code is publicly available at `https://github.com/tzco/Diffusion-wo-CFG`.

1 Introduction

Diffusion models [1–5] have become the cornerstone of many successful generative models, *e.g.* , image generation [6–10] and video generation [11–15] tasks. However, diffusion models trained with a standard denoising objective struggle to generate "low-temperature" samples [16, 17], and techniques such as classifier guidance [6] and Classifier-free Guidance (CFG) [16] have become standard *inference-time* fixes for improving sampling quality.

Despite its effectiveness, CFG operates as a post-hoc correction: conditional and unconditional predictions are interpolated at every denoising step on top of a model whose training never explicitly targeted the guidance-tilted score being sampled. This separation has known costs, including doubled inference compute, mixed conditional and unconditional tasks during training [17, 18], sensitivity to a guidance scale, and trajectories that no longer correspond to a valid heat diffusion of the data [19]. Subsequent remedies such as Autoguidance [17], guidance interval [18], CADS [20], and guided distillation [21] keep the training objective unchanged and instead modify the inference procedure or distil a CFG-corrected teacher into a student.

In this work, we argue the right place to address this mismatch is the *training objective itself* instead of inference-time procedure or post-hoc distillation, and propose Model-guidance (MG). MG moves the Bayes decomposition and correction of CFG into training, where condition dropping and an unconditional forward still exist, but uses them to construct an online self-bootstrapping target for the conditional prediction, anchored by the ground-truth denoising signal. The trained model then samples with a single forward at inference, with no CFG sampling or computation overhead.

As depicted in fig. 1, this single training-objective change yields multiple practical gains: inference speed is doubled (one forward per step instead of two), and the modification needs only a few lines of code on top of existing pipelines. MG is also distinct from conventional distillation: no pre-trained CFG teacher, no separate student net, and the guidance signal is generated online by the same model and anchored by the original denoising target. proposition 1 formalises this distinction: at the function-value level MG is a reparameterisation of standard denoising score matching, verified by recovering the vanilla score with reverse CFG.

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

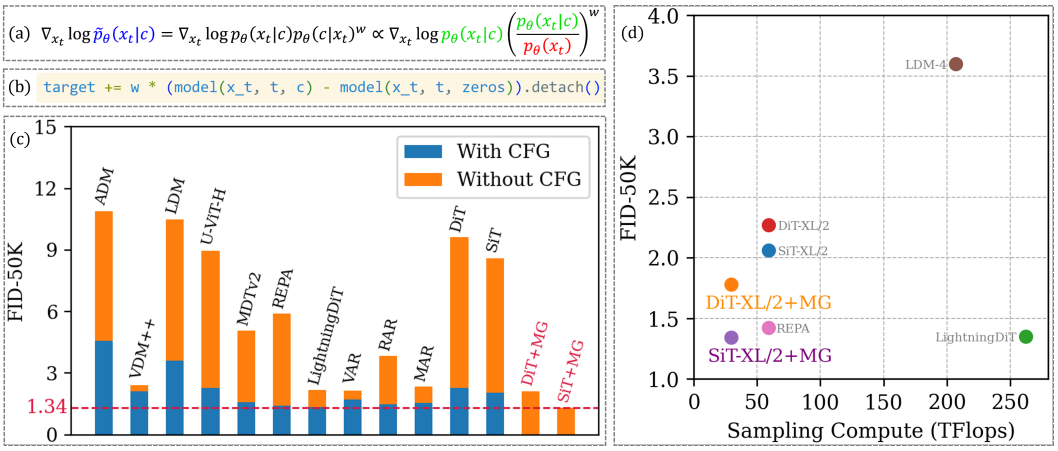


Figure 1: **We propose Model-guidance (MG), remove Classifier-free Guidance (CFG) during *inference-time*, and achieve state-of-the-art on ImageNet 256 generation with the FID of 1.34.** **(a)** While CFG needs two forwards (green and red), MG directly produces the final output (blue). **(b)** MG requires only few lines of code modification while providing significant quality improvements. **(c)** Comparing to concurrent methods, whether or not using CFG, MG yields the lowest FID results. **(d)** In addition to superior quality, the sampling cost of MG is also much lower than other methods.

We conduct comprehensive experiments on the prevalent ImageNet [22, 23] benchmarks at 256×256 and 512×512 resolution across DiT, SiT, and EDM2, and fine-tune Stable Diffusion 1.5 [8] on text-to-image. MG consistently matches or surpasses strong CFG-based baselines while halving inference compute. More broadly, we view MG as an instance of a wider opportunity: devices currently parked at inference time may be better expressed as training objectives, where they can interact with optimisation rather than be patched afterwards. In summary, our contributions are:

- We argue that CFG is a symptom of a training-sampling mismatch, and should better be addressed inside the training objective rather than inference-time or two-stage distillation.
- We propose Model-guidance (MG), an online self-bootstrapping objective with a reparameterisation property in proposition 1 that distincts from distillation-based surrogates.
- Extensive experiments across DiT, SiT, EDM2, and SD 1.5 fine-tuning achieve a state-of-the-art FID of 1.34 on ImageNet 256 with faster convergence and halved inference compute.

2 Background

2.1 Diffusion and Flow Models

Diffusion models [1–3] are a class of generative models that utilize forward and reverse stochastic processes to model complex data distributions. The forward process adds noise and transforms data samples into Gaussian distributions

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{1}$$

where x_t represents the noised data at timestep t and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ is the noise schedule. Conversely, the reverse process learns to denoise and finally recover the original data, which aims to reconstruct score [1, 5] from the noisy samples x_t by

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right), \tag{2}$$

where μ_θ and Σ_θ are mean and variance and predicted by neural networks. In common implementations, the training of diffusion models leverages a re-parameterized objective [3]

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\|\epsilon_\theta(x_t, t) - \epsilon\|^2. \tag{3}$$

Flow Models [24–27], an emerging type of generative models, utilize the concept of Ordinary Differential Equations (ODEs) and a forward process of Optimal Transport (OT) interpolant [28]. They learn the directions from noise pointing to ground-truth data by optimize

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,x_0,\epsilon}\left\|u_\theta(x_t) - u_t(x_t|x_0)\right\|^2, \quad \text{where} \quad x_t = (1-t)x_0 + t\epsilon \quad \text{and} \quad u_t(x_t|x_0) = x_0 - \epsilon. \tag{4}$$

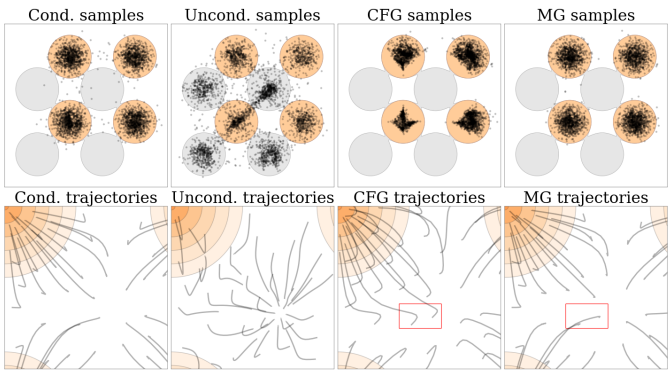


Figure 2: We use a grid 2D distribution with two classes, orange and gray, as an example and train diffusion models. **(a)** In the first row, CFG improves quality by eliminating the outliers, while the samples concentrate in the center and thus lost diversity. In contrast, MG yields less outliers, better coverage of data, and higher diversity than CFG. **(b)** In the second row, the trajectories of CFG turn sharply at the beginning, *e.g.* the samples inside the red box, while our method directly drives them to the closest data distributions.

2.2 Classifier-free Guidance

Classifier-free Guidance (CFG) [16] jointly trains conditional and unconditional diffusion models in a single network by randomly dropping the condition c with ratio λ and replacing it with an empty class $\varnothing$, so that $\epsilon_\theta(x_t, t, c)$ and $\epsilon_\theta(x_t, t, \varnothing)$ approximate the conditional and unconditional scores, respectively. At inference, CFG combines the two scores on-the-fly to guide the denoising process,

$$\tilde{\epsilon}_\theta(x_t, t, c) = \epsilon_\theta(x_t, t, c) + w \cdot (\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing)) \,, \tag{5}$$

where w is the guidance scale trading off generation quality and sampling diversity. CFG has become indispensable for modern diffusion models [29–32], but has several drawbacks [17, 18]: it doubles the number of function evaluations at inference, mixes conditional and unconditional tasks during training, and its trajectory does not represent a valid heat diffusion of the data distribution [19], leading to exaggerated truncation and mode dropping [33–35]. As fig. 2 shows, CFG improves quality at the cost of diversity and distorted trajectories.

3 Method

3.1 Model-guidance Loss

Instead of directly sampling from the learned conditional distribution, the core idea of CFG originates from classifier guidance [6] and effectively samples from a tilted joint distribution that emphasizes posterior alignment:

$$\tilde{p}_\theta(x_t|c) \propto p_\theta(x_t|c)\, p_\theta(c|x_t)^w, \tag{6}$$

where $w \geq 0$ is the weighting factor of the posterior. We emphasize that eq. (6) is *not* the standard Bayesian posterior $p(x|c)$, but the *implicit target sampling distribution* induced by the CFG inference rule (eq. (5)) [17, 19]. When w=0 it reduces to the model's conditional, and as w grows it places more probability mass on samples for which the model-implied posterior $p_\theta(c|x_t)$ is large. Since the classifier $p_\theta(c|x_t)$ of noisy samples is not directly available, CFG estimates it via Bayes' rule on-the-fly at inference,

$$p_\theta(c|x_t) = \frac{p_\theta(x_t|c)p_\theta(c)}{p_\theta(x_t)}, \tag{7}$$

where $p_\theta(x_t|c)$ and $p_\theta(x_t)$ are the conditional and unconditional distributions. The standard diffusion training objective only supervises the conditional score, while the posterior tilt that sampling actually targets is reconstructed at inference rather than learned. This is the structural mismatch we set out to close. It raises a natural question: rather than estimating $p_\theta(c|x_t)$ at every inference step, can we supervise the conditional prediction *during training* so that it directly produces a posterior-aligned score, eliminating the need for a CFG sampling rule?

As fig. 3 illustrates, we propose Model-guidance (MG) to directly learn the joint distribution $\tilde{p}_\theta(x_t|c)$, of which the score is decomposed as

$$\underbrace{\nabla_{x_t} \log \tilde{p}_\theta(x_t|c)}_{\text{joint score}} = \underbrace{\nabla_{x_t} \log p_\theta(x_t|c)}_{\text{conditional score}} + w \cdot \underbrace{\nabla_{x_t} \log p_\theta(c|x_t)}_{\text{posterior score}} \tag{8}$$

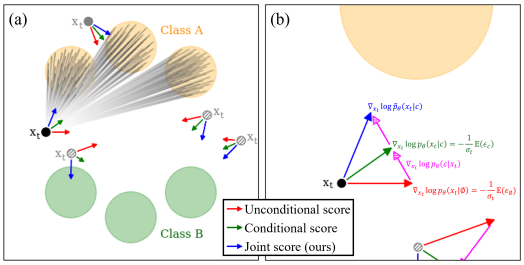


Figure 3: Illustration of CFG and proposed MG. **(a)** The conditional and unconditional scores point towards the centroids of data distribution, as the training pairs (x_0, ϵ) are randomly sampled. **(b)** While CFG provides better update directions by interpolating the two vectors during inference, MG aims to directly learn the joint score, $\nabla \log \tilde{p}_\theta(x_t|c)$, during training and thereby reduces inference costs.

During training, the ground-truth of the conditional score [2] in eq. (8) is

$$\nabla_{x_t} \log p_\theta(x_t|c) = -\frac{1}{\sigma_t}\epsilon, \tag{9}$$

where ϵ is the noise added to samples. However, the ground-truth of the posterior score, *e.g.* the score of posterior probability $p_\theta(c|x_t)$, cannot be directly obtained. Inspired by eq. (7), we transform the diffusion model itself into an implicit classifier. Specifically, we employ Bayes' rule during training to estimate the posterior probability

$$\begin{aligned} \log p_\theta(c|x_t) &= \log p_\theta(x_t|c) - \log p_\theta(x_t) + \log p_\theta(c) \\ &\propto \log p_\theta(x_t|c) - \log p_\theta(x_t) \end{aligned} \tag{10}$$

Next, we use the diffusion model itself to approximate the scores

$$\nabla_{x_t} \log p_t(x_t|c) = -\frac{1}{\sigma_t}\epsilon_\theta(x_t, t, c) \quad \text{and} \quad \nabla_{x_t} \log p_t(x_t) = -\frac{1}{\sigma_t}\epsilon_\theta(x_t, t, \varnothing), \tag{11}$$

where σ_t is the variance of the noise added to x_t at timestep t, $\varnothing$ is the empty class, and $\epsilon_\theta(\cdot)$ is the diffusion model. Substituting eq. (11) into eq. (10) yields

$$\nabla_{x_t} \log p_\theta(c|x_t) \propto -\frac{1}{\sigma_t}\left(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing)\right). \tag{12}$$

Combining eqs. (9) and (12) leads to a valid training target for eq. (8)

$$\nabla_{x_t} \log \tilde{p}_\theta(x_t|c) = -\frac{1}{\sigma_t}(\epsilon + w\left(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing)\right)). \tag{13}$$

The asymmetric use of ϵ and ϵ_θ in eq. (13) is dictated by the available supervision: the conditional score has the closed-form denoising target $-\epsilon/\sigma_t$ (eq. (9)), whereas the posterior score has no closed-form data target and is plugged in via the model's own predictions (eq. (11)).

Then, our method applies the Bayes' estimation in eq. (10) online and trains a conditional diffusion model to directly predict the score in eq. (8) according to eq. (13), instead of separately learning eq. (11) in the form of vanilla CFG. Consequently, our MG loss is

$$\mathcal{L}_{\text{MG}} = \mathbb{E}_{t,(x_0,c),\epsilon}\|\epsilon_\theta(x_t, t, c) - \epsilon'\|^2, \tag{14}$$

$$\epsilon' = \epsilon + w \cdot \text{sg}(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing)), \tag{15}$$

where the stop-gradient $\text{sg}(\cdot)$ treats the model-derived posterior correction in the manner of an online self-bootstrap target [36], yielding a *semi-gradient* objective in which the denoising target ϵ anchors the model to the data score while the model itself supplies a plug-in estimate of the posterior score.

Intuitively, this mirrors BYOL [36] that a network is supervised against a stop-gradient version of itself, and pure self-bootstrap can collapse without an asymmetric target. MG's data-side anchor, the ground-truth ϵ, plays the role of BYOL's EMA target, preventing trivial collapse to $\epsilon_\theta(c)$=$\epsilon_\theta(\varnothing)$ while the model's own posterior estimate steers the score. Section 3.2 formalises this construction as a function-value-level reparameterisation of standard denoising score matching that distinguishes MG from distillation-based surrogates whose students target a fixed teacher rather than the data score. Empirically, MG converges faster than the vanilla form in eq. (3).

For flow-based models, we have the analogous objective

$$\mathcal{L}_{\text{MG}} = \mathbb{E}_{t,(x_0,c),\epsilon}\|u_\theta(x_t, t, c) - u'\|^2, \tag{16}$$

$$u' = u + w \cdot \text{sg}(u_\theta(x_t, t, c) - u_\theta(x_t, t, \varnothing)). \tag{17}$$

Algorithm 1 Training with Model-guidance loss

Input: Dataset $\{\mathbf{X_i}, \mathbf{C_i}\}$, noise schedule $\bar{\alpha}$, learning rate η
Output: Model ϵ_θ
repeat
  Sample data $(x_0, c) \sim \{\mathbf{X_i}, \mathbf{C_i}\}$
  Sample noise $\epsilon \sim \mathcal{N}(0, 1)$ and time $t \sim \mathbf{U}(0, 1)$
  Add noise with $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
  Modify target $\epsilon' = \epsilon + w \cdot \text{sg}(\epsilon_\theta(x_t, c, t) - \epsilon_\theta(x_t, \varnothing, t))$
  Compute loss $\mathcal{L}_{\text{MG}} = \|\epsilon_\theta(x_t, c, t) - \epsilon'\|^2$
  Back propagation $\theta = \theta - \eta \nabla_\theta \mathcal{L}_{\text{MG}}$
until converged

where u is the ground-truth flow in eq. (4). During training, we randomly drop and replace the condition c in eqs. (14) and (16) with an additional empty class $\varnothing$ by a certain ratio λ to enable the estimation in eq. (11), sharing this design with CFG. At inference, however, the trained model samples with a single conditional forward and no CFG sampling rule nor any computation overhead.

3.2 A Reparameterisation View of MG

Proposition 1 (Unbiased prediction)**.** *At any optimum $\theta^\star$ of $\mathcal{L}_{\text{MG}}$ in eqs.* (14) *and* (15)*, $(1-w)\epsilon_{\theta^\star}(x_t, t, c) + w\,\epsilon_{\theta^\star}(x_t, t, \varnothing) = -\sigma_t \nabla_{x_t} \log p_t(x_t|c)$, and applying CFG with $w' = -w$ on an MG-trained model recovers the true conditional score at inference.*

Proof sketch. Expanding the square in eqs. (14) and (15) at the function-value level with gradient routing ignored gives

$$\mathcal{L}_{\text{MG}} = \mathbb{E}\big\|(1-w)\epsilon_\theta(x_t, t, c) + w\,\epsilon_\theta(x_t, t, \varnothing) - \epsilon\big\|^2,$$

which is the standard denoising loss applied to the linear combination $\hat{\epsilon}_\theta := (1-w)\epsilon_\theta(c) + w\,\epsilon_\theta(\varnothing)$. By the standard score-matching argument, at any optimum $\theta^\star$, $\hat{\epsilon}_{\theta^\star} = -\sigma_t \nabla_{x_t} \log p_t(x_t|c)$. Setting $w' = -w$ in eq. (5) then inverts the linear map and recovers the true conditional score. The stop-gradient changes gradient routing but not the loss value at any θ, so this is an optimum-level identity at the function-value level. Full derivation in section G.3. □

This identity has two consequences. First, MG is a reparameterisation of standard score matching, while distillation is not since the student targets a fixed teacher rather than the data score. Second, applying reverse CFG ($w' = -w$) at inference recovers vanilla FID and Recall in table 17, so MG retains conditional support and avoids mode collapse, as the high Recall in tables 1 and 2 confirms.

3.3 Compare MG with Distillation

Given a pre-trained diffusion model $\epsilon_\theta(x_t, t, c)$ as teacher, Guided Distillation (GD) [21] trains a separate student $\epsilon_\eta(x_t, t, c, w)$ that takes the guidance scale w as input, with the loss

$$\mathcal{L}_{\text{GD}} = \mathbb{E}_{w,t,(x_0,c),\epsilon} \left\|\epsilon_\eta(x_t, t, c, w) - \epsilon^w_\theta(x_t, t, c)\right\|^2, \tag{18}$$

$$\epsilon^w_\theta(x_t, t, c) = \epsilon_\theta(x_t, t, c) + w(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing)), \tag{19}$$

where $w \sim \mathbf{U}[w_{\text{min}}, w_{\text{max}}]$. The student is supervised against a fixed CFG-corrected output that is itself a surrogate for no well-defined distribution, so it is upper-bounded by this surrogate and cannot exceed it. MG differs structurally. There is no separate student, no fixed teacher, and by proposition 1 the supervision reduces to the standard denoising target at the function-value level. MG is therefore anchored by the data distribution rather than by a frozen surrogate. Empirically, offline GD and online distillation variants in section C all under-perform MG on identical architectures.

3.4 Implementation Variants

Scale-aware networks. Similar to other distillation-based methods [37, 21], the guidance scale w can optionally be fed into the network as an additional condition. At each training iteration, we

Table 1: Experiments on ImageNet 256 conditional generation without CFG. By employing MG, the performances of both DiT and SiT are greatly boosted, achieving state-of-the-art.

Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑	Tflops↓	#Params↓
ADM [6]	10.9	-	101.0	0.69	0.63	1120	554M
VDM++ [38]	2.40	-	225.3	0.78	0.66	-	2.0B
LDM [8]	10.5	-	103.5	0.71	0.62	103.6	400M
U-ViT [39]	8.97	-	136.7	0.69	0.63	-	501M
MDTv2 [40]	5.06	-	155.6	0.72	0.66	-	676M
REPA [41]	5.90	6.33	162.1	0.71	0.56	29.65	675M
LightningDiT [42]	2.17	4.36	205.6	0.77	0.65	131.2	675M
VAR [43]	2.16	-	288.7	0.81	0.61	-	2.0B
RAR [44]	3.83	-	274.5	0.79	0.61	-	1.5B
MAR [45]	2.35	-	227.8	0.79	0.62	-	943M
DiT-XL/2 [46]	9.62	6.85	121.5	0.67	0.67	29.65	675M
+MG$_{\text{(ours)}}$	**1.78**	4.46	298.5	0.80	0.66	29.65	675M
Improvement	81.5%	34.9%	146%	19.4%	−0.1%	0.0%	0.0%
SiT-XL/2 [47]	8.61	6.32	131.7	0.68	0.67	29.65	675M
+MG$_{\text{(ours)}}$	**1.34**	4.58	321.5	0.81	0.65	29.65	675M
Improvement	84.4%	27.5%	144%	19.1%	−3.0%	0.0%	0.0%

Table 2: Experiments on ImageNet 256 conditional generation with CFG. Comparing to models using CFG, our method still obtains excellent results and surpasses others with lower Tflops.

Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑	Tflops↓	#Params↓
ADM [6]	4.59	5.25	186.7	0.82	0.52	2240	544M
VDM++ [38]	2.12	-	267.7	0.81	0.65	-	2.0B
LDM [8]	3.60	-	247.7	0.87	0.48	207.2	400M
U-ViT [39]	2.29	5.68	263.9	0.82	0.57	-	501M
MDTv2 [40]	1.58	4.52	314.7	0.79	0.65	-	676M
REPA [41]	1.42	4.70	305.7	0.80	0.65	59.30	675M
LightningDiT [42]	1.35	4.15	295.3	0.79	0.65	262.3	675M
VAR [43]	1.73	-	350.2	0.82	0.60	-	2.0B
RAR [44]	1.48	-	326.0	0.80	0.63	-	1.5B
MAR [45]	1.55	-	303.7	0.81	0.62	-	943M
DiT-XL/2 [46]	2.27	4.60	278.2	0.83	0.57	59.30	675M
+MG$_{\text{(ours)}}$	**1.78**	4.46	298.5	0.80	0.66	29.65	675M
Improvement	21.6%	3.04%	7.30%	−3.6%	15.8%	50.0%	0.0%
SiT-XL/2 [47]	2.06	4.49	277.5	0.83	0.59	59.30	675M
+MG$_{\text{(ours)}}$	**1.34**	4.58	321.5	0.81	0.65	29.65	675M
Improvement	35.0%	−2.0%	15.9%	−2.4%	10.2%	50.0%	0.0%

sample $w \sim \mathbf{U}[w_{\text{min}}, w_{\text{max}}]$ *per batch* from a specified interval and use it both as the network input and as the weight in the MG target, modifying the loss to

$$\mathcal{L}_{\text{MG}} = \mathbb{E}_{w,t,(x_0,c),\epsilon} \|\epsilon_\theta(x_t, t, c, w) - \epsilon'\|^2, \tag{20}$$

$$\epsilon' = \epsilon + w \cdot \text{sg}(\epsilon_\theta(x_t, t, c, 0) - \epsilon_\theta(x_t, t, \varnothing, -1)). \tag{21}$$

After training, a *single* scale-aware model can sample with any $w \in [w_{\text{min}}, w_{\text{max}}]$ specified at inference, controlling the quality and diversity trade-off without retraining. Inference still requires only one forward per denoising step, since both the conditional prediction and the chosen w are passed to the model in a single call.

Automatic adjustment of w. While w in eqs. (15) and (17) plays an important role in balancing the posterior probability, manually searching for it during training is tedious and costly. We therefore introduce an automatic adjustment scheme. We begin with w=0, corresponding to vanilla diffusion training in eq. (3), and update w according to intermediate evaluations: w is raised when quality continues to improve and suppressed otherwise, converging to an optimum. We provide full details and pseudo-code in algorithm 2 and a robustness study in section F.1.

4 Experiment

We first present a system comparison on ImageNet 256×256 conditional generation, then conduct ablation on our detailed design choices. Specifically, we focus on the following questions:

Table 3: Experiments on model size.

Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-B/2	43.5	36.7	39.23	0.62	0.34
+CFG	9.67	9.14	160.6	0.79	0.36
+MG$_{(ours)}$	**7.24**	5.56	189.2	0.84	0.38
DiT-L/2	23.3	18.4	132.7	0.73	0.40
+CFG	6.76	7.10	203.4	0.78	0.38
+MG$_{(ours)}$	**5.43**	4.66	236.3	0.83	0.44
DiT-XL/2	19.5	15.6	163.5	0.79	0.46
+CFG	4.16	5.23	233.0	0.82	0.49
+MG$_{(ours)}$	**3.37**	4.73	257.2	0.84	0.51
SiT-B/2	33.0	27.8	65.24	0.68	0.35
+CFG	8.35	8.63	173.0	0.78	0.37
+MG$_{(ours)}$	**6.49**	5.69	212.3	0.86	0.38
SiT-L/2	18.9	16.3	173.2	0.71	0.42
+CFG	5.77	6.46	223.8	0.79	0.44
+MG$_{(ours)}$	**4.50**	4.03	243.9	0.85	0.46
SiT-XL/2	17.3	13.9	192.1	0.78	0.50
+CFG	3.47	4.21	247.9	0.83	0.53
+MG$_{(ours)}$	**2.89**	3.12	261.0	0.85	0.54

Table 4: Experiments on ImageNet 512.

Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-XL/2	12.0	7.12	105.3	0.75	0.64
+CFG	3.04	5.02	240.8	0.84	0.54
+MG$_{(ours)}$	**2.78**	4.86	257.2	0.83	0.58
SiT-XL/2	9.64	6.03	124.4	0.77	0.65
+CFG	2.62	4.18	252.2	0.84	0.57
+MG$_{(ours)}$	**2.24**	4.03	276.9	0.86	0.60
EDM2$_{XXL}$	1.91	4.65	226.7	0.80	0.68
+CFG	1.81	4.24	273.2	0.85	0.63
+MG$_{(ours)}$	**1.43**	4.03	299.1	0.84	0.64

Table 5: Experiments on text-to-image models.

Model	FID↓ (w/o CFG)	FID↓ (w/ CFG)
GLIDE [7]	-	12.32
LDM [8]	-	12.58
DALL·E 2 [48]	-	10.46
SD 1.5 [8]	23.11	8.32
+MG$_{(ours)}$	**7.86**	-

Table 6: Experiments on scale w.

Model	w	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-B/2	0.00	43.5	36.7	39.23	0.62	0.34
+CFG	2.00	9.67	9.14	160.6	0.79	0.36
+MG$_{(ours)}$	0.25	9.86	8.87	176.1	0.81	0.37
+MG$_{(ours)}$	0.50	**7.24**	5.56	189.2	0.84	0.38
+MG$_{(ours)}$	0.75	8.21	6.63	197.2	0.86	0.38
+MG$_{(ours)}$	1.00	9.66	7.90	224.7	0.85	0.39
+MG$_{(ours)}$	Auto	7.60	6.29	192.4	0.85	0.38
SiT-B/2	0.00	33.0	27.8	65.24	0.68	0.35
+CFG	2.00	8.35	8.63	173.0	0.78	0.37
+MG$_{(ours)}$	0.25	8.94	7.87	194.3	0.83	0.38
+MG$_{(ours)}$	0.50	**6.49**	5.69	212.3	0.86	0.38
+MG$_{(ours)}$	0.75	8.03	6.91	221.0	0.86	0.39
+MG$_{(ours)}$	1.00	9.14	7.99	236.7	0.88	0.40
+MG$_{(ours)}$	Auto	6.86	5.88	219.1	0.87	0.38

Table 7: Experiments on scale-aware networks.

Model	w-in	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-B/2	✗	43.5	36.7	39.23	0.62	0.34
+CFG	✗	9.67	9.14	160.6	0.79	0.36
+MG$_{(ours)}$	✗	**7.24**	5.56	189.2	0.84	0.38
+MG$_{(ours)}$	✓	8.13	6.03	175.1	0.84	0.39
SiT-B/2	✗	33.0	27.8	65.24	0.68	0.35
+CFG	✗	8.35	8.63	173.0	0.78	0.37
+MG$_{(ours)}$	✗	**6.49**	5.69	212.3	0.86	0.38
+MG$_{(ours)}$	✓	7.33	5.96	207.4	0.85	0.38

Table 8: Experiments on drop ratio λ.

Model	λ	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-B/2	0.10	43.5	36.7	39.23	0.62	0.34
+CFG	0.10	9.67	9.14	160.6	0.79	0.36
+MG$_{(ours)}$	0.05	11.7	9.90	156.7	0.78	0.33
+MG$_{(ours)}$	0.10	**7.24**	5.56	189.2	0.84	0.38
+MG$_{(ours)}$	0.15	7.62	5.99	183.4	0.83	0.38
+MG$_{(ours)}$	0.20	9.01	7.04	171.7	0.81	0.36
SiT-B/2	0.10	33.0	27.8	65.24	0.68	0.35
+CFG	0.10	8.35	8.63	173.0	0.78	0.37
+MG$_{(ours)}$	0.05	10.8	9.25	168.8	0.80	0.34
+MG$_{(ours)}$	0.10	**6.49**	5.69	212.3	0.86	0.38
+MG$_{(ours)}$	0.15	6.77	5.89	207.4	0.85	0.37
+MG$_{(ours)}$	0.20	8.87	8.06	199.6	0.84	0.37

- The generation quality of MG on existing diffusion models. (tables 1 and 2, section 4.2)
- Scaling to larger models, more datasets, and different tasks? (tables 3 to 5, section 4.3)
- Details of implementation, flexibility, and efficiency. (tables 6 to 8, figs. 4 and 5, section 4.4)

4.1 Setup

Implementation and dataset. We follow the experiment pipelines in DiT [46] and SiT [47]. We use ImageNet [22, 23] dataset and the Stable Diffusion [8] VAE to encode 256×256 images into the latent space of $\mathbb{R}^{32\times32\times4}$. We conduct ablation experiments with the B/2 variant of DiT and SiT models and train for 400K iterations. We train from scratch with AdamW [49, 50] optimizer and a batch size of 256, in consistent with DiT and SiT for fair comparisons. For inference, we use 250 sampling steps for both DiT and SiT models and Euler-Maruyama sampler for SiT. Full details are reported in appendix E, and we also provide the reproduction code in appendix A.

Baseline Models. We compare with several state-of-the-art image generation models, which form a strong baseline and can be classified into the following three classes: (a) *Pixel-space diffusion*: ADM [6], VDM++ [38]; (b) *Latent-space diffusion*: LDM [8], U-ViT [39], MDTv2 [40], REPA [41], LightningDiT [42], DiT [46], SiT [47]; (c) *Auto-regressive models*: VAR [43], RAR [44], MAR [45].

Evaluation metrics. We report the commonly used Frechet inception distance (FID) [51]. In addition, we report sFID [52], Inception Score (IS) [53], Precision (Pre.), and Recall (Rec.) [54] as supplementary metrics. We also report the computation costs to generate one sample with each model in terms of FLOPS to measure the trade-off between generation quality and inference speed.

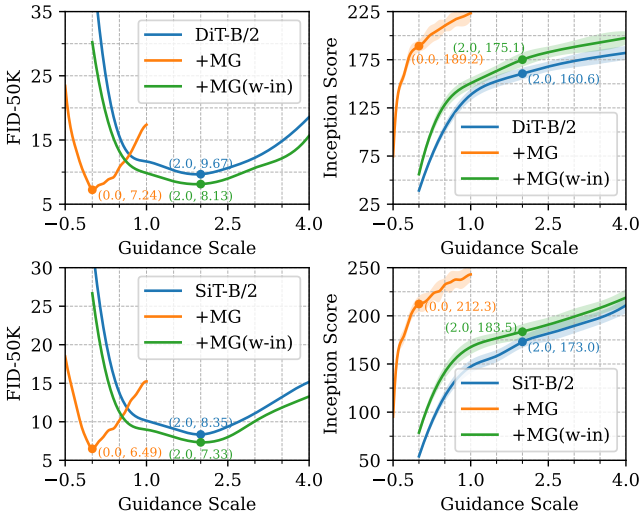


Figure 4: FID-50K and Inception Score with different scales during inference. MG also offers inference-time flexibility to control the tradeoff of quality and diversity.

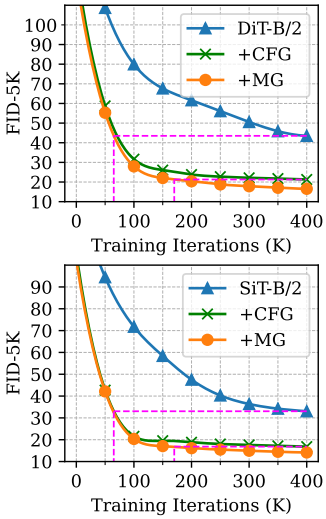


Figure 5: FID-5K results during training. MG is faster than both vanilla and CFG models.

More qualitative samples and additional metrics, including $\text{FD}_{\text{DINOv2}}$, KID, CLIP-T, and HPSv2, are provided in appendix H.

4.2 Overall Evaluation

System-level performance. We present a thorough comparison with recent state-of-the-art image generation approaches on ImageNet 256×256 dataset in tables 1 and 2. As shown in table 1, both DiT-XL/2 and SiT-XL/2 models greatly benefit from our method, achieving the performance gain of 81.5% and 84.4%. It is worth mentioning that our models do not apply modern techniques in the inference process, including rejection sampling [43], Classifier-free Guidance [3] and guidance interval [18]. Our models are also light-weight, *e.g.* 675M in contrast to RAR-XXL with 1.5B and MAR-H with 943M parameters, and consume less computational resources, for example, LightningDiT uses DiT-XL/1 to reduce patch size to 1×1 and needs $16\times$ computation in attention operations. We visualize our results in fig. 6.

Comparison with CFG. For a fair evaluation, we also compare with other methods that use Classifier-free Guidance, while it introduces an additional unconditional forward and doubles the sampling cost. CFG also typically requires a careful search over the guidance scale to achieve the best trade-off between quality and diversity. In contrast, our models still surpass other CFG-based methods.

Sampling cost. We report the computation consumption per sample in Tflops. CFG-based methods require two forwards per denoising step (one conditional and one unconditional), doubling sampling cost. MG samples with a single conditional forward and therefore halves inference compute without sacrificing quality.

4.3 Scaling Up

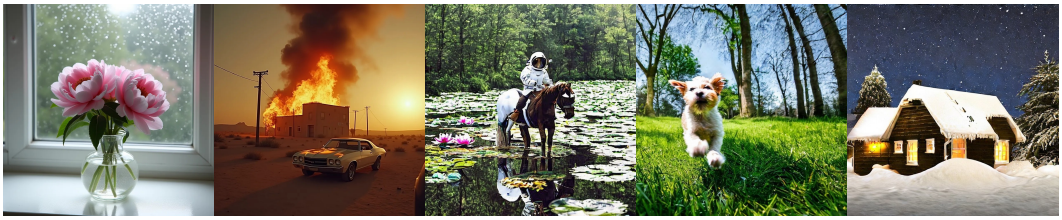
Model sizes. We study MG with different sizes of models in table 3. We train the B/2, L/2, and XL/2 variants of DiT and SiT models, and compare with both vanilla and CFG results. It demonstrates that our method is capable to boost the performance of models with different sizes and designs.

High-resolution images. Next, we apply MG on ImageNet 512×512 dataset in table 4 to validate our method in handling high resolution images and difficult distributions. In addition to DiT and SiT, we also conduct experiments with EDM2-XXL [55]. The results confirm that the generation quality is also significantly improved with MG and exceeds CFG by a remarkable margin in all cases. We also compare with Autoguidance [17] in appendix D.

Text-to-image models. Scaling MG to text-to-image models is also important. In table 5, we finetune Stable Diffusion (SD) 1.5 [8] with MG on the LAION-400M [56] dataset, and report FID of zero-shot text-to-image generation on COCO 2014 [57]. As shown, MG improves the FID of SD 1.5 from 23.11 to 7.86, even surpassing the 8.32 of CFG. This indicates that the training-objective change transfers to text conditioning (fig. 7).



Figure 6: Uncurated samples of SiT-XL/2+MG.



Elegant crystal vase holding pink peonies, soft raindrops tracing paths down the window behind it.

Burning building in the middle of a desert, a car parked outside at sunset, old movie still.

Photo of an astronaut riding a horse in the forest. There is a river in front of them with water lilies.

Under a sunny sky one puppy run freely in the park, surrounded by green grass and tall trees.

A cozy cottage rests in a snowy winter scene under a starry sky, emanating peaceful warm light.

Figure 7: Text-to-image samples of SD1.5+MG.

4.4 Ablation Study

Hyper-parameter w. The *training-time* hyperparameter w in eqs. (15) and (17) controls the strength of posterior alignment, analogous to the inference-time guidance scale in CFG. We conduct ablations with different w in table 6, where w=0 recovers vanilla diffusion. As w varies it balances the quality and diversity trade-off, and MG remains better even when w is not optimized (*e.g.* w=0.75).

Auto-tuning w. To overcome the tedious manual search of w, we propose to automatically adjust w, and obtain comparable performances with manual search in the highlighted rows in table 6.

Scale-aware networks. Whether to take w as an additional input is optional for MG. In table 7, models with w-input (orange) slightly lag behind counterparts without w-input (blue), but still exceed the quality of CFG, demonstrating the superiority of our method.

Hyper-parameter λ. The drop ratio λ is important for our method. To enable the estimation in eq. (11), we randomly drop the condition c of a fraction of training samples and replace it with the empty class $\varnothing$. In table 8, we find that $\lambda \in \{0.10, 0.15\}$ provides satisfactory performance.

Flexibility. In fig. 4, MG is also capable of sampling with different balances between quality and diversity. For MG with our proposed scale-aware networks, we can simply feed different guidance scales w as input. Otherwise, we can also wrap MG models with standard CFG, exactly as vanilla diffusion models. More details are provided in appendix G, and compatibility with other inference-time techniques such as CADS [20] is discussed in appendix G.4.

Efficiency. MG improves training and inference efficiency. With a $1.14\times$ per-iteration overhead from one extra unconditional forward, MG reaches the same FID in $1.77\times$ to $1.94\times$ less wall-clock time than CFG in fig. 5 and section F. At inference, MG requires fewer FLOPs than other methods in tables 1 and 2, on par with LightningDiT [42], which consumes $\approx 9\times$ more compute than MG.

5 Conclusion

The success of CFG reveals a training-sampling mismatch: standard diffusion training learns the unbiased conditional score, while the posterior tilt that sampling actually targets is patched at inference. This work argues the gap is better closed at the training-objective level. Model-guidance (MG) internalises CFG's posterior tilt into the training target while keeping the sampling procedure of vanilla conditional diffusion. proposition 1 shows MG remains a function-value-level reparameterisation of standard denoising score matching rather than a surrogate. Empirically, this single change yields faster convergence and better single-forward sampling across architectures and datasets, including FID 1.34 on ImageNet 256×256. More broadly, devices currently parked at inference time (guidance schedules, sampler corrections, trajectory fixes) may also benefit from being lifted into training objectives, where they can interact with optimisation rather than be patched afterwards.

Acknowledgments and Disclosure of Funding

Funding: This research received no external funding.

Competing interests: The authors declare no competing interests.

References

[1] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[2] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[7] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn 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.

[9] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[10] Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[12] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[13] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pages 393–411. Springer, 2025.

[14] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

[15] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, pages 1–20, 2024.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[17] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in neural information processing systems*, 2024.

[18] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in neural information processing systems*, 2024.

[19] Candi Zheng and Yuan Lan. Characteristic guidance: Non-linear correction for diffusion model at large guidance scale. In *International Conference on Machine Learning*. PMLR, 2024.

[20] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[25] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[26] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[27] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

[28] Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128 (1):153–179, 1997.

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

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

[31] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[32] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018. URL `https://api.semanticscholar.org/CorpusID:54482423`.

[34] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[35] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. URL `https://api.semanticscholar.org/CorpusID:246441861`.

[36] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[37] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

[38] Diederik P Kingma and Ruiqi Gao. Understanding the diffusion objective as a weighted integral of elbos. *arXiv preprint arXiv:2303.00848*, 2, 2023.

[39] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.

[40] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23164–23173, 2023.

[41] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[42] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

[43] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 2024.

[44] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024.

[45] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in neural information processing systems*, 2024.

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

[47] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[49] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[50] I Loshchilov. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[52] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, pages 7958–7968. PMLR, 2021.

[53] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[54] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

[55] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.

[56] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[57] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[58] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023.

[59] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[61] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

A Reproduction code

Training and evaluation code for SiT-XL/2+MG is publicly available at `https://github.com/tzco/Diffusion-wo-CFG`. The repository README provides environment setup and execution instructions, and `download.sh` downloads the released model checkpoints and FID reference statistics for reproducing the SiT-XL/2+MG results with FID of 1.34 in tables 1 and 2. The scripts `train.py` and `test.py` implement training and evaluation, respectively. The README also notes that changing the number of GPU ranks can affect evaluation results through rank-dependent random seeds.

B Limitations and Broader Impacts

Limitations MG removes the inference-time CFG sampling rule but does not eliminate every CFG-related design. During training, condition dropping and an unconditional forward are still used to estimate the posterior-alignment signal, and incur a $1.14\times$ per-iteration wall-time overhead (section F). The guidance weight w also remains a hyperparameter, now applied at training time. It is automated by algorithm 2 and shown robust to its sub-hyperparameters in section F.1, but supervising posterior alignment without an explicit unconditional branch (e.g. via implicit conditioning regularisation) remains an open direction. The method also still relies on existing encoders and decoders [8]. Concurrent end-to-end approaches [42] suggest a natural next step that integrates encoder/decoder training with the MG objective.

Broader Impacts While this work is mainly conducted on image generation diffusion models, autoregressive (AR) models and generative models on other modalities also employ CFG to improve performance. Extending this work to AR models and other modalities is also promising. On the negative side, our method learns from its training data and could therefore mirror any biases present. The image generation capabilities in this work might also be misused to spread false information, and we will limit the release of our model weights.

C Compare MG with Distillation

C.1 Offline Distillation

Table 9: Comparison between MG and Guided-Distillation.

Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-XL/2	9.62	6.85	121.5	0.67	0.67
+CFG	2.27	4.60	278.2	0.83	0.57
+GD	2.31	4.59	282.3	0.82	0.59
+MG$_{\text{(ours)}}$	**1.78**	4.46	298.5	0.80	0.66
SiT-XL/2	8.61	6.32	131.7	0.68	0.67
+CFG	2.06	4.49	277.5	0.83	0.59
+GD	2.09	4.67	275.4	0.81	0.56
+MG$_{\text{(ours)}}$	**1.34**	4.58	321.5	0.81	0.65

We empirically compare with two-stage Guided Distillation (GD) [21] in table 9. GD trains a student against a fixed CFG-corrected teacher and is upper-bounded by that surrogate. In our experiments GD slightly under-performs the original CFG model, whereas MG, which is anchored by the data score rather than a frozen surrogate (proposition 1), yields significant improvements. GD also incurs higher training cost from its two-stage nature.

C.2 Online Distillation

A natural baseline is to keep GD's distillation loss form (eqs. (22) and (23)) but apply it online during pre-training, substituting the fixed teacher with online predictions. We discuss two such online-distillation variants below. Both differ from MG in supervision structure and underperform it

in table 10.

$$\mathcal{L}_{\text{GD}} = \mathbb{E}_{w,t,(x_0,c),\epsilon} \left\| \epsilon_\eta(x_t, t, c, w) - \epsilon_\theta^w(x_t, t, c) \right\|^2, \tag{22}$$

$$\epsilon_\theta^w(x_t, t, c) = \epsilon_\theta(x_t, t, c) + w(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing)), \tag{23}$$

However, the student model $\epsilon_\eta(x_t, t, c, w)$ additionally consumes w as input, which is different from the teacher model $\epsilon_\theta(x_t, t, c)$. Here, we thoroughly discuss both options.

Case 1: We abandon the additional input w and train $\epsilon_\theta(x_t, t, c)$. In this case, the training loss is

$$\begin{aligned}\mathcal{L}_1 =& \mathbb{E}_{w,t,(x_0,c),\epsilon} \Bigg[\underbrace{\left\| \epsilon_\theta(x_t, t, c) - \epsilon \right\|^2}_{\text{first term}} \\ &+ \underbrace{\left\| \epsilon_\theta(x_t, t, c) - \text{sg}(\epsilon_\theta(x_t, t, c) + w(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing))) \right\|^2}_{\text{second term}} \Bigg].\end{aligned} \tag{24}$$

With $\mathcal{L}_1$, the model is confused by multiple update directions. The first term in eq. (24) directs the model to predict the ground-truth noise ϵ, while the second term in eq. (24) instructs the model to match the prediction modified by CFG. On the other hand, MG in eqs. (14) and (15) has the unified prediction target and update direction, leading to faster convergence and better performance.

Case 2: We adopt the form $\epsilon_\theta(x_t, t, c, w)$ and feed with the scale w. In this case, the training loss is

$$\begin{aligned}\mathcal{L}_2 =& \mathbb{E}_{w,t,(x_0,c),\epsilon} \Bigg[\underbrace{\left\| \epsilon_\theta(x_t, t, c, 0) - \epsilon \right\|^2}_{\text{first term}} \\ &+ \underbrace{\left\| \epsilon_\theta(x_t, t, c, w) - \text{sg}(\epsilon_\theta(x_t, t, c, 0) + w(\epsilon_\theta(x_t, t, c, 0) - \epsilon_\theta(x_t, t, \varnothing, -1))) \right\|^2}_{\text{second term}} \Bigg].\end{aligned} \tag{25}$$

With $\mathcal{L}_2$, the model is pretrained by the first term and distilled by the second term separately. The student $\epsilon_\theta(x_t, t, c, w)$ is supervised against another model output $\epsilon_\theta(x_t, t, c, 0)$, reintroducing the same surrogate-target issue as two-staged approaches. MG instead keeps a single conditional prediction and supervises it against the data score (proposition 1).

Table 10: Comparison between MG and online distillation.

Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
DiT-B/2	43.5	36.7	39.23	0.62	0.34
+CFG	9.67	9.14	160.6	0.79	0.36
+$\mathcal{L}_1$	11.3	12.4	147.9	0.74	0.34
+$\mathcal{L}_2$	10.1	10.9	157.1	0.78	0.36
+MG$_{\text{(ours)}}$	**7.24**	5.56	189.2	0.84	0.38
SiT-B/2	33.0	27.8	65.24	0.68	0.35
+CFG	8.35	8.63	173.0	0.78	0.37
+$\mathcal{L}_1$	9.76	10.7	154.3	0.72	0.35
+$\mathcal{L}_2$	8.38	8.94	169.6	0.76	0.36
+MG$_{\text{(ours)}}$	**6.49**	5.69	212.3	0.86	0.38

Experimental comparison. We additionally conduct experiments with the loss functions in eqs. (24) and (25) and compare with MG in table 10. As depicted, eq. (24) obtains the worst results, while MG outperforms both eqs. (24) and (25).

D Compare MG with Autoguidance

Autoguidance [17] is a close prior work to MG: both replace the explicit unconditional branch of CFG with a guidance signal derived from the model itself, rather than from an externally-defined empty class. The key differences are summarized below.

- **Single model vs. two models.** Autoguidance requires training and storing a separate "bad" model (a smaller or less-trained copy of the same architecture) to provide the guidance signal. MG uses a single model whose own conditional and unconditional predictions provide the guidance signal during training, with no extra model to maintain.
- **Inference cost.** Autoguidance still requires two model evaluations per denoising step at inference (the main model and the bad model). MG requires only one model evaluation per denoising step, halving the inference compute compared with both CFG and Autoguidance.
- **Where guidance is applied.** Autoguidance modifies only the inference procedure on top of a standard pre-trained model. MG modifies the training objective so that a single conditional prediction approximates a CFG-guided score, and the inference procedure is identical to vanilla diffusion sampling.

On ImageNet 512×512 with EDM2-XXL, Autoguidance reports FID 1.25 while MG reaches FID 1.43 in table 4. However, the inference cost of Autoguidance is still approximately $2\times$ in favour of MG. We view the two methods as largely orthogonal: Autoguidance is an inference-time technique on top of an existing model, while MG is a training-time objective.

E Hyperparameter and Implementation Details

Table 11: Setup for tables 1 and 2.

	DiT-XL/2	SiT-XL/2
Architecture		
Input dim.	32×32×4	32×32×4
Num. layers	28	28
Hidden dim.	1,152	1,152
Num. heads	16	16
Hyperparameters		
w	0.3	0.45
λ	0.1	0.1
Scale-aware	✗	✗
Optimization		
Prediction	ϵ	v
Batch size	256	256
Optimizer	AdamW	AdamW
lr	0.0001	0.0001
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)
Inference		
Sampler	ADM [6]	Euler-Maruyama
Steps	250	250

Implementations. We implement our method based on the code of DiT [46] and SiT [47, 41]. Throughout the experiments, we use the exact same structure as DiT [46] and SiT [47], and add an embedding for scale-aware networks that sums to the condition and timestep embedding. We use AdamW [49, 50] with constant learning rate of 1e-4, $(\beta_1, \beta_2) = (0.9, 0.999)$ without weight decay. We also pre-compute and save the latent vectors of images and use these latent vectors for training. As consequences, we only apply simple random horizontal flip as data augmentation. We use `stabilityai/sd-vae-ft-ema` for encoding and decoding images. The detailed hyper-parameter setup are provided in table 11.

Sampler. For DiT, we use the ADM [6] sampler with 250 steps the same as the original DiT paper [46]. For SiT, we use the Euler-Maruyama sampler with 250 steps the same as the original SiT paper [47].

Computing resources. We use 16x NVIDIA A100 40GB GPUs for experiments. We use a batch size of 256 and remain unchanged for all experiments.

Algorithm 2 Training with Model-guidance Loss: PyTorch-like Pseudo-code

```
def train(w=0.0, beta_1=0.05, beta_2=0.05, auto_adjust=False,
          drop_ratio=0.1, empty_label=0):
    # w is the guidance scale in Eq. 15 of the main text;
    # w=0 corresponds to vanilla diffusion training.
    # drop_ratio = lambda; empty_label is the index of the empty class.
    for step, (x, c) in enumerate(dataloader):
        # sample random noise and timestep
        noise = torch.randn(x.shape)
        timestep = torch.randint(0, diffusion.num_timesteps, x.size(0))

        # sample x_t from x
        x_t = diffusion.q_sample(x, timestep, noise)

        # randomly drop labels with ratio lambda
        z = c.clone()
        z[torch.randperm(len(z))[:int(len(z) * drop_ratio)]] = empty_label

        # predict conditional noise from x_t
        noise_pred = net(x_t, timestep, z)

        # compute MG learning target (Eq. 15 of the main text)
        with torch.no_grad():
            pred_w_cond = noise_pred.detach()
            # one additional unconditional forward, no gradient needed
            pred_wo_cond = net(x_t, timestep, torch.full_like(z, empty_label))
            target = noise + w * (pred_w_cond - pred_wo_cond)

        # compute loss and optimize
        loss = ((noise_pred - target) ** 2).mean()
        loss.backward()
        opt.step()
        opt.zero_grad()

        # automatically adjust w (additive update on w)
        if auto_adjust and step % freq_eval == 0:
            fid_eval = evaluate(net)
            # raise w when FID decreases (quality improves), suppress otherwise
            if fid_eval < history_fid:
                w = w + beta_1
            else:
                w = max(w - beta_1, 0.0)
            history_fid = history_fid * (1 - beta_2) + fid_eval * beta_2
```

Pseudo-code. We provide a torch-like pseudo-code of training models with MG in algorithm 2.

F Training Wall-time

Table 12: Training speed-up comparison on DiT-B/2.

	MG *vs.* vanilla DiT	MG *vs.* DiT+CFG
Training speedup (iterations to same FID)	6.50×	2.07×
Training speedup (wall-time to same FID)	5.71×	1.77×
Performance gain (same iterations)	62%	34%
Performance gain (same wall time)	57%	24%

Table 13: Training speed-up comparison on SiT-B/2.

	MG *vs.* vanilla SiT	MG *vs.* SiT+CFG
Training speedup (iterations to same FID)	7.04×	2.11×
Training speedup (wall-time to same FID)	6.18×	1.94×
Performance gain (same iterations)	57%	23%
Performance gain (same wall time)	53%	20%

In addition to fig. 5, we also report the training speed-up in terms of both iterations and wall-time in tables 12 and 13. When comparing with vanilla diffusion models, MG achieves $\geq 5.7\times$ training speed-up and $\geq 50\%$ performance gains. When comparing diffusion models using CFG sampling, MG still yields $\geq 1.7\times$ training speed-up and $\geq 20\%$ performance gains, whether in terms of training iterations or wall-time.

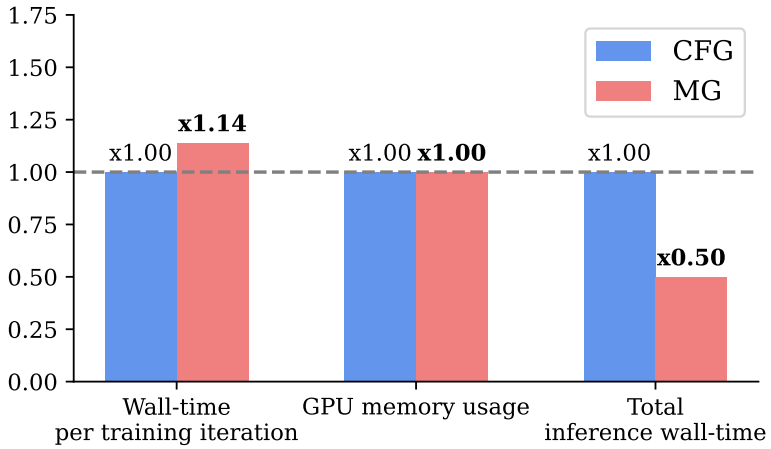


Figure 8: Comparing the computational efficiency of CFG and MG.

We also compare the wall-time per training iteration, GPU memory usage during training, and the total inference time to generate one image in fig. 8. Since MG requires only one additional forward of the online model during training (see algorithm 1), it significantly boosts convergence speed, improves performance, and reduce inference time by half at the cost of a mere overhead of $0.14\times$.

To clarify how MG only incurs a $0.14\times$ per-iteration overhead due to one additional unconditional forward, we report a breakdown of the per-iteration cost in table 14. The additional unconditional forward is executed under `torch.no_grad()` (see algorithm 2), which avoids storing intermediate activations and therefore costs substantially less than a full forward-backward pass.

Table 14: Per-iteration wall-time breakdown on DiT-B/2.

Operation	Time per iter (ms)	Relative to vanilla
Vanilla DiT-B/2 (forward + backward)	336	1.00×
MG: conditional forward (with grad)	97	28.9%
MG: unconditional forward (no grad)	48	14.3%
MG: backward (conditional branch only)	178	53.0%
MG total per iter	383	**1.14**×

Table 15: Comparison of total inference time in seconds to generate one sample. *CFG (separate)* means separately predicting conditional and unconditional scores during inference, and *CFG (parallel)* refers to concatenate conditional and unconditional predictions in batch dimension and forward once.

Batch size	vanilla SiT	SiT+CFG (separate)	SiT+CFG (parallel)	SiT+MG
32	1.42	2.86	2.86	**1.42**
64	1.37	2.76	2.75	**1.37**
128	1.33	2.67	2.64	**1.33**
256	1.30	2.61	Out of Memory	**1.30**

Note that while modern implementations of CFG concatenate conditional and unconditional predictions in the batch dimension and run a single forward pass with doubled batch size during inference,

their inference time is still proportional to TFLOPs in tables 1 and 2 and therefore doubled compared with MG. In table 15, we test and compare different implementations of CFG against MG on SiT-B/2 with one A100 80GB GPU at varying batch sizes. MG is approximately $2\times$ faster than CFG even with parallel implementation and sufficient GPU memory.

Finally, we report the computational overhead of the proposed automatic adjustment algorithm for the hyper-parameter w. As in algorithm 2, we evaluate FID-1K as an intermediate evaluation metric to adjust w, which introduces an inference process of 1 minutes per evaluation on DiT-B/2 and SiT-B/2 models. We evaluate and adjust every 20K iterations, which corresponds to an overhead of 20 minutes and $\approx 1.4\%$ of total training time.

F.1 Robustness of Auto-adjustment Hyper-parameters β_1, β_2

The auto-adjustment scheme of algorithm 2 introduces two additional hyper-parameters: β_1 controls the additive update step on w, and β_2 controls the smoothing of the running FID baseline. To verify that the scheme is not sensitive to these hyper-parameters, we sweep $\beta_1, \beta_2 \in \{0.01, 0.05, 0.1, 0.2\}$ on DiT-B/2 with auto-adjustment, keeping all other settings fixed. table 16 reports the resulting FID. Across the entire 4×4 grid the variance of final FID is small (< 0.3), and all configurations outperform the corresponding CFG baseline, indicating that the auto-adjustment scheme is robust to the choice of β_1 and β_2. We use the default $(\beta_1, \beta_2) = (0.05, 0.05)$ throughout the main experiments.

Table 16: Auto-adjustment robustness on DiT-B/2: final FID under different (β_1, β_2) choices.

	$\beta_2 = 0.01$	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$
$\beta_1 = 0.01$	7.44	7.30	7.35	7.50
$\beta_1 = 0.05$	7.39	**7.24**	7.33	7.42
$\beta_1 = 0.10$	7.35	7.30	7.28	7.43
$\beta_1 = 0.20$	7.39	7.40	7.47	7.52
DiT-B/2 + CFG baseline (as reference): 9.67				

G Inference-time Flexibility

G.1 Scale-Aware Networks

MG is also capable of sampling with different balances between quality and diversity. When implemented with the proposed scale-aware networks, we can simply change the input scale w. MG obtains better quality and diversity than CFG, as shown by the lower FID and higher IS scores across all guidance scale than CFG in fig. 4.

G.2 Applying CFG on MG models

While MG itself achieves better results in most cases, MG still offers inference-time flexibility even without scale-aware networks by applying standard CFG on top. Specifically, MG models ϵ_θ are fully compatible with the original CFG framework: we query the network twice to obtain $\epsilon_\theta(x_t, t, c)$ with condition and $\epsilon_\theta(x_t, t, \varnothing)$ with the empty class, then compute the final prediction as $\tilde{\epsilon}_\theta(x_t, t, c) = \epsilon_\theta(x_t, t, c) + w'(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing))$. Both $w' > 0$ and $w' < 0$ are valid options.

G.3 Proof of proposition 1: Population Fixed Point of MG

This section gives a careful population fixed-point treatment of proposition 1. The stop-gradient operator preserves forward values but changes gradient routing, so the rigorous statement is a fixed-point identity rather than an equality of gradient fields.

Setup. For compactness, write $f^c_\theta(x_t, t, c) \coloneqq \epsilon_\theta(x_t, t, c)$ and $f^u_\theta(x_t, t) \coloneqq \epsilon_\theta(x_t, t, \varnothing)$. The conditional MG loss is given by eqs. (14) and (15). With condition dropping at rate $\lambda > 0$, when $c = \varnothing$ the conditional and unconditional predictions coincide ($f^c_\theta(x_t, t, \varnothing) = f^u_\theta(x_t, t)$),

the MG correction term vanishes, and the dropped-condition contribution reduces to the standard unconditional denoising loss

$$\mathcal{L}^u_{\text{MG}}(\theta) = \mathbb{E}_{x_0,t,\epsilon}\big[\, \big\| f^u_\theta(x_t,t) - \epsilon \big\|_2^2 \,\big]. \tag{26}$$

We analyse the population objective under unlimited model capacity and exact population optimisation.

Theorem (Population Fixed Point of MG). Assume $0 \le w < 1$, $\lambda > 0$, unlimited capacity, and exact population optimisation. Define $\epsilon^\star_c(x_t,t,c) \coloneqq \mathbb{E}[\epsilon \mid x_t,t,c]$ and $\epsilon^\star_u(x_t,t) \coloneqq \mathbb{E}[\epsilon \mid x_t,t]$, which by the standard denoising-score identity satisfy $\epsilon^\star_c = -\sigma_t \nabla_{x_t} \log p_t(x_t \mid c)$ and $\epsilon^\star_u = -\sigma_t \nabla_{x_t} \log p_t(x_t)$. Then at any population fixed point $\theta^\star$ of the MG objective:

$$f^u_{\theta^\star}(x_t,t) = \epsilon^\star_u(x_t,t), \tag{27}$$

$$(1-w) f^c_{\theta^\star}(x_t,t,c) + w\, f^u_{\theta^\star}(x_t,t) = \epsilon^\star_c(x_t,t,c). \tag{28}$$

Step 1: The unconditional branch is anchored by condition dropout. A fraction $\lambda > 0$ of training examples have $c = \varnothing$, on which $f^c_\theta = f^u_\theta$ identically and the MG correction term in eq. (15) vanishes. The dropped-condition contribution to the total objective is therefore exactly eq. (26). Population optimality of squared-error regression gives

$$f^u_{\theta^\star}(x_t,t) = \mathbb{E}[\epsilon \mid x_t,t] = \epsilon^\star_u(x_t,t) = -\sigma_t \nabla_{x_t} \log p_t(x_t), \tag{29}$$

proving eq. (27).

Step 2: The conditional MG loss has an implicit denoising predictor. For non-empty c, the residual inside the conditional MG loss is

$$f^c_\theta(x_t,t,c) - \epsilon - w \cdot \text{sg}\big(f^c_\theta(x_t,t,c) - f^u_\theta(x_t,t)\big). \tag{30}$$

The stop-gradient changes gradient routing but not the forward value. At the function-value level,

$$\begin{aligned} & f^c_\theta - \epsilon - w \cdot \text{sg}\big(f^c_\theta - f^u_\theta\big) \\ &= f^c_\theta - \epsilon - w\big(f^c_\theta - f^u_\theta\big) \\ &= (1-w) f^c_\theta + w\, f^u_\theta - \epsilon. \end{aligned} \tag{31}$$

Defining the implicit denoising predictor

$$\hat{\epsilon}_\theta(x_t,t,c) \coloneqq (1-w) f^c_\theta(x_t,t,c) + w\, f^u_\theta(x_t,t), \tag{32}$$

the conditional MG loss equals the standard denoising loss applied to $\hat{\epsilon}_\theta$:

$$\mathcal{L}^c_{\text{MG}}(\theta) = \mathbb{E}_{x_0,c,t,\epsilon}\big[\, \|\hat{\epsilon}_\theta(x_t,t,c) - \epsilon\|_2^2 \,\big]. \tag{33}$$

Population optimality of squared-error regression then gives $\hat{\epsilon}_{\theta^\star} = \epsilon^\star_c$, which proves eq. (28).

Step 3: Effective guidance scale of MG. Combining eqs. (27) and (28) and using $w < 1$, we solve for $f^c_{\theta^\star}$:

$$f^c_{\theta^\star}(x_t,t,c) = \frac{\epsilon^\star_c(x_t,t,c) - w\, \epsilon^\star_u(x_t,t)}{1-w} = \epsilon^\star_c(x_t,t,c) + \frac{w}{1-w}\big(\epsilon^\star_c - \epsilon^\star_u\big). \tag{34}$$

This is exactly the CFG form $\epsilon_{\text{CFG}} = \epsilon^\star_c + s\big(\epsilon^\star_c - \epsilon^\star_u\big)$ with effective guidance scale

$$s_{\text{eff}} = \frac{w}{1-w}. \tag{35}$$

The MG conditional branch is therefore not the vanilla conditional predictor: it is a single-forward predictor that has already absorbed the CFG-style guidance direction.

Step 4: Reverse CFG recovers the vanilla conditional score. Standard CFG applied to an MG-trained model with scale w' gives $\tilde{\epsilon}_{\theta^\star} = f^c_{\theta^\star} + w'(f^c_{\theta^\star} - f^u_{\theta^\star})$. Setting $w' = -w$ and using eq. (28) yields

$$\tilde{\epsilon}_{\theta^\star} = (1-w) f^c_{\theta^\star} + w\, f^u_{\theta^\star} = \epsilon^\star_c(x_t,t,c) = -\sigma_t \nabla_{x_t} \log p_t(x_t \mid c), \tag{36}$$

recovering the vanilla conditional score and completing the proof. □

Remark on the stop-gradient. The stop-gradient in eqs. (14) and (15) preserves the forward value of the residual but changes which terms receive gradients. The practical semi-gradient loss and the

raw linear-combination loss therefore do *not* have identical gradient fields. The rigorous claim is that, under the population and unlimited-capacity assumptions above, the practical semi-gradient update has the same fixed point characterised by eqs. (27) and (28). At this point the conditional residual has zero population conditional mean, $\mathbb{E}[(1-w)f^c_{\theta^\star} + w\, f^u_{\theta^\star} - \epsilon \mid x_t, t, c] = 0$, and the unconditional branch is separately anchored by eq. (26). This is the precise sense in which MG acts as a function-value-level reparameterisation of standard denoising score matching. Empirical verification in table 17 confirms that the optima reached by SGD with stop-gradient indeed satisfy this property.

Remark on the case $w = 1$. The proof requires $w < 1$. When $w = 1$, eq. (28) degenerates into $f^u_\theta = \epsilon^\star_c$, which is generally incompatible with the unconditional optimum $f^u_\theta = \epsilon^\star_u$ unless $p_t(x_t \mid c) = p_t(x_t)$. The reparameterisation result is therefore stated for $0 \le w < 1$. Empirical experiments at $w = 1$ may still be reported, but they fall outside the scope of this population fixed-point theorem.

Empirical verification. To directly verify proposition 1, we apply standard CFG with $w' = -w$ at inference time on an MG-trained model and check whether FID and Recall recover to the vanilla (no-guidance) levels. table 17 shows that this is indeed the case, confirming that MG does not collapse the underlying distribution but only reparameterizes the sampling direction.

Table 17: Empirical verification of proposition 1 on SiT-XL/2. Applying standard CFG with $w' = -w$ at inference recovers vanilla FID/Recall, confirming that MG implicitly preserves the true data distribution.

Configuration	FID↓	Rec.↑
SiT-XL/2 (vanilla, no guidance)	8.61	0.67
SiT-XL/2 + MG (default $w = 0.45$)	1.34	0.65
SiT-XL/2 + MG, CFG inference $w' = -0.45$	8.56	0.66

G.4 Compatibility with Other Methods

Table 18: Compatibility of MG and CADS [20]. CADS is also applicable on MG models.

	FID↓
DiT [46]	9.62
DiT+CFG [46]	2.27
DiT+CADS [20]	1.70
DiT+MG+CADS	**1.57**

Besides the original CFG, MG is also orthogonal to and compatible with other techniques that improve DiT and SiT. As an example, we conduct ablation experiments in table 18 and show that MG is also orthogonal to and compatible with CADS [20] and CADS also enjoys benefits from MG.

H More Results

H.1 Additional Metrics

In addition to the commonly used FID and IS, we also report more evaluation metrics and results, including $\text{FD}_{\text{DINOv2}}$ [58] in and Kernel Inception Distance (KID) [59] in table 19. To assess the alignment of the generated images of MG and given conditions, we report CLIP-T [60] score in tables 19 and 20 by computing the CLIP similarity between images and the corresponding text prompts. The text prompt is "a photo of `[CLASS]`" for class-conditional image generation task on ImageNet in table 19, where `[CLASS]` is replaced by the label name. For text-to-image models, we additionally report the HPSv2 [61] results in table 20.

Table 19: FD_{DINOv2}, KID, and CLIP-T evaluation results.

	$FD_{DINOv2}\downarrow$	KID↓	CLIP-T↑
REPA [41]	58.71	0.052	0.334
LightningDiT [42]	55.04	0.050	0.325
DiT+CFG [46]	78.93	0.064	0.317
SiT+CFG [47]	70.52	0.053	0.326
DiT+MG (ours)	71.66	0.051	0.323
SiT+MG (ours)	**53.49**	**0.043**	**0.342**

Table 20: CLIP-T and HPSv2 evaluation results.

	CLIP-T↑	HPSv2↑
Stable Diffusion [8] 1.5	0.346	27.03
Stable Diffusion 1.5 +CFG	0.350	27.11
Stable Diffusion 1.5 +MG (ours)	**0.352**	**27.18**

H.2 2D Examples

We use a custom version of the visualization script in [17] to train models and plot figs. 2 and 9 to 12. Beside the 2D example in fig. 2, we also provide more detailed examples in figs. 9, 11 and 12. The default scale of CFG is $w = 2$ in figs. 2, 9, 11 and 12, and we also provide illustrations with different scales in fig. 10.

H.3 Generated Samples

We present more visualization of generated samples with our SiT-XL/2+MG model in figs. 13 to 20. We also present more text-to-image samples of our SD1.5+MG model in fig. 21.

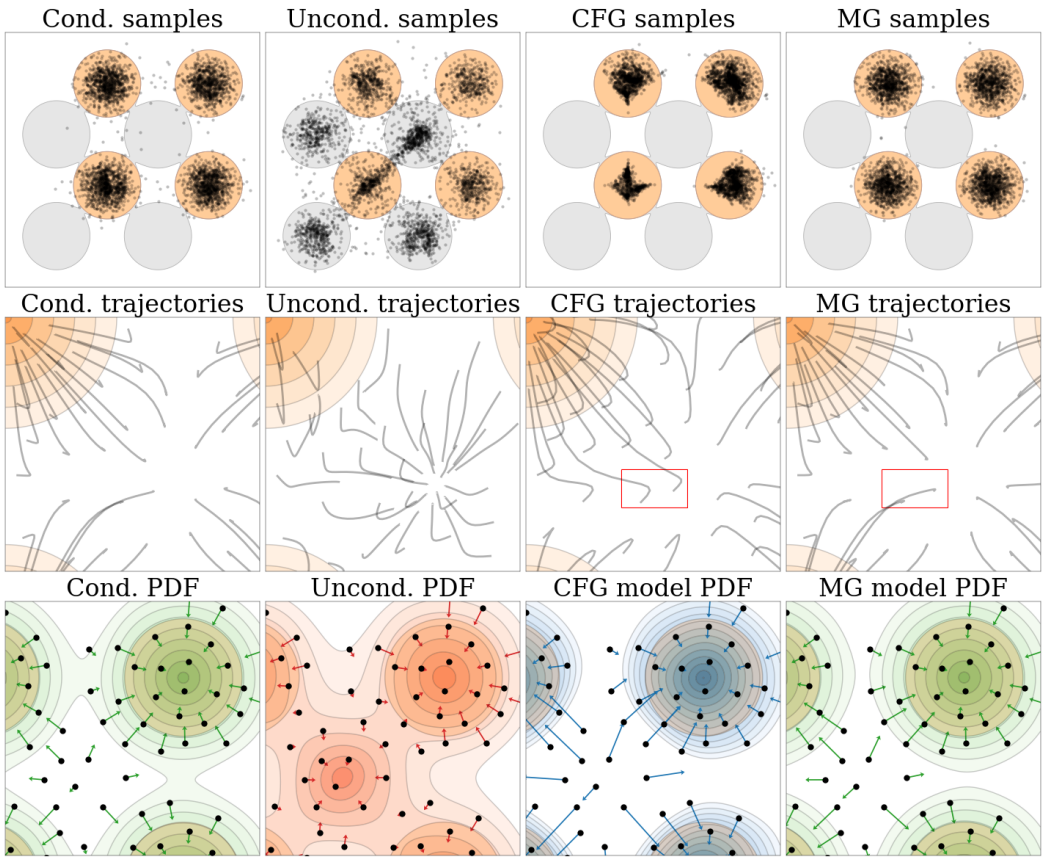


Figure 9: Diffusion models trained on a grid 2D distribution with two classes, which are marked with orange and gray colors, respectively. We plot the generated samples, trajectories, and probability density function (PDF) of conditional, unconditional, classifier-free guided model, and our approach.

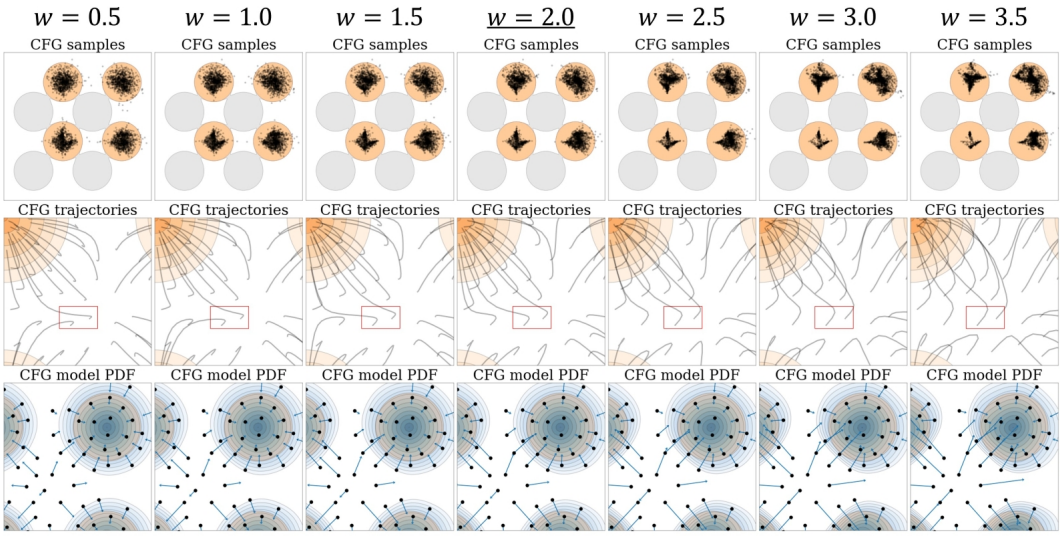


Figure 10: The generated samples, trajectories, and PDF at different guidance scales, where $w = 2$ corresponds to the CFG results in figs. 2 and 9.

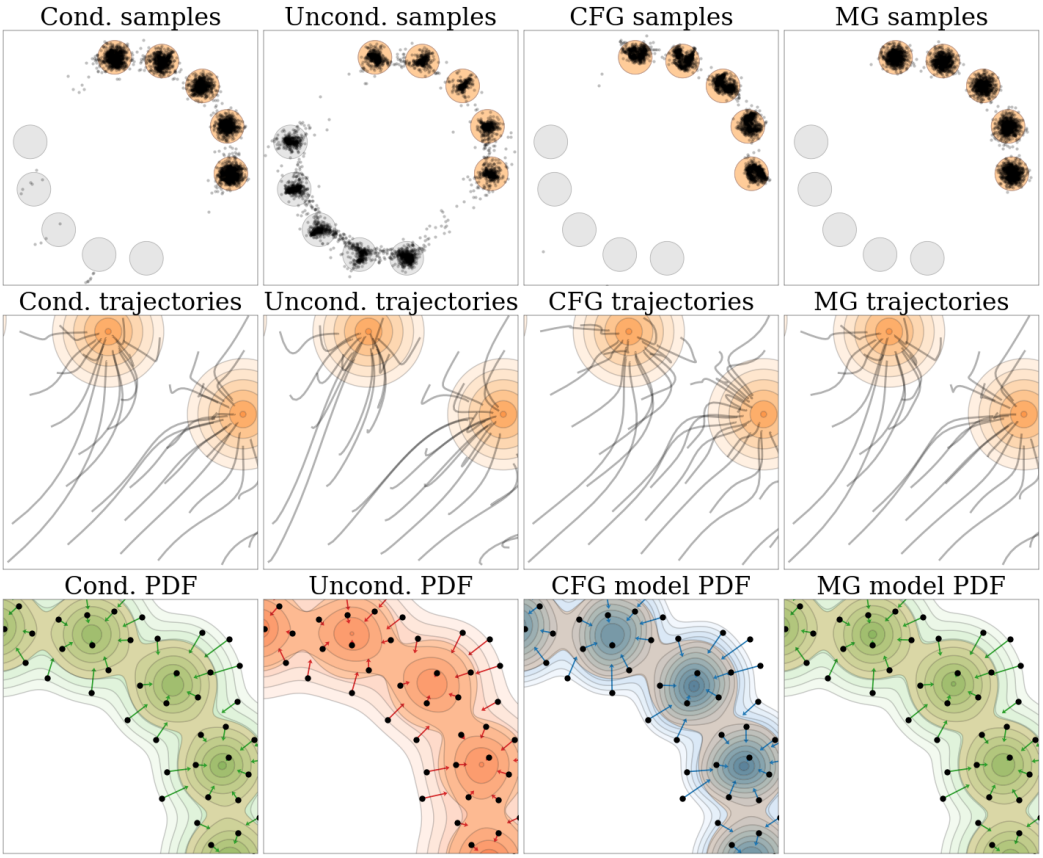


Figure 11: Diffusion models trained on a dots 2D distribution with two classes, which are marked with orange and gray colors, respectively. We plot the generated samples, trajectories, and probability density function (PDF) of conditional, unconditional, classifier-free guided model, and our approach.

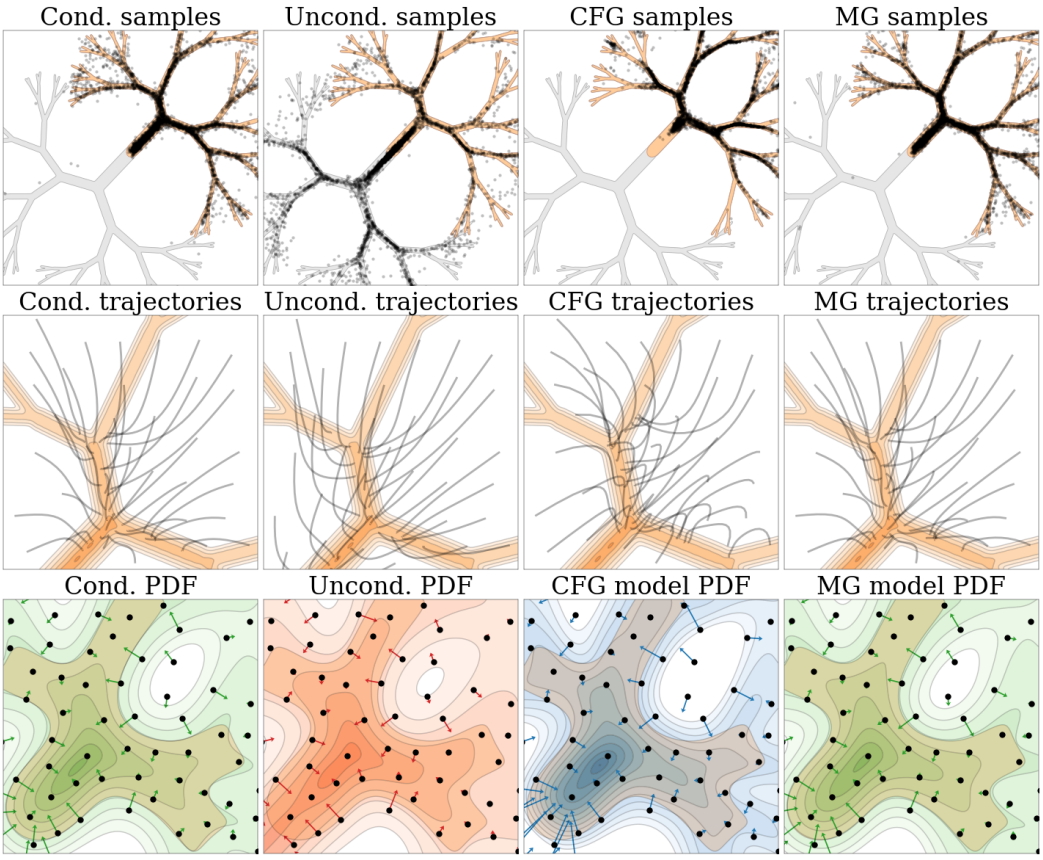


Figure 12: Diffusion models trained on a tree 2D distribution with two classes, which are marked with orange and gray colors, respectively. We plot the generated samples, trajectories, and probability density function (PDF) of conditional, unconditional, classifier-free guided model, and our approach.



Figure 13: Generated samples of SiT-XL/2+MG on the class American eagle (22).



Figure 14: Generated samples of SiT-XL/2+MG on the class macaw (88).



Figure 15: Generated samples of SiT-XL/2+MG on the class golden retriever (207).



Figure 16: Generated samples of SiT-XL/2+MG on the class lesser panda (387).



Figure 17: Generated samples of SiT-XL/2+MG on the class coral reef (973).

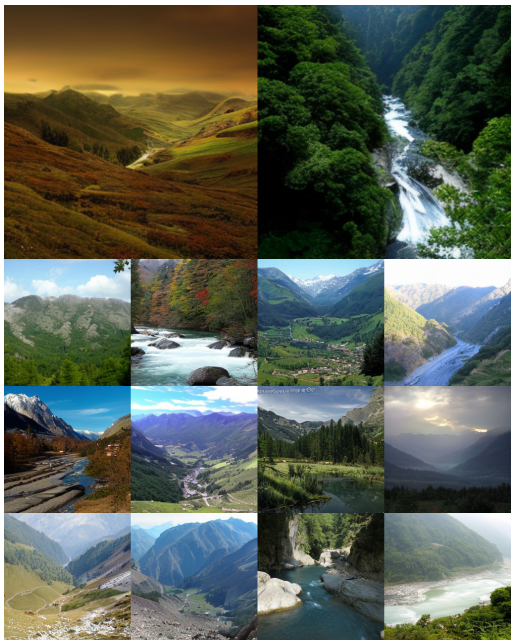


Figure 18: Generated samples of SiT-XL/2+MG on the class valley (979).



Figure 19: Generated samples of SiT-XL/2+MG on the class geyser (974).



Figure 20: Generated samples of SiT-XL/2+MG on the class volcano (980).



A hidden waterfall cascades into a crystal-clear pool within a lush, emerald grotto, sunlight filtering through leaves.

A surreal underwater library has bookshelves made of coral and fish swimming gracefully among floating, open books.

A nomadic desert caravan travels across vast, sun-drenched sand dunes under a sky painted with hues of orange and purple.

An enchanting fairy tale cottage with a thatched roof is nestled beside a sparkling stream in a sun-dappled forest.

A highly detailed macro photograph captures the intricate patterns and iridescent colors of a butterfly's delicate wings.



A serene mountain monastery perches precariously on a cliffside, overlooking a vast valley shrouded in morning mist.

A whimsical candy land constructed entirely of lollipops, chocolate rivers, and gingerbread houses under a cotton candy sky.

A tranquil scene shows a wooden rowboat gently drifting on a still lake, reflecting the vibrant autumn foliage perfectly.

A dramatic scene shows a lone lighthouse bravely standing against a raging, stormy sea with crashing waves.

A detailed miniature world exists inside a discarded glass bottle, with tiny houses, trees, and a flowing river.



A surreal dreamscape depicts floating islands connected by shimmering, ethereal bridges under a sky of swirling pastel colors.

A cozy, book-filled library with a crackling fireplace offers a warm and inviting atmosphere on a snowy winter day.

A hyperrealistic close-up shows a dew-covered dragonfly resting delicately on a vibrant, freshly bloomed lotus flower.

An impressionistic painting captures a bustling Parisian street cafe scene on a rainy evening with glowing reflections.

A lone astronaut stands on a desolate alien planet, gazing at a swirling nebula in the distant, star-filled sky.



An ancient, moss-covered stone archway opens to a mystical forest bathed in the soft glow of a full moon.

A cozy, sun-drenched kitchen with a freshly baked pie on the windowsill, and a cat peacefully napping nearby.

A vast, windswept desert landscape with ancient, crumbling statues partially buried in the shifting golden sands under a blazing sun.

A dramatic volcanic eruption, with molten lava flowing down the mountainside under a stormy, ash-filled sky.

An old, weathered wooden ship sails courageously through a tumultuous, storm-tossed sea towards the horizon.



A tranquil Japanese garden in springtime, featuring a koi pond, cherry blossom trees, and traditional stone lanterns.

A tabby cat with yellow eyes sits behind a plate piled with fresh potatoes, beside a lady wearing a hat.

An abandoned spaceship, overgrown with alien flora, rests silently on the crimson plains of an unexplored, distant planet.

A surreal desert oasis appears with palm trees and shimmering water under a sky filled with impossibly large moons.

An old-fashioned steam train chugs through a picturesque snowy mountain landscape, smoke billowing from its chimney.

Figure 21: Text-to-image samples of SD1.5+MG.

NeurIPS Paper Checklist

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 claims are supported by the method, experiments, and reported results in the paper.

   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 appendix discusses limitations and broader impacts of the proposed method.

   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: The paper includes mathematical derivations, formal proposition, and its proof.

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: The paper provides algorithms, hyperparameters, and pseudo-code. Public training and evaluation code, environment setup instructions, and checkpoint downloads for SiT-XL/2+MG are available at `https://github.com/tzco/Diffusion-wo-CFG`.

   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: [Yes].

Justification: Code and model checkpoint downloads for SiT-XL/2+MG are publicly available at `https://github.com/tzco/Diffusion-wo-CFG`. The repository README gives environment setup, training, and evaluation instructions.

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: The paper reports datasets, models, optimizer, hyperparameters, samplers, and evaluation metrics.

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: The paper reports standard metrics but does not include error bars or significance tests.

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: [No].

   Justification: The paper reports FLOPs and speedups but not complete hardware or runtime details.

   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: The work uses standard benchmarks and discusses potential misuse risks.

   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: The appendix discusses efficiency benefits, data biases, and possible misuse of image generation.

    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: The paper states that model weights will have limited release to reduce misuse risks.

    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: [No].

    Justification: Existing assets are cited, but their licenses are not fully enumerated.

    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: [Yes].

    Justification: The public repository at `https://github.com/tzco/Diffusion-wo-CFG` documents the SiT-XL/2+MG reproduction asset, including dependencies, training and evaluation commands, and checkpoint downloads.

    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: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    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: [N/A].

    Justification: The paper does not involve human-subject research requiring IRB approval.

    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: [N/A].

    Justification: LLMs are not used as a core component of the research.

    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.