Direct Discriminative Optimization: Your Likelihood-Based Visual Generative Model is Secretly a GAN Discriminator

Kaiwen Zheng [1 2] **Yongxin Chen** [1] **Huayu Chen** [2] **Guande He** [3] **Ming-Yu Liu** [1] **Jun Zhu** [2] **Qinsheng Zhang** [1]
https://research.nvidia.com/labs/dir/ddo/

Abstract

While likelihood-based generative models, particularly diffusion and autoregressive models, have achieved remarkable fidelity in visual generation, the maximum likelihood estimation (MLE) objective, which minimizes the forward KL divergence, inherently suffers from a mode-covering tendency that limits the generation quality under limited model capacity. In this work, we propose Direct Discriminative Optimization (DDO) as a unified framework that integrates likelihood-based generative training and GAN-type discrimination to bypass this fundamental constraint by exploiting reverse KL and self-generated negative signals. Our key insight is to parameterize a discriminator implicitly using the likelihood ratio between a learnable target model and a fixed reference model, drawing parallels with the philosophy of Direct Preference Optimization (DPO). Unlike GANs, this parameterization eliminates the need for joint training of generator and discriminator networks, allowing for direct, efficient, and effective fine-tuning of a well-trained model to its full potential beyond the limits of MLE. DDO can be performed iteratively in a self-play manner for progressive model refinement, with each round requiring less than 1% of pretraining epochs. Our experiments demonstrate the effectiveness of DDO by significantly advancing the previous SOTA diffusion model EDM, reducing FID scores from 1.79/1.58/1.96 to new records of 1.30/0.97/1.26 on CIFAR-10/ImageNet-64/ImageNet 512×512 datasets without any guidance mechanisms, and by consistently improving both guidance-free and CFG-enhanced FIDs of visual autoregressive models on ImageNet 256×256.

[1]NVIDIA [2]Tsinghua University [3]The University of Texas at Austin. Correspondence to: Jun Zhu <dcszj@tsinghua.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

(a) EDM2-L (Karras et al., 2024b) (FID 1.96)

(b) EDM2-L + DDO (Ours, FID **1.26**)

Figure 1. Samples on ImageNet 512×512, *without any guidance*.

1. Introduction

Modeling the distribution of high-dimensional data is a fundamental challenge in machine learning (Bishop & Nasrabadi, 2006; Goodfellow et al., 2016). Recent years have witnessed the domination of diffusion (Ho et al., 2020; Song et al., 2021b) and autoregressive (Van Den Oord et al., 2016) paradigms in generative modeling of continuous data

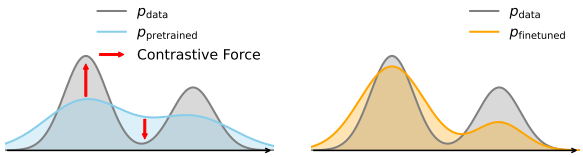


Figure 2. Toy example illustrating DDO. (a) Models pretrained via maximum likelihood estimation (MLE) exhibit dispersed density, while DDO imposes contrastive forces toward the data distribution. (b) The finetuned model concentrates better on the main mode.

and discrete data. They have achieved both theoretical and empirical success in visual tasks including image and video synthesis (Dhariwal & Nichol, 2021; Esser et al., 2021; Ramesh et al., 2021; Karras et al., 2022; Ho et al., 2022; Rombach et al., 2022; Balaji et al., 2022; Gupta et al., 2023; Esser et al., 2024; Brooks et al., 2024; Bao et al., 2024; Tian et al., 2024), forming the cornerstone of large-scale generation systems in the era of AI-generated content.

Diffusion and autoregressive models are representatives of likelihood-based generative models. Compared to Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) which often face unstable training and mode collapse issues, these models are distinguished by their stability, scalability, and generalizability. Besides, their iterative generation process imposes fewer constraints on network Lipschitzness, potentially facilitating superior generative capability over GAN's single-step generation. Likelihood-based generative models aim to learn the underlying data distribution p_{data} by maximizing the likelihood of the observed data under a parameterized probabilistic model p_θ, which is equivalent to minimizing the forward Kullback–Leibler (KL) divergence:

$$\max_\theta \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x})\right] \Longleftrightarrow \min_\theta D_{\text{KL}}(p_{\text{data}} \parallel p_\theta)$$

However, this maximum likelihood estimation (MLE) objective entails inherent limitations. Forward KL is known to prioritize "mode-covering" and imposes extreme penalties if the model severely underestimates the likelihood of any training sample (Karras et al., 2024a). Under limited model capacity, this property forces the learned density to spread out excessively (Figure 2(a)), potentially leading to blurry samples—a phenomenon commonly observed in Variational Autoencoders (VAEs) (Kingma & Welling, 2014) and in likelihood training of diffusion models (Song et al., 2021a; Kingma et al., 2021; Lu et al., 2022a; Zheng et al., 2023b). Consequently, these models often rely heavily on guidance methods (Ho & Salimans, 2021; Kim et al., 2023a; Karras et al., 2024a) to steer the samples away from unlikely low-probability regions and toward the core of the data manifold in order to improve overall generation quality. In contrast, GANs, which are theoretically grounded in Jensen–Shannon (JS) divergence or Wasserstein distance (Arjovsky et al., 2017), tend to produce sharper and more realistic samples.

To bypass MLE's mode-covering nature, we aim to leverage GAN-type loss to discriminate between the model and data distributions and produce contrastive forces that guide the model. However, typical GANs require parameterizing extra discriminator networks and alternating optimization, creating engineering complications. Applying GAN-type training trivially to diffusion or autoregressive models is especially inefficient due to their iterative sampling processes.

In this work, we introduce Direct Discriminative Optimization (DDO), a framework that bridges likelihood-based generative models with GANs to push their performance beyond the limits of MLE. Our key insight is to implicitly parameterize the discriminator using the likelihood ratio between a learnable target model and a fixed reference model, both initialized from the pretrained model. This parameterization, inspired by Direct Preference Optimization (DPO) (Rafailov et al., 2024), offers theoretical guarantees of optimality, divergence bounds, and connections to guidance methods. It also enables direct finetuning of the pretrained model without altering the network architecture or inference protocol and supports iterative refinement via multi-round self-play.

DDO achieves significant performance gains for both diffusion and autoregressive models sufficiently pretrained on standard image benchmarks. By finetuning state-of-the-art diffusion models EDM (Karras et al., 2022) and EDM2 (Karras et al., 2024b), we achieve unprecedented guidance-free FID scores of 1.30/0.97/1.26 on CIFAR-10/ImageNet-64/ImageNet 512×512 datasets. Finetuning the visual autoregressive model VAR (Tian et al., 2024) on ImageNet 256×256 reduces the FID from 1.92 to 1.73 while removing sampling tricks. Notably, even without classifier-free guidance (CFG) (Ho & Salimans, 2021), the finetuned model achieves an FID of 1.79, surpassing the CFG-enhanced pretrained model while cutting the inference cost by half.

2. Background

2.1. Likelihood-Based Generative Models

Likelihood-based generative models parameterize a probability distribution p_θ to learn the data distribution p_{data}, enabling explicit likelihood evaluation and density estimation. Among them, diffusion and autoregressive models are two prominent types that excel in visual generation.

Autoregressive (AR) models (Van Den Oord et al., 2016; Brown et al., 2020) learn discrete[1] data distributions via the next-token prediction mechanism:

$$\log p_\theta(\boldsymbol{x}) = \sum_{n=1}^{d} \log p_\theta(x^{(n)} | \boldsymbol{x}^{(<n)}) \tag{1}$$

[1]AR can also be adapted to model continuous data (Tschannen et al., 2024; Li et al., 2024).

where d denotes the data dimension (sequence length). It factorizes the joint distribution into a product of conditional probabilities, allowing exact likelihood computation. Each $p_\theta(\cdot|\boldsymbol{x}^{(<n)})$ is parameterized via a Softmax operation over the model's output logits and optimized using cross-entropy loss against the ground-truth token. In visual autoregressive modeling, images are first quantized to discrete tokens within a compact latent space using autoencoders (Van Den Oord et al., 2017; Esser et al., 2021).

Diffusion models (Ho et al., 2020; Song et al., 2021b) learn continuous data distributions by gradually perturbing clean data $\boldsymbol{x}_0 \sim p_{\text{data}}$ with Gaussian noise, which generates a trajectory $\{\boldsymbol{x}_t\}_{t=0}^T$, and then learning to reverse this process. The forward and backward dynamics can be formulated as either stochastic or ordinary differential equations (SDEs or ODEs) (Song et al., 2021b). The forward process follows a closed-form transition kernel $q_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\alpha_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I})$ with predefined noise schedule α_t, σ_t, enabling reparameterization as $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The model is typically parameterized as a noise prediction network $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ trained to estimate $\boldsymbol{\epsilon}$ via mean squared error (MSE) regression, which forms an evidence (or variational) lower bound (ELBO) on the likelihood (Song et al., 2021a):

$$\log p_\theta(\boldsymbol{x}_0) \geq C - \mathbb{E}_{t\sim p(t), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[w(t)\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2\right] \tag{2}$$

where $\boldsymbol{x}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}$, C is a constant irrelevant to θ, and $p(t), w(t)$ are certain time distribution and weighting function. The ELBO provides a reasonable likelihood approximation compared to the exact but cumbersome instantaneous change-of-variable formula in neural ODEs (Chen et al., 2018). Moreover, while the likelihood bound is tight only for specific $p(t), w(t)$, alternative choices share the same optimum and can serve as surrogate objectives (Kingma & Gao, 2024).

From the perspective of score matching (Song et al., 2021b), the optimal noise predictor is linked to the *score function* $\boldsymbol{s}^*(\boldsymbol{x}_t, t) := \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ by $\boldsymbol{\epsilon}^*(\boldsymbol{x}_t) = -\sigma_t \boldsymbol{s}^*(\boldsymbol{x}_t, t)$, where q_t denotes the marginal distribution at time t in the forward process. Due to the properties of MSE, the network can be parameterized in alternative yet theoretically equivalent forms, such as a velocity predictor (Salimans & Ho, 2022; Zheng et al., 2023b) that estimates the tangent of the diffusion trajectory, commonly known as flow matching (Lipman et al., 2022). In our experiments, we adopt the more generalized F-parameterization introduced in EDM (Karras et al., 2022) (detailed in Appendix C).

2.2. GANs

GANs (Goodfellow et al., 2014) do not explicitly model the likelihood p_θ but instead directly optimize the data generation process through adversarial training. Specifically, the optimization involves an adversarial interplay between a generator network $\boldsymbol{g}_\theta : \mathbb{R}^{d_z} \mapsto \mathbb{R}^d$ that maps latent variables $\boldsymbol{z} \in \mathbb{R}^{d_z} \sim p(\boldsymbol{z})$ (typically Gaussian noise) into synthetic samples, and a discriminator network $d_\phi : \mathbb{R}^d \mapsto [0, 1]$ that classifies samples as real or fake:

$$\min_\theta \max_\phi \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\log d_\phi(\boldsymbol{x})\right] + \mathbb{E}_{p_\theta(\boldsymbol{x})}\left[\log(1 - d_\phi(\boldsymbol{x}))\right]. \tag{3}$$

Here $p_\theta(\boldsymbol{x})$ is the generator distribution, whose exact density is intractable but can be easily sampled from via $\boldsymbol{x} = \boldsymbol{g}_\theta(\boldsymbol{z}), \boldsymbol{z} \sim p(\boldsymbol{z})$. In the inner loop, the discriminator is optimized using binary cross-entropy loss (also known as noise contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010)), and its optimal solution can be derived as:

$$d^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_\theta(\boldsymbol{x})} \tag{4}$$

under which the minimax game becomes

$$\min_\theta 2D_{\text{JS}}(p_{\text{data}} \parallel p_\theta) - 2\log 2 \tag{5}$$

where $D_{\text{JS}}(p \parallel q) = \frac{1}{2}D_{\text{KL}}(p \parallel \frac{p+q}{2}) + \frac{1}{2}D_{\text{KL}}(q \parallel \frac{p+q}{2})$ is the Jensen–Shannon (JS) divergence. This theoretically ensures that the optimal generator distribution matches the data distribution. However, in practice, training instability arises due to gradient vanishing and mode collapse, inspiring variants such as Wasserstein GANs (Arjovsky et al., 2017).

GANs can be incorporated to enhance other generative models. For example, Discriminator Guidance (Kim et al., 2023a) utilizes the gradient information from the discriminator as a corrective term to refine the score function in diffusion models (discussed in Section 4.2). Additionally, GANs are commonly employed as an auxiliary loss to improve one-step generation such as in diffusion distillation (Kim et al., 2023b; Yin et al., 2024; Zhou et al., 2024b).

3. Direct Discriminative Optimization

Motivated by the benefits of adversarial training in enhancing generation quality, we aim to bridge likelihood-based generative models with GANs to derive an alternative training paradigm to MLE. Unlike prior works that incorporate GAN as an auxiliary loss and require additional engineering overhead, our approach seeks to (1) directly optimize likelihood-based generative models without modifying the network architecture, adding extra discriminators, complicating the training procedure or increasing inference costs, and (2) eliminate the need for backpropagation through the sampling process, making it applicable to diffusion and autoregressive models that rely on iterative sampling.

3.1. Your Likelihood-Based Generative Model is Secretly a Discriminator

Unlike one-step generators that learn a direct mapping from noise to data, likelihood-based generative models are

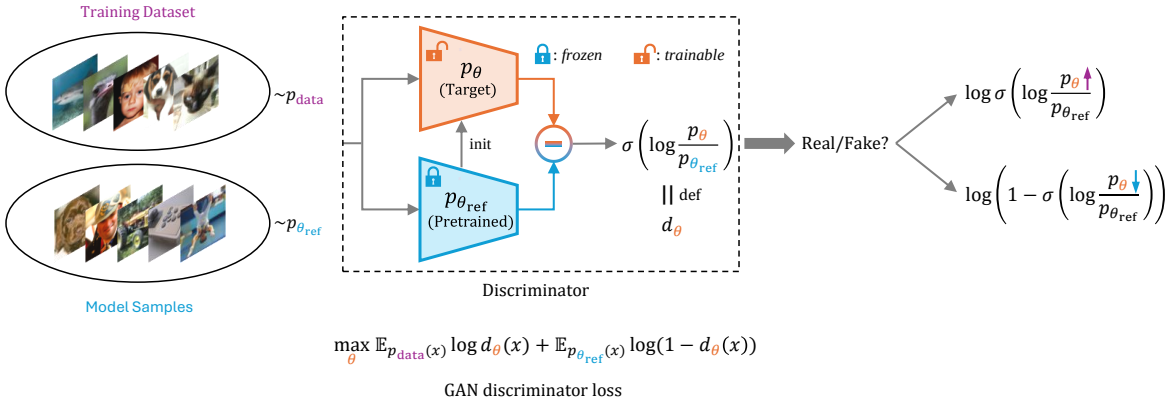


Figure 3. Illustration of DDO. (1) Models. θ_{ref} is the (pretrained) reference model frozen during training. θ is the learnable model initialized as θ_{ref}. (2) Data. Samples from p_{data} are drawn from the training dataset. Samples from $p_{\theta_{\text{ref}}}$ are generated by the reference model, either offline or online. (3) Objective. The target model θ is optimized by applying the GAN discriminator loss with the implicitly parameterized discriminator d_θ to distinguish between real samples from p_{data} and fake samples from $p_{\theta_{\text{ref}}}$.

grounded in the probabilistic definition of the likelihood function p_θ, which enables both the generation of samples $\boldsymbol{x} \sim p_\theta$ and the evaluation of the likelihood $p_\theta(\boldsymbol{x})$, either exactly or approximately, while retaining the tractability of backpropagation through the likelihood computation. This inspires us to utilize the likelihood information embedded in the optimal discriminator (Eqn. (4)).

Specifically, consider a pretrained model $p_{\theta_{\text{ref}}}$ as a reference to generate fake samples. The optimal discriminator d_θ for

$$\min_\theta -\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\log d_\theta(\boldsymbol{x})\right] - \mathbb{E}_{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\left[\log(1-d_\theta(\boldsymbol{x}))\right] \quad (6)$$

can be rewritten as:

$$d^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} = \sigma\left(\log \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right) \quad (7)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the Sigmoid function. The data distribution p_{data} is available from d^*. Therefore, if we parameterize the discriminator d_θ using a likelihood-based target generative model p_θ as:

$$d_\theta(\boldsymbol{x}) \coloneqq \sigma\left(\log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right) \quad (8)$$

then the optimal target model that minimizes the GAN discriminator loss matches the data distribution. We formalize this induced objective in the following theorem.

Theorem 3.1 (Optimality)**.** *With unlimited model capacity, the optimal likelihood-based model p_θ under the objective*

$$\begin{aligned}\min_\theta \mathcal{L}(\theta) = &-\mathbb{E}_{p_{data}(\boldsymbol{x})}\left[\log \sigma\left(\log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{ref}}(\boldsymbol{x})}\right)\right] \\ &-\mathbb{E}_{p_{\theta_{ref}}(\boldsymbol{x})}\left[\log\left(1-\sigma\left(\log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{ref}}(\boldsymbol{x})}\right)\right)\right]\end{aligned} \quad (9)$$

satisfies $p_{\theta^*} = p_{data}$.

In contrast to previous GAN-based methods that introduce a separate discriminator network d_ϕ, our approach implicitly defines the discriminator through a target generative model p_θ. While it is theoretically feasible to initialize $\theta, \theta_{\text{ref}}$ arbitrarily and train from scratch, strong initial conditions facilitate optimization (Section 3.2). In practice, we initialize $\theta, \theta_{\text{ref}}$ from widely available pretrained models, promoting steady improvement. We refer to this approach as *Direct Discriminative Optimization* (DDO), drawing parallels with Direct Preference Optimization (DPO) (Rafailov et al., 2024), which aligns language models with human preferences by expressing the reward model in terms of the likelihood ratio between two policies (discussed in Section 4.1). The DDO pipeline is illustrated in Figure 3.

What does the DDO update do? For a mechanistic understanding of DDO, we can analyze the gradient of the loss function with respect to parameters θ:

$$\nabla_\theta \mathcal{L}(\theta) = \int \underbrace{(1-d_\theta(\boldsymbol{x}))}_{\in[0,1]} \underbrace{(p_\theta(\boldsymbol{x}) - p_{\text{data}}(\boldsymbol{x}))}_{p_\theta(\boldsymbol{x})\uparrow \text{ when } <0} \nabla_\theta \log p_\theta(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \quad (10)$$

Intuitively, gradient descent increases the model likelihood $p_\theta(\boldsymbol{x})$ for data points $\boldsymbol{x}$ that satisfy $p_\theta(\boldsymbol{x}) < p_{\text{data}}(\boldsymbol{x})$, and decreases it otherwise, pushing p_θ closer to p_{data}. Furthermore, the gradient magnitude is weighted by both the distance $|p_\theta(\boldsymbol{x}) - p_{\text{data}}(\boldsymbol{x})|$ and $1 - d_\theta(\boldsymbol{x})$, assigning higher weights to samples discriminated as fake.

3.2. Theoretical Analysis

Apart from the optimality guarantee, we also examine the behavior of the DDO objective when θ is not optimal. Specifically, we investigate the following question:

Is p_θ closer to p_{data} with a lower $\mathcal{L}(\theta)$?

Under certain assumptions, we can establish bounds on the divergence between p_θ and p_{data} in terms of the difference

between $\mathcal{L}(\theta)$ to the optimal loss value $\mathcal{L}^*$, as formalized in the following theorem.

Theorem 3.2 (Divergence Bounds)**.** *If* $\log \frac{p_{\theta_{ref}}}{p_{data}}$ *and* $\log \frac{p_\theta}{p_{\theta_{ref}}}$ *are bounded, there exist some constants* C_1, C_2 *such that*

$$D_{\mathrm{KL}}(p_{data} \parallel p_\theta) \leq C_1 \sqrt{\mathcal{L}(\theta) - \mathcal{L}^*} \tag{11}$$

$$D_{\mathrm{KL}}(p_\theta \parallel p_{data}) \leq C_2 \sqrt{\mathcal{L}(\theta) - \mathcal{L}^*} \tag{12}$$

The assumption of bounded $\log \frac{p_\theta}{p_{\theta_{\text{ref}}}}$ implies that the optimized distribution does not deviate significantly from the reference distribution, which is reasonable when finetuning for a short duration. The assumption of bounded $\log \frac{p_{\theta_{\text{ref}}}}{p_{\text{data}}}$ imposes a constraint to the reference model regarding its mutual density coverage with the data distribution. We can expect $\log \frac{p_{\theta_{\text{ref}}}}{p_{\text{data}}}$ to be lower bounded, i.e., $p_{\theta_{\text{ref}}}$ sufficiently covers p_{data}, which aligns with the characteristics of MLE-trained models. Under this condition, the forward KL $D_{\mathrm{KL}}(p_{\text{data}} \parallel p_\theta)$ remains bounded by $\sqrt{\mathcal{L}(\theta) - \mathcal{L}^*}$. However, bounding the reverse KL requires an upper bound on $\frac{p_{\theta_{\text{ref}}}}{p_{\text{data}}}$, which imposes a stronger constraint on $p_{\theta_{\text{ref}}}$.

3.3. Practical Implementation

We introduce several practical techniques that make DDO applicable to high-dimensional real-world data and diffusion models whose likelihood computation is expensive. [2]

Generalized Objective with Extra Coeffecients The log-likelihood $\log p_\theta(\boldsymbol{x})$ of likelihood-based generative models often scales with the data dimension and can reach magnitudes of 10^3. As the DDO objective in Eqn. (9) involves a Sigmoid operation on $\log p_\theta(\boldsymbol{x})$, this can lead to gradient vanishing and numerical precision issues. To address this, we add hyperparameters α, β to control the relative weights of loss terms and scale the probability ratio:

$$\begin{aligned}\mathcal{L}_{\alpha,\beta}(\theta) = &- \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\log \sigma\left(\beta \log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right)\right] \\ &- \alpha \mathbb{E}_{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\left[\log\left(1 - \sigma\left(\beta \log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right)\right)\right]\end{aligned} \tag{13}$$

The modified loss retains the same optimization trend, namely, increasing $p_\theta(\boldsymbol{x})$ for $\boldsymbol{x} \sim p_{\text{data}}$ and decreasing $p_\theta(\boldsymbol{x})$ for $\boldsymbol{x} \sim p_{\theta_{\text{ref}}}$, but the optimum may "overshoot" the data distribution for $\beta < 1$. Specifically, we have:

Theorem 3.3. *With unlimited model capacity, the optimal likelihood-based generative model* θ *that minimizes* $\mathcal{L}_{\alpha,\beta}(\theta)$ *satisfies* $p_{\theta^*} \propto p_{\theta_{ref}}^{1-1/\beta} p_{data}^{1/\beta}$ *for certain* α.

This establishes a deep connection with guidance methods (discussed in Section 4.2). In practice, we observe that α and β across a wide range of values[3] yield reasonable performance. We sweep over them for the best results.

Handling Compute-Intensive Likelihood Evaluating the model likelihood for a specific data point can be computationally intensive. In particular, unlike autoregressive models, which only require a single forward pass through the network to compute $\log p_\theta(\boldsymbol{x})$ (Eqn. (1)) due to the causal structure imposed by attention masks, diffusion models necessitate multiple forward passes over different timesteps to approximate $\log p_\theta(\boldsymbol{x})$ through the ELBO (Eqn. (2)). Specifically, the log-likelihood ratio in the DDO loss is:

$$\log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})} \approx \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\Delta_{\boldsymbol{x}_t,t,\boldsymbol{\epsilon}}\right] \tag{14}$$

where $\boldsymbol{x}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}$ and

$$\Delta_{\boldsymbol{x}_t,t,\boldsymbol{\epsilon}} = -w(t)\left(\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{\epsilon}\|_2^2 - \|\boldsymbol{\epsilon}_{\theta_{\text{ref}}}(\boldsymbol{x}_t,t) - \boldsymbol{\epsilon}\|_2^2\right) \tag{15}$$

We apply Jensen's inequality pointwise to derive an upper bound for the loss using the convexity of the function $-a\log\sigma(x) - b\log(1-\sigma(x))$ for any $a, b \geq 0$:

$$\begin{aligned}&\mathcal{L}(\theta) \\ &\approx -\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})} \log \sigma\left(\mathbb{E}_{t,\boldsymbol{\epsilon}}[\Delta]\right) - \mathbb{E}_{p_{\theta_{\text{ref}}}(\boldsymbol{x})} \log(1 - \sigma\left(\mathbb{E}_{t,\boldsymbol{\epsilon}}[\Delta]\right)) \\ &\leq -\mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})} \log\sigma(\Delta) + \mathbb{E}_{p_{\theta_{\text{ref}}}(\boldsymbol{x})} \log(1 - \sigma(\Delta)))\right]\end{aligned} \tag{16}$$

This treatment, analogous to the one used in Diffusion-DPO (Wallace et al., 2024), enables us to approximate the diffusion DDO loss using a single forward pass for each $\boldsymbol{x}$.

Multi-Round Refinement via Self-Play Due to the practical modifications for applicability, the optimization process of DDO provides useful gradient information in the early stage but does not converge to the data distribution in the final. To maximize the fine-tuning performance, we adopt a multi-round refinement strategy, where the reference model $p_{\theta_{\text{ref}}}$ is iteratively updated by replacing it with an improved version from the previous round:

$$\begin{aligned}&\text{Round } n\text{:} \quad \ldots \rightarrow \underbrace{p_{\theta^*_{n-1}}}_{\text{Reference}} \rightarrow \underbrace{\sigma\left(\beta \log \frac{p_{\theta_n}}{p_{\theta^*_{n-1}}}\right)}_{\text{Discriminator}} \\ &\text{Round } n+1\text{:} \quad \rightarrow \underbrace{p_{\theta^*_n}}_{\text{Reference}} \rightarrow \ldots\end{aligned}$$

where θ^*_n represents the best-performing model across different hyperparameter configurations and training iterations in round n. In each round, the reference model acts as a fixed generator, making the multi-round optimization analogous to the generator-discriminator interplay in GANs. However, unlike GANs, where both networks are explicitly optimized, we never update the reference (generator) model directly. Instead, the generator is obtained from the discriminator in the previous round, leading to a form of self-play. This iterative refinement process is conceptually similar to Iterative

[2] A code example is provided in Appendix D.

[3] Typical choices are $\alpha \in [0.5, 50]$ and $\beta \in [0.01, 0.1]$, while the optimal values depend on the specific model and settings.

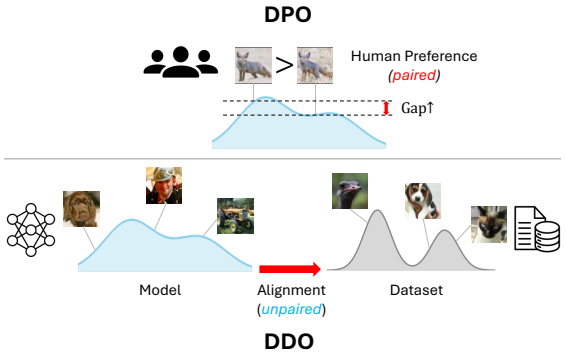


Figure 4. Comparison with DPO.

DPO (Xu et al., 2023) and SPIN (Chen et al., 2024c), which extend DPO for better language model alignment.

3.4. Discussion

Connection to RL At a high level, DDO enables visual generative models to **utilize negative signals from self-generated samples**—a characteristic deeply rooted in reinforcement learning (RL) that underpins modern language models (Achiam et al., 2023; Guo et al., 2025). Distinguished from works that employ a similar contrastive loss merely to off-the-shelf data (Chen et al., 2024a), DDO can fundamentally improve the base model's ability.

Extension to f-divergence The GAN discriminator loss can be generalized to f-divergences (Nowozin et al., 2016):

$$D_f(p \parallel q) = \sup_T \mathbb{E}_{p(x)}\left[T(x)\right] - \mathbb{E}_{q(x)}\left[f^*(T(x))\right] \quad (17)$$

where f^* is the convex conjugate of f. DDO can be extended to this family as the optimal $T^*(x) = f'\left(\frac{p(x)}{q(x)}\right)$ explicitly involves the density ratio.

4. Comparison with Existing Methods

4.1. Direct Preference Optimization (DPO)

DPO (Rafailov et al., 2024) is a lightweight surrogate objective designed for reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Achiam et al., 2023) that enhances the instruction-following ability of pretrained language models. Standard RLHF involves two stages: (1) learning a reward model r_θ and (2) aligning the reference policy $\pi_{\theta_{\text{ref}}}$ to the target policy $\pi_\theta(y|x) \propto \pi_{\theta_{\text{ref}}}(y|x)e^{r_\theta(x,y)/\beta}$ using RL, where x is the prompt and y is the response. The Bradley-Terry preference mode (Bradley & Terry, 1952) links preferences and rewards by

$$p(y_w \succ y_l|x) := \frac{e^{r(x,y_w)}}{e^{r(x,y_l)}+e^{r(x,y_w)}} = \sigma(r(x,y_w) - r(x,y_l)) \quad (18)$$

where y_w and y_l denote the winning and losing responses for a given prompt x, annotated by human. DPO enables direct optimization of pretrained language models on preference data without training a separate reward model:

$$\begin{aligned}&\mathcal{L}^{\text{DPO}}(\theta)\\ &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\theta_{\text{ref}}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\theta_{\text{ref}}}(y_l|x)}\right)\end{aligned} \quad (19)$$

where the reward function $r_\theta(y|x)$ is implicitly parameterized by the log-likelihood ratio $\beta\log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{ref}}}(y|x)}$.

Despite sharing similar insights in parameterization, DDO is fundamentally different from DPO. As illustrated in Figure 4, DPO is designed for *preference learning*, requiring additional paired human-annotated data and maximizing the likelihood gap between preferred (winning) and non-preferred (losing) responses without considering the whole distribution. In contrast, DDO focuses on *distribution alignment*, directly aligning the model with the ground-truth data distribution. It requires only the original training data that are unpaired with the model-generated samples.

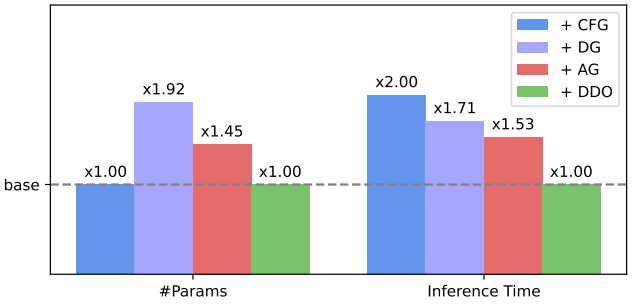


Figure 5. Comparison of model parameter counts and inference time across different guidance methods and DDO. For DG, we measure the statistics on class-conditional CIFAR-10. For AG, we measure the statistics on ImageNet-64.

4.2. Guidance Methods

We review several types of guidance methods that enhance diffusion models[4] at inference time. Let $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ denote the score function representation introduced in Section 2.1.

Classifier-Free Guidance (CFG) (Ho & Salimans, 2021) combines the unconditional/conditional model to obtain a new score predictor $\boldsymbol{s}'_\theta(\boldsymbol{x}_t, t, c) := \boldsymbol{s}_\theta(\boldsymbol{x}_t, t, c) + w(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t, c) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t, \emptyset))$, where $\emptyset$ represents the unconditional case, and w is the guidance scale. The unconditional model shares parameters with the conditional model and is learned by random label dropout during training.

Discriminator Guidance (DG) (Kim et al., 2023a) trains a time-dependent discriminator network d_ϕ that distinguishes between perturbed real data and model-generated samples. Its gradient is then used to refine the score function: $\boldsymbol{s}_{\theta,\phi}(\boldsymbol{x}_t, t) = \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + w\nabla_{\boldsymbol{x}_t}\log\frac{d_\phi(\boldsymbol{x}_t,t)}{1-d_\phi(\boldsymbol{x}_t,t)}$. Similar to

[4]Guidance can be easily adapted to autoregressive models.

DDO, DG leverages the optimal discriminator (Eqn. (4)) for theoretical guarantees. However, it explicitly parameterizes the discriminator as a separate, time-aware network.

Autoguidance (AG) (Karras et al., 2024a) operates similarly to CFG but refines the score function by guiding the base model with an inferior variant: $\boldsymbol{s}_{\theta,\phi}(\boldsymbol{x}_t, t) := \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + w(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{s}_\phi(\boldsymbol{x}_t, t))$, where $\boldsymbol{s}_\phi$ is a degraded version of $\boldsymbol{s}_\theta$ obtained via reduced model capacity or under-training.

For a unified perspective, all these guidance methods improve a well-trained distribution p_θ by amplifying its difference from a degraded or less informative distribution p_ϕ: $p_{\theta,\phi} \propto p_\theta \left(\frac{p_\theta}{p_\phi}\right)^w$. This superposition sharpens the MLE-optimized model distribution and suppresses low-probability outliers (Karras et al., 2024a). According to Theorem 3.3, DDO with $\beta < 1$ induces a similar overshooting effect, highlighting the theoretical connection.

Unlike guidance methods, DDO enhances sample quality without increasing inference costs compared to the base model (Figure 5). Moreover, in scenarios where CFG is crucial for balancing image-condition alignment and diversity (e.g., FID-IS curve), DDO can be seamlessly integrated with CFG to achieve an overall improved trade-off (Section 5.3).

5. Experiments

Our experiments aim to investigate the following aspects:

1. The effectiveness and efficiency of DDO in enhancing the visual quality of well-trained diffusion models (Section 5.2) and autoregressive (Section 5.3) models.
2. The impact of the hyperparameters α, β, as well as the benefits of multi-round refinement.

5.1. Experimental Setups

Datasets & Models We experiment on standard image benchmarks including CIFAR-10 (Krizhevsky et al., 2009) in 32×32 resolution and ImageNet (Deng et al., 2009) in multiple resolutions (64×64, 256×256, 512×512). For each dataset, we apply DDO to finetune state-of-the-art diffusion or autoregressive models, including EDM (Karras et al., 2022), EDM2 (Karras et al., 2024b) and VAR (Tian et al., 2024). We compare with a range of advanced generative baselines, including diffusion models, autoregressive (AR) models, masked models, and GAN-based approaches.

Training & Evaluation We evaluate Fréchet inception distance (FID) (Heusel et al., 2017) on 50k images as the primary benchmark metric for all experiments, and additionally measure Inception Score (IS) (Barratt & Sharma, 2018) for ImageNet 256×256. We report the number of function evaluations (NFE) as a quantification of inference efficiency. We find strict class balance crucial for FIDs on ImageNet and slightly modify the original EDM sampling scripts to enforce it. For diffusion models, we finetune over multiple rounds until further improvement is negligible. For VAR, we observe rapid convergence and only finetune for 2 rounds. The finetuning is highly efficient, with each round requiring less than 1% of pretraining iterations. Further experiment details can be found in Appendix C, and visualizations of generated samples are provided in Appendix E.

5.2. Results on Diffusion Models

Table 1. Results on unconditional and class-conditional CIFAR-10. †Including diffusion distillation methods with auxiliary GAN loss. ‡ The reported parameter count excludes those of the discriminator (for GANs) and VAE encoder/decoder (for latent-space models).

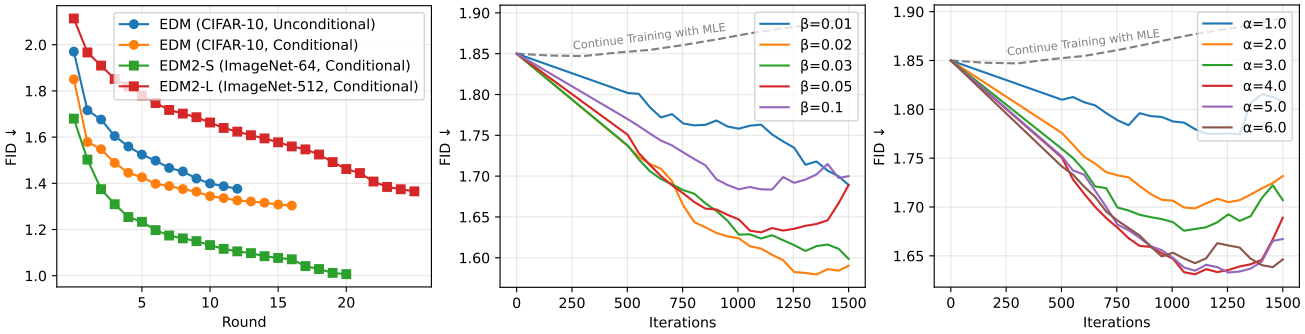
Type	Model	#Params‡	NFE	Uncond	Cond
				FID↓	FID↓
GAN†	StyleGAN2-ADA (Karras et al., 2020)	20M	1	2.92	2.42
	StyleGAN-XL (Sauer et al., 2022)	18M	1	-	1.85
	R3GAN (Huang et al., 2025)	21M	1	-	1.96
	CTM (Kim et al., 2023b)	59M	1	1.98	1.73
	SiD^{2}A (Zhou et al., 2024b)	56M	1	1.50	1.40
Diffusion	DDPM (Ho et al., 2020)	36M	1000	3.17	-
	iDDPM (Nichol & Dhariwal, 2021)	50M	4000	2.90	-
	DDIM (Ho et al., 2020)	36M	100	4.16	-
	DPM-Solver (Lu et al., 2022b)	107M	48	2.65	-
	DPM-Solver-v3 (Zheng et al., 2023a)	56M	12	2.24	-
	NCSN++ (Song et al., 2021b)	108M	2000	2.20	-
	LSGM (Vahdat et al., 2021)	376M	138	2.10	-
	VDM (Kingma et al., 2021)	31M	1000	7.41	-
	Flow Matching (Lipman et al., 2022)	-	142	6.35	-
	i-DODE (Zheng et al., 2023b)	139M	215	3.76	-
	EDM (Karras et al., 2022)	56M	35	1.97	1.79
	+ DG (Kim et al., 2023a)	107M	53	1.77	1.64
Ours	EDM (retested)	56M	35	1.97	1.85
	+ DDO	56M	35	**1.38**	**1.30**

Table 2. Results on class-conditional ImageNet-64.

Type	Model	#Params	NFE	FID↓
GAN	StyleGAN-XL (Sauer et al., 2022)	135M	1	1.51
	R3GAN (Huang et al., 2025)	104M	1	2.09
	CTM (Kim et al., 2023b)	324M	1	1.92
	DMD2 (Yin et al., 2024)	296M	1	1.28
	PaGoDA (Kim et al., 2024)	296M	1	1.21
	SiD^{2}A (Zhou et al., 2024b)	296M	1	1.11
Diffusion	iDDPM (Nichol & Dhariwal, 2021)	270M	250	2.92
	ADM (Dhariwal & Nichol, 2021)	296M	250	2.07
	RIN (Jabri et al., 2022)	281M	1000	1.23
	EDM (Karras et al., 2022)	296M	511	1.36
	VDM++ (Kingma & Gao, 2024)	296M	511	1.43
	DisCo-Diff (Xu et al., 2024)	-	623	1.22
	EDM2-S (Karras et al., 2024b)	280M	63	1.58
	+ CFG (Ho & Salimans, 2021)	560M	126	1.48
	+ AG (Karras et al., 2024a)	405M	126	1.01
	EDM2-M	498M	63	1.43
	EDM2-L	777M	63	1.33
	EDM2-XL	1.1B	63	1.33
Ours	EDM2-S (retested)	280M	63	1.60
	+ DDO	280M	63	**0.97**

The EDM and EDM2 base models on CIFAR-10 and ImageNet are implemented as separate unconditional or class-conditional networks. Since CFG provides limited benefits on these datasets, we directly apply the diffusion DDO loss without considering the interaction with CFG.

Main Results Table 1, Table 2 and Table 3 present the quantitative results on CIFAR-10 and ImageNet. Figure 6(a) illustrate the FID reduction over multiple rounds. We highlight the advantages of DDO as follows:



(a) FID-Round (b) $\alpha = 4.0$ (c) $\beta = 0.05$

Figure 6. Illustrations of DDO on diffusion models, including (a) multi-round refinement and (b)(c) training curves under different α, β. The ImageNet FIDs in this figure are evaluated without class rebalancing.

Table 3. Results on class-conditional ImageNet 512× 512. "G" denotes guidance (CFG by default). The reported NFE values are without guidance; applying guidance doubles them. †With classifier guidance. ‡With autoguidance (AG) (Karras et al., 2024a).

Type	Model	w/o G			w/ G
		#Params	NFE	FID↓	FID↓
GAN	BigGAN-deep (Brock, 2018)	112M	1	8.43	-
	StyleGAN-XL (Sauer et al., 2022)	168M	1	2.41	-
	SiD^{2}A (Zhou et al., 2024b)	1.5B	1	1.37	-
Diffusion	ADM (Dhariwal & Nichol, 2021)	559M	250	23.24	7.72†
	ADM-U	731M	500	9.96	3.85†
	DiT-XL/2 (Peebles & Xie, 2023)	675M	250	12.03	3.04
	SiT-XL (Ma et al., 2024)	675M	250	-	2.62
	+ REPA (Yu et al., 2024b)	675M	250	-	2.08
	RIN (Jabri et al., 2022)	410M	1000	3.95	-
	U-ViT, L (Hoogeboom et al., 2023)	2B	512	3.54	3.02
	VDM++ (Kingma & Gao, 2024)	2B	512	2.99	2.65
	USiT-2B (Chen et al., 2024b)	2B	-	2.90	1.72
	EDM2-XS (Karras et al., 2024b)	125M	63	3.53	2.91
	EDM2-S	280M	63	2.56	2.23
	+ AG				1.34‡
	EDM2-L	777M	63	2.06	1.88
	EDM2-XXL	1.5B	63	1.91	1.81
	+ AG				1.25‡
Masked	MaskGIT (Chang et al., 2022)	227M	12	7.32	-
	MAGVIT-v2 (Yu et al., 2023)	307M	64	3.07	1.91
	MAR-L (Li et al., 2024)	481M	1024	2.74	1.73
AR	VAR-d36-s (Tian et al., 2024)	2.3B	10	-	2.63
Ours	EDM2-L (retested)	777M	63	1.96	1.77
	+ DDO	777M	63	**1.26**	**1.21**‡
	+ DPM-Solver-v3 (Zheng et al., 2023a)	777M	25	1.29	**1.21**‡

(1) Effectiveness. With multi-round refinement, DDO achieves record-breaking guidance-free FID scores of 1.38/1.30 on CIFAR-10, 0.97 on ImageNet-64, and 1.26 on ImageNet 512×512, significantly improving upon the EDM and EDM2 base models by 30%, 40% and 36%, respectively. Additionally, DDO outperforms all guidance-based and GAN-based methods requiring complex GAN-specific tuning or increasing inference costs.

(2) Efficiency. Although we employ substantial training over dozens of rounds to maximize the performance and explore the upper bound of DDO, FID improves significantly within just a few rounds. Notably, DDO in a single round achieves FIDs of 1.72/1.58 on CIFAR-10 with the EDM base model, surpassing DG. On ImageNet-64, the compact EDM2-S attains an FID of 1.31 after only 3 rounds, surpassing the 4× larger EDM2-XL. On ImageNet 512×512, guidance-free EDM2-L matches CFG-enhanced EDM2-XXL (2× larger) with only 4 rounds, demonstrating the parameter efficiency unlocked by DDO.

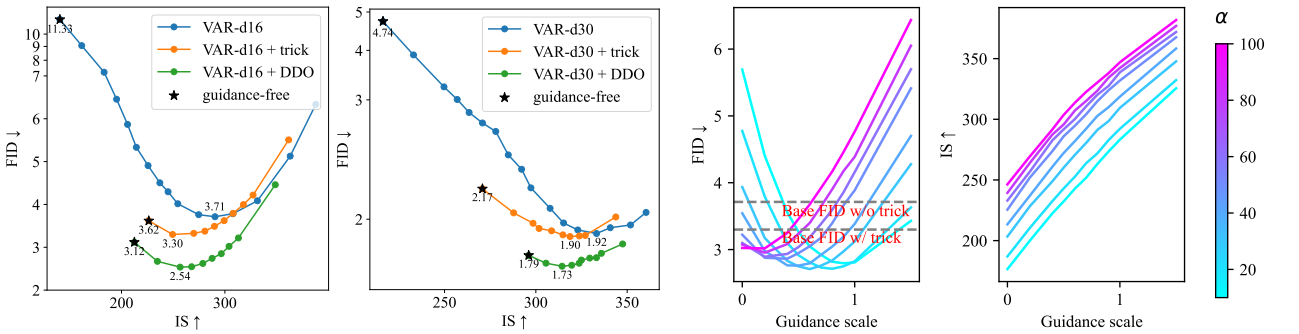
Effects of α, β As shown in Figure 6(b)(c), we visualize the training curves under different α, β for class-conditional CIFAR-10 during the first round. We empirically find that a wide range of α, β consistently improves the base model, though identifying the optimal hyperparameters requires grid searching. Moreover, tuning α while keeping β fixed or adjusting β under appropriate α yields similar effects.

Comparison to MLE Figure 6(b)(c) also includes the FID curve of extended training using the original diffusion loss (i.e., MLE). Unlike DDO, continued MLE training fails to improve performance and even leads to degradation when we do not retain previous optimizer states. This is partly because the base model is already extremely optimized and more fundamentally due to the forward KL objective's inherent limitations. In contrast, DDO with various α, β configurations consistently demonstrates clear advancements.

Accelerate Sampling By leveraging the advanced sampler DPM-Solver-v3 (Zheng et al., 2023a), we reduce the inference steps of EDM2-L+DDO to 25 while preserving generation quality. Our model achieves FID scores of 1.29/1.21 for 512×512 images, with an end-to-end per-image latency of 1.03s/1.96s on a single H100 GPU (batch size = 1).

5.3. Results on Autoregressive Models

The VAR models rely heavily on CFG to enhance generation quality. A distinctive feature of CFG is its ability to balance diversity and fidelity by adjusting the guidance scale, which is essential for creating the FID-IS trade-off. Consequently, we need to accommodate DDO to ensure that the finetuned models remain compatible with CFG. To this end, we choose the reference and target distributions $p_{\theta_{\text{ref}}}, p_\theta$ as the guidance-free model corresponding to $w = 0$. To preserve the model's ability for unconditional generation, we incorporate random label dropout during DDO fine-tuning. We set $\alpha = 0$ for the unconditional part to prevent it from receiving negative signals from reference samples $\boldsymbol{x} \sim p_{\theta_{\text{ref}}}$.



(a) VAR-d16 (b) VAR-d30 (c) FID and IS w.r.t. guidance scale w

Figure 7. Illustrations of DDO on autoregressive models. (a)(b) FID-IS trade-off curves and (c) the impact of α under $\beta = 0.02$.

Table 4. Results on class-conditional ImageNet 256× 256. *Samples are generated at 512 × 512 resolution and resized to 256 × 256 using OpenCV's `INTER_LANCZOS4` method.

Type	Model	w/o G			w/ G
		#Params	NFE	FID↓	FID↓
GAN	BigGAN-deep (Brock, 2018)	112M	1	6.95	-
	GigaGAN (Kang et al., 2023)	569M	1	3.45	-
	StyleGAN-XL (Sauer et al., 2022)	166M	1	2.30	-
Diffusion	ADM (Dhariwal & Nichol, 2021)	554M	250	10.94	4.59
	ADM-U	608M	500	7.49	3.94
	LDM-4 (Rombach et al., 2022)	400M	250	10.56	3.60
	DiT-XL/2 (Peebles & Xie, 2023)	675M	250	9.62	2.27
	SiT-XL (Ma et al., 2024)	675M	250	8.3	2.06
	+ REPA (Yu et al., 2024b)	675M	250	5.90	1.42
	RIN (Jabri et al., 2022)	410M	1000	3.42	-
	U-ViT, L (Hoogeboom et al., 2023)	2B	512	2.77	2.44
	VDM++ (Kingma & Gao, 2024)	2B	512	2.40	2.12
	LightningDiT (Yao et al., 2025)	675M	250	2.17	1.35
Masked	MaskGIT (Chang et al., 2022)	227M	8	6.18	-
	MAGVIT-v2 (Yu et al., 2023)	307M	64	3.65	1.78
	MAR-H (Li et al., 2024)	943M	256	2.35	1.55
	MaskBit (Weber et al., 2024)	305M	256	-	1.52
AR	VQGAN (Esser et al., 2021)	1.4B	256	15.78	-
	ViT-VQGAN (Yu et al., 2021)	1.7B	1024	4.17	-
	RQ-Transformer (Lee et al., 2022)	3.8B	68	7.55	-
	LlamaGen-3B (Sun et al., 2024)	3.1B	256	13.58	3.05
	Open-MAGVIT2-XL (Luo et al., 2024)	1.5B	256	9.63	2.33
	VAR-d16 (Tian et al., 2024)	310M	10	3.62	3.30
	VAR-d30	2.0B	10	2.17	1.90
	RAR-XXL (Yu et al., 2024a)	1.5B	256	3.91	1.48
	xAR-H (Ren et al., 2025)	1.1B	50	-	1.24
Ours	VAR-d16 (w/o sampling tricks)	310M	10	11.33	3.71
	+ DDO	310M	10	**3.12**	**2.54**
	VAR-d30 (w/o sampling tricks)	2.0B	10	4.74	1.92
	+ CCA (Chen et al., 2024a)	2.0B	10	2.54	-
	+ DDO	2.0B	10	**1.79**	**1.73**
	EDM2-L + DDO (*downsampled**)	777M	63	**1.26**	**1.21**
	+ DPM-Solver-v3 (Zheng et al., 2023a)	777M	25	1.29	**1.21**

Main Results Table 4 presents the quantitative results on ImageNet 256×256. Figure 7(a)(b) illustrates the FID-IS trade-off varying the CFG scale. We summarize the advantages of DDO as follows:

(1) Eliminating sampling tricks. It is worth noting that the original VAR results are based on top-k and top-p sampling strategies, which artificially lower the temperature. These heuristics introduce a training-inference gap and fail to reflect the genuine capability of pretrained models. In contrast, when evaluating models finetuned with DDO, we discard all such tricks, ensuring a more principled assessment of generative performance.

(2) Guidance-free performance. DDO significantly reduces the guidance-free FID from 11.33/4.74 to 3.12/1.79 for VAR-d16 and VAR-d30, achieving 3.6× and 2.6× improvement. Notably, the guidance-free FIDs even outperform CFG-enhanced FIDs (3.30/1.90) of the original VAR results, indicating that higher-quality samples can be generated at half the inference cost with DDO. This is also superior to methods like CCA (Chen et al., 2024a) which only aim to remove the CFG but harm the model performance.

(3) CFG-enhanced performance. When combined with CFG, the finetuned VAR models achieve significantly better FID-IS trade-offs than the pretrained counterparts, even when the latter employ sampling tricks. The lowest FID improves from 3.30/1.90 to 2.54/1.79. Furthermore, the finetuned VAR-d16 outperforms the 2× larger VAR-d20 (600M parameters) which has a CFG-enhanced FID of 2.57.

Effects of α, β Figure 7(c) visualizes FID and IS varying the CFG scale, where we finetune VAR-d16 under $\beta = 0.02$ and different α for 60 iterations. The results indicate that all $\alpha \in [10, 100]$ consistently achieve a CFG-enhanced FID lower than that of the base model. Larger values of α tend to yield lower guidance-free FIDs but may slightly weaken performance when combined with CFG.

6. Conclusion

In this work, we introduce Direct Discriminative Optimization (DDO), a universal enhancement technique designed for visual likelihood-based generative models. Inspired by the GAN framework and the parameterization insights from DPO, DDO breaks the curse of forward KL and substantially improves generation quality. Extensive experiments demonstrate its remarkable effectiveness and efficiency, surpassing state-of-the-art diffusion and autoregressive models and achieving record-breaking FID scores on standard image benchmarks. There remain promising directions for future exploration, such as eliminating the need for hyperparameter searching, improving inference efficiency through distillation, and scaling to tasks like text-to-image generation. We leave such avenues for future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Bao, F., Xiang, C., Yue, G., He, G., Zhu, H., Zheng, K., Zhao, M., Liu, S., Wang, Y., and Zhu, J. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

Barratt, S. and Sharma, R. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brock, A. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

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

Chen, H., Su, H., Sun, P., and Zhu, J. Toward guidance-free ar visual generation via condition contrastive alignment. *arXiv preprint arXiv:2410.09347*, 2024a.

Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., and Han, S. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024b.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024c.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Dhariwal, P. and Nichol, A. Q. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680, 2014.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Fei-Fei, L., Essa, I., Jiang, L., and Lezama, J. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Hoogeboom, E., Heek, J., and Salimans, T. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023.

Huang, Y., Gokaslan, A., Kuleshov, V., and Tompkin, J. The gan is dead; long live the gan! a modern gan baseline. *arXiv preprint arXiv:2501.05441*, 2025.

Jabri, A., Fleet, D., and Chen, T. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

Kang, M., Zhu, J.-Y., Zhang, R., Park, J., Shechtman, E., Paris, S., and Park, T. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10124–10134, 2023.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Karras, T., Aittala, M., Kynkäänniemi, T., Lehtinen, J., Aila, T., and Laine, S. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024a.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024b.

Kim, D., Kim, Y., Kwon, S. J., Kang, W., and Moon, I.-C. Refining generative process with discriminator guidance in score-based diffusion models. In *International Conference on Machine Learning*, pp. 16567–16598. PMLR, 2023a.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *The Twelfth International Conference on Learning Representations*, 2023b.

Kim, D., Lai, C.-H., Liao, W.-H., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y., and Ermon, S. Pagoda: Progressive growing of a one-step generator from a low-resolution diffusion teacher. *arXiv preprint arXiv:2405.14822*, 2024.

Kingma, D. and Gao, R. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *Advances in Neural Information Processing Systems*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lee, D., Kim, C., Kim, S., Cho, M., and Han, W.-S. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11523–11532, 2022.

Li, T., Tian, Y., Li, H., Deng, M., and He, K. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

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

Lou, A., Meng, C., and Ermon, S. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., and Zhu, J. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *International Conference on Machine Learning*, pp. 14429–14460. PMLR, 2022a.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022b.

Luo, Z., Shi, F., Ge, Y., Yang, Y., Wang, L., and Shan, Y. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024.

Ma, N., Goldstein, M., Albergo, M. S., Boffi, N. M., Vanden-Eijnden, E., and Xie, S. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Nowozin, S., Cseke, B., and Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Ren, S., Yu, Q., He, J., Shen, X., Yuille, A., and Chen, L.-C. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Sahoo, S. S., Deschenaux, J., Gokaslan, A., Wang, G., Chiu, J. T., and Kuleshov, V. The diffusion duality. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Sauer, A., Schwarz, K., and Geiger, A. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. K. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1415–1428, 2021a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., and Yuan, Z. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

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

Tschannen, M., Eastwood, C., and Mentzer, F. Givt: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pp. 292–309. Springer, 2024.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11287–11302, 2021.

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

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Wang, Z., Zheng, H., He, P., Chen, W., and Zhou, M. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

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

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022.

Xie, J., Mao, W., Bai, Z., Zhang, D. J., Wang, W., Lin, K. Q., Gu, Y., Chen, Z., Yang, Z., and Shou, M. Z. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Xu, J., Lee, A., Sukhbaatar, S., and Weston, J. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

Xu, Y., Corso, G., Jaakkola, T., Vahdat, A., and Kreis, K. Disco-diff: Enhancing continuous diffusion models with discrete latents. *arXiv preprint arXiv:2407.03300*, 2024.

Yao, J., Yang, B., and Wang, X. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., and Freeman, W. T. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024.

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

Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Birodkar, V., Gupta, A., Gu, X., et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

Yu, Q., He, J., Deng, X., Shen, X., and Chen, L.-C. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024a.

Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., and Xie, S. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024b.

Zhang, J., Huang, H., Zhang, P., Wei, J., Zhu, J., and Chen, J. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In *International Conference on Machine Learning (ICML)*, 2025a.

Zhang, J., Wei, J., Zhang, P., Xu, X., Huang, H., Wang, H., Jiang, K., Zhu, J., and Chen, J. Sageattention3: Microscaling fp4 attention for inference and an exploration of 8-bit training. *arXiv preprint arXiv:2505.11594*, 2025b.

Zhang, J., Wei, J., Zhang, P., Zhu, J., and Chen, J. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations (ICLR)*, 2025c.

Zhang, J., Xiang, C., Huang, H., Wei, J., Xi, H., Zhu, J., and Chen, J. Spargeattn: Accurate sparse attention accelerating any model inference. In *International Conference on Machine Learning (ICML)*, 2025d.

Zheng, K., Lu, C., Chen, J., and Zhu, J. DPM-Solver-v3: Improved diffusion ode solver with empirical model statistics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Zheng, K., Lu, C., Chen, J., and Zhu, J. Improved techniques for maximum likelihood estimation for diffusion odes. In *International Conference on Machine Learning*, pp. 42363–42389. PMLR, 2023b.

Zheng, K., Chen, Y., Mao, H., Liu, M.-Y., Zhu, J., and Zhang, Q. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

Zhou, C., Yu, L., Babu, A., Tirumala, K., Yasunaga, M., Shamis, L., Kahn, J., Ma, X., Zettlemoyer, L., and Levy, O. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024a.

Zhou, M., Zheng, H., Gu, Y., Wang, Z., and Huang, H. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. *arXiv preprint arXiv:2410.14919*, 2024b.

A. Related Work

Paradigms of Generative Models Beyond autoregressive (AR) and diffusion models introduced in Section 2.1, other likelihood-based generative models have been explored in the literature. Variational autoencoders (VAEs) (Kingma & Welling, 2014), energy-based models (Du & Mordatch, 2019), and normalizing flows (Dinh et al., 2016) are once popular for generative modeling of continuous data, but have since fallen out of favor due to their limited expressiveness and lack of scalability in modern large-scale generation tasks. In particular, VAEs are now primarily used as dimensionality reduction tools that compress data into latent spaces (Esser et al., 2021; Rombach et al., 2022), rather than generating samples from scratch. For discrete data generation, masked models, such as BERT (Devlin et al., 2019), CMLM (Ghazvininejad et al., 2019) for masked language modeling and MaskGIT (Chang et al., 2022) for masked image generation, offer an alternative likelihood-based paradigm to AR. While discrete diffusion models (Austin et al., 2021; Lou et al., 2023; Shi et al., 2024; Sahoo et al., 2024) have recently regained interest, they are largely equivalent to the simpler masked models and suffer from hidden numerical precision issues that lead to unfair evaluation when measured by the generative perplexity metric alone (Zheng et al., 2024), suggesting that the notion of "diffusion" is actually unnecessary or even harmful in these models[5]. There have also been pioneering efforts to combine different generative paradigms. MAR (Li et al., 2024) integrates masked modeling with the diffusion loss, enabling autoregressive image generation with continuous tokens. Transfusion (Zhou et al., 2024a) and Show-o (Xie et al., 2024) combine AR with diffusion/masked models for multi-modal generation, effectively synthesizing a mixture of text and image data. DDO is potentially applicable to these models for quality enhancement, and we leave such explorations for future work.

Improving Generation Quality with GAN Except for using GAN as an auxiliary loss for enhancing one-step or few-step generation in diffusion distillation (Kim et al., 2023b; Yin et al., 2024; Zhou et al., 2024b), as mentioned in Section 2.2, several works have explored directly integrating diffusion models with GANs. Notably, Xiao et al. (2022) replaces the reverse denoising steps in diffusion models with a sequence of conditional GAN generators, enabling few-step generation. Wang et al. (2022) modifies the GAN discriminator to distinguish between diffused real and generated samples in a time-aware manner. Kim et al. (2023a) leverages the gradient information from a trained discriminator to refine pretrained diffusion models. Chen et al. (2024a) adopts a binary classification loss with likelihood ratio parameterization similar to our objective, but its applicability is limited to removing CFG in autoregressive models and degrades the model performance. Apart from quality, quantized or sparse attention (Zhang et al., 2025c;a;b;d) can be employed in parallel to accelerate inference.

B. Theoretical Analyses of the DDO Objective

In this section, we investigate the theoretical properties of the DDO objective and provide informal proofs to Theorem 3.1, Theorem 3.2, and Theorem 3.3 in the main text.

B.1. Analyses of $\mathcal{L}(\theta)$

Optimal Solution It is straightforward to show that the optimal θ minimizing $\mathcal{L}(\theta)$ satisfies $p_{\theta^*} = p_{\text{data}}$ following the common GAN literature (Goodfellow et al., 2014). Specifically, let $r_\theta(\boldsymbol{x}) \coloneqq \log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}$ denote the log-likelihood ratio between the learnable and reference distribution. The objective $\mathcal{L}(\theta)$ can be expressed as an integral form:

$$\mathcal{L}(\theta) = \int \mathcal{L}(\theta)_{\boldsymbol{x}} \mathrm{d}\boldsymbol{x} \tag{20}$$

where

$$\mathcal{L}(\theta)_{\boldsymbol{x}} = -p_{\text{data}}(\boldsymbol{x}) \log \sigma(r_\theta(\boldsymbol{x})) - p_{\theta_{\text{ref}}}(\boldsymbol{x}) \log(1 - \sigma(r_\theta(\boldsymbol{x}))) > 0 \tag{21}$$

is the pointwise loss, and we only consider $\boldsymbol{x}$ in the valid range where p_{data} and $p_{\theta_{\text{ref}}}$ have nonzero support. For any $(a, b) \in \mathbb{R}^2 \backslash \{(0, 0)\}$, the function $y \to -a \log y - b \log(1 - y), y \in [0, 1]$ achieves its minimum at $\frac{a}{a+b}$. Applying this to the pointwise loss $\mathcal{L}(\theta)_{\boldsymbol{x}}$, the minimizer satisfies

$$\sigma(r_{\theta^*}(\boldsymbol{x})) = \frac{p_{\theta^*}(\boldsymbol{x})}{p_{\theta^*}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} \Rightarrow p_{\theta^*}(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x}) \tag{22}$$

Since the global minimizer of $\mathcal{L}(\theta)$ is the pointwise minimizer of $\mathcal{L}(\theta)_{\boldsymbol{x}}$ for all $\boldsymbol{x}$, it follows that $p_{\theta^*} = p_{\text{data}}$.

[5] Apart from the masked case, other discrete diffusion variants, such as uniform, have a more genuine connection to diffusion (Sahoo et al., 2025).

Loss Gradient Using the derivative identity $\frac{\mathrm{d}\sigma(x)}{\mathrm{d}x} = \sigma(x)(1-\sigma(x))$, we obtain $\frac{\mathrm{d}\log\sigma(x)}{\mathrm{d}x} = 1-\sigma(x)$, $\frac{\mathrm{d}\log(1-\sigma(x))}{\mathrm{d}x} = -\sigma(x)$. Applying these to the pointwise loss, we derive the gradient w.r.t. $r_\theta(\boldsymbol{x})$:

$$\frac{\mathrm{d}\mathcal{L}(\theta)_{\boldsymbol{x}}}{\mathrm{d}r_\theta(\boldsymbol{x})} = p_{\theta_{\text{ref}}}(\boldsymbol{x})\sigma(r_\theta(\boldsymbol{x})) - p_{\text{data}}(\boldsymbol{x})(1-\sigma(r_\theta(\boldsymbol{x}))) = \frac{p_\theta(\boldsymbol{x}) - p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} p_{\theta_{\text{ref}}}(\boldsymbol{x}) = (1-d_\theta(\boldsymbol{x}))(p_\theta(\boldsymbol{x}) - p_{\text{data}}(\boldsymbol{x})) \tag{23}$$

Thus, the full loss gradient is given by

$$\nabla_\theta \mathcal{L}(\theta) = \int \nabla_\theta \mathcal{L}(\theta)_{\boldsymbol{x}} \mathrm{d}\boldsymbol{x} = \int \frac{\mathrm{d}\mathcal{L}(\theta)_{\boldsymbol{x}}}{\mathrm{d}r_\theta(\boldsymbol{x})} \nabla_\theta r_\theta(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int (1-d_\theta(\boldsymbol{x}))(p_\theta(\boldsymbol{x}) - p_{\text{data}}(\boldsymbol{x})) \nabla_\theta \log p_\theta(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{24}$$

Divergence Bounds We aim to derive bounds for the divergence between p_θ and p_{data} when θ is not optimal, using the difference between the loss value $\mathcal{L}(\theta)$ and its optimal counterpart $\mathcal{L}^*$. Without any assumptions, the forward KL divergence $D_{\mathrm{KL}}(p_{\text{data}} \parallel p_\theta)$ is lower bounded by $\mathcal{L}(\theta) - \mathcal{L}^*$. By definition, $\mathcal{L}^*$ is the minimum loss value achieved when $p_\theta = p_{\text{data}}$. The difference $\mathcal{L}(\theta) - \mathcal{L}^*$ can be decomposed as follows:

$$\begin{aligned}
\mathcal{L}(\theta) - \mathcal{L}^* =& -\int p_{\text{data}}(\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} + p_{\theta_{\text{ref}}}(\boldsymbol{x}) \log \frac{p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\
&+ \int p_{\text{data}}(\boldsymbol{x}) \log \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} + p_{\theta_{\text{ref}}}(\boldsymbol{x}) \log \frac{p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\
=& \int p_{\text{data}}(\boldsymbol{x}) \log \frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} + (p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})) \log \frac{p_\theta(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\
=& D_{\mathrm{KL}}(p_{\text{data}} \parallel p_\theta) - D_{\mathrm{KL}}(\frac{p_{\text{data}} + p_{\theta_{\text{ref}}}}{2} \parallel \frac{p_\theta + p_{\theta_{\text{ref}}}}{2})
\end{aligned} \tag{25}$$

Therefore, $D_{\mathrm{KL}}(p_{\text{data}} \parallel p_\theta) = \mathcal{L}(\theta) - \mathcal{L}^* + D_{\mathrm{KL}}(\frac{p_{\text{data}}+p_{\theta_{\text{ref}}}}{2} \parallel \frac{p_\theta+p_{\theta_{\text{ref}}}}{2}) \geq \mathcal{L}(\theta) - \mathcal{L}^*$. While this result establishes a lower bound for the divergence, additional assumptions are required to derive an upper bound. Specifically, we assume that $\log \frac{p_{\theta_{\text{ref}}}}{p_{\text{data}}}$ and $\log \frac{p_\theta}{p_{\theta_{\text{ref}}}}$ are bounded, i.e., there exist constants M, M_1, M_2 such that $|r_\theta(\boldsymbol{x})| = \left|\log \frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right| \leq M, M_1 \leq \log \frac{p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})} \leq M_2$ for all $\boldsymbol{x}$. The pointwise loss can be expressed as a function of $r_\theta(\boldsymbol{x})$:

$$\mathcal{L}(\theta)_{\boldsymbol{x}} = f(r_\theta(\boldsymbol{x})) \tag{26}$$

where

$$f(y) \coloneqq -p_{\text{data}}(\boldsymbol{x}) \log \sigma(y) - p_{\theta_{\text{ref}}}(\boldsymbol{x}) \log(1-\sigma(y)) = (p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})) \log(1+e^y) - p_{\text{data}} y \tag{27}$$

The first and second order derivatives of f are given by:

$$f'(y) = \frac{p_{\theta_{\text{ref}}}(\boldsymbol{x})e^y - p_{\text{data}}(\boldsymbol{x})}{1+e^y}, \quad f''(y) = \frac{(p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x}))e^y}{(1+e^y)^2} = \frac{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})}{2 + e^y + e^{-y}} \tag{28}$$

Applying Taylor's expansion at $y = r_{\theta^*}(\boldsymbol{x}) = \log \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}$, we obtain:

$$f(r_\theta(\boldsymbol{x})) = f(r_{\theta^*}(\boldsymbol{x})) + f'(r_{\theta^*}(\boldsymbol{x}))(r_\theta(\boldsymbol{x}) - r_{\theta^*}(\boldsymbol{x})) + \frac{1}{2} f''(\xi)(r_\theta(\boldsymbol{x}) - r_{\theta^*}(\boldsymbol{x}))^2 \tag{29}$$

where $\xi \in [\min\{r_\theta(\boldsymbol{x}), r_{\theta^*}(\boldsymbol{x})\}, \max\{r_\theta(\boldsymbol{x}), r_{\theta^*}(\boldsymbol{x})\}]$. Since $f'(r_{\theta^*}(\boldsymbol{x})) = 0$ and $r_\theta(\boldsymbol{x}) - r_{\theta^*}(\boldsymbol{x}) = \log \frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}$, we get:

$$\left(\log \frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\right)^2 = \frac{2}{f''(\xi)} (\mathcal{L}(\theta)_{\boldsymbol{x}} - \mathcal{L}(\theta^*)_{\boldsymbol{x}}) \tag{30}$$

Note that $f''(y)$ is a monotonically decreasing function w.r.t. $|y|$ and attains its maximum at the boundary of the given range, we have:

$$\begin{aligned}
\frac{2}{f''(\xi)} &\leq \max\left\{\frac{2}{f''(r_\theta(\boldsymbol{x}))}, \frac{2}{f''(r_{\theta^*}(\boldsymbol{x}))}\right\} \leq \max\left\{\frac{2}{f''(M)}, \frac{2}{f''(r_{\theta^*}(\boldsymbol{x}))}\right\} \\
&= 2\max\left\{\frac{2 + e^{-M} + e^M}{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})}, \frac{p_{\text{data}}(\boldsymbol{x}) + p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right\}
\end{aligned} \tag{31}$$

Therefore,

$$\begin{aligned} p_{\text{data}}(\boldsymbol{x})\left(\log\frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\right)^2 &\le 2\max\left\{\frac{2+e^{-M}+e^{M}}{1+p_{\theta_{\text{ref}}}(\boldsymbol{x})/p_{\text{data}}(\boldsymbol{x})}, \frac{p_{\text{data}}(\boldsymbol{x})+p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right\}(\mathcal{L}(\theta)_{\boldsymbol{x}}-\mathcal{L}(\theta^*)_{\boldsymbol{x}}) \\ &\le 2\max\left\{\frac{2+e^{-M}+e^{M}}{1+e^{M_1}}, 1+e^{-M_1}\right\}(\mathcal{L}(\theta)_{\boldsymbol{x}}-\mathcal{L}(\theta^*)_{\boldsymbol{x}}) \end{aligned} \tag{32}$$

Applying Jensen's inequality, we derive an upper bound for the forward KL divergence:

$$\begin{aligned} D_{\text{KL}}(p_{\text{data}} \parallel p_\theta) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\log\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right] &\le \sqrt{\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\left(\log\frac{p_{\text{data}}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right)^2\right]} = \sqrt{\int p_{\text{data}}(\boldsymbol{x})\left(\log\frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\right)^2 \mathrm{d}\boldsymbol{x}} \\ &\le C_1\sqrt{\mathcal{L}(\theta)-\mathcal{L}^*} \end{aligned} \tag{33}$$

where $C_1 = \sqrt{2\max\left\{\frac{2+e^{-M}+e^{M}}{1+e^{M_1}}, 1+e^{-M_1}\right\}}$ is related to the lower bound of $\log\frac{p_{\theta_{\text{ref}}}}{p_{\text{data}}}$. Similarly,

$$\begin{aligned} p_\theta(\boldsymbol{x})\left(\log\frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\right)^2 &\le 2\frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\max\left\{\frac{2+e^{-M}+e^{M}}{1+p_{\text{data}}(\boldsymbol{x})/p_{\theta_{\text{ref}}}(\boldsymbol{x})}, \frac{p_{\text{data}}(\boldsymbol{x})+p_{\theta_{\text{ref}}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\right\}(\mathcal{L}(\theta)_{\boldsymbol{x}}-\mathcal{L}(\theta^*)_{\boldsymbol{x}}) \\ &\le 2e^{M}\max\left\{\frac{2+e^{-M}+e^{M}}{1+e^{M_2}}, 1+e^{-M_2}\right\}(\mathcal{L}(\theta)_{\boldsymbol{x}}-\mathcal{L}(\theta^*)_{\boldsymbol{x}}) \end{aligned} \tag{34}$$

By integrating over $\boldsymbol{x}$, we obtain $D_{\text{KL}}(p_\theta \parallel p_{\text{data}}) \le C_2\sqrt{\mathcal{L}(\theta)-\mathcal{L}^*}$, where $C_2 = \sqrt{2e^{M}\max\left\{\frac{2+e^{-M}+e^{M}}{1+e^{M_2}}, 1+e^{-M_2}\right\}}$ is related to the upper bound of $\log\frac{p_{\theta_{\text{ref}}}}{p_{\text{data}}}$.

B.2. Analyses of $\mathcal{L}_{\alpha,\beta}(\theta)$

Once we introduce additional coefficients α, β, the generalized DDO objective $\mathcal{L}_{\alpha,\beta}(\theta)$ may become intractable and no longer admit $p_{\theta^*} = p_{\text{data}}$ as the optimal solution. Specifically, the pointwise loss with α, β is

$$\mathcal{L}_{\alpha,\beta}(\theta)_{\boldsymbol{x}} = -p_{\text{data}}(\boldsymbol{x})\log\sigma(\beta r_\theta(\boldsymbol{x})) - \alpha p_{\theta_{\text{ref}}}(\boldsymbol{x})\log(1-\sigma(\beta r_\theta(\boldsymbol{x}))) \tag{35}$$

The optimal θ should satisfy

$$\begin{aligned} &\frac{\mathrm{d}\mathcal{L}_{\alpha,\beta}(\theta)_{\boldsymbol{x}}}{\mathrm{d}r_\theta(\boldsymbol{x})} = \alpha\beta p_{\theta_{\text{ref}}}(\boldsymbol{x})\sigma(\beta r_\theta(\boldsymbol{x})) - \beta p_{\text{data}}(\boldsymbol{x})(1-\sigma(\beta r_\theta(\boldsymbol{x}))) = 0 \\ \Rightarrow &\sigma(\beta r_\theta(\boldsymbol{x})) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+\alpha p_{\theta_{\text{ref}}}(\boldsymbol{x})} = \sigma\left(\log\frac{p_{\text{data}}(\boldsymbol{x})}{\alpha p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right) \\ \Rightarrow &p_\theta(\boldsymbol{x}) = p_{\theta_{\text{ref}}}(\boldsymbol{x})\left(\frac{p_{\text{data}}(\boldsymbol{x})}{\alpha p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right)^{1/\beta} \end{aligned} \tag{36}$$

However, since p_θ is parameterized as a likelihood-based generative model, it must have self-normalized density. This optimality condition is only achieved when α is a proper normalizing constant satisfying

$$\int p_\theta(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1 \Rightarrow \alpha = \left(\int p_{\theta_{\text{ref}}}^{1-1/\beta}(\boldsymbol{x})p_{\text{data}}^{1/\beta}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^{\beta} \tag{37}$$

Under this specific choice of α, the optimal solution $p_{\theta^*} \propto p_{\theta_{\text{ref}}}^{1-1/\beta}p_{\text{data}}^{1/\beta}$. Otherwise, the optimization is subject to the constraint $\int p_\theta(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1$. To enforce this, we introduce a Lagrange multiplier λ and define the Lagrangian:

$$\mathcal{L} = \int \mathcal{L}_{\alpha,\beta}(\theta)_{\boldsymbol{x}}\mathrm{d}\boldsymbol{x} + \lambda\left(1-\int p_\theta(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right) \tag{38}$$

To find the optimal p_θ, we take the functional derivative of $\mathcal{L}$ w.r.t. p_θ and set it to zero:

$$\frac{\delta \mathcal{L}}{\delta p_\theta(\boldsymbol{x})} = \frac{\mathrm{d}\mathcal{L}_{\alpha,\beta}(\theta)_{\boldsymbol{x}}}{\mathrm{d}p_\theta(\boldsymbol{x})} - \lambda = (\alpha\beta p_{\theta_{\text{ref}}}(\boldsymbol{x})\sigma(\beta r_\theta(\boldsymbol{x})) - \beta p_{\text{data}}(\boldsymbol{x})(1-\sigma(\beta r_\theta(\boldsymbol{x})))) \frac{\mathrm{d}r_\theta(\boldsymbol{x})}{\mathrm{d}p_\theta(\boldsymbol{x})} - \lambda = 0 \tag{39}$$

which can be simplified to

$$\alpha \left(\frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right)^\beta - \frac{\lambda}{\beta}\frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\left[1 + \left(\frac{p_\theta(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})}\right)^\beta\right] = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\theta_{\text{ref}}}(\boldsymbol{x})} \tag{40}$$

This equation, combined with the constraint $\int p_\theta(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1$, determines both λ and the optimal p_θ. However, no closed-form solution exists for this problem. Despite this, we can expect certain ranges of α to skew the optimal solution away from p_{data} toward the direction of $p_{\theta_{\text{ref}}}^{1-1/\beta} p_{\text{data}}^{1/\beta}$, even if the exact equality does not hold.

C. Experiment Details

Throughout all experiments, each training run under a given set of configurations (certain reference model and hyperparameters α, β) is conducted on a single node with 8 NVIDIA A100 (SXM4-80GB) GPUs.

Diffusion Models We follow the parameterization and noise schedule of EDM (Karras et al., 2022) and EDM2 (Karras et al., 2024b). Specifically, EDM introduces a time-dependent skip connection that preconditions the denoiser $\boldsymbol{D}_\theta$ (which predicts clean data $\boldsymbol{x}_0$) using a free-form network $\boldsymbol{F}_\theta$, allowing $\boldsymbol{F}_\theta$ to predict an adaptive mixture of signal and noise:

$$\boldsymbol{D}_\theta(\boldsymbol{x}_t, t) = c_{\text{skip}}(t)\boldsymbol{x}_t + c_{\text{out}}(t)\boldsymbol{F}_\theta(c_{\text{in}}(t)\boldsymbol{x}_t, c_{\text{noise}}(t)) \tag{41}$$

where

$$c_{\text{skip}}(t) = \frac{\sigma_{\text{data}}^2}{\sigma_{\text{data}}^2 + t^2}, \quad c_{\text{out}}(t) = \frac{\sigma_{\text{data}} t}{\sqrt{\sigma_{\text{data}}^2 + t^2}}, \quad c_{\text{in}}(t) = \frac{1}{\sqrt{\sigma_{\text{data}}^2 + t^2}}, \quad c_{\text{noise}}(t) = \frac{1}{4}\log t \tag{42}$$

EDM employs a simple variance exploding (VE) noise schedule satisfying $\alpha_t = 1, \sigma_t = t$. It's worth noting that the preconditioning used in EDM actually transforms it into v-prediction under the variance preserving (VP) noise schedule, owing to the normalizing factor $c_{\text{in}}(t)$ and the skip connection coefficients $c_{\text{skip}}(t), c_{\text{out}}(t)$ (Zheng et al., 2023b). The EDM models are pretrained by a F-prediction MSE loss:

$$\mathcal{L}^{\text{EDM}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0 \sim p_{\text{data}}, t\sim p(t), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\|\boldsymbol{F}_\theta(c_{\text{in}}(t)\boldsymbol{x}_t, c_{\text{noise}}(t)) - \hat{\boldsymbol{F}}(\boldsymbol{x}_0, \boldsymbol{x}_t, t)\|_2^2\right] \tag{43}$$

where $\boldsymbol{x}_t = \boldsymbol{x}_0 + t\boldsymbol{\epsilon}, \hat{\boldsymbol{F}}(\boldsymbol{x}_0, \boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_0 - c_{\text{skip}}(t)\boldsymbol{x}_t}{c_{\text{out}}(t)}$ is the prediction target, and $p(t)$ is a time distribution satisfying $\log t \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$, where $P_{\text{mean}}, P_{\text{std}}$ are hyperparameters.

We adopt similar settings for DDO finetuning. Specifically, we use the approximation in Eqn. (16) and set the weighting $w(t)$ to satisfy $w(t)\|\boldsymbol{\epsilon}_\theta - \boldsymbol{\epsilon}\|_2^2 = \|\boldsymbol{F}_\theta - \hat{\boldsymbol{F}}\|_2^2$, leading to the following objective ($1 - \sigma(x) = \sigma(-x)$):

$$\begin{aligned}\mathcal{L}_{\alpha,\beta}^{\text{EDM-DDO}}(\theta) = -\mathbb{E}_{t\sim p(t), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\Big[&\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)} \log\sigma\left(-\beta\left(\|\boldsymbol{F}_\theta - \hat{\boldsymbol{F}}\|_2^2 - \|\boldsymbol{F}_{\theta_{\text{ref}}} - \hat{\boldsymbol{F}}\|_2^2\right)\right) \\ &+\alpha\mathbb{E}_{p_{\theta_{\text{ref}}}(\boldsymbol{x}_0)} \log\sigma\left(\beta\left(\|\boldsymbol{F}_\theta - \hat{\boldsymbol{F}}\|_2^2 - \|\boldsymbol{F}_{\theta_{\text{ref}}} - \hat{\boldsymbol{F}}\|_2^2\right)\right)\Big]\end{aligned} \tag{44}$$

where we use the same form of time distribution $\log t \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$, and $P_{\text{mean}}, P_{\text{std}}$ are typically the same as pertaining. See Appendix D for the specific implementation. For each finetuning round, we launch $\sim$20 nodes to sweep over the hyperparameters α, β in $[0.5, 6.0] \times [0.01, 0.1]$. We disable all dropout layers in the network to ensure steady improvement. We also find numerical precision crucial for the diffusion DDO loss and disable mixed-precision training. For each round, 50k images are generated offline from the reference model as the reference dataset.

For EDM on CIFAR-10, we finetune the unconditional and class-conditional model for 12 and 16 rounds, respectively. We set $P_{\text{mean}} = -1.2, P_{\text{std}} = 1.2$ throughout all rounds, which is the same as pretraining. Each round has a duration of 1.5M images (30 epochs, 0.75% of pretraining) with a batch size of 512. The learning rate warms up linearly from 0 to $1.5e-4$ during each round, and the data augmentation probability is set to 12% as pretraining. We find the exponential

moving average (EMA) beneficial for stabilizing the model performance and choose a relatively small EMA half-life (0.25M images) as we finetune less duration than pretraining. We evaluate the FID each time trained with 50k images and save the best model for the next round. Each round takes $\sim$3h including both training and evaluation.

For EDM2-S on ImageNet-64, we finetune the model for 24 rounds, where we set $P_{\text{mean}} = -0.8, P_{\text{std}} = 1.6$ for the first 16 rounds following pretraining, and increase P_{std} to 3.0 in the last 8 rounds. Each round has a duration of 6.4M images (5 epochs, 0.6% of pretraining) with a batch size of 512. We use a learning rate of $5e-5$ for the first 16 rounds and $2e-5$ for the last 8 rounds, along with the learning rate scheduler in EDM2 which is a mix of linear warmup and inverse square root decay, where we set the ramp-up to 1M images and the decay reference to 2000 iterations. We follow the power function EMA introduced in EDM2 and set the EMA length to 0.05. We evaluate the FID each time trained with 2^{17} ($\approx$ 131k) images and save the best model for the next round. Each round takes $\sim$1d including both training and evaluation.

For EDM2-L on ImageNet 512×512, we finetune the model for 28 rounds. Following the pretraining setup, we fix $P_{\text{mean}} = -0.4$, while progressively increasing the variance P_{std} to 1.6, 2.0, and 3.0 starting from the 1st, 18th, and 22nd round, respectively. Each round has a duration of 6.4M images (5 epochs, 0.34% of pretraining) with a batch size of 2048. We use a learning rate of $1e-4$ for the first 24 rounds and $5e-5$ for the last 4 rounds, along with the learning rate scheduler in EDM2 which is a mix of linear warmup and inverse square root decay, where we set the ramp-up to 1M images and the decay reference to 800 iterations. We follow the power function EMA introduced in EDM2 and set the EMA length to 0.05. We evaluate the FID each time trained with 2^{18} ($\approx$ 262k) images and save the best model for the next round. Each round takes $\sim$2d including both training and evaluation. We further boost the finetuned model with autoguidance (Karras et al., 2024a). We choose `edm2-img512-xs-0134217-0.165.pkl` from Karras et al. (2024a) as the bad version model and use a small guidance scale of 1.1, which slightly improves the FID from 1.26 to 1.21.

Autoregressive Models We finetune VAR-d16 and VAR-d30 (Tian et al., 2024) both for only 2 rounds. VAR is highly efficient at inference, enabling us to sample from the reference distribution online during training by generating random latent tokens with the reference model conditioned on the same class labels as those in the dataset batch. We disable all dropout layers in the network and set the label dropout probability to 50% for unconditional training. Unlike diffusion DDO, we enable mixed-precision when finetuning VAR. For each round, we launch $\sim$10 nodes to sweep over the hyperparameters α, β in $[10.0, 100.0] \times \{0.02\}$. Each round has a duration of 80 iterations (0.064 epoch, less than 0.03% of pretraining) with a batch size of 1024. We follow the learning rate scheduler in VAR pretraining and set the peak learning rate to a smaller value of $4e-6$. We evaluate the FIDs (corresponding to guidance-free/a moderate CFG scale) every 4 iterations and save the best model for the next round. Each round takes $\sim$5h/7.5h for VAR-d16/d30 including both training and evaluation.

D. Code Example

```python
import torch
import torch.nn.functional as F


# Original diffusion loss of EDM
class EDMLoss:
    def __init__(self, P_mean=-0.4, P_std=1.0, sigma_data=0.5):
        self.P_mean = P_mean
        self.P_std = P_std
        self.sigma_data = sigma_data

    def __call__(self, net, images, labels=None):
        """
        net: the target denoiser network
        images: real samples from the dataset, shape (B, C, H, W)
        """
        # Sample diffusion time
        rnd_normal = torch.randn([images.shape[0], 1, 1, 1], device=images.device)
        sigma = (rnd_normal * self.P_std + self.P_mean).exp()
        # Diffusion loss weighting
        weight = (sigma**2 + self.sigma_data**2) / (sigma * self.sigma_data) ** 2
        # Sample Gaussian noise
        noise = torch.randn_like(images) * sigma
```

```
        # Denoise
        denoised = net(images + noise, sigma, labels)
        # Compute loss
        loss = torch.sum(weight * (denoised - images) ** 2, dim=(1, 2, 3))
        return loss.mean()


# Diffusion DDO loss of EDM
class EDMLoss_DDO:
    def __init__(self, P_mean=-0.4, P_std=1.0, sigma_data=0.5, alpha=1.0, beta=0.02):
        self.P_mean = P_mean
        self.P_std = P_std
        self.sigma_data = sigma_data
        self.alpha = alpha
        self.beta = beta

    def __call__(self, net, ref_net, images, fake_images, labels=None, fake_labels=None):
        """
        net: the target denoiser network
        ref_net: the reference denoiser network (frozen)
        images: real samples from the dataset, shape (B, C, H, W)
        fake_images: fake samples generated by the reference model, shape (B, C, H, W)
        """
        # Sample diffusion time
        rnd_normal = torch.randn([images.shape[0], 1, 1, 1], device=images.device)
        sigma = (rnd_normal * self.P_std + self.P_mean).exp()
        # Diffusion loss weighting
        weight = (sigma**2 + self.sigma_data**2) / (sigma * self.sigma_data) ** 2
        # Sample Gaussian noise
        noise = torch.randn_like(images) * sigma
        # Denoise by the target model
        D = net(images + noise, sigma, labels)
        net.eval()
        D_fake = net(fake_images + noise, sigma, fake_labels)
        net.train()
        D_logp = -torch.sum(weight * (D - images) ** 2, dim=(1, 2, 3))
        D_fake_logp = -torch.sum(weight * (D_fake - fake_images) ** 2, dim=(1, 2, 3))
        # Denoise by the reference model
        with torch.no_grad():
            ref_D = ref_net(images + noise, sigma, labels)
            ref_D_fake = ref_net(fake_images + noise, sigma, fake_labels)
        ref_D_logp = -torch.sum(weight * (ref_D - images) ** 2, dim=(1, 2, 3))
        ref_D_fake_logp = -torch.sum(weight * (ref_D_fake - fake_images) ** 2, dim=(1, 2, 3))
        # Compute loss
        loss = -F.logsigmoid(self.beta * (D_logp - ref_D_logp)) \
               -self.alpha * F.logsigmoid(-self.beta * (D_fake_logp - ref_D_fake_logp))
        return loss.mean()
```

E. Additional Results

Figure 8. Random samples of EDM (CIFAR-10, Unconditional), FID 1.97.

Figure 9. Random samples of EDM + DDO (CIFAR-10, Unconditional), FID 1.38.

Figure 10. Random samples of EDM (CIFAR-10, Class-conditional), FID 1.85.

Figure 11. Random samples of EDM + DDO (CIFAR-10, Class-conditional), FID 1.30.

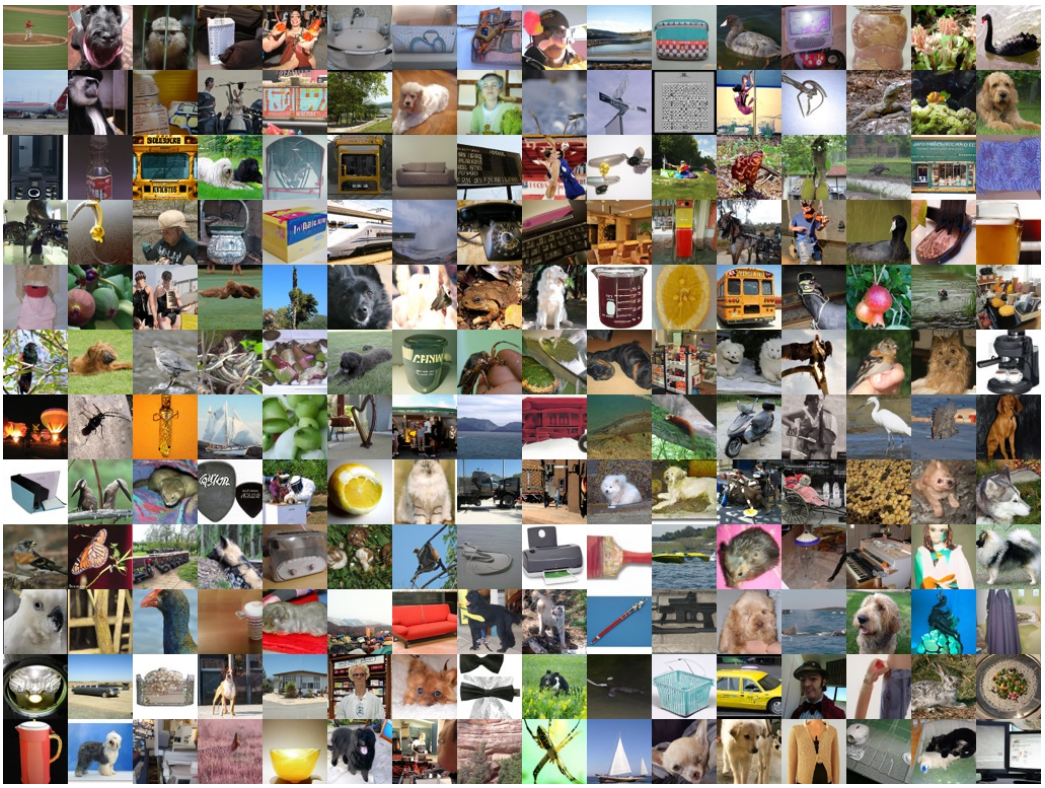


Figure 12. Random samples of EDM2-S (ImageNet-64, Class-conditional), FID 1.60.

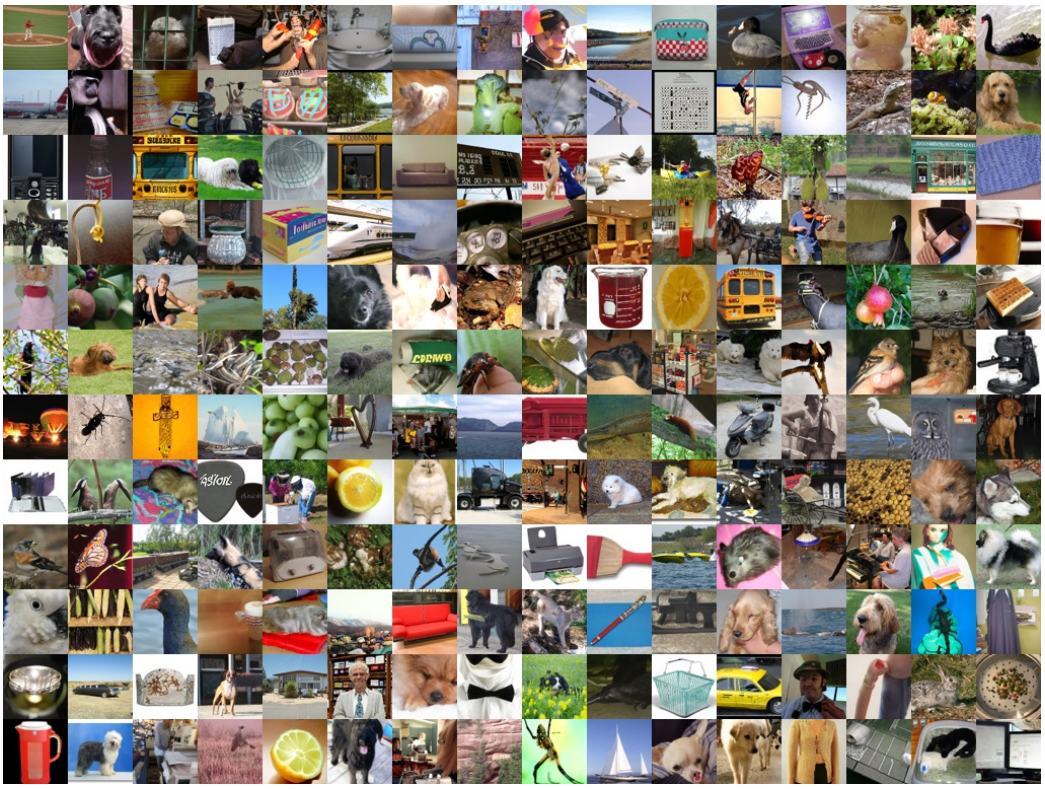


Figure 13. Random samples of EDM2-S + DDO (ImageNet-64, Class-conditional), FID 0.97.

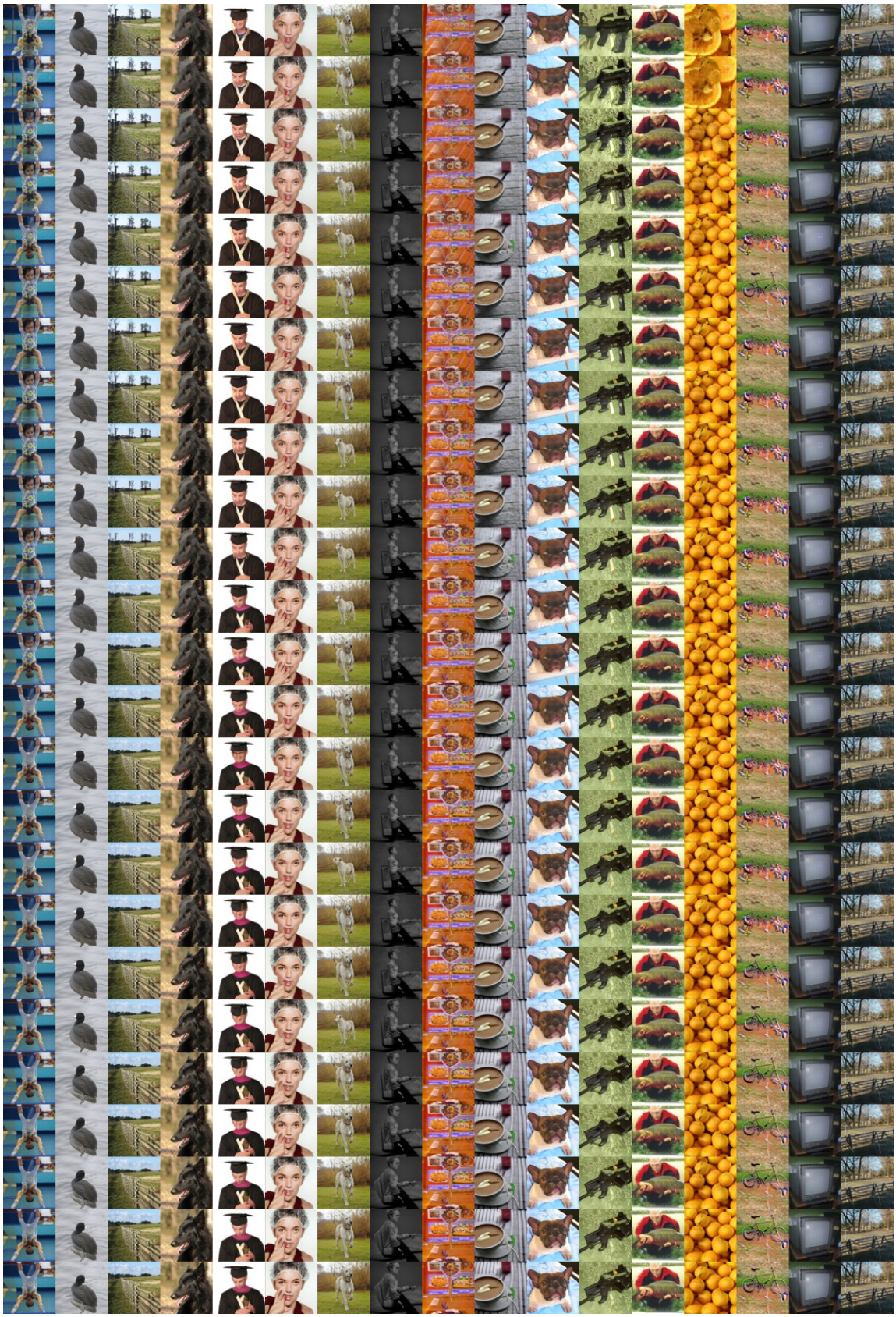
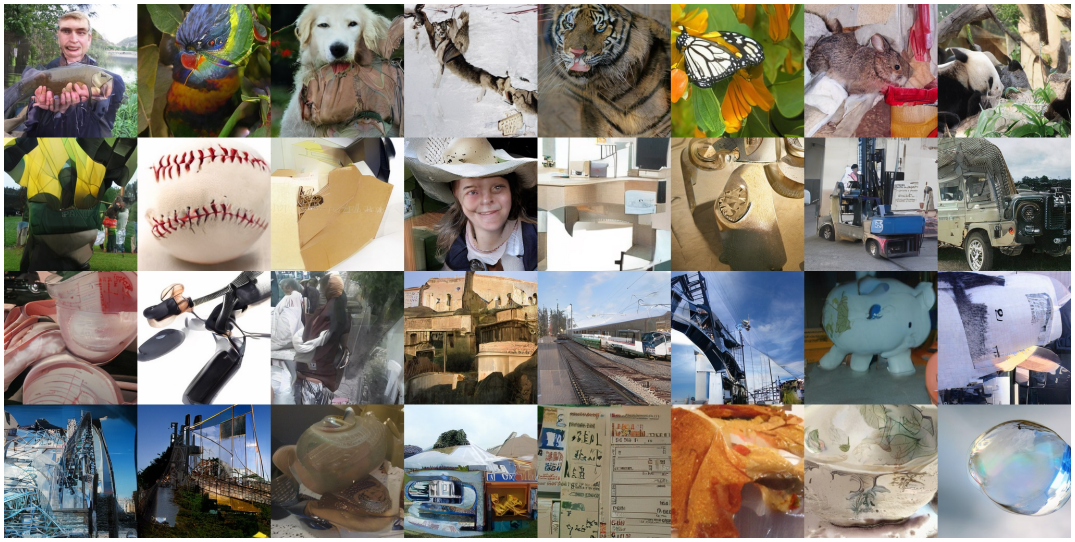
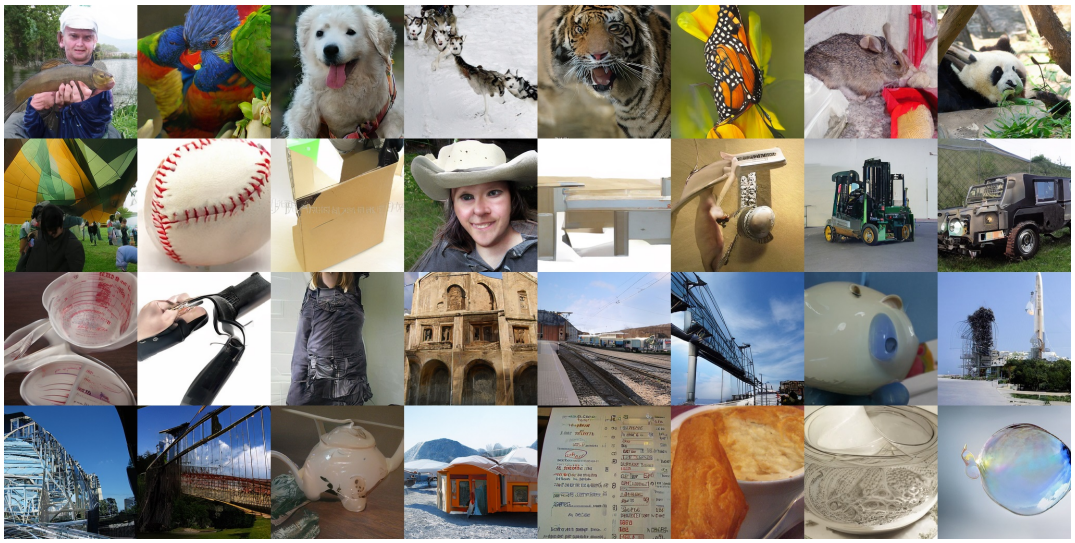


Figure 14. Illustration of the multi-round refinement process on EDM2-S (ImageNet-64).

VAR-d16 w/o trick
(FID 11.33)



VAR-d16 w/ trick
(FID 3.62)



VAR-d16 + DDO
(FID 3.12)

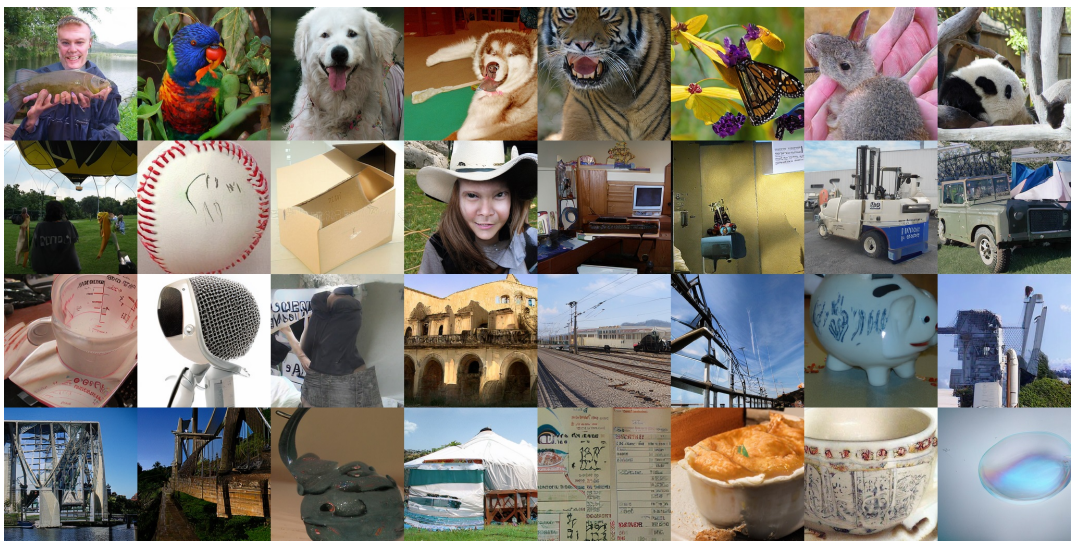
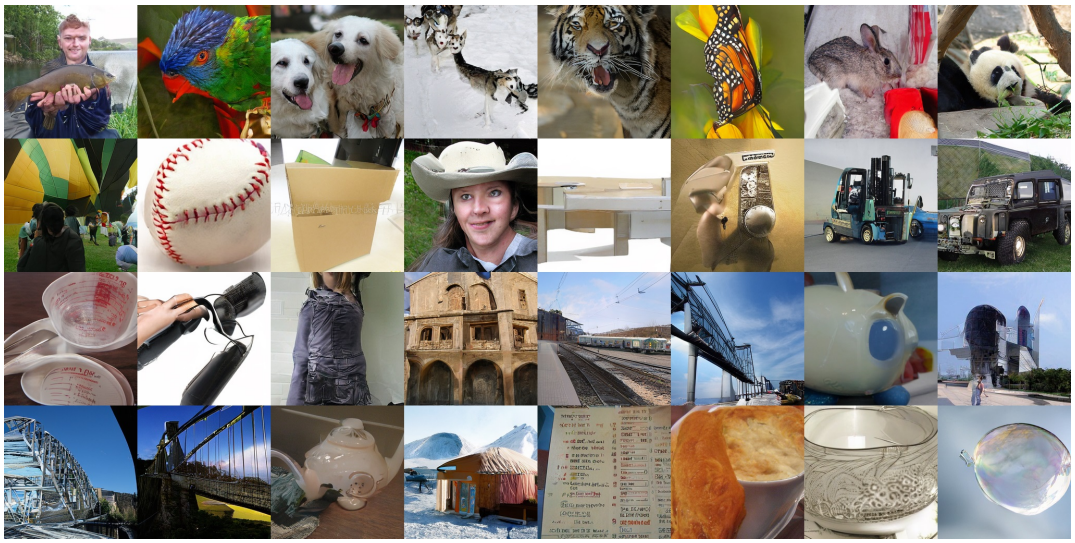


Figure 15. Guidance-free samples by pretrained and finetuned VAR-d16.

VAR-d16 w/o trick
(FID 3.71)

VAR-d16 w/ trick
(FID 3.30)



VAR-d16 + DDO
(FID 2.54)

Figure 16. CFG-enhanced samples by pretrained and finetuned VAR-d16.

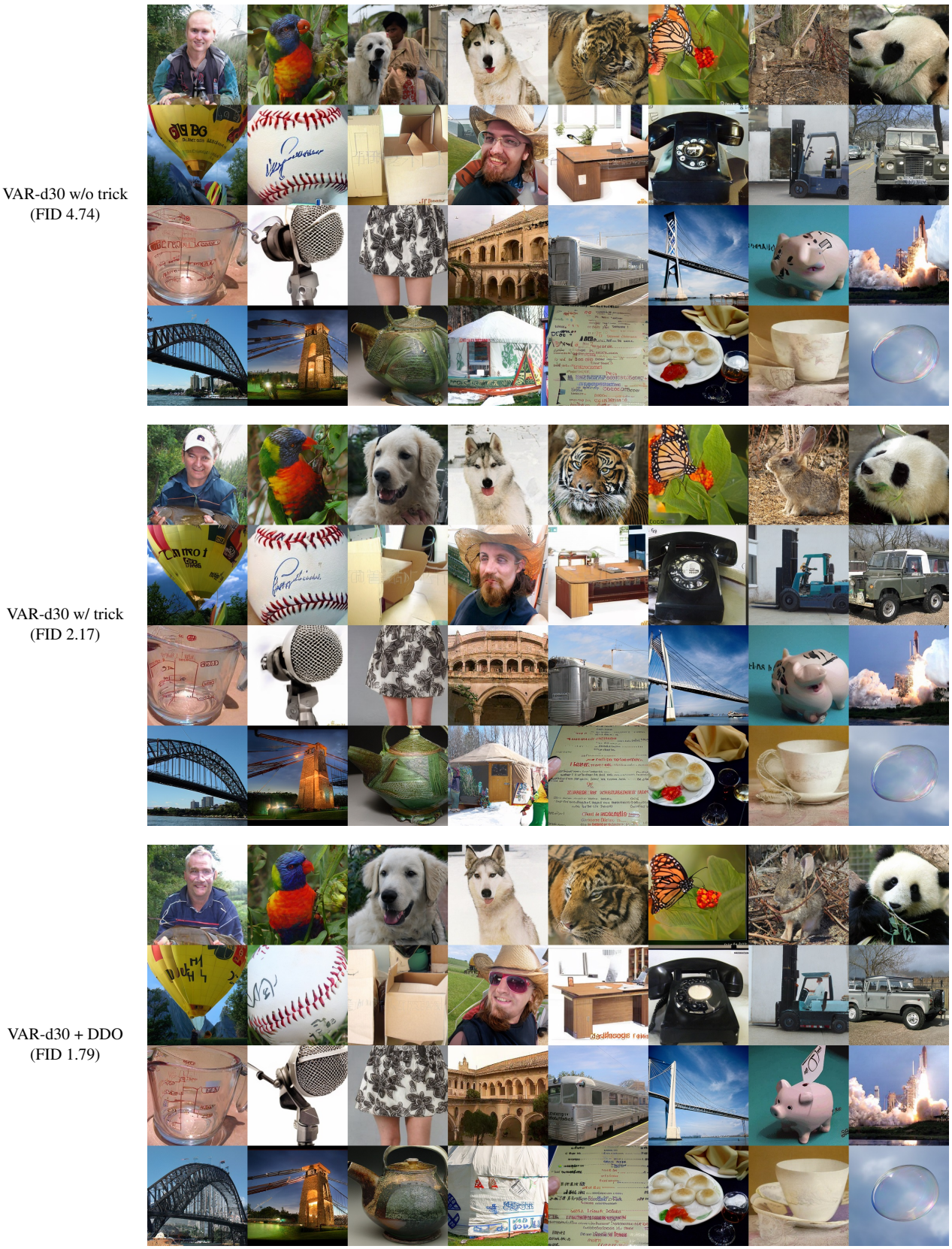
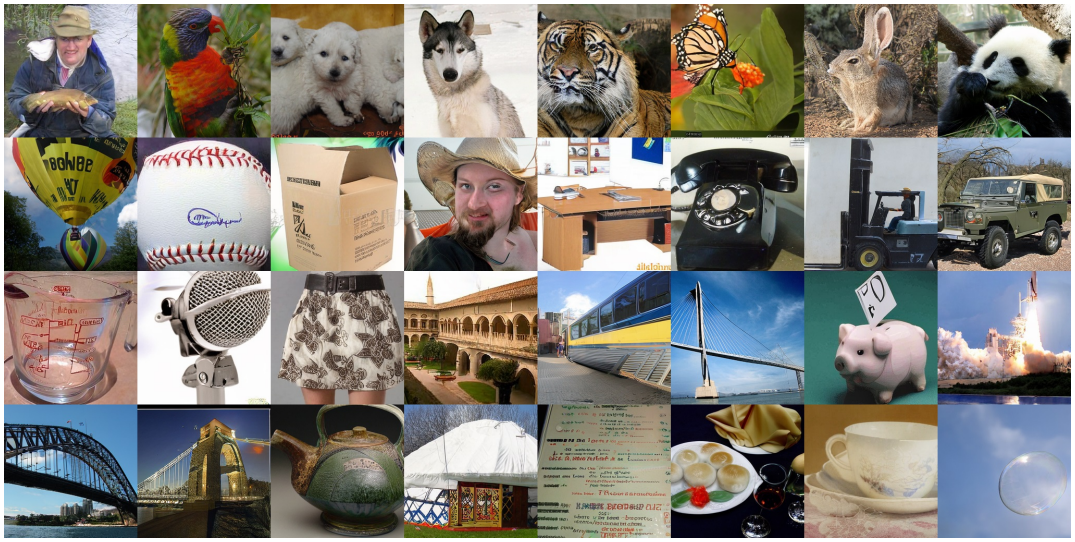
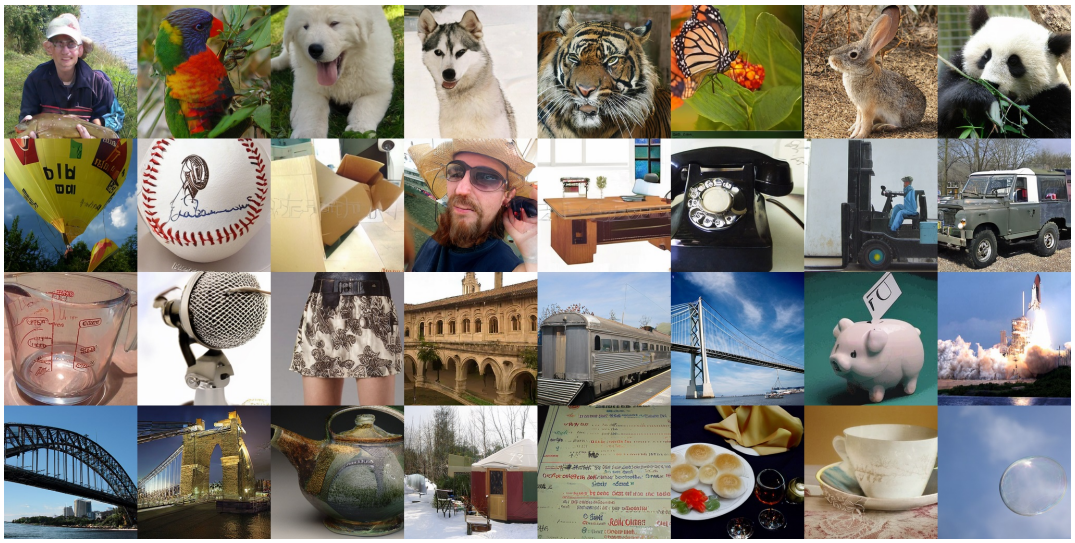


Figure 17. Guidance-free samples by pretrained and finetuned VAR-d30.

VAR-d30 w/o trick
(FID 1.92)



VAR-d30 w/ trick
(FID 1.90)



VAR-d30 + DDO
(FID 1.73)

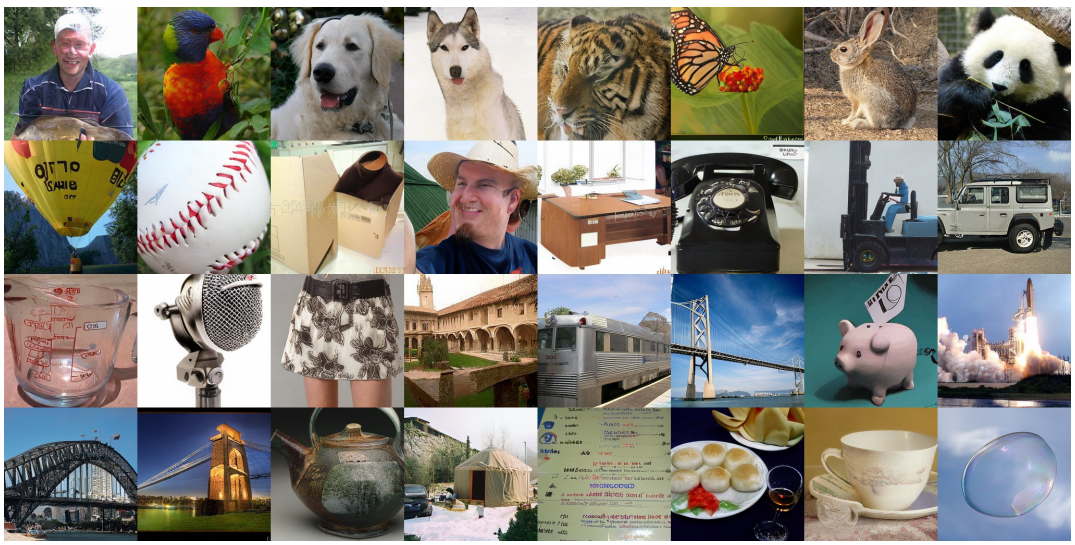


Figure 18. CFG-enhanced samples by pretrained and finetuned VAR-d30.