Generative Data Augmentation via Diffusion Distillation, Adversarial Alignment, and Importance Reweighting

Ruyi An[1*] **Haicheng Huang**[2*] **Huangjie Zheng**[3] **Mingyuan Zhou**[1]
[1]The University of Texas at Austin [2]Shanghai Jiao Tong University [3]Apple
ruyian@utexas.edu mingyuan.zhou@mccombs.utexas.edu

Abstract

Generative data augmentation (GDA) leverages generative models to enrich training sets with entirely new samples drawn from the modeled data distribution to achieve performance gains. However, the usage of the mighty contemporary diffusion models in GDA remains impractical: *i)* their thousand-step sampling loop inflates wall-time and energy cost per image augmentation; and *ii)* the divergence between synthetic and real distributions is unknown–classifier trained on synthetic receive biased gradients. We propose DAR-GDA, a three-stage augmentation pipeline that unites model **D**istillation, **A**dversarial alignment, and importance **R**eweighting that makes diffusion-quality augmentation both fast and optimized for improving downstream learning outcomes. In particular, a teacher diffusion model is compressed into a one-step student via score distillation, slashing the time per-image cost by $> 100\times$ while preserving FID. During this distillation (D), the student model additionally undergoes adversarial alignment (A) by receiving direct training signals against real images, supplementing the teacher's guidance to better match the true data distribution. The discriminator from this adversarial process inherently learns to assess the synthetic-to-real data gap. Its calibrated probabilistic outputs are then employed in reweighting (R) by importance weights that quantify the distributional gap and adjust the empirical loss when training downstream models; we show that reweighting yields an unbiased stochastic estimator of the real-data risk, fostering training dynamics akin to those of genuine samples. Experiments validate DAR-GDA's synergistic design through progressive accuracy gains with each D-A-R stage. Our approach not only surpasses conventional non-foundation-model GDA baselines but also remarkably matches or exceeds the GDA performance of large, web-pretrained text-to-image models, despite using solely in-domain data. DAR-GDA thus offers diffusion-fidelity GDA samples efficiently, while correcting synthetic-to-real bias to benefit downstream tasks.

1 Introduction

From early datasets of a few hundred instances [23, 22, 62] to today's web-scale corpora [17, 56, 73], data has always been the engine of machine learning. Discriminative models are only as good as the quantity, quality, and diversity of the samples they see. Data augmentation eases this dependency by synthesizing additional training examples, aiming to improve the generalization and robustness of models by exposing them to a more varied set of training instances. Classic hand-crafted transforms, such as geometric warps, color jitter, flips, exploit invariances of natural images [50, 77] but cannot create entirely new content. Generative data augmentation (GDA) closes this gap by sampling fresh, high-fidelity images that improve downstream performance and unlock learning under privacy [111, 94, 35], security [107, 87], or copyright constraints [55, 75].

[*]These authors contributed equally.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Among generative models, diffusion models [31, 84] have emerged as the state-of-the-art, delivering unparalleled sample quality and mode coverage [18, 32]. However, two obstacles limit their practical use for augmentation: *i)* diffusion sampling requires hundreds to thousands of denoising iterations, significantly inflating computational cost per image [104]; *ii)* the generator distribution q_G inherently differs from the true data distribution p_{data}. Unmeasured discrepancies between q_G and p_{data} can seed spurious artifacts [19] and coverage bias [93, 68], and ultimately skew classifier gradients, leading to biased generalization.

To overcome these challenges and unlock the potential of diffusion models for GDA, we introduce DAR-GDA, a unified framework that renders diffusion-quality, efficient generative augmentation by intertwining diffusion model **D**istillation, **A**dversarial alignment, and **R**eweighting by importance in GDA. First, score-based **D**istillation compresses a full teacher diffusion model trained on the true dataset into an efficient, single-step student generator, drastically cutting wall-time per sample by more than two orders of magnitude while preserving the Fréchet Inception Distance. Second, when integrated with the distillation process, **A**dversarial training further aligns the student model with the underlying distribution: the student generator not only imitates the teacher's scores but also competes against real images, which directly minimizes the Jensen–Shannon divergence with the true data, further narrowing the gap between q_G and p_{data}. Crucially, the discriminator learned in this process doubles as a density-ratio estimator; its calibrated output approximates the density ratio between two distributions, yielding per-sample importance weights. Third, for **R**eweighting, these importance weights are applied when training downstream learners on synthesized images from the fixed student generator, for which reweighting the classification objective with these importance weights yields an unbiased stochastic estimator of the true data risk. Furthermore, we show that the combined effect of adversarial alignment and importance reweighting tightens an upper bound on generalization error relative to conventional GDA. While the proposed DAR-GDA framework is generally applicable to discriminative modeling, in this work, we focus on one specific task of classification as the fundamental benchmark for investigating its impact on model generalization. The source code is available at `https://github.com/ruyianry/gda-dar`.

Our contributions are as follows:

- We introduce DAR-GDA, a unified framework enabling practical GDA with diffusion models by concurrently addressing i) their slow sampling speeds and ii) the synthetic-to-real data bias, integrated via adversarial training.
- We demonstrate how integrating adversarial training with score distillation not only improves the distilled model through real-data guidance by minimizing JS divergence, but also yields a discriminator. This discriminator, in turn, enables a sample importance reweighting to correct for bias present in empirical risk on synthetic samples, forming our three-stage DAR-GDA approach.
- Experiments confirm progressive performance gains across D-A-R stages, with DAR-GDA simultaneously demonstrating superior classification performance and efficiency for GDA on CIFAR-10 and ImageNet-1K.

2 Related work

2.1 Generative Data Augmentation

Early GDA approaches rely on VAEs and GANs [47, 25, 9], which proved valuable for niche scenarios such as class imbalance, low-shot recognition, and domain shifts [119, 90, 72, 76, 7, 66, 53, 114, 115, 52, 1, 36]. Their impact on large-scale, high-resolution tasks remained limited because those models struggled to capture complex data distributions. Diffusion models overcome that limitation: they faithfully reproduce fine detail and avoid mode collapse, enabling advances in robustness [6, 26, 10, 61], privacy preservation [64], data imbalance [3], and self- or semi-supervised learning [97, 89, 112, 8]. A growing line of work harnesses foundation models—diffusion generators pre-trained on web-scale corpora such as Stable Diffusion and GLIDE [67, 63]—and then aligns them to a downstream domain [5, 34, 98, 71, 122, 6, 29, 108]. While providing strong priors, this strategy raises several concerns such as licensing and usage constraints and limited applicability in fields with no web counterpart like scientific diffraction imaging [96]. Crucially, the high inference cost of iterative diffusion sampling remains, rendering it an economically challenging solution. These

limitations motivate a task-centric GDA pipeline that *i)* learns an expressive generator directly from the classifier's own training set; and *ii)* produces diffusion-quality samples at reduced cost.

2.2 Adversarial Training

GANs cast generation as a two-player game in which a discriminator learns to distinguish between real and generated samples, outputting a probability of a sample being real [25]. This adversarial process is recognized for minimizing the Jensen-Shannon divergence between the true data distribution and the generator's distribution, which is distinct from the reconstruction losses-based generative models [21]. Beyond its role in training the generator, the GAN discriminator's output can be interpreted as a likelihood-free density-ratio estimator [92, 27, 80, 91], which has spurred various techniques such as post-hoc discriminator-guided rejection sampling or weakly supervised discriminator trained on reference set to refine generator outputs, yielding fairer or higher-fidelity sample sets [11, 43, 51, 4]. Both rely on training a GAN from scratch and therefore inherit mode-collapse and stability issues [104] that limit their usefulness for large-scale augmentation. More recent work attaches a separately trained, GAN-style discriminator to a diffusion backbone to steer sampling [41]; this requires a distinct secondary training phase for the discriminator, and the valuable density-ratio information it learns is typically not propagated to inform downstream tasks. Collectively, we see a need for a more integrated adversarial mechanism that can be synergistically employed with diffusion models, and applied for enhancing downstream applications.

3 Preliminaries

3.1 Diffusion Models

Diffusion (or score-based) generative models [79, 31, 84, 83] construct a latent Markov chain that gradually corrupts a data point $\boldsymbol{x}_0 \in \mathbb{R}^d$ into noise, and learns to reverse this process to generate new data. In the forward diffusion, a deterministic schedule $\{\alpha_t, \sigma_t\}_{t=1}^T$ mixes signal and Gaussian noise:

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

with α_t decreases and σ_t increases with t such that $\boldsymbol{x}_T$ approaches standard Gaussian noise.

The reverse process is modeled by a parameterized conditional distribution $p_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ whose noise-prediction network ϵ_ϕ is trained to predict the noise $\boldsymbol{\epsilon}_t$ added at t [31, 84]:

$$\mathcal{L}^{(\text{diffusion})}(\phi) \;=\; \mathbb{E}_{\boldsymbol{x}_0 \sim \mathcal{P},\, \boldsymbol{\epsilon}_t,\, t}\big[\|\boldsymbol{\epsilon}_t - \epsilon_\phi(\boldsymbol{x}_t, t)\|_2^2\big], \tag{2}$$

The trained noise predictor $\boldsymbol{\epsilon}_\phi$ is directly related to the score function $\nabla_{\boldsymbol{x}_t} \log p_\phi(\boldsymbol{x}_t)$, often approximated as $s_\phi(\boldsymbol{x}_t, t) = -{\sigma_t}^{-1} \epsilon_\phi(\boldsymbol{x}_t, t)$. To generate a new sample, the model iteratively applies this reverse process, starting from pure noise $\boldsymbol{x}_T$ and gradually denoising it using the learned score over T steps to reconstruct an initial data point $\boldsymbol{x}_0$. While this iterative process yields high-fidelity samples, performing T (often hundreds to thousands) steps incurs considerable computational cost.

3.2 Score Distillation

Score distillation aims to accelerate the inference of the trained T-step teacher diffusion sampler by condensing them into an efficient student generator, G_θ. This student, often capable of few- or single-step synthesis, is trained to replicate the teacher's learned data distribution, typically by using the teacher's score predictions to guide the student's outputs at various noise levels. This is commonly obtained by training an auxiliary fake diffusion model, parameterized by ψ, on student's outputs,

$$\boldsymbol{y}_t \;=\; \alpha_t G_\theta(\boldsymbol{z}) + \sigma_t \boldsymbol{\epsilon}_t, \qquad \boldsymbol{z}, \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

via the same objective as Eq. 2 [58, 110, 121, 105], *i.e.*, $\mathcal{L}^{(\text{fake score})}(\psi) \;=\; \mathbb{E}_{\boldsymbol{z}, \boldsymbol{\epsilon}_t, t}\big[\|\boldsymbol{\epsilon}_t - \epsilon_\psi(\boldsymbol{y}_t, t)\|_2^2\big]$. Let $\psi^*(\theta)$ be the minimizer w.r.t. ψ for fixed θ. The generator is then updated by minimizing the score distill (SD) loss, the divergence $\mathcal{D}$ between the fake score $p_{\psi^*(\theta)}$ and the teacher score p_ϕ:

$$\mathcal{L}^{(\text{SD})}(\theta) \;=\; \mathbb{E}_{\boldsymbol{z}, \boldsymbol{\epsilon}_t, t}\big[\mathcal{D}\big(p_{\psi^*(\theta)}(\boldsymbol{y}_t\,;\, t) \,\|\, p_\phi(\boldsymbol{y}_t\,;\, t)\big)\big]. \tag{4}$$

The divergence $\mathcal{D}$ can be instantiated as a Kullback-Leibler (KL) divergence [100, 58, 110], Fisher divergence [121, 120], or score-constructed trajectory-level divergence [42]. In practice, ψ and θ are initialized from ϕ and alternately updated. This process can yield a one-step generator reproducing the teacher's quality at significantly reduced sampling cost [58, 121, 120], offering a promising path to overcome the efficiency bottleneck of diffusion models in practical GDA.

3.3 Generalization Risk in Supervised Learning

Let $\mathcal{P}$ be the unknown underlying data distribution over input-label pairs $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$. Given a hypothesis $h : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|} \in \mathcal{H}$, *e.g.*, a ω-parameterized neural network function h_ω, the generalization risk $\mathcal{R}(h; \mathcal{P})$ quantifies the expected loss on unseen data using a loss function $\ell : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$(such as cross-entropy), quantifying the penalty for the predicted output $h(x)$ given the true label y.

Since $\mathcal{P}$ is inaccessible, learning relies on a finite i.i.d. sample set $\mathcal{S}_\mathcal{P} = \{(x_i, y_i)\}_{i=1}^n \sim \mathcal{P}$. The true risk is then approximated by the empirical risk, an unbiased estimation $\mathcal{R}_{\text{emp}}(h; \mathcal{S}_\mathcal{P})$:

$$\mathcal{R}(h; \mathcal{P}) = \mathbb{E}_\mathcal{P}\big[\ell\big(h(x), y\big)\big] \approx n^{-1} \sum_{i=1}^{n} \ell\big(h(x_i), y_i\big) = \mathcal{R}_{\text{emp}}(h; \mathcal{S}_\mathcal{P}). \tag{5}$$

Empirical risk minimization (ERM) [95, 59] is a fundamental principle of statistical learning that aims to find a hypothesis that minimizes this empirical risk, $\mathcal{R}_{\text{emp}}$, with the expectation that a small empirical risk translates into low true risk and thus good generalization. The ERM principle forms the operational basis for the vast majority of supervised learning algorithms [60].

4 Distillation, Adversarial Alignment, and Reweighting for GDA

Before presenting our unified framework in Section 4.4, we first motivate its design by detailing the logical progression that leads to its components. This section begins with a critical reflection on the implicit biases in conventional GDA (Section 4.1). We then establish a principled, bias-correction mechanism via density-ratio reweighting (Section 4.2). This solution's efficacy demands a generator with high fidelity and mode coverage, pointing directly to state-of-the-art diffusion models. However, their practical use is barred by prohibitive sampling cost and the lack of a built-in density-ratio estimator. We then employ adversarial score distillation (Section 4.3) as the technique that simultaneously resolves both challenges, providing the foundation for the DAR-GDA pipeline.

4.1 Rethinking Learning from Generated Data Augmentation

Given the unknown data distribution $(\boldsymbol{x}, y)$, the generator G defines a distribution $\mathcal{Q}_G$, from which we can draw sets of augmenting synthetic samples $\mathcal{S}_{\mathcal{Q}_G} = \{(\boldsymbol{x}_i, y_i)\} \sim \mathcal{Q}_G$. A common practice to train a hypothesis, *e.g.*, a classifier, is to minimize the empirical risk on these synthetic samples:

$$|\mathcal{S}_{\mathcal{Q}_G}|^{-1} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{S}_{\mathcal{Q}_G}} \ell\big(h(\boldsymbol{x}_i), y_i\big) \;\approx\; \mathbb{E}_{\mathcal{Q}_G}\big[\ell\big(h(\boldsymbol{x}), y\big)\big], \tag{6}$$

This strategy implicitly assumes that the synthetic distribution $\mathcal{Q}_G$ is a faithful proxy for the true data distribution $\mathcal{P}$. However, in practice, $\mathcal{Q}_G$ inevitably differs from $\mathcal{P}$. Consequently, optimizing the objective in Eq. 6 amounts to optimizing with respect to a biased approximation of the true risk $\mathcal{R}(h; \mathcal{P}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{P}}[\ell(h(\boldsymbol{x}), y)]$. This inherent bias reflects the error in the learning objective that arises directly from the distributional misalignment between $\mathcal{Q}_G$ and $\mathcal{P}$:

$$\Delta_{\text{bias}} \;=\; \mathbb{E}_{\mathcal{Q}_G}\big[\ell(h(\boldsymbol{x}), y)\big] \;-\; \mathbb{E}_\mathcal{P}\big[\ell(h(\boldsymbol{x}), y)\big]. \tag{7}$$

This distributional misalignment can manifest in several ways. For instance, if certain features or modes are overrepresented in $\mathcal{Q}_G$ relative to $\mathcal{P}$–that is, in regions where $q_G(\boldsymbol{x}) > p_{\text{data}}(\boldsymbol{x})$, the hypothesis h may overfit to these dominant synthetic patterns, thereby neglecting rarer but potentially crucial characteristics of the true data. This issue of imbalance is a recognized challenge, even in today's advanced diffusion models [93, 68, 104]. Furthermore, the deviation is exacerbated if $\mathcal{Q}_G$ generates spurious, incoherent, or low-quality samples, *i.e.*, samples $\boldsymbol{x}_s$ for which $q_G(\boldsymbol{x}_s, y) > 0$ but $p_{\text{data}}(\boldsymbol{x}_s, y) \approx 0$. Training on such artifacts, also reported in recent diffusion models [19, 37, 44], can lead the hypothesis to learn incorrect correlations, thereby undermining the effectiveness of GDA.

Given these inherent distributional discrepancies, naively treating all synthetic samples as perfect and equally valuable representatives of $\mathcal{P}$–as implied by Eq. 6–can misguide the learning process. Therefore, to obtain a more faithful estimate of the true risk $\mathcal{R}(h; \mathcal{P})$, and thus mitigate $\Delta_{\text{bias}}(h)$, it is crucial to account for variations among individual synthetic samples in their fidelity and alignment with $\mathcal{P}$. A principled approach involves reweight the contribution of each synthetic sample in the empirical risk according to its estimated alignment. This naturally gives rise to the need for a mechanism that can quantify the degree of alignment between each synthetic sample and the true data distribution, enabling the corresponding correction of the loss terms $\ell(h(\boldsymbol{x}), y)$.

4.2 Density-Ratio Reweighting

A natural way to perform this quantification is to compare the marginal densities of synthetic samples under the two distributions. However, since $p_{\text{data}}(\cdot)$ is only known through its empirical distribution and $q_G(\cdot)$ is defined implicitly by the generative process, directly quantifying their discrepancy–and thus correcting for the induced learning bias–is challenging.

To circumvent the need for explicit likelihood estimation, we draw inspiration from the Generative Adversarial Network (GAN) framework [25], which provides a likelihood-free mechanism for distinguishing between real and synthetic samples. In GANs, a discriminator D and a generator G engage in a minimax game: $\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim q_G(\boldsymbol{x})}[\log(1 - D(\boldsymbol{x}))]$. For a fixed generator G, the optimal discriminator D^* satisfies [25, 113]:

$$D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + q_G(\boldsymbol{x})}. \tag{8}$$

This optimal discriminator D^* captures the probability that a sample $\boldsymbol{x}$ originates from $p_{\text{data}}(\boldsymbol{x})$ rather than $q_G(\boldsymbol{x})$, thereby providing an implicit comparison between the two densities.

Leveraging this, the density ratio $r(\boldsymbol{x})$, which compares the likelihoods of a sample $\boldsymbol{x}$ under p_{data} versus q_G can be derived from $D^*(\boldsymbol{x})$:

$$r(\boldsymbol{x}) := \frac{D^*(\boldsymbol{x})}{1 - D^*(\boldsymbol{x})} = \frac{p_{\text{data}}(\boldsymbol{x})/(p_{\text{data}}(\boldsymbol{x}) + q_G(\boldsymbol{x}))}{1 - p_{\text{data}}(\boldsymbol{x})/(p_{\text{data}}(\boldsymbol{x}) + q_G(\boldsymbol{x}))} = \frac{p_{\text{data}}(\boldsymbol{x})}{q_G(\boldsymbol{x})}. \tag{9}$$

This ratio acts as an importance weight briding the expectations of the two involved distributions: for any integrable f:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}}\big[f(\boldsymbol{x},y)\big] \;=\; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{Q}_G}\big[r(\boldsymbol{x})\, f(\boldsymbol{x},y)\big], \tag{10}$$

so weighting synthetic samples by $r(\boldsymbol{x})$ converts expectations under q_G into those under p_{data}. Intuitively, an $r(\boldsymbol{x}) > 1$ amplifies regions underrepresented by q_G, whereas an ratio $r(\boldsymbol{x}) < 1$ downweights overrepresented or atypical synthetic examples–thereby correcting distributional misalignment.

To apply this approach to GDA, we estimate $r(\boldsymbol{x})$ via the discriminator D (denote the estimate $r_D(\boldsymbol{x})$) and reweight the synthetic loss contributions:

$$\begin{aligned}\mathcal{L}^{(\text{reweight})}(h; G, D) :=& |\mathcal{S}_{\mathcal{Q}_G}|^{-1} \sum_{\boldsymbol{x}\in\mathcal{S}_{\mathcal{Q}_G}} r_D(\boldsymbol{x})\ell(h(\boldsymbol{x}), y) \\ \approx& \mathbb{E}_{\mathcal{Q}_G}\left[r_D(\boldsymbol{x})\ell(h(\boldsymbol{x}), y)\right] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}}[\ell(h(\boldsymbol{x}), y)]\end{aligned} \tag{11}$$

Optimizing h on synthetic samples under this reweighted objective aligns training on the synthetic distribution $\mathcal{Q}_G$ with learning directly from the true data distribution $\mathcal{P}$, thereby mitigating the bias $\Delta_{\text{bias}}(h)$ that arises from their distributional discrepancy.

In practice, $r(\boldsymbol{x})$ is estimated using a parameterized discriminator $D_\eta(\boldsymbol{x}) \approx D^*(\boldsymbol{x})$ which introduces additional variance into the loss computation. To manage the inherent bias-variance trade-off, we employ two variance reduction techniques:

Truncation: To prevent large $r(\boldsymbol{x})$ values from dominating the loss and destabilizing training, we clip the importance weights as $\bar{r}(\boldsymbol{x}) = \min(r(\boldsymbol{x}), \gamma)$, for a threshold $\gamma \geq 1$.

Self-normalization: Within each mini-batch of k synthetic samples $\{\boldsymbol{x}_j\}_{j=1}^k$, we apply batch-wise self-normalization to the truncated weight: $\tilde{r}(\boldsymbol{x}) = \bar{r}(\boldsymbol{x})/\sum_{j=1}^k \bar{r}(\boldsymbol{x}_j)$. This normalizes the sum of weights in a mini-batch, further stabilizing the updates.

The efficacy of importance reweighting hinges on a well-behaved density ratio $r(\boldsymbol{x})$, which requires significant overlap between the generative ($\mathcal{Q}_G$) and true ($\mathcal{P}$) distributions [24]. Generators trained with purely adversarial objectives often fail this prerequisite, suffering from mode collapse or support mismatch, especially in the early, noise-producing training stages [116, 70, 85, 86, 38, 2, 48]. This makes their discriminators poor estimators for $r(\boldsymbol{x})$. Diffusion models, on the other hand, with their high sample quality and excellent mode coverage [82, 46], emerge as ideal candidates for q_G. However, they present two fundamental obstacles: *i)* Efficiency: they are computationally prohibitive for augmentation sampling; and *ii)* Mechanism: they are trained without an adversarial component, leaving no mechanism to estimate the $r(\boldsymbol{x})$ required for importance reweighting. These challenges motivate a unified approach that simultaneously addresses the efficiency problem and the density-ratio-estimation problem. We therefore bridge this by integrating adversarial training directly within

the score distillation framework to compress a pre-trained diffusion teacher model into a fast student generator, while concurrently learning the discriminator needed for the bias-correcting reweighting.

4.3 Adversarial Score Distillation for Diffusion Models

The Reweighting stage of our DAR-GDA framework necessitates a discriminator capable of estimating the density ratio $r(\boldsymbol{x})$. As established, vanilla diffusion models not only lack this adversarial component but are also prohibitively slow due to their iterative sampling process, leading to high per-sample costs. To address both the need for a density-ratio importance estimator and the demand for computationally efficient augmentation, we therefore bridge the score diffusion distillation procedure, detailed in Section 3.2, with adversarial training, drawing inspiration from methodologies that integrate GAN-like objectives with diffusion processes [99].

Critically, the student generator G_θ is initialized from and guided by the high-fidelity, pre-trained teacher. This provides a powerful starting point, allowing the adversarial training to bypass the unstable, error-prone early stages–where a generator starting from noise struggles to produce informative samples–that typically plague purely adversarially trained models [85, 86].

In this process, the trained iterative teacher parameterized by ϕ is compressed into a one-step, fast student generator G_θ, while a discriminator D_η is learned simultaneously. This is framed as the following minimax objective:

$$\min_{\theta} \max_{\eta} \; \lambda_1 \, \mathcal{L}^{(\mathrm{SD})}_{\phi,\psi}(\theta) \; + \; \lambda_2 \, \mathcal{L}^{(\mathrm{adv.})}(\theta, \eta), \tag{12}$$

where $\mathcal{L}^{(\mathrm{SD})}_{\phi,\psi}$ is the score-distillation loss defined in Eq. 4, compelling the student G_θ to match the teacher's score estimates. The adversarial loss, $\mathcal{L}^{(\mathrm{adv.})}(\theta, \eta)$ is defined as:

$$\mathcal{L}^{(\mathrm{adv.})}(\theta, \eta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P},\, \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\big[\log D_\eta(\alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}; t)\big] + \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\, \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\big[\log\big(1 - D_\eta(\boldsymbol{y}_t; t)\big)\big], \tag{13}$$

where $\boldsymbol{y}_t = \alpha_t G_\theta(\boldsymbol{z}) + \sigma_t \boldsymbol{\epsilon}$ is a noisy sample from the student generator G_θ at t, and $t \sim \pi$ is a sampling distribution over timesteps. This design allows the discriminator to compare real and fake samples across various noise levels–a design shown to enhance training stability [99, 106]. In the specific case where the adversarial game operates only on the clean image space, π simplifies to a Dirac delta distribution centered at $t = 0$. The adversarial gradient of the student is obtained by minimizing the negative of the second expectation in Eq. 13, *i.e.*, $\mathcal{L}^{(\mathrm{adv.})}_G(\theta) = -\mathbb{E}_{\boldsymbol{z},t}\big[\log D_\eta(\boldsymbol{y}_t; t)\big]$.

The discriminator D_η can be realized as a standalone network that processes the entire input to produce a single global probability [42], or as an integrated component sharing the student's U-Net encoder [118]. In this latter, integrated approach, a final probability is obtained by applying patch-wise aggregation to the encoder's output logits. The parameter vector η denotes all the weights of this discriminator, including any parameters shared with the student generator.

The adversarial score distillation strategy offers several synergistic advantages. The student generator G_θ produces samples in a single step, drastically reducing computational costs compared to its iterative teacher and making diffusion models practical for GDA. The synergy of adversarial training and score distillation also drives the student distribution q_{G_θ} closer to p_{data} by directly minimizing the Jensen–Shannon divergence, while preserving the teacher's high-fidelity details. Empirically, adversarial distillation has shown stable behaviour, with one-step students generally matching–and frequently reported to surpass–their teachers' quality in FID [120, 42, 109, 58]. Crucially, this process yields the discriminator D_η as a concurrent byproduct, eliminating the need for a separate training phase. This discriminator provides the exact mechanism required for the reweighting stage, furnishing the density-ratio estimate in Eq. 9 needed to debias the downstream classifier's training.

4.4 The DAR-GDA Framework

Having introduced the modular components of Distillation, Adversarial alignment, and Reweighting, we now synergistically combine them into the DAR-GDA framework. The DAR-GDA framework provides a unified pipeline: an efficient student generator G_θ (from Distillation) that is aligned with p_{data} (via Adversarial training), and a co-trained discriminator D_η that enables bias-correcting reweighting. This integrated design enhances generalization by mitigating the key error sources inherent in learning from synthetic data. Furthermore, initializing G_θ from a proficient diffusion teacher significantly stabilizes adversarial training and mitigates classic GAN pathologies such

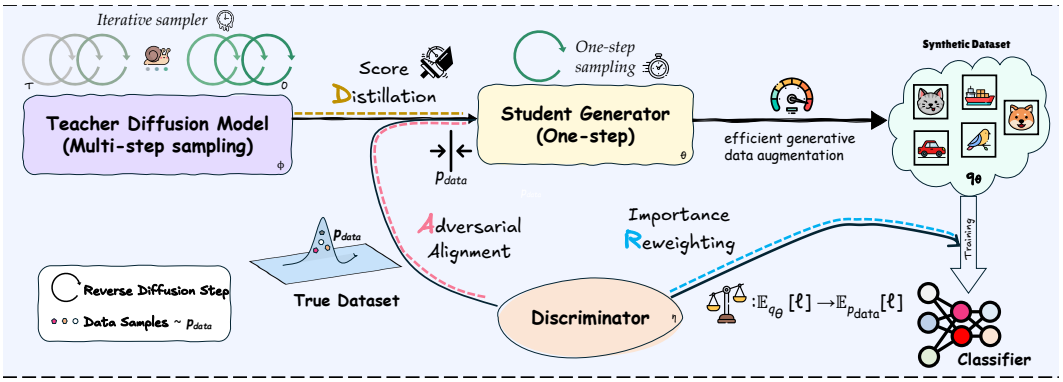


Figure 1: Overview of the DAR-GDA framework. **(D)** A multi-step teacher diffusion model is distilled into an efficient one-step student generator. **(A)** The student is adversarially aligned with the true data distribution using a discriminator. **(R)** The discriminator's outputs are then used as importance weights to train a downstream classifier.

as mode collapse and catastrophic forgetting [88]. We illustrate the overall idea in Figure 1 and summarize the complete algorithmic procedure in Appendix B.

This integrated design enhances generalization by mitigating the key error sources inherent in learning from synthetic data. We can formalize this improvement by analyzing the expected risk $\mathbb{E}_{\mathcal{P}}[\ell(h)]$ following [10, 117, 74]. This true risk can be decomposed as two summands:

$$\mathbb{E}_{\mathcal{P}}\big[\ell(h)\big] = \Big[\mathbb{E}_{\mathcal{P}}\big[\ell(h)\big] - \mathbb{E}_{\mathcal{P}}\big[\ell(h)_{\mathcal{Q}_{G_\theta}}\big]\Big] \;+\; \Big[\mathbb{E}_{\mathcal{P}}\big[\ell(h)_{\mathcal{Q}_{G_\theta}}\big]\Big], \tag{14}$$

The first summand reflects the impact of distributional mismatch. For a bounded loss $|\ell| \le L$:

$$\Big|\mathbb{E}_{\mathcal{P}}\big[\ell(h)\big] - \mathbb{E}_{\mathcal{P}}\big[\ell(h)_{\mathcal{Q}_{G_\theta}}\big]\Big| \le 2L\,\mathcal{D}_{\mathrm{TV}}(\mathcal{P} \,\|\, \mathcal{Q}_{G_\theta}) \;\le\; 2L\sqrt{2\,\mathcal{D}_{\mathrm{JS}}\big(\mathcal{P} \,\|\, \mathcal{Q}_{G_\theta}\big)}, \tag{15}$$

by variational characterization of total variation (TV) distance and Pinsker's inequality. Adversarial score distillation directly minimizes the Jensen-Shannon (JS) divergence between $\mathcal{P}$ and $\mathcal{Q}_G$, hence tightening the upper bound on the distribution mismatch compared to using non-adversarially trained generators or the original teacher diffusion model. The second term, the biased synthetic risk, is the objective naively optimized by conventional GDA (Eq. 6). Our Reweighting stage directly corrects this. By employing the objective in Eq. 11, we replace the minimization of this biased term with an unbiased stochastic estimator of the true data risk, $\mathbb{E}_{\mathcal{P}}[\ell(h)]$.

Thus, DAR-GDA enhances generalization through a dual mechanism: the Adversarial component minimizes the distributional bias (Term 1), while the Reweighting component provides an unbiased risk estimator for hypothesis training (correcting Term 2). Coupled with the practical benefit of $> 100\times$ reduction in per-sample generation cost from Distillation, DAR-GDA is positioned as a powerful and practical tool for GDA.

5 Experiments

We evaluate DAR-GDA on CIFAR-10 and ImageNet-1K, comparing it with *i)* standard data-augmentation pipelines, *ii)* strong diffusion-based GDA baselines, and *iii)* a state-of-the-art GAN. The analysis proceeds in two stages. We benchmark it against conventional data augmentation methods and existing GDA baselines. We also assess its performance and adaptability with various underlying generative models. We additionally probe how DAR scales under dynamic augmentation on smaller datasets for its practical feasibility.

Datasets. We use the canonical train/val splits of CIFAR-10 [49] and ImageNet-1K [17]; no external or test data are introduced at any stage, including teacher training, distillation, or classification.

Evaluation. Classification performance is reported as top-1 accuracy for CIFAR–10 and top-1/top-5 accuracy for ImageNet-1K. Generator quality is measured with Fréchet Inception Distance (FID)[30].

Generative Models. For diffusion-based GDA, we adopt the publicly released checkpoints of DDPM++–EDM for CIFAR-10 [39] and DDPM++–EDM2-XXL for ImageNet-1K [40]. On CIFAR-10 we include R3GAN [33], GAN state-of-the-art achieving the best reported FID. We omit GANs on ImageNet because no open-source 512×512 model matches diffusion quality, and de novo training is

Table 1: CIFAR-10 classification accuracy with ResNet-18 and VGG-16, using training sets augmented by synthetic data equal to 1× the original dataset size (additive augmentation) on various generative models and adversarial-distillation methods under static versus dynamic data generation schemes. $*$ indicates the use of a web-pretrained text-to-image model, such as GLIDE [63] and Stable Diffusion (SD) [67] which are to be carefully compared to other generative models trained only on the original data. The best result is shown in **bold**.

Generative Model	**DAR progression**			**static** (gen. once)		**dynamic** (re-gen. each epoch)	
	Distill	Adv. align	Reweight	ResNet-18	VGG-16	ResNet-18	VGG-16
Real only	-	-	-	95.00±0.15	93.76±0.20	95.00±0.45	93.76±0.50
R3GAN [33] (FID=1.96)	-	✔	✗	95.18±0.50	93.90±0.42	95.61±0.98	94.11±1.12
		✔	✔	95.32±0.47	93.86±0.39	95.72±1.00	93.73±1.26
GLIDE + real guidance [29]*	-	-	-	95.77±0.25	94.50±0.41	-	-
SD + ActGen [34]*	-	-	-	95.92±0.42	94.68±0.39	-	-
Diffusion Model [39] (FID=1.81)	✗	✗	✗	95.37±0.23	93.95±0.32	96.16±0.76	94.98±0.79
	CTM [42] (FID=1.73)	✗	✗	95.32±0.39	93.97±0.29	95.84±0.86	95.08±1.02
		✔	✗	95.43±0.21	94.09±0.26	96.35±0.98	95.15±1.13
		✔	✔	95.88±0.22	94.28±0.24	96.57±1.01	95.33±1.07
	SiDA [120] (FID=1.39)	✗	✗	95.48±0.25	94.18±0.28	96.29±0.68	95.22±0.79
		✔	✗	95.84±0.36	94.53±0.36	96.40±1.16	95.40±1.02
		✔	✔	**96.21**±0.25	**94.64**±0.20	**96.73**±1.15	**95.73**±1.08

notoriously unstable at that scale. We also include selected finetuned text-to-image models pre-trained on web-scale image data. Please note that comparisons with these models may be subject to concerns like additional information from pre-training data and potential test-set leakage.

Implementation Details.

We instantiate the (D) and (A) steps of our DAR-GDA with two recent leading algorithms: CTM [42], a trajectory-based score distillation with a standalone discriminator, and SiDA [120], a Fisher-divergence-minimizing score-based distillation with encoder-sharing discriminator operating in noise space. Both are tested on CIFAR-10; SiDA alone is used on ImageNet-1K owing to CTM's current memory footprint. On CIFAR-10, we set $\alpha = 1.2$ for SiDA and train with Adam (lr=1e-5) [45] optimizer. CTM is trained for 256 steps per batch with a student learning rate of 3e-4 and the discriminator learning rate 2e-3 (batch size 128). For ImageNet-1K we distill EDM2-XXL with SiDA across 8 A100-80GB GPUs: $\alpha = 1.0$, per-GPU batch size 64, gradient accumulation every 128 iterations using Adam (lr=5e-5) optimizer. For CIFAR-10 we train ResNet-18 [28] and VGG-16 [78]; for ImageNet-1K we use ResNet-50 [28] and ViT-S/16 [20] to evaluate the performance with and without the (R)eweighting component with self-normalization and $\gamma = 1$. CIFAR-10 models are trained for 300 epochs with batch size 128 using momentum SGD (lr=0.1). The hyperparameters λ_1 and λ_2 for the adversarial alignment objective follow the settings from prior work. On ImageNet-1K, ResNet-50 is trained for 90 epochs with batch size 4096 and initial learning rate 1.6, while ViT-S/16 is trained for 300 epochs with batch size 1024, AdamW [57], and initial learning rate 3e-3. We apply self-normalization and set γ to be 1 for obtaining $r(\boldsymbol{x})$ for reweighting. Diffusion outputs are generated at 32×32 for CIFAR-10 and 512×512 for ImageNet; the latter are down-sampled to 224×224 before classification to match baseline protocols. Experiments use NVIDIA A100-80GB GPUs-single-GPU for CIFAR-10, 8-GPU (DDP) for ImageNet-1K implemented in PyTorch 2.1 [65].

Table 2: ImageNet-1K classification accuracy with ResNet-50 and ViT-S/16, using training sets augmented by synthetic data equal to 1× the original dataset size (additive augmentation) on various generative models under the static generation. $*$ denotes the use of a web-pretrained text-to-image model, Imagen [69] or Stable Diffusion (SD) [67] which are to be carefully compared to other generative models trained only on the original data. The best result is shown in **bold**.

Generative Model	**DAR progression**				
	Distill	Adv. align	Reweight	Top-1 Acc.	Top-5 Acc.
ResNet-50 classifier					
Real only	-	-	-	76.37±0.03	92.86±0.08
Imagen + finetune [5]*	-	-	-	78.17	-
SD + ActGen [34]*	-	-	-	**78.34±0.32**	94.12±0.38
Diffusion Model [40] (FID=1.91)	✗	✗	✗	77.12±0.15	93.95±0.20
	SIDA [120] (FID=1.37)	✗	✗	77.15±0.27	93.74±0.19
		✔	✗	77.89±0.18	93.90±0.24
		✔	✔	78.03±0.23	**94.13±0.32**
ViT-S/16 classifier					
Real only	-	-	-	79.91±0.04	94.48±0.08
Imagen + finetune[5]*	-	-	-	81.00	-
SD + ActGen [34]*	-	-	-	**81.17±0.51**	95.32±0.31
Diffusion Model [40] (FID=1.91)	✗	✗	✗	80.50±0.18	95.18±0.25
	SIDA [120] (FID=1.37)	✗	✗	80.46±0.23	95.19±0.23
		✔	✗	80.99±0.32	**95.38±0.43**
		✔	✔	**81.17±0.45**	95.36±0.40

Table 4: CIFAR-10 classification accuracy with ResNet-18 and VGG-16, using training sets *substituted by* synthetic data equal to 1× the original dataset size on various generative models and adversarial-distillation methods under static and dynamic data generation. The best result is in **bold**.

Generative Model	**DAR progression**			**static** (gen. once)		**dynamic** (re-gen. each epoch)	
	Distill	Adv. align	Reweight	ResNet-18	VGG-16	ResNet-18	VGG-16
Real only	-	-	-	95.00±0.15	93.76±0.20	95.00±0.45	93.76±0.50
R3GAN [33] (FID=1.96)	-	✔	✗	90.09±0.65	87.75±0.39	90.26±0.84	89.79±0.58
		✔	✔	90.29±0.71	87.97±0.58	92.91±0.72	88.03±0.61
Diffusion Model [39] (FID=1.81)	✗	✗	✗	91.78±0.25	90.60±0.22	95.88±0.19	94.26±0.21
	CTM [42] FID=1.73	✗	✗	90.83±0.39	89.65±0.30	93.70±0.24	93.18±0.28
		✔	✗	91.03±0.44	89.78±0.46	94.00±0.70	93.27±0.67
		✔	✔	91.37±0.43	90.26±0.52	94.32±0.72	93.38±0.52
	SiDA [120] FID=1.39	✗	✗	93.20±0.28	91.71±0.33	95.78±0.35	94.46±0.27
		✔	✗	93.50±0.35	91.88±0.38	96.39±0.60	95.25±0.52
		✔	✔	**93.77**±0.37	**92.31**±0.48	**96.68**±0.63	**95.72**±0.48

More detailed experimental setups are provided in Appendix C. DAR-GDA thus offers a practical, drop-in GDA solution, achieving diffusion-level fidelity with efficiency bias correction.

5.1 Additive Augmentation Results

We evaluate DAR-GDA by supplementing the original training set with an equal volume of synthetic samples that replicate the original label distribution, using both static (one-time) and dynamic (per-epoch) generation strategies. Tables 1 and 2 present classification results for CIFAR-10 and ImageNet-1K. DAR-GDA consistently boosts accuracy across both datasets and hypothesis sets. On CIFAR-10, it improves ResNet-18 by +1.7 pp and VGG-16 by +1.9 pp. Similar gains are observed on ImageNet-1K for ResNet-50 (+1.7 pp) and ViT-S/16 (+1.3 pp). Notably, our in-domain DAR-GDA (trained solely on task-specific data) matches or outperforms pre-trained large text-to-image models, particularly where baseline diffusion models without DAR are suboptimal. Dynamic generation consistently yields better results than static, underscoring the value of increased diversity. Also, SiDA leads to better classification performance than CTM, correlating with SiDA's superior FID score.

Table 3: GPU hours to generate 1:1 augmenting training data replica. SD involves downsampling for CIFAR-10.

Dataset	R3GAN	EDM/EDM2	CTM-EDM	SiD-EDM	SD
CIFAR-10	26.0s	3102s	58.3s	39.1s	6.8h
IN1K	-	291h	-	8.7h	1290h

Table 3 highlights the substantial GPU time savings for synthetic data generation. Distillation achieves synthesizing speeds comparable to fast GANs, positioning (D)istillation as a key to achieve an economical solution for high-quality GDA.

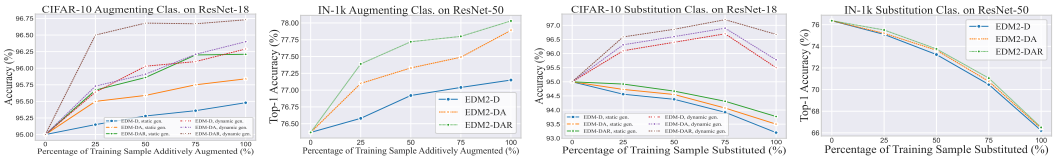
5.2 Substituting Augmentation Results

We next consider evaluate DAR-GDA in a full data replacement scenario, where hypotheses are trained solely on synthetic samples equivalent in volume and label distribution to the original training set. Tables 4 and 5 report the classification performance for CIFAR-10 and ImageNet-1K, respectively. While DAR-GDA consistently boosts accuracy over baseline synthetic data, a performance gap to training on real data is observed, this drop being more significant on ImageNet-1K. Notably, on ImageNet-1K, DAR-GDA enables hypotheses trained on synthetic data to match the performance achieved with data from the original, non-distilled multi-step diffusion teacher model.

Table 5: ImageNet-1K classification accuracy with ResNet-50 and ViT-S/16, using training sets *substituted by* synthetic data equal to 1× the original dataset size on various generative models and adversarial-distillation methods under static versus dynamic data generation schemes.

Generative Model	**DAR progression**				
	Distill	Adv. align	Reweight	Top-1 Acc.	Top-5 Acc.
ResNet-50 classifier					
Real only	-	-	-	76.37±0.03	92.86±0.08
Diffusion Model [40] (FID=1.91)	✗	✗	✗	66.41±0.47	86.58±0.35
	SIDA [120] FID=1.37	✗	✗	66.22±0.50	86.30±0.42
		✔	✗	66.42±0.59	86.54±0.38
		✔	✔	**66.50±0.87**	**86.82±0.66**
ViT-S/16 classifier					
Real only	-	-	-	79.10±0.03	94.43±0.08
Diffusion Model [40] (FID=1.91)	✗	✗	✗	67.89±0.38	85.55±0.29
	SIDA [120] FID=1.37	✗	✗	67.54±0.43	85.09±0.39
		✔	✗	67.73±0.50	85.30±0.47
		✔	✔	**68.01±0.64**	**85.89±0.56**

On CIFAR-10, SiDA-generated data significantly outperforms CTM data, despite both models being distilled from the same teacher. Remarkably, fully-synthetic GDA with SiDA with dynamic generation can surpass the performance of GDA training on the real dataset alone.



(a) (b) (c) (d)

Figure 2: Classification accuracy of SiDA-distilled EDM/EDM2 models on CIFAR-10/IN-1K with varying synthetic dataset sizes (measured as a % of the original training set). Performance is shown under two schemes: additive augmentation (Figs. 2a, 2b) and full data substitution (Figs. 2c, 2d).

5.3 Further Empirical Studies

We present an ablation study on augmenting size in Fig. 2. For additive augmentation, performance increases with more synthetic data, and our method's components yield incremental gains. Conversely, under data substitution, performance generally declines with increasing synthetic data. This decline is more significant for ImageNet, where generative models face greater difficulty matching the true data distribution. Notably, on CIFAR-10, a dynamic generation schedule improves performance with up to 75% of the synthetic data replacement, where diversity from new per-epoch samples appears to offset the loss of real data. Additional empirical studies are presented in Appendix D.

6 Discussion and Conclusion

Limitations and Ethical Considerations. Our methodology concentrates on GDA itself, rather than on the development of new generative models. Hence, we leveraged state-of-the-art EDM diffusion models as the teacher. While this approach entails a high, one-time pre-training cost, our framework does not address this; rather, our focus is on making these powerful, pre-trained models highly efficient for low-latency applications like GDA. A related limitation is that our validation, while rigorous, was constrained to the two large-scale datasets for which these specific teacher weights are publicly available, though the observed patterns were consistent and supported our hypotheses. Ethically, our GDA framework contributes positively to content safety by enabling stronger and more reliable discriminative models for detecting and filtering harmful or NSFW material. Nevertheless, because the generator component is trained to replicate the visual characteristics of harmful content for the purpose of improving filtering, it inherently carries the capability to reproduce such material. To ensure responsible use, the trained generator should be securely stored to prevent misuse.

Future Work. Looking ahead, the core principles of DAR-GDA are fundamentally domain-agnostic. Our framework treats the teacher model as a "black box" for score predictions. This modularity allows the framework to be applied beyond vision, for instance, to emerging diffusion models in domains like protein [101] and molecule generation [12], simply by replacing the U-Net backbone with a domain-specific architecture. This is particularly promising for data-scarce fields like medicine (e.g., MRI or X-ray synthesis), where a distilled student model could be shared to enable "pseudo data sharing" without violating patient confidentiality. Beyond new domains, the framework is applicable to a broader range of conditional tasks. For instance, in text-to-image generation, the discriminator component could be adapted to evaluate image quality and prompt alignment, *e.g.*, assessing $p(\text{image} \mid \text{prompt})$. The same principle applies to dense prediction tasks like segmentation, where the challenge shifts towards pseudo-labeling from foundation models or other forms of weak supervision to create the necessary training signals.

Conclusion. In this paper, we addressed two primary hindrances to the practical application of current diffusion models in GDA: suboptimal sampling efficiency and potential misalignment between generated and target data distributions. We introduced DAR-GDA, a three-stage synergistic framework where adversarial alignment (A) serves as a crucial bridge connecting model distillation (D)—to enhance sampling speed—and sample reweighting (R)—to correct distributional shifts. Our experiments demonstrated that the progressive integration of these DAR components leads to consistent improvements in GDA performance. Beyond the class-conditional models explored, this work may inspire the application of similar principled strategies to a wider array of diffusion-based generative models for effective data augmentation.

Acknowledgments

M. Zhou acknowledges the support of a gift grant from Apple.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

[3] Reyhane Askari-Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *Trans. Mach. Learn. Res.*, 2024.

[4] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian J. Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *ICLR*, 2019.

[5] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves ImageNet classification. *Trans. Mach. Learn. Res.*, 2023.

[6] Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. In *ICLR workshop*, 2023.

[7] Christopher Beckham, Sina Honari, Vikas Verma, Alex Lamb, Farnoosh Ghadiri, R. Devon Hjelm, Yoshua Bengio, and Chris Pal. On adversarial mixup resynthesis. In *NeurIPS*, 2019.

[8] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. One-shot unsupervised domain adaptation with personalized diffusion models. In *CVPRW*, 2023.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[10] Zhen Cheng, Fei Zhu, Xu-Yao Zhang, and Cheng-Lin Liu. Breaking the limits of reliable prediction via generated data. *Int. J. Comput. Vis.*, 2024.

[11] Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *ICML*, 2020.

[12] François Cornet, Grigory Bartosh, Mikkel N. Schmidt, and Christian A. Naesseth. Equivariant neural diffusion for molecule generation. In *NeurIPS*, 2024.

[13] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[14] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation strategies from data. In *CVPR*, 2019.

[15] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

[16] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[18] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

[19] Anh-Dung Dinh, Daochang Liu, and Chang Xu. Representative guidance: Diffusion model sampling with consistency. In *ICLR*, 2025.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[21] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

[22] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[23] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *CVPRW*, 2004.

[24] John Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020.

[26] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A. Mann. Improving robustness using generated data. In *NeurIPS*, 2021.

[27] Aditya Grover and Stefano Ermon. Boosted generative models. In *AAAI*, 2018.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip H. S. Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023.

[30] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017.

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[32] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47:1–47:33, 2022.

[33] Nick Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The GAN is dead; long live the GAN! a modern GAN baseline. In *NeurIPS*, 2024.

[34] Tao Huang, Jiaqi Liu, Shan You, and Chang Xu. Active generation for image classification. In *ECCV*, 2024.

[35] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. DeepPrivacy: A generative adversarial network for face anonymization. In *ISVC*, 2019.

[36] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. In *ICLR*, 2022.

[37] Yazid Janati, Badr Moufad, Mehdi Abou El Qassime, Alain Oliviero Durmus, Eric Moulines, and Jimmy Olsson. A mixture-based framework for guiding diffusion models. In *ICML*, 2025.

[38] Simon Jenni and Paolo Favaro. On stabilizing generative adversarial training with noise. In *CVPR*, 2019.

[39] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

[40] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024.

[41] Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. In *ICML*, 2023.

[42] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *ICLR*, 2024.

[43] Yeongmin Kim, Byeonghu Na, Minsang Park, JoonHo Jang, Dongjun Kim, Wanmo Kang, and Il-Chul Moon. Training unbiased diffusion models from biased dataset. In *ICLR*, 2024.

[44] Yujin Kim, Hyunsoo Kim, Hyunwoo J. Kim, and Suhyun Kim. When model knowledge meets diffusion model: Diffusion-assisted data-free image synthesis with alignment of domain and class. In *ICML*, 2025.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[46] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. On density estimation with diffusion models. In *NeurIPS*, 2021.

[47] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[48] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215*, 2017.

[49] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Canadian Institute for Advanced Research, 2009.

[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[51] Jinhee Lee, Haeri Kim, Youngkyu Hong, and Hye Won Chung. Self-diagnosing GAN: diagnosing underrepresented samples in generative adversarial networks. In *NeurIPS*, 2021.

[52] Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *NeurIPS*, 2017.

[53] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. BigDatasetGAN: Synthesizing ImageNet with pixel-wise annotations. In *CVPR*, 2022.

[54] Yuting Li, Yingyi Chen, Xuanlong Yu, Dexiong Chen, and Xi Shen. SURE: survey recipes for building reliable and robust deep networks. In *CVPR*, 2024.

[55] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *ICML*, 2023.

[56] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[58] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-Instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *NeurIPS*, 2023.

[59] Mehryar Mohri. *Foundations of machine learning*. MIT press, 2018.

[60] Andrea Montanari and Basil N Saeed. Universality of empirical risk minimization. In *CoLT*, 2022.

[61] Byeonghu Na, Yeongmin Kim, HeeSun Bae, Jung Hyun Lee, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Label-noise robust diffusion models. In *ICLR*, 2024.

[62] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996.

[63] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

[64] Jinseong Park, Yujin Choi, and Jaewook Lee. In-distribution public data synthesis with diffusion models for differentially private image classification. In *CVPR*, 2024.

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[66] Suman V. Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *NeurIPS*, 2019.

[67] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[68] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. CADS: unleashing the diversity of diffusion models through condition-annealed sampling. In *ICLR*, 2024.

[69] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[70] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

[71] Mert Bülent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic ImageNet clones. In *CVPR*, 2023.

[72] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

[73] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[74] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *ICLR*, 2022.

[75] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by Text-to-Image models. In *USENIX*, 2023.

[76] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my GAN? In *ECCV*, 2018.

[77] Patrice Y. Simard, David Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, 2003.

[78] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[79] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[80] Jiaming Song and Stefano Ermon. Bridging the gap between f-GANs and Wasserstein GANs. In *ICML*, 2020.

[81] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[82] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *NeurIPS*, 2021.

[83] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

[84] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[85] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. VEEGAN: reducing mode collapse in GANs using implicit variational learning. In *NeurIPS*, 2017.

[86] Ruoyu Sun, Tiantian Fang, and Alexander G. Schwing. Towards a better global loss landscape of GANs. In *NeurIPS*, 2020.

[87] Vajira Thambawita, Pegah Salehi, Sajad Amouei Sheshkal, Steven A Hicks, Hugo L Hammer, Sravanthi Parasa, Thomas de Lange, Pål Halvorsen, and Michael A Riegler. Singan-seg: Synthetic training data generation for medical image segmentation. *PLOS ONE*, 17(5):e0267976, 2022.

[88] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in GANs. In *IJCNN*, 2020.

[89] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. In *NeurIPS*, 2023.

[90] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle J. Palmer, and Ian D. Reid. A Bayesian data augmentation approach for learning deep models. In *NeurIPS*, 2017.

[91] Ryan D. Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *ICML*, 2019.

[92] Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, 2016.

[93] Soobin Um, Suhyeon Lee, and Jong Chul Ye. Don't play favorites: Minority guidance for diffusion models. In *ICLR*, 2024.

[94] Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection. In *AISTATS*, 2023.

[95] Vladimir Vapnik. Principles of risk minimization for learning theory. In *NeurIPS*, 1991.

[96] Benquan Wang, Ruyi An, Jin-Kyu So, Sergei Kurdiumov, Eng Aik Chan, Giorgio Adamo, Yuhan Peng, Yewen Li, and Bo An. OpticalNet: An optical imaging dataset and benchmark beyond the diffraction limit. In *CVPR*, 2025.

[97] Yifei Wang, Jizhe Zhang, and Yisen Wang. Do generated data always help contrastive learning? In *ICLR*, 2024.

[98] Zerun Wang, Jiafeng Mao, Xueting Wang, and Toshihiko Yamasaki. Training data synthesis with difficulty controlled diffusion model. *arXiv preprint arXiv:2411.18109*, 2024.

[99] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training gans with diffusion. In *ICLR*, 2023.

[100] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In *NeurIPS*, 2023.

[101] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[102] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *ICLR*, 2022.

[103] Ruixuan Xiao, Yiwen Dong, Haobo Wang, Lei Feng, Runze Wu, Gang Chen, and Junbo Zhao. ProMix: Combating label noise via maximizing clean sample utility. In *IJCAI*, 2023.

[104] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *ICLR*, 2022.

[105] Sirui Xie, Zhisheng Xiao, Diederik P. Kingma, Tingbo Hou, Ying Nian Wu, Kevin P. Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. EM distillation for one-step diffusion models. In *NeurIPS*, 2024.

[106] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. UFOGen: You forward once large scale text-to-image generation via diffusion GANs. In *CVPR*, 2024.

[107] Zhexin Yao, Qiuming Liu, Jingkang Yang, Yanan Chen, and Zhen Wu. PPUP-GAN: A gan-based privacy-protecting method for aerial photography. *Future Gener. Comput. Syst.*, 145:284–292, 2023.

[108] Teresa Yeo, Andrei Atanov, Harold Luc Benoit, Aleksandr Alekseev, Ruchira Ray, Pooya Esmaeil Akhoondi, and Amir Zamir. Controlled training data generation with diffusion models. *Trans. Mach. Learn. Res.*, 2025.

[109] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

[110] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

[111] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2019.

[112] Zebin You, Yong Zhong, Fan Bao, Jiacheng Sun, Chongxuan Li, and Jun Zhu. Diffusion models and semi-supervised learners benefit mutually with few labels. In *NeurIPS*, 2023.

[113] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, 2018.

[114] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. DatasetGAN: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

[115] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with GAN. In *NeurIPS Workshop on Synthetic Data for Empowering ML Research*, 2022.

[116] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah D. Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. In *NeurIPS*, 2018.

[117] Chenyu Zheng, Guoqiang Wu, and Chongxuan Li. Toward understanding generative data augmentation. In *NeurIPS*, 2023.

[118] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. In *ICLR*, 2023.

[119] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In *ICCV*, 2017.

[120] Mingyuan Zhou, Huangjie Zheng, Yi Gu, Zhendong Wang, and Hai Huang. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. In *ICLR*, 2024.

[121] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *ICML*, 2024.

[122] Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. In *ICML Workshop on Data-centric Machine Learning*, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See the abstract and Section 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The paper provides the full set of assumptions and a complete (and correct) proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Experimental settings, parameters, and environment are described in Section 5 to enable reproducibility of results.

Guidelines:

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

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

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

Answer: [Yes]

Justification: The experimental data are all from public datasets. Open access to the code is provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Data splits, hyperparameters, and training details are thoroughly described in Section 5.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Error bars were reported and the results are discussed comprehensively.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides sufficient information for the computational resources in Section 5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics with no ethical issues.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both potential societal impacts and negative societal impacts of the method proposed in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper does not involve releasing models or data with high misuse risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All third-party assets used are properly licensed and cited, including datasets and model weights used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: No new asset is introduced.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subjects research.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No human subjects research involved; IRB approval not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [No]

    Justification: Large Language Models are not used as a core component of this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A More Mathematical Discussions

A.1 Detailed Deriavations for Eq. 11

By the law of large numbers, the empirical reweighted loss converges to its population expectation under $\mathcal{Q}_G$:

$$|\mathcal{S}_{\mathcal{Q}_G}|^{-1} \sum_{(\boldsymbol{x},y)\in\mathcal{S}_{\mathcal{Q}_G}} r_D(\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y) \;\approx\; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{Q}_G}\big[r_D(\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y)\big]. \tag{16}$$

Substituting the definition of the density ratio $r(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})/q_G(\boldsymbol{x})$ gives

$$\begin{aligned}
\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{Q}_G}\big[r(\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y)\big] &= \int r(\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y)\,q_G(\boldsymbol{x},y)\,d\boldsymbol{x}\,dy \\
&= \int \frac{p_{\text{data}}(\boldsymbol{x})}{q_G(\boldsymbol{x})}\,\ell(h(\boldsymbol{x}),y)\,q_G(\boldsymbol{x},y)\,d\boldsymbol{x}\,dy \\
&= \int p_{\text{data}}(\boldsymbol{x})\,q_G(y\mid\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y)\,d\boldsymbol{x}\,dy.
\end{aligned} \tag{17}$$

The term $p_{\text{data}}(\boldsymbol{x})$ adjusts for the marginal discrepancy between p_{data} and q_G, ensuring that the contribution of each sample $\boldsymbol{x}$ in the expectation reflects its true data likelihood. Hence, Eq. (17) represents the unbiased risk with respect to the true data marginal:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{Q}_G}\big[r(\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y)\big] \;=\; \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\Big[\mathbb{E}_{y\sim q_G(y|\boldsymbol{x})}[\ell(h(\boldsymbol{x}),y)]\Big]. \tag{18}$$

Assume the generator is sufficiently well-trained based on the strong conditionally pre-trained teacher and is class faithful, *i.e.*, $(\boldsymbol{x},y)\sim\mathcal{Q}_G$, the inner expectation naturally approximates $\mathbb{E}_{y\sim p_{\text{data}}(y|\boldsymbol{x})}[\ell(h(\boldsymbol{x}),y)]$. Consequently, the reweighted risk converges to the true expected risk:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{Q}_G}\big[r_D(\boldsymbol{x})\,\ell(h(\boldsymbol{x}),y)\big] \;\approx\; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}}[\ell(h(\boldsymbol{x}),y)]. \tag{19}$$

This establishes that the reweighting mechanism based on the discriminator-estimated ratio $r_D(\boldsymbol{x})$ corrects for the marginal distributional shift between $q_G(\boldsymbol{x})$ and $p_{\text{data}}(\boldsymbol{x})$, thereby aligning optimization on $\mathcal{Q}_G$ with learning on $\mathcal{P}$.

A.2 Detailed Derivation for Eq.15

Let $(\Omega,\mathcal{F},\mu)$ be a measurable space and assume $\mathcal{P},\mathcal{Q}$ admit densities $p = \frac{d\mathcal{P}}{d\mu}$ and $q = \frac{d\mathcal{Q}}{d\mu}$ w.r.t. μ.

Define the total variational distance:

$$\mathcal{D}_{\text{TV}}(\mathcal{P}\|\mathcal{Q}) = \tfrac{1}{2}\int_\Omega |p-q|\,d\mu = 1 - \int_\Omega \min\{p,q\}\,d\mu. \tag{20}$$

For the bounded loss $|\ell|\le L$,

$$\begin{aligned}
\Big|\mathbb{E}_{\mathcal{P}}[\ell(h)] - \mathbb{E}_{\mathcal{Q}}[\ell(h)]\Big| &= \left|\int \ell(h(\boldsymbol{x}),y)\,d(\mathcal{P}-\mathcal{Q})\right| && (21)\\
&\le \int |\ell(h(\boldsymbol{x}),y)|\,\big|d(\mathcal{P}-\mathcal{Q})\big| && (22)\\
&\le L\int \big|d(\mathcal{P}-\mathcal{Q})\big| && (23)\\
&= 2L\,\mathcal{D}_{\text{TV}}(\mathcal{P}\|\mathcal{Q}). && (24)
\end{aligned}$$

By Pinsker's inequality [13], for any $\mathcal{R},\mathcal{S}$ with densities r,s,

$$\mathcal{D}_{\text{TV}}(\mathcal{R}\|\mathcal{S}) \;\le\; \sqrt{\tfrac{1}{2}\,D_{\text{KL}}(\mathcal{R}\|\mathcal{S})}. \tag{25}$$

Equivalently, $\mathcal{D}_{\text{KL}}(\mathcal{R}\|\mathcal{S}) \ge 2\,\mathcal{D}_{\text{TV}}(\mathcal{R}\|\mathcal{S})^2$.

Fix $\pi\in(0,1)$ and set the mixture $\mathcal{M} = \pi\mathcal{P} + (1-\pi)\mathcal{Q}$ with density $m = \pi p + (1-\pi)q$. Define the π-Jenson-Shannon divergence

$$D_{\text{JS},\pi}(\mathcal{P}\|\mathcal{Q}) = \pi\,D_{\text{KL}}(\mathcal{P}\|\mathcal{M}) + (1-\pi)\,D_{\text{KL}}(\mathcal{Q}\|\mathcal{M}). \tag{26}$$

Applying Pinsker's to each KL term and then expressing the two TVs through $\mathcal{D}_{\text{TV}}(\mathcal{P},\mathcal{Q})$:

$$\begin{aligned}
\mathcal{D}_{\text{TV}}(\mathcal{P}\|\mathcal{M}) &= \tfrac{1}{2}\int |p-m|\,d\mu = \tfrac{1}{2}\int |(1-\pi)(p-q)|\,d\mu = (1-\pi)\,\mathcal{D}_{\text{TV}}(\mathcal{P},\mathcal{Q}),\\
\mathcal{D}_{\text{TV}}(\mathcal{Q}\|\mathcal{M}) &= \tfrac{1}{2}\int |q-m|\,d\mu = \tfrac{1}{2}\int |\pi(q-p)|\,d\mu = \pi\,\mathcal{D}_{\text{TV}}(\mathcal{P},\mathcal{Q}).
\end{aligned} \tag{27}$$

Hence,

$$D_{\mathrm{JS},\pi}(\mathcal{P}\|\mathcal{Q}) \geq \pi \cdot 2\,\mathcal{D}_{\mathrm{TV}}(\mathcal{P}\|\mathcal{M})^2 + (1-\pi)\cdot 2\,\mathcal{D}_{\mathrm{TV}}(\mathcal{Q}\|\mathcal{M})^2 \tag{28}$$

$$= 2\Big(\pi(1-\pi)^2 + (1-\pi)\pi^2\Big)\,\mathcal{D}_{\mathrm{TV}}(\mathcal{P}\|\mathcal{Q})^2 \tag{29}$$

$$= 2\,\pi(1-\pi)\,\mathcal{D}_{\mathrm{TV}}(\mathcal{P}\|\mathcal{Q})^2. \tag{30}$$

For the standard Jensen-Shannon divergence, *i.e.*, $\pi = \frac{1}{2}$,

$$D_{\mathrm{JS}}(\mathcal{P}\|\mathcal{Q}) \;\geq\; \tfrac{1}{2}\,\mathcal{D}_{\mathrm{TV}}(\mathcal{P}\|\mathcal{Q})^2 \tag{31}$$

$$\mathcal{D}_{\mathrm{TV}}(\mathcal{P}\|\mathcal{Q}) \;\leq\; \sqrt{2\,D_{\mathrm{JS}}(\mathcal{P}\|\mathcal{Q})}\,. \tag{32}$$

Combining the results, we have

$$\Big|\mathbb{E}_{\mathcal{P}}[\ell(h)] - \mathbb{E}_{\mathcal{Q}}[\ell(h)]\Big| \;\leq\; 2L\,\mathcal{D}_{\mathrm{TV}}(\mathcal{P}\|\mathcal{Q}) \;\leq\; 2L\sqrt{2\,D_{\mathrm{JS}}(\mathcal{P}\|\mathcal{Q})}\,. \tag{33}$$

Lastly, since any quantity x is trivially upper-bounded by its absolute value, *i.e.*, $x \leq |x|$, we relate Eq. 33 to the first summand in Eq. 15 by $\Big|\mathbb{E}_{\mathcal{P}}[\ell(h)] - \mathbb{E}_{\mathcal{Q}}[\ell(h)]\Big| \geq \mathbb{E}_{\mathcal{P}}[\ell(h)] - \mathbb{E}_{\mathcal{Q}}[\ell(h)]$, giving a valid upper bound for the non-absolute risk difference.

B Algorithmic Description of GDA-DAR

We summarize the overall training scheme in Algorithm 1, which illustrates a dynamic, alternating adversarial optimization. Kindly note that the discriminator update is presented in this explicit, separate form of alternating optimization for conceptual clarity, which corresponds to an architecture where D_η is a standalone network [42]. In alternative formulations mentioned in Section 4.3–when the discriminator is integrated into the student's U-Net and shares an encoder [118, 120]–its parameters η are shared with the student's parameters θ. In that scenario, the separate discriminator update is omitted, and the adversarial loss is incorporated into the total generator loss to be optimized jointly.

C Experimental Details

C.1 Details of the Datasets

CIFAR-10 consists of 60,000 RGB images, each with a resolution of 32×32 pixels. The dataset is evenly distributed across 10 distinct classes, with each class containing 6,000 images. By default, these are split into 5,000 training samples and 1,000 test samples per class, offering a balanced and computationally efficient benchmark for evaluating machine learning algorithms.

ImageNet-1K is a dataset comprising 1.28 million images labeled across 1,000 object categories. Each image is annotated with one or more class labels, enabling comprehensive studies of object recognition in diverse real-world scenarios.

C.2 Details of the Metrics

We assess model performance based on accuracy. For ImageNet-1K, which contains 1,000 possible classes, we report both Top-1 and Top-5 accuracies.

We evaluate the generative models based on the Fréchet Inception Distance (FID) [30], a widely used metric to evaluate the quality of images generated by generative models. This is based on the distance between the feature distributions of real and generated images in the latent space of a pretrained Inception network.

Table 3 details the GPU hours required to generate the augmenting dataset. For CIFAR-10, computations were performed on a single 80GB-A100 GPU. For other datasets, we utilized eight 80GB-A100 GPUs, employing the maximum possible batch size that is a power of two.

C.3 Details of the Implementation

For CIFAR-10, we set $\alpha = 1.2$ for SiDA and train using the Adam optimizer with a learning rate of 1×10^{-5}. CTM is trained with a batch size of 128 for 256 steps per batch, using a student

Algorithm 1 Generative data augmentation in supervised training with GDA-DAR framework.

Input: Teacher diffusion model ϵ_ϕ; real dataset $\mathcal{S}_\mathcal{P}$; hyperparameters: learning rates $\alpha_\theta, \alpha_\eta, \alpha_h$, loss weights $\lambda_1, \lambda_2, \lambda_{\text{aug}}$, truncation threshold γ; number of training iterations $N_{\text{gen}}, N_{\text{clf}}$.

Phase 1: Adversarial Score Distillation (D & A stages)
Initialize student generator G_θ and auxiliary score network ϵ_ψ from teacher ϵ_ϕ.
Initialize discriminator D_η.
for $i = 1, \ldots, N_{\text{gen}}$ **do**
  Sample real batch $\{\boldsymbol{x}_j\}_{j=1}^B \sim \mathcal{S}_\mathcal{P}$; noise batch $\{\boldsymbol{z}_j\}_{j=1}^B \sim \mathcal{N}(0, \mathbf{I})$.
  Generate synthetic batch $\{\hat{\boldsymbol{x}}_j = G_\theta(\boldsymbol{z}_j)\}_{j=1}^B$.
  // Update Discriminator
  Compute adversarial loss $\mathcal{L}^{(\text{adv.})}(\theta, \eta)$ using Eq. 13 on real and synthetic batches.
  Update discriminator: $\eta \leftarrow \eta - \alpha_\eta \nabla_\eta \left(-\mathcal{L}^{(\text{adv.})}(\theta, \eta)\right)$.
  // Update Student Generator
  Compute score distillation loss $\mathcal{L}^{(\text{SD})}(\theta)$ using Eq. 4.
  Compute generator's adversarial loss $\mathcal{L}_G^{(\text{adv.})}(\theta) = -\mathbb{E}_{\boldsymbol{z},t}[\log D_\eta(\boldsymbol{y}_t)]$.
  Compute total generator loss $\mathcal{L}_G = \lambda_1 \mathcal{L}^{(\text{SD})}(\theta) + \lambda_2 \mathcal{L}_G^{(\text{adv.})}(\theta)$.
  Update generator: $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta \mathcal{L}_G$.
end for

Phase 2: Reweighted Classifier Training (R stage)
Initialize discriminative hypothesis (classifier) h.
Freeze parameters θ and η.
for $i = 1, \ldots, N_{\text{clf}}$ **do**
  // Real Data Path
  Sample real batch $\{\boldsymbol{x}_j, y_j\}_{j=1}^B \sim \mathcal{S}_\mathcal{P}$.
  Compute real loss $\mathcal{L}^{(\text{real})} = \frac{1}{B} \sum_{j=1}^B \ell(h(\boldsymbol{x}_j), y_j)$.
  // Synthetic Augmentation Path
  Let synthetic batch size $B_{\text{aug}} = \lfloor \lambda_{\text{aug}} B \rfloor$.
  Sample noise batch $\{\boldsymbol{z}_s, y_s\}_{j=1}^{B_{\text{aug}}} \sim \mathcal{N}(0, \mathbf{I}) \times \text{Cat}(\mathcal{Y})$.
  Generate synthetic batch $\{\hat{\boldsymbol{x}}_s\}_{j=1}^{B_{\text{aug}}} = G_\theta(\{\boldsymbol{z}_s\}_{j=1}^{B_{\text{aug}}})$.
  // Compute Importance Weights for synthetic batch
  for $j = 1, \ldots, B_{\text{aug}}$ **do**
    Compute density ratio $r_j = D_\eta(\hat{\boldsymbol{x}}_{s,j}) / (1 - D_\eta(\hat{\boldsymbol{x}}_{s,j}))$.
    Truncate weight $\tilde{r}_j = \min(r_j, \gamma)$.
  end for
  if self_norm **then**
    Self-normalize: $\tilde{r}_j \leftarrow \bar{r}_j \Big/ \sum_{k=1}^{B_{\text{aug}}} \tilde{r}_k \quad \text{for } j = 1, \ldots, B_{\text{aug}}$
  end if
  Compute reweighted loss $\mathcal{L}^{(\text{reweight})} = \sum_{j=1}^{B_{\text{aug}}} \tilde{r}_j \cdot \ell(h(\hat{\boldsymbol{x}}_{s,j}), y_{s,j})$.
  // Combined ppdate of the hypothesis h
  Compute total loss $\mathcal{L}_{\text{total}} = \mathcal{L}^{(\text{real})} + \mathcal{L}^{(\text{reweight})}$.
  Update hypothesis: $h \leftarrow h - \alpha_h \nabla_h \mathcal{L}_{\text{total}}$.
end for

Output: Trained hypothesis h.

learning rate of 3×10^{-4} and a discriminator learning rate of 2×10^{-3}. For ImageNet-1K, we distill EDM2-XXL using SiDA across 8 A100-80GB GPUs with $\alpha = 1.0$, per-GPU batch size 64, and gradient accumulation every 128 iterations, using Adam with learning rate 5×10^{-5}. CIFAR-10 models are trained for 300 epochs with a batch size of 128 using momentum SGD (learning rate 0.1). Hyperparameters λ_1 and λ_2 for adversarial alignment follow prior work.

For ImageNet-1K, ResNet-50 is trained for 90 epochs with batch size 4096 and initial learning rate 1.6, while ViT-S/16 is trained for 300 epochs with batch size 1024, using AdamW [57] and initial learning rate 3×10^{-3}. Self-normalization is applied with $\gamma = 1$ to obtain $r(\boldsymbol{x})$ for reweighting.

Diffusion outputs are generated at 32×32 resolution for CIFAR-10 and 512×512 for ImageNet, with the latter downsampled to 224×224 before classification to match baseline protocols.

C.3.1 Classifier Training

The hyperparameters used for training the hypotheses, *i.e.*, the classification models, on the CIFAR-10 and ImageNet-1K datasets are detailed in Tables A1 and A2, respectively. Models are trained using a Stochastic Gradient Descent (SGD) optimizer across 90 epochs. The learning rate is programmed to initially ramp up from 0.0 to 0.4 over the first five epochs, followed by decrements of 0.1 at the 30th, 60th, and 80th epochs.

For CIFAR-10 training, we utilize standard augmentation techniques including padding each image by 4 pixels on all sides, then performing a random 32×32 crop from either the padded image or its horizontal flip, following the methods described in [28]. For ImageNet-1K, the models undergo training using a 224×224 random crop, applied with bilinear interpolation, from either the original image or its horizontal flip.

The detailed recipes of the hypothesis training are reported in Table A1 and Table A2 respectively.

Table A1: Training recipes for ResNet-18 and VGG-16 on CIFAR-10.

Model	ResNet-18	VGG-16
Batch size	128	128
Epochs	300	300
Optimizer	Momentum SGD	Momentum SGD
Learning rate	0.1	0.1
LR scheduler	CosineAnnealingLR	CosineAnnealingLR
Nesterov	True	True
Weight decay	5e-4	5e-4

Table A2: Training recipes for ResNet-50 and ViT-S/16 on ImageNet-1K.

Model	ResNet-50	ViT-S/16
Batch size	4096	1024
Epochs	130	300
Optimizer	Momentum SGD	AdamW
Learning rate	1.6	0.003
LR scheduler	CosineannealingLR	CosineAnnealingLR
Weight decay	1e-4	0.3
Warmup epochs	5	30
Label smoothing	0	0.11
Mixup probability	0	0.2
Cutmix α	0	1.0

C.3.2 Generative models training

For the base diffusion model, we adopt the public weight for EDM Diffusion on CIFAR-10 and EDM2 on ImageNet-1K for their state-of-the-art performance.

For experiments on the CIFAR-10 dataset (see Tables 1 and 4), we set $\alpha = 1.2$ for SiDA and train using the Adam optimizer with a learning rate of 1e-5. The batch size is set to 256, and gradient accumulation is performed every iteration (i.e., accumulation round = 1). We do not use mixed precision training (fp16) for SiDA on CIFAR-10. Dropout and data augmentation settings follow those used in EDM for CIFAR-10. All other SiDA-specific hyperparameters are kept consistent with the original implementation as described in [120].

For CTM, CTM is trained with batch size 128 for 256 steps per batch, using a student learning rate of 3×10^{-4} and a discriminator learning rate of 2×10^{-3} Mixed precision training (fp16) was employed to match the original setup. The exponential moving average (EMA) decay rate is 0.999. Other CTM-specific parameters adhere to the settings reported in [42].

On the ImageNet-1K dataset, we distilled EDM2-XXL with a batch size of 2048 and gradient accumulation performed every 128 iterations. train using the Adam optimizer with a learning rate of 5e-5. Mixed precision training (fp16) was used for SiDA on this dataset. Dropout parameters follow those of EDM2 on ImageNet-1K, and all other SiDA-specific hyperparameters remain consistent with those in the original work [120].

C.3.3 Training Procedure

In our experiments on the CIFAR-10 dataset (shown in Tables 1 and 4), we investigated two distinct strategies for generating synthetic data: the static approach and the dynamic approach. For the static approach, we created 50,000 synthetic samples, equivalent in size to the original dataset, which remained consistent throughout the entire training process. In contrast, the dynamic approach involved generating a new set of 50,000 synthetic samples at the start of each epoch, ensuring that the synthetic dataset used for training was entirely refreshed every epoch. For the experiments on the ImageNet-1K dataset, we generated 1.28 million synthetic samples for training, which is equivalent to the size of the original dataset. Given the large scale of ImageNet-1K, regenerating the synthetic dataset at each epoch would be computationally expensive and time-consuming. Therefore, we opted for the static approach.

C.3.4 Implementation Details and Notes

While our framework comprises three stages, its implementation is manageable due to its modular design. The (D), (A), and (R) stages are sequential and decoupled. As noted in relevant work [120, 42], stability is observed in the adversarial distillation. This can be attributed to the design of initializing the student with a pre-trained teacher, which bypasses the unstable early stages of typical GAN training. Hence, for the student generator and discriminator training phase, a single, fixed set of distillation hyperparameters was used across all datasets. For the final classifier training (R) stage, we simply follow standard, existing recipes from the literature. In addition,

The primary tuning consideration *i)* balancing the loss weights λ_1 and λ_2 remain at a comparable magnitude; and *ii)* considering truncation threshold γ and the use of self-normalization to achieve a favorable bias-variance trade-off in the estimated importance ratios.

Additionally, a key in-process indicator of a successful D+A phase is the student generator's own performance. We (and others [120, 42, 109, 58]) observe that the student's FID score remains stable and often improves beyond that of the original teacher. This provides direct evidence that the discriminator is providing high-quality gradients and has successfully captured the real-versus-generated distribution gap. We recommend users monitor the student's generation quality as a primary check for a healthy run.

D Additional Experiment

D.1 Ablation Study of Variance Reduction Techniques

We ablate on the two variance-reducing techniques used, namely *self-normalization* and *trucation*, with respect to the hyperparameter that controls the upper bound of γ in Table A3. Our observations indicate that removing self-normalization generally leads to a decrease in classification performance. This decrease is particularly significant on ImageNet-1K, likely because weight variance can be greater with such a massive dataset. Furthermore, we observe that $\gamma = 1$ appears to be the optimal upper bound for truncation. At this value, a synthetic image contributes equally to a real image in the loss computation and subsequent optimization.

Table A3: Classification accuracies (top-1) under different weight variance reduction techniques for additive, static augmentation using EDM/EDM2-SiD distillation.

Self-norm	γ	CIFAR-10		ImageNet-1K	
		ResNet-18	VGG-16	ResNet-50	ViT-S/16
✗	0.5	95.30	94.08	77.09	80.05
✗	1	96.15	94.60	77.75	80.90
✗	2	95.72	94.10	77.23	80.12
✔	0.5	95.70	94.37	77.73	80.47
✔	1	96.21	94.64	78.03	81.17
✔	2	95.86	94.40	77.64	80.68

In addition, we conducted an empirical study reporting the sample variance of the weights for 100,000 generated ImageNet samples. As shown in Table A4, both truncation and self-normalization progressively reduce the variance.

Table A4: Sample variance of importance weights $r(\boldsymbol{x})$ for 100,000 generated ImageNet samples, under different truncation (γ) and self-normalization (SN) schemes.

Truncation γ	**Variance of $r(\boldsymbol{x})$ (no SN)**	**Variance after SN**
∞ (no clip)	1.21	0.42
2.0	0.37	0.12
1.0	0.18	0.05
0.5	0.07	0.02

D.2 Discriminator Noising Scheme

When employing the discriminator as the encoder for the student generator's backbone, it's necessary to transform the clean input, $\boldsymbol{x}_{\text{clean}}$, into a noisy version, $\boldsymbol{x}_{\text{noisy}}$. This is achieved by corrupting $\boldsymbol{x}_{\text{clean}}$ with a predetermined level of noise corresponding to a specific timestep t_0. The choice of this noising scheme, specifically the selection of t_0, can influence the features learned by the discriminator-encoder and subsequently the performance of the student generator.

We investigated several approaches for selecting t_0, as detailed in Table D.2. We observe that the uniform sampling strategy is the best among all ablated ones with $t_0 = 0.1T$ having comparable performance on both datasets. The approach of treating the shared encoder as the discriminator offers a trade-off: it obviates the need to tune the noising scheme, while avoiding the potential complexities and additional computational overhead associated with employing a separate, dedicated discriminator network.

Table A5: Classification accuracies (top-1) under different discriminator noising schemes for static additive augmentation using EDM/EDM2-SiD distillation.

π	CIFAR-10		ImageNet-1K	
	ResNet-18	VGG-16	ResNet-50	ViT-S/16
$\delta(0.05T)$	96.08	94.48	77.87	81.02
$\delta(0.1T)$	96.18	94.66	77.89	81.10
$\delta(0.5T)$	95.33	94.07	77.23	80.51
$\mathcal{U}(0, 0.5T)$	96.21	94.64	78.03	81.17

D.3 Noisy and Imbalanced Setting

While our initial goal was to focus on challenges like distribution mismatch and sampling cost in GDA, we further investigated how our approach would be suitable for these real-world scenarios such as on imbalanced and noisy data.

Class Imbalance We conducted experiments on CIFAR-10-LT [16] with an imbalance factor of 10 and 100 using the sa, me training protocol as in the balanced-data setting the dynamic scheme, with

ResNet-18 as the backbone in Tables A6 and A7 respectively. We also included a long-tailed-based method, SURE [54], for comparison. The evaluation tracks the progressive impact of each stage in our DAR-GDA pipeline (+D), (+A), and Reweighting (+R) with SiD [121]-distilled EDM [39] diffusion model. This performance likely stems from our method's ability to leverage the strong generative priors of the diffusion model teacher to populate tail classes, a different paradigm from methods like SURE, which are designed to learn directly from the imbalanced data.

Table A6: Results on CIFAR-10-LT with Imbalance Factor = 10.

Method	**Augment**	**Substitute**
Baseline	86.75	86.75
SURE [54]	95.15	95.15
EDM-SID (+D)	94.88	94.47
EDM-SID (+D+A)	95.53	94.98
EDM-SID (+D+A+R)	**95.78**	95.02

Table A7: Results on CIFAR-10-LT with Imbalance Factor = 100.

Method	**Augment**	**Substitute**
Baseline	69.76	69.76
SURE [54]	86.75	86.75
EDM-SID (+D)	91.45	91.02
EDM-SID (+D+A)	91.84	91.35
EDM-SID (+D+A+R)	**92.01**	91.64

Noisy Label We also conducted an experiment on the CIFAR-10-N dataset [102] using the "aggre" noise labels to simulate a real-world scenario with untrusted data. We used a ResNet-18 backbone with the augmenting data protocol. We compare our result with a noisy-label learning method, ProMix [103] in Table A8.

Table A8: Results on CIFAR-10-N with ResNet-18.

Method	**Static**	**Dynamic**
Baseline	87.77	87.77
ProMix [103]	**97.65**	**97.65**
EDM-SID (+D)	93.48	95.79
EDM-SID (+D+A)	93.80	96.26
EDM-SID (+D+A+R)	93.81	96.30

Our method contains no explicit mechanism for label correction; the discriminator is designed to down-weight visually atypical samples, not to detect label noise. Consequently, while the strong generative prior provides a robust foundation, the performance does not surpass that of specialist methods like ProMix, which are explicitly architected to handle label noise.

D.4 Comparisons with Non-generative Data Augmentation Methods

Our primary work focused on challenges within the GDA paradigm. To provide a broader context, we conducted additional experiments comparing our DAR-GDA framework against two widely-used non-generative data augmentation baselines: AutoAugment [14] and RandAugment [15].

Table A9: Comparison with non-generative augmentation on CIFAR-10 with ResNet-18.

Method	Static	Dynamic
Real-only	95.00	95.00
AutoAugment [14]	95.85	95.85
RandAugment [15]	95.79	95.79
EDM-SID (+D)	95.48	96.29
EDM-SID (+D+A)	95.84	96.40
EDM-SID (+D+A+R)	96.21	96.73
EDM-SID (+D+A+R) + RandAugment	**96.48**	**96.97**

Table A10: Comparison with non-generative augmentation on ImageNet-1K with ResNet-50.

Method	Top-1
Real-only	76.37
AutoAugment [14]	77.62
RandAugment [15]	77.65
EDM-SID (+D)	77.15
EDM-SID (+D+A)	77.89
EDM-SID (+D+A+R)	78.03
EDM-SID (+D+A+R) + RandAugment	**78.29**

We notice that our full pipeline (DAR-GDA +D+A+R) outperforms both AutoAugment and RandAugment. Furthermore, we highlight that our generative approach is complementary to these traditional, non-generative data augmentation techniques. These transforms can be applied to synthetic images just as they are to real ones, further diversifying the training set. To demonstrate this, we applied RandAugment on top of our generated data, which yields further improvement.

D.5 On FFHQ Dataset

To further validate our framework's robustness, we conducted experiments on a gender classification task using the FFHQ 64x64 dataset [**?**]. Since the EDM teacher and our student generator are unconditional, we first created a high-quality labeled dataset by using an external classifier of ResNet-50 pretrained on Celeb-A to generate raw predictions, which were then human-verified. We then applied our "static" scheme with results shown in Table A11. We observe monotonic performance gains as each stage of the DAR pipeline is applied.

Table A11: Gender classification accuracy on FFHQ 64x64. The baseline was trained on real data with human-verified pseudo-labels.

Method	ResNet-18	VGG-16
Real-only	94.08	93.10
EDM-SID (+D)	94.20	93.25
EDM-SID (+D+A)	94.63	93.62
EDM-SID (+D+A+R)	**94.87**	**93.70**

E Computational Complexity Analysis

We perform a formal analysis of the computational complexity for our framework. We define the following computational costs:

T: The number of denoising steps in the teacher sampler, *e.g.*, $T \approx 1000$.

c_f: The wall-clock cost of one forward pass of the diffusion model backbone (assumed the same for the teacher and the one-step student).

N: The total number of synthetic instances to be generated.

C_{distill}: The one-time, upfront cost of distilling the teacher into the student generator.

C_{pretrain}: The cost of pre-training the teacher model, which is shared/sunk for both methods and cancels out in the comparison.

The core of our efficiency gain comes from reducing the per-sample generation cost from the $\mathcal{O}(T)$ complexity of iterative samplers, like DDPM [31] or DDIM [81], to an $\mathcal{O}(1)$ complexity. Table A12 contrasts these approaches.

Table A12: Comparison of sampling time complexity and theoretical speed-up.

Sampler	**# Reverse Steps**	**Asymptotic Cost**	**Theoretical Speed-up (vs. 1000-step DDPM)**
DDPM	1000	$\mathcal{O}(T)$	Baseline (1×)
DDIM (fast)	50	$\mathcal{O}(T)$	≈ 20×
One-step distilled	1	$\mathcal{O}(1)$	≈ 1000×

With these variables, we can compare the total cost of generating N samples using a naive iterative approach versus our DAR-GDA framework. The costs are summarized in Table A13.

Table A13: Computational cost comparison for generating N synthetic samples.

Pipeline	**Total Cost Equation**
Naive Diffusion GDA	$C_{\text{naive}} = N \cdot T \cdot c_f$
DAR-GDA	$C_{\text{DAR}} = C_{\text{distill}} + N \cdot c_f$

Our framework (DAR-GDA) is faster than the naive approach whenever $C_{\text{DAR}} < C_{\text{naive}}$, which implies:

$$C_{\text{distill}} + Nc_f < NTc_f \quad \Longrightarrow \quad C_{\text{distill}} < Nc_f(T-1) \tag{34}$$

We can estimate the one-time distillation cost as $C_{\text{distill}} = P \cdot N \cdot k \cdot c_f$, where P is the number of data passes (epochs) during distillation, N is the number of data samples in the training set, and k is the number of model evaluations per sample per pass.

Empirically, recent works [120, 42] suggest that one-step distillation converges in at most $P \leq 150$ passes, with $k = 3$, *i.e.*, teacher, fake, and student score network forwards. Assuming a teacher with $T = 1000$ steps, the ratio of distillation cost to savings is:

$$\frac{C_{\text{distill}}}{Nc_f(T-1)} = \frac{PNkc_f}{Nc_f(T-1)} \approx \frac{Pk}{T-1} \approx \frac{150 \cdot 3}{999} \approx 0.45 < 1 \tag{35}$$

Thus, the inequality is comfortably satisfied. Furthermore, if synthetic data is dynamically refreshed each epoch , *i.e.*, the "dynamic" scheme, the savings compound. In that case, the effective N becomes $N \cdot E$ (where E is the number of classifier training epochs), making the efficiency gain of DAR-GDA even more significant.