UNIVERSAL INVERSE DISTILLATION FOR MATCHING MODELS WITH REAL-DATA SUPERVISION (NO GANS)

Anonymous authors
Paper under double-blind review

ABSTRACT

While achieving exceptional generative quality, modern diffusion, flow, and other matching models suffer from slow inference, as they require many steps of iterative generation. Recent distillation methods address this by training efficient one-step generators under the guidance of a pre-trained teacher model. However, these methods are often constrained to only one specific framework, e.g., only to diffusion or only to flow models. Furthermore, these methods are naturally data-free, and to benefit from the usage of real data, it is required to use an additional complex adversarial training with an extra discriminator model. In this paper, we present **RealUID**, a universal distillation framework for all matching models that seamlessly incorporates real data into the distillation procedure without GANs. Our **RealUID** approach offers a simple theoretical foundation that covers previous distillation methods for Flow Matching and Diffusion models, and is also extended to their modifications, such as Bridge Matching and Stochastic Interpolants.

1 INTRODUCTION

In generative modeling, the goal is to learn to sample from complex data distributions (e.g., images), and two powerful paradigms for it are the **Diffusion Models** (DM) and the **Flow Matching** (FM) models. While they share common principles and are even equivalent under certain conditions (Holderrieth et al., 2024; Gao et al., 2025), they are typically studied separately. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) transform data into noise through a forward process and then learn a reverse-time stochastic differential equation (SDE) to recover the data distribution. Training minimizes score-matching objectives, yielding unbiased estimates of intermediate scores. Sampling requires simulating the reverse dynamics, which is computationally heavy but delivers high-quality and diverse results. Flow Matching (Lipman et al., 2023; Liu, 2022) instead interpolates between source and target distributions by learning the vector field of an ordinary differential equation (ODE). The field is estimated through unbiased conditional objectives, but the resulting ODE often has curved trajectories, making sampling costly due to expensive integration. Beyond these, **Bridge Matching** (Peluchetti, 2023; Liu et al., 2022b) and **Stochastic Interpolants** (Albergo et al., 2023) generalize the framework and naturally support *data couplings*, which are crucial for data-to-data translation. Since all of the above optimize *conditional matching* objectives to recover an ODE/SDE for generation, we refer to them collectively as *matching models*.

Despite their success, matching models share a major drawback: sampling is slow, as generation requires integrating many steps of an SDE or ODE. To address this, a range of distillation techniques have been proposed to compress multi-step dynamics into efficient one-step or few-step generators. Although matching models follow a similar mathematical framework, many distillation works consider only one particular framework, e.g., only Diffusion Models (Zhou et al., 2024a;b), Flow Matching (Huang et al., 2024), or Bridge Matching (Gushchin et al., 2025). Furthermore, these distillation methods are data-free by construction and cannot benefit from the utilization of real data without using additional GAN-based losses. *Thus, the following problems remain:*

1. Similar distillation techniques developed separately for similar matching models frameworks.
2. Absence of a natural way to incorporate real data in distillation procedures (without GANs).

Contributions. In this paper, we address these issues and present the following **main contributions**:

1. We present the *Universal Inverse Distillation with real data (RealUID)* framework for matching models, including diffusion and flow matching models (§3) as well as Bridge Matching and Stochastic Interpolants (Appendix C.). It unifies previously introduced Flow Generator Matching (FGM), Score Identity Distillation (SiD) and Inverse Bridge Matching Distillation (IBMD) methods (§3.2) for flow, score and bridge matching models respectively, provides simple yet rigorous theoretical explanations based on a linearization technique, and reveals the connections between these methods and inverse optimization (§3.3).
2. Our RealUID introduces a novel and natural way to incorporate real data directly into the distillation loss, eliminating the need for extra adversarial losses which require additional discriminator networks used in GANs from the previous works (§3.4).

2 Backgrounds on Training and Distilling Matching Models

We describe the Diffusion Models and Flow Matching frameworks (§2.1) and distillation methods for them (§2.3). Then, we discuss how real data can be added to distilling methods via GANs (§2.4)

Preliminaries. We work on the D-dimensional Euclidean space $\mathbb{R}^D$. This space is equipped with the standard scalar product $\langle x, y\rangle = \sum_{d=1}^D x_d y_d$, the ℓ_2-norm $\|x\| = \sqrt{\langle x, x\rangle}$ and ℓ_2-distance $\|x - y\|, \forall x, y \in \mathbb{R}^D$. We consider probability distributions from the set $\mathcal{P}(\mathbb{R}^D)$ of absolutely continuous distributions with finite variance and support on the whole $\mathbb{R}^D$.

2.1 Diffusion and Flow Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) consider a forward noising process that gradually transforms clean data p_0 into a noise p_T on the time interval $[0, T]$:

$$dx_t = f_t \cdot x_t dt + g_t \cdot dw_t, \quad x_0 \sim p_0,$$

where f_t and g_t are time-dependent scalars. This process defines a conditional distribution $p_t(x_t|x_0)$:

$$p_t(x_t|x_0) = \mathcal{N}(\alpha_t x_0 | \sigma_t^2 \mathbf{I}), \quad \text{where}$$

$$\alpha_t = \exp\left(\int_0^t f_s\, ds\right), \quad \sigma_t = \left(\int_0^t g_s^2 \exp\left(-2\int_0^s f_u\, du\right) ds\right)^{1/2}.$$

Each conditional distribution admits a conditional score function, describing it:

$$s_t(x_t|x_0) := \nabla_{x_t} \log p_t(x_t|x_0) = -(x_t - \alpha_t x_0)/\sigma_t^2.$$

The reverse dynamics from the noise distribution p_T to the data distribution p_0 is provided by the following reverse-time SDE:

$$dx_t = (f_t \cdot x_t - g_t^2 \cdot s_t(x_t))dt + g_t d\bar{w}_t,$$

where $s_t(x_t)$ is the unconditional score function of $p_t(x_t) = \int p(x_t|x_0)p(x_0)dx_0$ given by $s_t(x_t) = \mathbb{E}_{x_0 \sim p_0(\cdot|x_t)}[s_t(x_t|x_0)]$. This conditional expectation is learned via denoising score matching:

$$\mathcal{L}_{\text{DSM}}(s', p_0) = \mathbb{E}_{t\sim[0,T], x_0\sim p_0, x_t\sim p_t(\cdot|x_0)}\left[w_t \|s'_t(x_t) - s_t(x_t|x_0)\|_2^2\right], \tag{1}$$

where w_t are some positive weights. The reverse dynamics admits a probability flow ODE (PF-ODE):

$$dx_t = (f_t \cdot x_t - g_t^2 \cdot s_t(x_t)/2)dt, \quad u_t(x_t) := (f_t \cdot x_t - g_t^2 \cdot s_t(x_t)/2),$$

which provides faster inference than the SDE formulation.

Flow Matching framework (Lipman et al., 2023; Liu et al., 2023) constructs the flow directly by learning the drift $u_t(x_t)$. Specifically, for each data point $x_0 \sim p_0$, one defines a conditional flow $p_t(x_t|x_0)$ with the corresponding conditional vector field $u_t(x_t|x_0)$ generating it via ODE:

$$dx_t = u_t(x_t|x_0)dt.$$

Then to construct the flow between the noise p_T and data p_0, one needs to compute the unconditional vector field $u_t(x_t) = \mathbb{E}_{x_0\sim p_0(\cdot|x_t)}[u_t(x_t|x_0)]$ which generates the flow $p_t(x_t) = \int p(x_t|x_0)p(x_0)dx_0$. It can be done by solving the following Conditional Flow Matching problem:

$$\mathcal{L}_{\text{CFM}}(v, p_0) = \mathbb{E}_{t\sim[0,T], x_0\sim p_0, x_t\sim p_t(x_t|x_0)}\left[w_t \|v_t(x_t) - u_t(x_t|x_0)\|_2^2\right].$$

In practice, the most popular choice is the Gaussian conditional flows $p_t(x_t|x_0) = \mathcal{N}(\alpha_t x_0, \sigma_t^2 \mathbf{I})$. For this conditional flow samples can be obtained as $x_t = \alpha_t x_0 + \sigma_t \epsilon$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and the conditional drift can be calculated as $u_t(x_t|x_0) = \dot{\alpha}_t x_0 + \dot{\sigma}_t \epsilon$.

2.2 UNIVERSAL LOSS FOR MATCHING MODELS

From a mathematical point of view, it was shown in (Holderrieth et al., 2024; Gao et al., 2025) that flow and diffusion models basically share the same loss structure. We recall this structure but use our own notation. We call diffusion and flow models and their extensions as matching models.

Matching models work with a probability path $\{p_t\}_{t\in[0,T]}$ on the time interval $[0,T]$, transforming the desired data $p_0 \in \mathcal{P}(\mathbb{R}^D)$ to the noise $p_T \in \mathcal{P}(\mathbb{R}^D)$. This path is built as a mixture of simple conditional paths $\{p_t(\cdot|x_0)\}_{t\in[0,T]}$ conditioned on samples $x_0 \sim p_0$, i.e., $p_t(x_t) = \int_{\mathbb{R}^D} p_t(x_t|x_0)p_0(x_0)dx_0, \forall x_t \in \mathbb{R}^D$. The path $\{p_t\}_{t\in[0,T]}$ determines the function $f^{p_0} : [0,T]\times\mathbb{R}^D \to \mathbb{R}^D$ which recovers it (e.g., score function or drift generating it). The conditional paths also determine their own simple conditional functions $f^{p_0}(\cdot|x_0)$ so that they express $f_t^{p_0}(x_t) = \mathbb{E}_{x_0\sim p_0(\cdot|x_t)} f_t^{p_0}(x_t|x_0)$, where $p_0(\cdot|x_t)$ denotes data distribution p_0 conditioned on the sample x_t at time t. Since f^{p_0} cannot be computed directly, it is approximated by function $f : [0,T]\times\mathbb{R}^D \to \mathbb{R}^D$ via minimizing the squared ℓ_2-distance between the functions:

$$\|f_t(x_t) - f_t^{p_0}(x_t)\|^2 = \|f_t(x_t) - \mathbb{E}_{x_0\sim p_0(\cdot|x_t)} f_t^{p_0}(x_t|x_0)\|^2 \propto \mathbb{E}_{x_0\sim p_0(\cdot|x_t)}\|f_t(x_t) - f_t^{p_0}(x_t|x_0)\|^2.$$

Definition 1. *We define **Universal Matching (UM)** loss $\mathcal{L}_{UM}(f, p_0)$ that takes fake function f and distribution $p_0 \in \mathcal{P}(\mathbb{R}^D)$ as arguments and upon minimization over f returns the function f^{p_0}*

$$\mathcal{L}_{UM}(f,p_0) := \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0\sim p_0, x_t\sim p_t(\cdot|x_0)}\|f_t(x_t) - f_t^{p_0}(x_t|x_0)\|^2,\ f^{p_0} := \arg\min_f \mathcal{L}_{UM}(f,p_0), \quad (2)$$

where $t \sim [0,T]$ denotes uniform or weighted sampling of time t from the interval $[0,1]$.

2.3 DISTILLATION OF MATCHING-BASED MODELS

To solve the long inference problem of matching models, a line of distillation approaches sharing similar principles was introduced: **Score Identity Distillation (SiD)** (Zhou et al., 2024b), **Flow Generator Matching (FGM)** (Huang et al., 2024), and **Inverse Bridge Matching Distillation (IBMD)** (Gushchin et al., 2025), for diffusion, flow, and bridge matching models, respectively.

The **Score Identity Distillation (SiD)** approach (Zhou et al., 2024b;a) trains a student generator $G_\theta : \mathcal{Z} \to \mathbb{R}^D$ (parameterized by θ) that produces a distribution p_0^θ from a latent distribution $p^{\mathcal{Z}}$ on $\mathcal{Z}$. This approach minimizes the squared ℓ_2-distance between the known teacher score function $s^* := \arg\min_{s'} \mathcal{L}_{\text{DSM}}(s', p_0^*)$ on real data p_0^* and the unknown student score function s^θ:

$$\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}\|s_t^\theta(x_t^\theta) - s_t^*(x_t^\theta)\|^2, \quad \text{s.t. } s^\theta = \arg\min_{s'}\mathcal{L}_{\text{DSM}}(s', p_0^\theta), \quad (3)$$

where p_t^θ is the forward noising process for the generator distribution p_0^θ. The authors propose the tractable loss without $\arg\min$ and with parameter α_{SiD} to approximate the real gradients of (3) :

$$\begin{aligned}\mathcal{L}_{\text{SiD}}(\theta) &:= \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{z\sim p^{\mathcal{Z}}, x_0^\theta = G_\theta(z), x_t^\theta\sim p_t^\theta}[-2\omega_t\alpha_{\text{SiD}}\|s_t^*(x_t^\theta) - s_t^{sg[\theta]}(x_t^\theta)\|^2 \\ &+ 2\omega_t\langle s_t^*(x_t^\theta) - s_t^{sg[\theta]}(x_t^\theta), s_t^*(x_t^\theta) - s_t^\theta(x_t^\theta|x_0^\theta)\rangle], \quad s^\theta = \arg\min_{s'}\mathcal{L}_{\text{DSM}}(s', p_0^\theta), \quad (4)\end{aligned}$$

where gradients w.r.t. θ are not calculated for the variables under stop-gradient $sg[\cdot]$ operator. The SiD pipeline is two alternating steps: first, refine the fake score $s^{sg[\theta]}$ by minimizing the DSM loss (1) on new p_0^θ from the previous step. Then, update the generator G_θ using the gradient of (4) with the frozen $s^{sg[\theta]}$. The α_{SiD} parameter is chosen from the range $[0.5, 1.2]$, although theoretically only the value $\alpha_{\text{SiD}} = 0.5$ restores true gradient as we show in our paper.

The authors of **FGM** considered a similar approach, but for the Flow Matching models. Specifically, they also use a generator G_θ to produce a distribution p_0^θ, but instead of denoising score matching loss, consider conditional FM loss. The method minimizes the squared ℓ_2-distance between the fields:

$$\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t\sim p_t^\theta}\|u_t^\theta(x_t) - u_t^*(x_t)\|^2, \quad \text{s.t. } u^\theta := \arg\min_v \mathcal{L}_{\text{CFM}}(v, p_0^\theta), \quad (5)$$

where the interpolation path $\{p_t^\theta\}_{t\in[0,T]}$ is constructed between the noise p_T and generator p_0^θ distributions. To avoid the same problem of differentiating through $\arg\min$ operator as in SiD, the authors derive a tractable loss whose gradients match those of (5):

$$\mathcal{L}_{\text{FGM}}(\theta) := \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{z\sim p^{\mathcal{Z}}, x_0^\theta = G_\theta(z), x_t^\theta\sim p_t^\theta}[-\|u_t^*(x_t^\theta) - u_t^{sg[\theta]}(x_t^\theta)\|^2 \quad (6)$$

$$+ \quad 2\langle u_t^*(x_t^\theta) - u_t^{sg[\theta]}(x_t^\theta), u_t^*(x_t^\theta) - u_t^\theta(x_t^\theta|x_0^\theta)\rangle], \text{ s.t. } u^\theta = \arg\min_v \mathcal{L}_{\text{CFM}}(v, p_0^\theta).$$

We consider distillation of matching models working with data couplings such as Inverse Bridge Matching Distillation for Bridge Matching models and Stochastic Interpolants in Appendix C. Notably, all these approaches (SiD, FGM, IBMD) are *data-free*, i.e., they do not use any real data from p_0^* to train a generator by construction of the used objective functions.

2.4 GANs FOR REAL DATA INCORPORATION

FGM and SiD methods exhibit strong performance in one-step generation tasks. However, the generator in these methods is trained under the guidance of the teacher model alone. This means the generator cannot get more information about the real data that the teacher has learned. For example, it cannot correct the teacher's errors. To address this, recent works (Yin et al., 2024a; Zhou et al., 2024a) propose adding real data via a GAN framework (Goodfellow et al., 2014). In such approaches, the encoder of fake model f is typically augmented with an additional head to serve as a discriminator D with the following adversarial loss:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{t\sim[0,T]}\left[\mathbb{E}_{x_t^*\sim p_t^*}\left[\ln D_t\left(x_t^*\right)\right] + \mathbb{E}_{x_t^\theta\sim p_t^\theta}\left[\ln[1 - D_t\left(x_t^\theta\right)]\right]\right]. \tag{7}$$

The overall objective in such hybrid frameworks (Zhou et al., 2024a) consists of:

Generator loss:

$$\mathcal{L}_{G_\theta} = \lambda_{\text{dist}}\mathcal{L}_{\text{FGM/SiD}}^{G_\theta} + \lambda_{\text{adv}}^{G_\theta}\mathcal{L}_{\text{adv}}^{G_\theta}, \tag{8}$$

Fake model loss:

$$\mathcal{L}_D = \lambda_{\text{dist}}\mathcal{L}_{\text{FGM/SiD}}^{f} + \lambda_{\text{adv}}^{D}\mathcal{L}_{\text{adv}}^{D}. \tag{9}$$

Here, λ_{dist}, $\lambda_{\text{adv}}^{G_\theta}$, and λ_{adv}^{D} are weighting coefficients for the distillation and adversarial components. Despite empirical gains, the GAN-augmented formulation entails nontrivial costs: it necessitates architectural modifications, such as an auxiliary discriminator head, and inherits the well-known optimization problems of adversarial training, such as non-stationary objectives, mode collapse, and sensitivity to training dynamics.

3 UNIVERSAL DISTILLATION OF MATCHING MODELS WITH REAL DATA

In this section, we present our novel RealUID approach for matching models enhanced by real data. First, we show that the previous data-free distillation methods can be unified under the single UID framework (§3.1). Then, we describe how this framework is connected to prior works (§3.2) and inverse optimization (§3.3). Using this intuition, we propose and discuss the real data modified UID framework (RealUID) with a natural way to incorporate real data without GANs (§3.4).

3.1 UNIVERSAL INVERSE DISTILLATION

To learn a complex real data distribution p_0^*, one usually trains a teacher function $f^* := \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*)$ which is then used in a multi-step sampling procedure (Def. 1). To avoid time-consuming sampling, one can train a simple *student generator* $G_\theta : \mathcal{Z} \to \mathbb{R}^D$ with parameters θ to reproduce the real data p_0^* from the distribution $p^{\mathcal{Z}}$ on the latent space $\mathcal{Z}$. The teacher function serves as a guide that shows how close the student distribution p_0^θ and the real data p_0^* are. FGM and SiD methods (§2.3) train such generator via minimizing the squared ℓ_2-distance between the known teacher function f^* and an unknown student function $f^\theta := \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^\theta)$:

$$\begin{aligned}
&\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}\|f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta)\|^2 = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}\|f_t^*(x_t^\theta) - \mathbb{E}_{x_0^\theta\sim p_0^\theta(\cdot|x_t^\theta)} f_t^\theta(x_t^\theta|x_0^\theta)\|^2 \\
&= \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}[\|f_t^*(x_t^\theta)\|^2] - 2\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta, x_0^\theta\sim p_0^\theta(\cdot|x_t^\theta)}[\langle f_t^*(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle] \\
&+ \underbrace{\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}[\|\mathbb{E}_{x_0^\theta\sim p_0^\theta(\cdot|x_t^\theta)}[f_t^\theta(x_t^\theta|x_0^\theta)]\|^2]}_{\text{not tractable}},
\end{aligned} \tag{10}$$

where $\{p_t^\theta\}_{t\in[0,T]}$ is the probability path constructed between generator distribution p_0^θ and noise p_T. The problem is that the final term (10) cannot be calculated directly. This is because it involves the

math expectation inside the squared norm, unlike the other terms which are linear in the expectations. It means that a simple estimate of $\|f_t^\theta(x_t^\theta|x_0^\theta)\|^2$ using samples x_0^θ and x_t^θ will be *biased*. Moreover, to differentiate through the math expectation inside the norm, an explicit dependence of p_0^θ on θ is required, while, in practice, usually only dependence of samples x_0^θ on θ is known.

Making loss tractable via linearization. To resolve this, we use a linearization technique. For a fixed point x_t^θ and time t, we reformulate the squared norm as a maximization problem. We achieve this by introducing an auxiliary function $\delta : [0, T] \times \mathbb{R}^D \to \mathbb{R}^D$ and using the identity

$$\|f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta)\|^2 = \max_{\delta_t(x_t^\theta)} \left\{-\|\delta_t(x_t^\theta)\|^2 + 2\langle \delta_t(x_t^\theta), f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta)\rangle\right\}$$

$$= \max_{\delta_t(x_t^\theta)} \mathbb{E}_{x_0^\theta \sim p_0^\theta(\cdot|x_t^\theta)} \left\{-\|\delta_t(x_t^\theta)\|^2 + 2\langle \delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\langle \delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle\right\}. \quad (11)$$

The reparameterization $\delta = f^* - f$ with a *fake function* $f : [0, T] \times \mathbb{R}^D \to \mathbb{R}^D$ allows to get:

$$(11) = \max_{f_t(x_t^\theta)} \mathbb{E}_{x_0^\theta \sim p_0^\theta(\cdot|x_t^\theta)} \left\{-\|f_t^*(x_t^\theta) - f_t(x_t^\theta)\|^2 + 2\langle f_t^*(x_t^\theta) - f_t(x_t^\theta), f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta|x_0^\theta)\rangle\right\} \quad (12)$$

$$= \max_{f_t(x_t^\theta)} \mathbb{E}_{x_0^\theta \sim p_0^\theta(\cdot|x_t^\theta)} \Big\{ \underbrace{\|f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta|x_0^\theta)\|^2}_{=\mathcal{L}_{\text{UM}}(f^*, p_0^\theta)} - \underbrace{\|f_t(x_t^\theta) - f_t^\theta(x_t^\theta|x_0^\theta)\|^2}_{=\mathcal{L}_{\text{UM}}(f, p_0^\theta)} \Big\}. \quad (13)$$

Since now all expectations are linear and can be estimated, the final step is to compute the expectation over all points x_t^θ and times t and minimize it over the generator distribution p_θ.

Summary. We build a universal distillation framework as a single min-max optimization (14), implicitly minimizing squared ℓ_2-distance between teacher and student functions. When real and generated probability paths match, these functions match as well, and the distance attains its minimum.

Theorem 1 (Real data generator minimizes UID loss). *Let teacher $f^* := \arg\min_f \mathcal{L}_{UM}(f, p_0^*)$ be the minimizer of UM loss (Def. 1) on real data $p_0^* \in \mathcal{P}(\mathbb{R}^D)$. Then real data generator G_{θ^*}, s.t. $p_0^{\theta^*} = p_0^*$, is a solution to the min-max optimization of **Universal Inverse Distillation (UID) loss** $\mathcal{L}_{UID}(f, p_0^\theta)$ over fake function f and generator distribution p_0^θ*

$$\min_\theta \max_f \left\{\mathcal{L}_{UID}(f, p_0^\theta) := \mathcal{L}_{UM}(f^*, p_0^\theta) - \mathcal{L}_{UM}(f, p_0^\theta)\right\}. \quad (14)$$

Lemma 1 (UID loss minimizes squared ℓ_2-distance). *Maximization of UID loss (14) over fake function f retrieves student function $f^\theta := \arg\min_f \mathcal{L}_{UM}(f, p_0^\theta)$ and represents the squared ℓ_2-distance between it and the teacher f^*:*

$$f^\theta = \arg\max_f \mathcal{L}_{UID}(f, p_0^\theta), \quad \max_f \mathcal{L}_{UID}(f, p_0^\theta) = \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{x_t^\theta \sim p_t^\theta} \|f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta)\|^2. \quad (15)$$

Note that the distance (15) mostly captures mismatches for the points from generator main domain which do not cover real data, i.e., points x_t^θ s.t. $p^\theta(x_t^\theta) \gg 0, p^*(x_t^\theta) \to 0$. For out-of-domain points $p_t^\theta(x_t^\theta) \to 0$, the generator cannot receive feedback, because distance (15) for x_t^θ also vanishes. Moreover, if teacher function is inaccurate, the generator will learn it with all inaccuracies.

3.2 Relation to prior distillation works

FGM and SiD approaches formulate distillation as a constraint minimization of generator loss subject to the optimal fake model. For generator updates, the explicit UID loss (12) matches SiD loss (4) with $\alpha_{\text{SiD}} = 0.5$ and FGM loss (6) up to weighting. For a fake model, it also minimizes the UM loss on the generated data. The work (Gushchin et al., 2025) was the first to formulate the distillation of Bridge Matching models in their IBMD framework as a min-max optimization of the single loss (13).

Although previous works derive the same losses, we give a new, simple explanation using a linearization technique. *This technique is more powerful and general for handling intractable math expectations than complex proofs for concrete models from FGM, SiD, IBMD.* Furthermore, it allows adding real data directly into the distillation loss (see §3.4 and Appendix A.2) and extending it, e.g., deriving a loss for minimizing the ℓ_2-distance instead of the squared one (Appendix A.4).

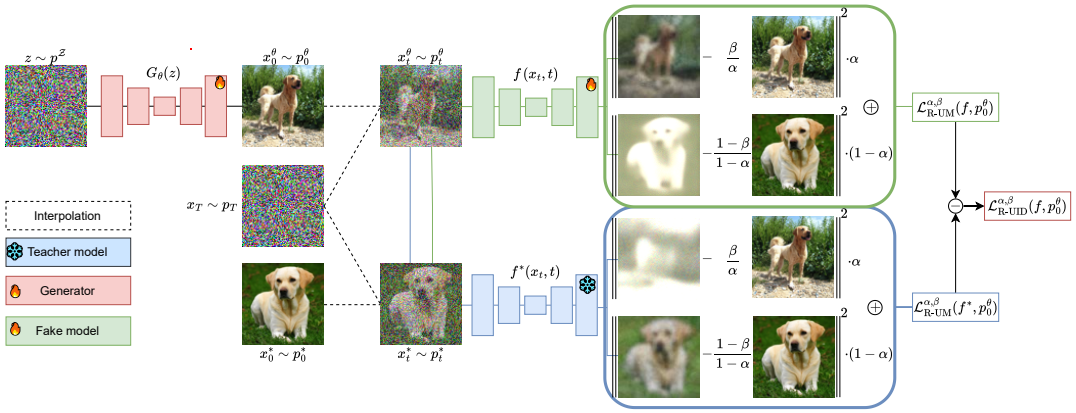

Figure 1: Pipeline of **our RealUID distillation framework** (§3) with the direct incorporation of real data p_0^* adjusted by hyperparameters $\alpha, \beta \in (0, 1]$. In the figure, it is depicted for Flow Matching models predicting denoised samples. It distills a costly frozen teacher model f^* (blue) into a one-step generator G_θ (red) upon min-max optimization of $\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(f, p_0^\theta)$ loss over fake model f (green) and generator distribution p_0^θ with parameters θ. We use alternating optimization, updating the fake model several times per one generator update for stability. Algorithm's pseudocode is located in Appendix B.

3.3 Connection with Inverse Optimization

We derived UID loss (14) by minimizing the squared ℓ_2-distance between teacher and student functions. However, this loss admits another interpretation: its structure is typical for inverse optimization (Chan et al., 2025). In this framework, one considers a parametric family of optimization problems $\min_f \mathcal{L}(f, \theta)$ with objective loss $\mathcal{L}(f, \theta)$ depending on argument f and parameters θ. The goal is to find the parameters θ^* that yield a known, desired solution $f^* = \arg\min_f \mathcal{L}(f, \theta^*)$. One standard way to recover the required parameters is to solve the same min-max problem as (14):

$$\min_\theta \max_f \{\mathcal{L}(f^*, \theta) - \mathcal{L}(f, \theta)\} \sim \min_\theta \big\{\mathcal{L}(f^*, \theta) - \min_f \{\mathcal{L}(f, \theta)\}\big\}. \tag{16}$$

The inverse problem (16) always has minimum 0 which is attained when $\theta = \theta^*$.

Although the inverse optimization can handle arbitrary losses $\mathcal{L}$, it does not describe the properties of the optimized functions or how to find solutions. In our case, we show that all losses are tractable and minimize the distances between teacher and student functions (Lemmas 1 and 2). Furthermore, in Appendix A, we provide and justify a list of extensions of our framework that cannot be stated as inverse problems. All our proofs are self-contained and do not rely on inverse optimization, which only provides intuition and understanding.

3.4 RealUID: natural approach for real data incorporation

Previous distillation methods add real data during training only via GANs with extra discriminator and adversarial loss. We propose a simpler, more natural way that requires no extra models or losses.

Based on intuition from inverse optimization (§3.3), we see that the min-max inverse problem (16) is compatible with other losses. This allows us to redesign the UM loss (2) to incorporate real data into it. A key constraint is that the loss must still yield the same teacher upon minimization on the real data. Thus, we derive a novel Unified Matching loss with real data - a weighted sum of two UM-like losses on generated and real data parameterized by $\alpha, \beta \in (0, 1]$ which control the weights.

Definition 2. *We define* ***Universal Matching loss with real data*** *on generated data $p_0^\theta \in \mathcal{P}(\mathbb{R}^D)$ with $\alpha, \beta \in (0, 1]$ (when $\alpha = 1$ the real data term becomes $2(1-\beta)\langle f_t(x_t^*), f_t^*(x_t^*|x_0^*)\rangle$):*

$$\begin{aligned}\mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f, p_0^\theta) \;=\; & \underbrace{\alpha \cdot \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0^\theta \sim p_0^\theta, x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta)}\left[\|f_t(x_t^\theta) - \frac{\beta}{\alpha} f^\theta(x_t^\theta|x_0^\theta)\|^2\right]}_{\text{generated data } p_0^\theta \text{ term}} \\ + \; & \underbrace{(1-\alpha) \cdot \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0^* \sim p_0^*, x_t^* \sim p_t^*(\cdot|x_0^*)}\left[\|f_t(x_t^*) - \frac{1-\beta}{1-\alpha} f_t^*(x_t^*|x_0^*)\|^2\right]}_{\text{real data } p_0^* \text{ term}}.\end{aligned} \tag{17}$$

RealUM loss (17) for all α, β and UM loss (2) yield the same teacher when input distribution is real data p_0^*, i.e., $\arg\min_f \mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f, p_0^*) = \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*) = f^*$. Hence, the min-max inverse scheme (16) with RealUM loss and the old teacher f^* will still have a real data generator as a solution:

$$\min_\theta \{\underbrace{\mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f^*, p_0^\theta) - \min_f \{\mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f, p_0^\theta)\}}_{\geq 0}\} = \mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f^*, p_0^*) - \underbrace{\min_f \{\mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f, p_0^*)\}}_{=\mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f^*, p_0^*)} = 0.$$

But now distillation loss will incorporate real data through the real data terms of $\mathcal{L}^{\alpha,\beta}_{\text{R-UM}}(f, p_0^\theta)$.

Theorem 2 (Real data generator minimizes RealUID loss). *Let teacher $f^* := \arg\min_f \mathcal{L}_{UM}(f, p_0^*)$ be the minimizer of UM loss on real data p_0^*. Then real data generator G_{θ^*}, s.t. $p_0^{\theta^*} = p_0^*$, is a solution to the min-max optimization of **Universal Inverse Distillation loss with real data (RealUID)** $\mathcal{L}^{\alpha,\beta}_{R\text{-}UID}(f, p_0^\theta)$ over fake function f and generator distribution p_0^θ:*

$$\min_\theta \max_f \left\{\mathcal{L}^{\alpha,\beta}_{R\text{-}UID}(f, p_0^\theta) := \mathcal{L}^{\alpha,\beta}_{R\text{-}UM}(f^*, p_0^\theta) - \mathcal{L}^{\alpha,\beta}_{R\text{-}UM}(f, p_0^\theta)\right\}. \tag{18}$$

We provide analysis of RealUID in Appendix A.1, below we highlight the most important findings.

Role of coefficients α, β. The RealUID framework uses real data samples only to minimize RealUM loss for the fake model. As shown in Lemma 2, RealUID also implicitly minimizes the rescaled distance (20) between the teacher and generator functions. This distance is still minimal when $p_0^\theta = p_0^*$, alternatively proving Theorem 2. The proof of Lemma 2 is located in Appendix A.1.

Lemma 2 (Distance minimized by RealUID loss). *Maximization of RealUID loss $\mathcal{L}^{\alpha,\beta}_{R\text{-}UID}$ (17) over fake function f returns the weighted sum between the teacher f^* and student function $f^\theta := \arg\min_f \mathcal{L}_{UM}(f, p_0^\theta)$ and represents the weighted squared ℓ_2-distance between them:*

$$\left[\arg\max_f \mathcal{L}^{\alpha,\beta}_{R\text{-}UID}(f, p_0^\theta)\right](t, x_t) = \frac{(1-\beta)p_t^*(x_t)\cdot f_t^*(x_t) + \beta p_t^\theta(x_t)\cdot f_t^\theta(x_t)}{(1-\alpha)p_t^*(x_t) + \alpha p_t^\theta(x_t)}, \tag{19}$$

$$\max_f \mathcal{L}^{\alpha,\beta}_{R\text{-}UID}(f, p_0^\theta) = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^*\sim p_t^*}\left[\frac{\|\frac{\beta}{\alpha}\cdot[p_t^*(x_t^*)f_t^*(x_t^*) - p_t^\theta(x_t^*)f_t^\theta(x_t^*)] + (p_t^\theta(x_t^*) - p_t^*(x_t^*))\cdot f_t^*(x_t^*)\|^2}{p_t^*(x_t^*)((1-\alpha)p_t^*(x_t^*) + \alpha p_t^\theta(x_t^*))/\alpha^2}\right]. \tag{20}$$

With the help of real data, the distance (20) captures mismatches for both incorrectly generated points from the generator's main domain and the real data points, which the generator fails to cover. Thus, unlike data-free UID loss (Lemma 1), RealUID loss provides the generator with feedback also on the real data domain it needs to cover (see Appendix A.1.2). Moreover, if teacher function is inaccurate, RealUID can now provably fix teacher's errors (see Appendix A.1.3).

Choice of coefficients α, β. Lemma 2 shows that, instead of values α and β, actually the values α and β/α determine the balance between real and generated data in the minimized distance (20). Furthermore, coefficient α only sets the general scaling of the distance, while β/α *plays the most important role*, as it determines the relation between f_t^θ and f_t^* inside the distance.

Value $\beta/\alpha = 1$ yields the distance identical to the data-free distance (15) up to scaling. Even when $\alpha = \beta < 1$ and real data is formally added, it has no, or negative, effect on the generator. Excessively low α and β diminish the effect of the generated data term f_t^θ in the optimal fake (19), leading to vanishing gradients. The same issue occurs with $\beta/\alpha \ll 1$ in (20), while $\beta/\alpha \gg 1$ diminish the effect of the right real data term f_t^*. Plus, configurations $\beta < \alpha = 1$ may be unstable due to out-of-domain samples. See Appendix A.1.2 for more details of the distance analysis. Moreover, if teacher function is inaccurate, only the choice $\beta/\alpha \neq 1$ can fix teacher's errors (see Appendix A.1.3).

Hence, good coefficients $\alpha, \beta \in (0, 1]$ can be chosen by first finding good $\beta/\alpha \neq 1$, as it has the largest impact, and then adjusting $\alpha < 1$. Both β/α and α should be close to 1.

Comparison with GAN-based methods. Unlike SiD and FGM with GANs, we do not use extra adversarial losses and discriminator to incorporate real data. We only modify UM loss, preserving its core structure and fake model architecture. While general adversarial loss is unrelated to the main distillation loss and has uninterpretable scaling hyperparameters, our RealUID loss and weighting

Generation	$\alpha \diagdown \beta$	0.94	0.96	0.98	1.0
Unconditional	0.94	2.66	**2.28**	2.58	2.98
	0.96	2.37	2.58	2.29	2.65
	0.98	2.97	2.33	2.62	2.38
	1.0	5.81	4.51	3.29	2.58
Conditional	0.94	2.35	2.19	2.25	2.47
	0.96	2.09	2.32	2.13	2.27
	0.98	2.34	**2.02**	2.26	2.05
	1.0	4.32	3.27	2.43	2.21

Generation	$\lambda_{\text{adv}}^{G_\theta}$	λ_{adv}^{D}	FID (↓)
Unconditional	0.1	0.3	2.42
	0.3	1	**2.29**
	1	3	2.39
	5	15	2.54
Conditional	0.1	0.3	2.22
	0.3	1	**2.12**
	1	3	2.15
	5	15	2.40

Table 1: Ablation studies of (α, β) coefficients in the left table and adversarial weighting parameters $(\lambda_{\text{adv}}^{G_\theta}, \lambda_{\text{adv}}^{D})$ in the right table for CIFAR-10 in both unconditional and conditional settings. The baseline RealUID ($\alpha = 1.0, \beta = 1.0$) does not use real data. Configurations that outperform the baseline are highlighted. All values report FID ↓, where lower is better. The best configuration in each case is **bolded**.

coefficients $\alpha, \beta \in (0, 1]$ come naturally from the data-free UID loss. The original UID loss (14), equivalent to SiD (4) with $\alpha_{\text{SiD}} = 0.5$ and FGM (6), is obtained when $\alpha = \beta = 1$.

Extension for Bridge Matching and Stochastic Interpolants framework. In Appendix C, we demonstrate that our framework can be easily extended to other matching models by parametrizing the generated data coupling $\pi^\theta(x_0, x_T)$ instead of the data distribution p_0^θ.

4 Experiments

All implementations were developed in PyTorch, and the code will be made publicly available.

This section provides an ablation study and evaluation of our RealUID, assessing both its performance and computational efficiency. We begin in (§4.1) by detailing the experimental setup. In (§4.2), we show that our incorporation of real data via coefficients α, β improves performance, speeds up convergence, and enables effective fine-tuning. In (§4.3), we assess the benchmark performance and computational demands of RealUID relative to SOTA methods. Additional experimental details and results are provided in Appendix D and Appendix E, respectively.

4.1 Experimental Setup

Datasets and Evaluation Protocol. Due to computational resources constraints, the experiments were conducted only on the conditional/unconditional CIFAR-10 dataset with 32×32 resolution (Krizhevsky et al., 2009) and on the CelebA dataset with 64×64 resolution (Liu et al., 2015), see Appendix E.2. In line with the prior works (Karras et al., 2019; 2022), we report test FID scores (Heusel et al., 2017), computed using 50k generated samples.

Implementation Details. In contrast to prior studies (Zhou et al., 2024b;a; Huang et al., 2024), which employ the computationally demanding EDM architecture (Karras et al., 2022), our work adopts a more lightweight alternative (Tong et al., 2023) due to resource constraints (see (§4.3) for efficiency analysis). We also trained our own flow-matching model, denoted by f^*, which served as the teacher. Further implementation details are provided in Appendix D.

4.2 Benchmarking Methods under a Unified Experimental Configuration

We evaluate RealUID under a unified experimental protocol (fixed architecture and implementation). We begin by (i) conducting an ablation over α, β to assess the influence of real-data incorporation. We then (ii) compare RealUID to a GAN-based alternative, showing that RealUID achieves comparable or superior accuracy. Furthermore, (iii) we analyze convergence, indicating that RealUID variants with real data train substantially faster than baselines without real-data. Finally, (iv) we explore a fine-tuning stage initialized from strong RealUID checkpoints, showing further performance gains.

Ablation study of coefficients α, β. The search for optimal α and β parameters was restricted to values near 1, specifically $\alpha, \beta \in [0.9, 1.0]$ with increments of 0.02 to cover the full grid. Setting these parameters too low prevents the student from accurately capturing the true generator gradient, which

Table 2: This table presents the results of our ablation study on the RealUID framework, evaluated using the FID metric under both unconditional and conditional generation setups. The Teacher Flow model with 100 NFE is reported as a reference. The performance of the baseline RealUID ($\alpha = 1.0, \beta = 1.0$) without real-data incorporation is indicated in *italic*. For emphasis, we underline the two counterparts that incorporate real data: the GAN-based and our RealUID methods. The best-performing configurations, obtained via an additional fine-tuning stage with adjusted $(\alpha_{\text{FT}}, \beta_{\text{FT}})$, are highlighted in **bold**. Qualitative results are presented in § E.3.

Model	FID (↓)
Teacher Flow (NFE=100)	3.57
RealUID ($\alpha = 1.0, \beta = 1.0$)	*2.58*
RealUID ($\alpha = 1.0, \beta = 1.0$) + GAN ($\lambda_{\text{adv}}^{G_\theta} = 0.3, \lambda_{\text{adv}}^{D} = 1$)	2.29
RealUID ($\alpha = 0.94, \beta = 0.96$)	2.28
RealUID ($\alpha = 0.94, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$)	**2.03**

Model	FID (↓)
Teacher Flow (NFE=100)	5.56
RealUID ($\alpha = 1.0, \beta = 1.0$)	*2.21*
RealUID ($\alpha = 1.0, \beta = 1.0$) + GAN ($\lambda_{\text{adv}}^{G_\theta} = 0.3, \lambda_{\text{adv}}^{D} = 1$)	2.12
RealUID ($\alpha = 0.98, \beta = 0.96$)	2.02
RealUID ($\alpha = 0.98, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$)	**1.91**

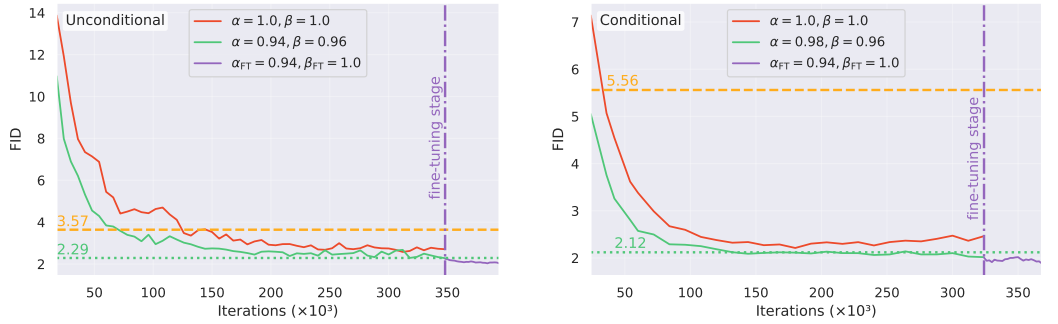

Figure 2: Evolution of FID during CIFAR-10 distillation for (i) the baseline RealUID ($\alpha = 1.0, \beta = 1.0$), (ii) the best-performing RealUID configurations, and (iii) subsequent fine-tuning, evaluated in both unconditional and conditional settings. The performances of Teacher Flow and UID+GAN are indicated by horizontal reference lines in their respective colors. Methods that incorporate real data—best-performing RealUID and UID+GAN—are highlighted in green to facilitate comparison.

in turn leads the generator to produce noisy samples. The results are reported in Table 1. As a baseline, we highlight the model without data incorporation our RealUID ($\alpha = 1.0, \beta = 1.0$). As shown in the table, using real data with $\alpha = \beta < 1.0$ or with $\alpha = 1.0, \beta < 1.0$ or with substantially different α and β consistently degraded performance. In contrast, parameter settings close to the diagonal $\alpha/\beta = 1.02$ or $\alpha/\beta = 0.98$ produced improved results, with the best performance achieved by our RealUID ($\alpha = 0.94, \beta = 0.96$) for the unconditional case and our RealUID ($\alpha = 0.98, \beta = 0.96$) for the conditional case. Note that the practical results for various α, β match the theoretical description from (§3.4.)

Comparison with GAN-based method. We integrated the GAN-based approach proposed by Zhou et al. (2024a) into our experimental framework as an alternative method for incorporating real data, enabling a direct comparison with our RealUID formulation. Specifically, we combined the GAN loss with the baseline RealUID ($\alpha = 1.0, \beta = 1.0$). As shown in Table 1, the best-performing configurations are achieved with GAN losses ($\lambda_{\text{adv}}^{G_\theta} = 0.3$, $\lambda_{\text{adv}}^{D} = 1$). While this setup performs comparably to RealUID ($\alpha = 0.94, \beta = 0.96$) in the unconditional setting, it remains clearly inferior to RealUID ($\alpha = 0.98, \beta = 0.96$) in the conditional case.

Convergence Speed. Our RealUID (α, β) with parameters, which are highlighted in Table 1, achieves faster convergence than the baseline RealUID ($\alpha = 1.0, \beta = 1.0$). For clarity, we present qualitative comparisons of the best-performing configurations against their baselines in Figure 2. As shown in figure, the best RealUID configurations reach the saturated performance level of the baseline after ∼100k iterations, whereas the baseline requires ∼300k iterations to achieve comparable metrics. These results demonstrate that incorporating real data substantially accelerates convergence.

Fine-tuning stage. We observe that the RealUID framework offers substantial flexibility for fine-tuning. In this procedure, the generator G_θ is initialized from the best-performing RealUID checkpoint obtained during training from scratch, while the fake model f is initialized from the teacher model f^*. Fine-tuning then proceeds with new hyperparameter values α_{FT} and β_{FT}, allowing for refined control over the degree of real-data incorporation during this stage. We find that the configurations RealUID ($\alpha = 0.94, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$) and RealUID ($\alpha = 0.98, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$) produced the best results in the unconditional and conditional cases, respectively, as shown in Tables 2. Ablation studies analyzing the effect of α_{FT} and β_{FT} are provided in Appendix E.1.

Table 3: Comparison of *unconditional* generation on CIFAR-10. The best method under the FID metric in each section is highlighted with **bold**.

Family	Model	NFE	FID (↓)
Diffusion & GAN	DDPM (Ho et al., 2020)	1000	3.17
	VP-EDM (Karras et al., 2022)	35	1.97
	StyleGAN2+ADA+Tune (Karras et al., 2020)	1	2.92
	StyleGAN2+ADA+Tune+DI (Luo et al., 2023)	1	2.71
	Diffusion ProjectedGAN (Wang et al., 2022)	1	2.54
	iCT-deep (Song & Dhariwal, 2023)	1	2.51
	Diff-Instruct (Luo et al., 2023)	1	4.53
	DMD (Yin et al., 2024b)	1	3.77
	CTM (Kim et al., 2023)	1	1.98
	sCD (Lu & Song, 2024)	1	3.66
	sCT (Lu & Song, 2024)	1	2.85
	SiD, $\alpha_{\text{SiD}} = 1.0$ (Zhou et al., 2024b)	1	2.03
	SiD, $\alpha_{\text{SiD}} = 1.2$ (Zhou et al., 2024b)	1	1.92
	SiDA, $\alpha_{\text{SiD}} = 1.0$ (Zhou et al., 2024a)	1	1.52
	SiD^{2}A, $\alpha_{\text{SiD}} = 1.2$ (Zhou et al., 2024a)	1	1.52
	SiD^{2}A, $\alpha_{\text{SiD}} = 1.0$ (Zhou et al., 2024a)	1	**1.50**
Flow-based	CFM (Yang et al., 2024)	2	5.34
	IMM (Zhou et al., 2025)	1	3.20
	MeanFlow (Geng et al., 2025)	1	2.92
	FACM (Peng et al., 2025)	1	2.69
	1-ReFlow (+Distill) (Liu et al., 2022a)	1	6.18
	2-ReFlow (+Distill) (Liu et al., 2022a)	1	4.85
	3-ReFlow (+Distill) (Liu et al., 2022a)	1	5.21
	FGM (Huang et al., 2024)	1	3.08
	RealUID ($\alpha = 0.94, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$) (**Ours**)	1	**2.03**

Table 4: Comparison of *conditional* generation on CIFAR-10. The best method under the FID metric in each section is highlighted with **bold**.

Family	Model	NFE	FID (↓)
Diffusion & GAN	VP-EDM (Karras et al., 2022)	35	1.79
	GET-Base (Geng et al., 2023)	1	6.25
	BigGAN (Brock et al., 2018)	1	14.73
	BigGAN+Tune (Brock et al., 2018)	1	8.47
	StyleGAN2+ADA (Karras et al., 2020)	1	3.49
	StyleGAN2+ADA+Tune (Karras et al., 2020)	1	2.42
	StyleGAN2+ADA+Tune+DI (Luo et al., 2023)	1	2.27
	StyleGAN-XL (Sauer et al., 2022)	1	1.85
	StyleSAN-XL (Takida et al., 2023)	1	**1.36**
	Diff-Instruct (Luo et al., 2023)	1	4.19
	DMD (Yin et al., 2024b)	1	2.66
	DMD (*w.o.* KL) (Yin et al., 2024b)	1	3.82
	DMD (*w.o. reg.*) (Yin et al., 2024b)	1	5.58
	GDD-I (Zheng et al., 2024)	1	1.44
	CTM (Kim et al., 2023)	1	1.73
	SiD, $\alpha_{\text{SiD}} = 1.0$ (Zhou et al., 2024b)	1	1.93
	SiD , $\alpha_{\text{SiD}} = 1.2$ (Zhou et al., 2024b)	1	1.71
	SiDA, $\alpha_{\text{SiD}} = 1.0$ (Zhou et al., 2024b)	1	1.44
	SiD^{2}A, $\alpha_{\text{SiD}} = 1.0$ (Zhou et al., 2024a)	1	1.40
	SiD^{2}A, $\alpha_{\text{SiD}} = 1.2$ (Zhou et al., 2024a)	1	1.39
Flow-based	FGM (Huang et al., 2024)	1	2.58
	RealUID ($\alpha = 0.98, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$) (**Ours**)	1	**1.91**

Methods	Inference Time (ms)	# Total Param (M)	Max GPU Mem Alloc (MB)	Max GPU Mem Reserved (MB)
RealUID (**Ours**)	**18.636**	**36.784**	**165**	**172**
FGM (Huang et al., 2024) / SiD (Zhou et al., 2024b;a)	30.745	55.734	242	276

Table 5: Inference complexity on an Ascend 910B3 (65 GB) NPU. For each method, we report (i) the mean inference time per image (bs=1, fp32), averaged over 10,000 iterations; (ii) the total number of parameters (Millions); and (iii) peak NPU memory usage (maximum allocated and reserved, in MB). Best values are **bolded**.

Scaling to larger datasets. In Appendix E.2, we provide the similar results of the same ablation studies on the CelebA dataset with 64×64 resolution.

4.3 Benchmark performance and Computational comparisons

As shown in Tables 3 and 4, RealUID consistently outperforms all prior flow-based models on CIFAR-10, significantly surpassing the strongest flow distillation baseline, FGM. Despite its compact architecture (§4.1), it achieves performance comparable to leading diffusion distillation methods-matching SiD (α_{SiD}=1.0) and closely approaching SiD (α_{SiD}=1.2), while falling short of adversarially enhanced models such as SiD^{2}A. Based on ablation studies and comparisons with GANs (§4.2), we hypothesis that this performance gap is attributed to architectural and teacher capacity differences rather than the lack of adversarial loss. In terms of efficiency, RealUID leverages a lightweight architecture based on Tong et al. (2023). Therefore, as summarized in Table 5, it achieves nearly $2\times$ faster inference, lower memory usage, and reduced model size compared to recent distillation approaches (Zhou et al., 2024b;a; Huang et al., 2024). The results indicate that our approach achieves competitive performance while maintaining a lower computational footprint.

5 Discussion, Extension, Future Works

Extensions. Our RealUID (§3.4) framework can distill Flow/Bridge Matching, Diffusion models, and Stochastic Interpolants enhanced by a novel natural way to incorporate real data. In Appendix A, we provide three extensions of our RealUID beyond the inverse scheme: General RealUID with 3 coefficients (Appendix A.2), SiD framework with real data for $\alpha_{\text{SiD}} \neq \frac{1}{2}$ (Appendix A.3) and Normalized RealUID for minimizing non-squared ℓ_2-distance between teacher and student (Appendix A.4).

Relation to DMD. Instead of minimizing the squared ℓ_2-distance between the score functions, *Distribution Matching Distillation* (Luo et al., 2023; Wang et al., 2023; Yin et al., 2024b;a) (DMD) approach minimizes the KL divergence between the real and generated data. Its gradients are computed using the generator and teacher score functions, leading to the similar alternating updates. *We would like to highlight that DMD does not fit UID framework.* Nevertheless, we investigated an opportunity to incorporate real data into DMD without GANs in Appendix A.5.

REFERENCES

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

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. *Operations Research*, 73(2):1046–1074, 2025.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025.

Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. *Advances in Neural Information Processing Systems*, 36:41914–41931, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry P Vetrov, and Evgeny Burnaev. Entropic neural optimal transport via diffusion processes. *Advances in Neural Information Processing Systems*, 36, 2024.

Nikita Gushchin, David Li, Daniil Selikhanovych, Evgeny Burnaev, Dmitry Baranchuk, and Alexander Korotin. Inverse bridge matching distillation. 2025.

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

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

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. *arXiv preprint arXiv:2410.20587*, 2024.

Zemin Huang, Zhengyang Geng, Weijian Luo, and Guo-jun Qi. Flow generator matching. *arXiv preprint arXiv:2410.19310*, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.

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

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. *arXiv preprint arXiv:2310.02279*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022a.

Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*, 2022b.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

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. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023.

Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. Flow-anchored consistency models. *arXiv preprint arXiv:2507.03738*, 2025.

A Sauer, K Schwarz, and A StyleGAN-XL Geiger. scaling stylegan to large diverse datasets. In *Proceedings of the SIGGRAPH Conference. ACM*, pp. 1–10, 2022.

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

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yuhta Takida, Masaaki Imaizumi, Takashi Shibuya, Chieh-Hsin Lai, Toshimitsu Uesaka, Naoki Murata, and Yuki Mitsufuji. San: Inducing metrizability of gan with discriminative normalized linear layer. *arXiv preprint arXiv:2301.12811*, 2023.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

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. *Advances in neural information processing systems*, 36:8406–8441, 2023.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

Mingyuan Zhou, Huangjie Zheng, Yi Gu, Zhendong Wang, and Hai Huang. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. *arXiv preprint arXiv:2410.14919*, 2024a.

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 *Forty-first International Conference on Machine Learning*, 2024b.

CONTENTS

A Theoretical proofs and extensions

In this appendix, we discuss our RealUID framework (Appendix A.1) in theoretical details and provide three extensions of it: *General RealUID* framework with 3 degrees of freedom (Appendix A.2), *SiD framework with real data* (Appendix A.3) and *Normalized RealUID* framework for minimizing ℓ_2-distance between teacher and student functions instead of the squared one (Appendix A.4). All proofs are based on the linearization technique and splitting terms in linearized decomposition between real and generated data.

We also propose an *approach to incorporate real data into DMD* framework, which is unsuitable for our RealUID Appendix A.5.

A.1 RealUID theoretical properties

In this section, we discuss our RealUID loss in detail. We begin by presenting its explicit form and how it connects linearization technique and real data incorporation. We then demonstrate that the loss minimizes a squared ℓ_2-distance between the rescaled teacher and student functions (Appendix A.1.1). Finally, we provide the motivation of the best choice of coefficients $\alpha \neq \beta$ from the perspectives of the better distance (Appendix A.1.2) and the correction of the teacher's errors (Appendix A.1.3).

A.1.1 Proof of RealUID Distance Lemma 2

Putting explicit values for RealUM loss (17) in RealUID loss (18) and denoting $\delta_t = f_t^* - f_t$, we get:

$$\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\delta, p_0^\theta) = \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta\sim p_0^\theta, \\ x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}}[-\alpha\|\delta_t(x_t^\theta)\|^2 + 2\alpha\langle\delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\beta\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle]$$

$$+\mathbb{E}_{\substack{t\sim[0,T], x_0^*\sim p_0^*, \\ x_t^*\sim p_t^*(\cdot|x_t^*)}}[-(1-\alpha)\|\delta_t(x_t^*)\|^2 + 2(1-\alpha)\langle\delta_t(x_t^*), f_t^*(x_t^*)\rangle - 2(1-\beta)\langle\delta_t(x_t^*), f_t^*(x_t^*|x_0^*)\rangle].$$

This form provides an alternative definition of coefficients α and β: they define the proportion in which each summand in the data-free linearized representation (11) of the squared ℓ_2-distance is split between the real and generated data. The idea of splitting coefficients between two data types helps extend RealUID to extra coefficients (Appendix A.2), new distances (Appendix A.4) and SiD framework with $\alpha_{SiD} \neq \frac{1}{2}$ (Appendix A.3).

Proof of Lemma 2. First, we take math expectation over data points x_0^*. Since the expectation can be taken in a reverse order, i.e., $\mathbb{E}_{x_0^*\sim p_0^*, x_t^*\sim p_t^*(\cdot|x_0^*)} = \mathbb{E}_{x_t^*\sim p_t^*, x_0^*\sim p_0^*(\cdot|x_t^*)}$, we see that

$$\begin{aligned}\mathbb{E}_{x_0^*\sim p_0^*, x_t^*\sim p_t^*(\cdot|x_0^*)}[\langle\delta_t(x_t^*), f_t^*(x_t^*|x_0^*)\rangle] &= \mathbb{E}_{x_t^*\sim p_t^*}\langle\delta_t(x_t^*), \mathbb{E}_{x_0^*\sim p_0^*(\cdot|x_t^*)}[f_t^*(x_t^*|x_0^*)]\rangle \\ &= \mathbb{E}_{x_t^*\sim p_t^*}[\langle\delta_t(x_t^*), f_t^*(x_t^*)\rangle]. \end{aligned} \tag{21}$$

For the generated data term $\mathbb{E}_{x_0^\theta\sim p_0^\theta, x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}[\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle] = \mathbb{E}_{x_t^\theta\sim p_t^\theta}[\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta)\rangle]$, the reasoning is similar. Thus, we can write down RealUID loss in an explicit form with $\delta_t = f_t^* - f_t$

$$\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\delta, p_0^\theta) = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}[-\alpha\|\delta_t(x_t^\theta)\|^2 + 2\alpha\langle\delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\beta\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta)\rangle]$$

$$+\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^*\sim p_t^*}[-(1-\alpha)\|\delta_t(x_t^*)\|^2 + 2(1-\alpha)\langle\delta_t(x_t^*), f_t^*(x_t^*)\rangle - 2(1-\beta)\langle\delta_t(x_t^*), f_t^*(x_t^*)\rangle]. \tag{22}$$

Then, we rescale the generated data terms in RealUID loss (22) using the equality $p_t^\theta(x_t) = \frac{p_t^\theta(x_t)}{p_t^*(x_t)} p_t^*(x_t)$ for $x_t \in \mathbb{R}^D$ (we assume $p_t^*(x_t) > 0, \forall x_t, t$) leaving only math expectation w.r.t. the real data, i.e,

$$\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\delta, p^\theta_0) = \mathbb{E}_{\substack{t\sim[0,T]\\ x^*_t\sim p^*_t}}\left[-[(1-\alpha)+\alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]\|\delta_t(x^*_t)\|^2\right]$$
$$-\mathbb{E}_{\substack{t\sim[0,T]\\ x^*_t\sim p^*_t}}\left[2\beta\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}\langle\delta_t(x^*_t), f^\theta_t(x^*_t)\rangle + 2[(\beta-\alpha)+\alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]\langle\delta_t(x^*_t), f^*_t(x^*_t)\rangle\right].$$

Finally, we maximize the loss w.r.t. $\delta_t(x^*_t)$ for each x^*_t and t as a quadratic function. The maximum is achieved when

$$\delta_t(x^*_t) = \frac{[(\beta-\alpha)+\alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]f^*_t(x^*_t) - \beta\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}f^\theta_t(x^*_t)}{[(1-\alpha)+\alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]}$$

or in terms of the fake model $f = f^* - \delta$

$$\left(\arg\max_f \mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(f, p^\theta_0)\right)(t, x_t) = \frac{f^*_t(x_t)\cdot(1-\beta) + f^\theta_t(x_t)\cdot\beta\frac{p^\theta_t(x_t)}{p^*_t(x_t)}}{(1-\alpha)+\alpha\frac{p^\theta_t(x_t)}{p^*_t(x_t)}}. \tag{23}$$

The maximum itself equals to

$$\max_f \mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(f, p^\theta_0) = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x^*_t\sim p^*_t}\left[\frac{\|f^*_t(x^*_t)\cdot((\beta-\alpha)+\alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}) - f^\theta_t(x^*_t)\cdot\beta\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}\|^2}{(1-\alpha)+\alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}}\right].$$

It is easy to see that when $p^\theta_0 = p^*_0$ and $f^\theta = f^*$ this distance achieves its minimal value 0. Moreover, optimal fake model in this case matches the teacher f^*, i.e.,

$$\left(\arg\max_f \mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(f, p^*_0)\right)(t, x_t) = \frac{f^*_t(x_t)\cdot(1-\beta) + f^*_t(x_t)\cdot\beta\frac{p^*_t(x_t)}{p^*_t(x_t)}}{(1-\alpha)+\alpha\frac{p^*_t(x_t)}{p^*_t(x_t)}} = f^*_t(x_t).$$

□

A.1.2 Explanation of the choice of coefficients α and β

Here we show that the best way to incorporate real data during generator training is to set $\beta/\alpha \neq 1$.

Following Lemma 2, we know exactly what distance our RealUID loss implicitly minimizes. Below we examine it for various $\alpha, \beta \in (0, 1]$:

$$\max_f \mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(f, p^\theta_0) = \int_{x_t} l_t(x_t, \beta, \alpha)dx_t,$$
$$l_t(x_t, \beta, \alpha) := \frac{\alpha^2\|(p^*_t(x_t)(\frac{\beta}{\alpha}-1) + p^\theta_t(x_t))\cdot f^*_t(x_t) - \frac{\beta}{\alpha}\cdot p^\theta_t(x_t)\cdot f^\theta_t(x_t)\|^2}{(1-\alpha)p^*_t(x_t) + \alpha p^\theta_t(x_t)},$$

where $l_t(x_t, \beta, \alpha)$ denotes the distance for the particular point x_t.

The total distance mostly sums up from the two groups of points: incorrectly generated points from the generator's main domain, i.e., $p^\theta_t(x_t) \gg 0, p^*(x_t) \to 0$, and real data points which are not covered by the generator, i.e., $p^\theta_t(x_t) \to 0, p^*(x_t) \gg 0$. For the points out of both domains $p^\theta_t(x_t) \to 0, p^*_t(x_t) \to 0$, the distance tends to 0, as well as for matching points $p^\theta_t(x_t) \approx p^*_t(x_t)$.

Choice of coefficients α, β. Next, we consider various coefficients $\alpha, \beta \in (0, 1]$ and how they affect two main groups of points.

- All configurations affect the incorrectly generated points $x_t : p^*_t(x_t) \to 0, p^\theta(x_t) \gg 0$:

$$l_t(x_t, \beta, \alpha) \approx \frac{\|\alpha p^\theta_t(x_t)\cdot f^*_t(x_t) - \beta p^\theta_t(x_t)\cdot f^\theta_t(x_t)\|^2}{\alpha p^\theta_t(x_t)} \approx \frac{\beta^2\|f^\theta_t(x_t)\|^2}{\alpha}p^\theta_t(x_t) \gg 0. \tag{24}$$

  Note that increasing $\beta/\alpha > 1$ will diminish the weight of the distance in comparison with $\alpha = \beta = 1$, while decreasing otherwise will lift the weight up.

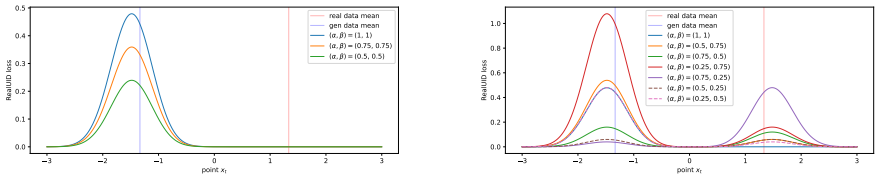

Figure 3: RealUID loss for $1D$-Gaussians under various coefficients (α, β).

- Configuration $\beta < \alpha = 1$ is unstable for uncovered real data points $x_t : p_t^\theta(x_t) \to 0, p^*(x_t) \gg 0$:

$$l_t(x_t, \beta, \alpha) \approx \frac{\|p_t^*(x_t)(\beta - 1) \cdot f_t^*(x_t) - \beta p_t^\theta(x_t) \cdot f_t^\theta(x_t)\|^2}{p_t^\theta(x_t)} \to \infty.$$

- Configuration $\beta = \alpha = 1$ (UID loss) does not affect uncovered real data points $x_t : p_t^\theta(x_t) \to 0, p^*(x_t) \gg 0$:

$$l_t(x_t, \beta, \alpha) \approx \frac{\|p_t^\theta(x_t) \cdot f_t^*(x_t) - p_t^\theta(x_t) \cdot f_t^\theta(x_t)\|^2}{p_t^\theta(x_t)} = \|f_t^*(x_t) - f_t^\theta(x_t)\|^2 p_t^\theta(x_t) \to 0.$$

- Configuration $\beta = \alpha < 1$ does not affect uncovered real data points $x_t : p_t^\theta(x_t) \to 0, p^*(x_t) \gg 0$:

$$l_t(x_t, \beta, \alpha) \approx \frac{\|\alpha p_t^\theta(x_t) f_t^*(x_t) - \beta p_t^\theta(x_t) f_t^\theta(x_t)\|^2}{(1 - \alpha) p_t^*(x_t)} = \frac{\|\alpha f_t^*(x_t) - \beta f_t^\theta(x_t)\|^2}{(1 - \alpha)} \frac{(p_t^\theta(x_t))^2}{p_t^*(x_t)} \to 0.$$

  Notably, in this configuration, the distance drops even faster than when $\alpha = \beta = 1$, what makes it even less preferable.

- *Only configuration* $\beta/\alpha \neq 1$ *affects the uncovered real data points* $x_t : p_t^\theta(x_t) \to 0, p^*(x_t) \gg 0$:

$$l_t(x_t, \beta, \alpha) \approx \frac{\|p_t^*(x_t)(\beta - \alpha) \cdot f_t^*(x_t) - \beta p_t^\theta(x_t) \cdot f_t^\theta(x_t)\|^2}{(1 - \alpha) p_t^*(x_t)} \gg 0.$$

Visual illustration. We analytically calculate the loss surface $l_t(x_t, \alpha, \beta)$ between the FM models transforming one-dimensional real data Gaussian $\mathcal{N}(\mu^*, 1)$ and generated Gaussian $\mathcal{N}(\mu^\theta, 1)$ to noise $\mathcal{N}(0, 1)$ on the time interval $[0, 1]$. In this case, the generated and real data interpolations are $p_t^\theta(x_t) = \mathcal{N}(x_t | \mu^\theta(1 - t), t^2 + (1 - t)^2)$ and $p_t^*(x_t) = \mathcal{N}(x_t | \mu^*(1 - t), t^2 + (1 - t)^2)$. The unconditional vector field $u = f$ between $\mathcal{N}(0, 1)$ and $\mathcal{N}(\mu, 1)$ can be calculated as

$$u_t(x_t) = \mathbb{E}_{x_0 \sim p_0(\cdot|x_t)} \left[\frac{x_t - x_0}{t} \right] = \int_{x_0} \left(\frac{x_t - x_0}{t} \right) \cdot \mathcal{N} \left(\frac{x_t - x_0(1 - t)}{t} | 0, 1 \right) \cdot \mathcal{N}(x_0 | \mu, 1) dx_0$$
$$= \frac{a(2t^2 - 2t) - bt^2}{\sqrt{2\pi}(1 - 2t + 2t^2)^{\frac{3}{2}}} \exp \left(-\frac{(x_t - \mu(1 - t))^2}{2(1 - 2t + 2t^2)^2} \right). \tag{25}$$

In Figure 3, we depict the loss surfaces for the fixed time $t = 1/3$, real data $\mu^* = 2$, generated data $\mu^\theta = -2$ and various pairs of (α, β). We can see that configurations $\beta/\alpha = 1$ do not detect the real data sample, even when $\alpha = \beta < 1$ and real data is formally used. while $\beta/\alpha \neq 1$ actually spots both domains, increasing the weight of generator domain when $\beta/\alpha > 1$ and decreasing it otherwise.

A.1.3 Correction of teacher's errors

In this chapter, we assume that instead of accurate teacher $f^* = \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*)$ we have access only to the arbitrary corrupted teacher $\tilde{f}^*$. *We will show that adding real data via our approach with* $\alpha \neq \beta$ *provably mitigates the teacher's errors in the final generator.*

Minimized distance. With the corrupted teacher $\tilde{f}^*$ and $\tilde{\delta} = \tilde{f}^* - f$, our corrupted Real-UID loss (see Appendix A.1.1) has the explicit form

$$\mathcal{L}_{\text{R-UID}}^{\alpha,\beta}(\tilde{\delta}, p_0^\theta) = \mathbb{E}_{\substack{t \sim [0,T], x_0^\theta \sim p_0^\theta, \\ x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta)}} [-\alpha \|\tilde{\delta}_t(x_t^\theta)\|^2 + 2\alpha \langle \tilde{\delta}_t(x_t^\theta), \tilde{f}_t^*(x_t^\theta) \rangle - 2\beta \langle \tilde{\delta}_t(x_t^\theta), f_t^\theta(x_t^\theta | x_0^\theta) \rangle]$$

$$+\mathbb{E}_{\substack{t\sim[0,T],x_0^*\sim p_0^*,\\ x_t^*\sim p_t^*(\cdot|x_t^*)}}[-(1-\alpha)\|\tilde{\delta}_t(x_t^*)\|^2+2(1-\alpha)\langle\tilde{\delta}_t(x_t^*),\tilde{f}_t^*(x_t^*)\rangle-2(1-\beta)\langle\tilde{\delta}_t(x_t^*),f_t^*(x_t^*|x_0^*)\rangle].$$

Note that sampled terms $f_t^*(x_t^*|x_0^*)$ and $f_t^\theta(x_t^\theta|x_0^\theta)$ are not affected by the corruption and give the accurate functions $f_t^*(x_t^*)=\mathbb{E}_{x_0^*\sim p_0^*(\cdot|x_t^*)}[f_t^*(x_t^*|x_0^*)]$ and $f_t^\theta(x_t^\theta)=\mathbb{E}_{x_0^\theta\sim p_0^\theta(\cdot|x_t^\theta)}[f_t^\theta(x_t^\theta|x_0^\theta)]$:

$$\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\tilde{\delta},p_0^\theta)=\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta\sim p_t^\theta}[-\alpha\|\tilde{\delta}_t(x_t^\theta)\|^2+2\alpha\langle\tilde{\delta}_t(x_t^\theta),\tilde{f}_t^*(x_t^\theta)\rangle-2\beta\langle\delta_t(x_t^\theta),f_t^\theta(x_t^\theta)\rangle]$$
$$+\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^*\sim p_t^*}[-(1-\alpha)\|\tilde{\delta}_t(x_t^*)\|^2+2(1-\alpha)\langle\tilde{\delta}_t(x_t^*),\tilde{f}_t^*(x_t^*)\rangle-2(1-\beta)\langle\tilde{\delta}_t(x_t^*),f_t^*(x_t^*)\rangle].$$

Then, we rescale the generated data terms using the equality $p_t^\theta(x_t)=\frac{p_t^\theta(x_t)}{p_t^*(x_t)}p_t^*(x_t)$ for $x_t\in\mathbb{R}^D$ (we assume $p_t^*(x_t)>0,\forall x_t,t$) leaving only math expectation w.r.t. the real data, i.e,

$$\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\tilde{\delta},p_0^\theta)=\mathbb{E}_{t\sim[0,T],x_t^*\sim p_t^*}\left[-[(1-\alpha)+\alpha\frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)}](\|\tilde{\delta}_t(x_t^*)\|^2+2\langle\tilde{\delta}_t(x_t^*),\tilde{f}_t^*(x_t^*)\rangle)\right]$$
$$-\mathbb{E}_{\substack{t\sim[0,T],\\ x_t^*\sim p_t^*}}\left[2\langle\tilde{\delta}_t(x_t^*),(1-\beta)f_t^*(x_t^*)+\beta\frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)}f_t^\theta(x_t^*)\rangle\right].$$

Finally, we maximize the loss w.r.t. $\tilde{\delta}_t(x_t^*)$ for each x_t^* and t as a quadratic function $\max_{\tilde{\delta}}\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\tilde{\delta},p_0^\theta)=$

$$\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^*\sim p_t^*}\left[\frac{\|\tilde{f}_t^*(x_t^*)\cdot((1-\alpha)+\alpha\frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)})-(1-\beta)f_t^*(x_t^*)-\beta\frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)}f_t^\theta(x_t^*)\|^2}{(1-\alpha)+\alpha\frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)}}\right]. \quad (26)$$

Hence, max-min optimization of the corrupted RealUID loss implicitly minimizes expected distance (26). However, due to arbitrary function $\tilde{f}$, we now cannot guarantee that minimum is achived when the relation inside the norm equals 0. Previously, we could use the solution $p^\theta=p^*$ which obviously achieved a minimum of 0. Now, due to the implicit and complex relationship between f^θ and p^θ, we can neither find an explicit form for the optimal p^θ nor guarantee the minimum of 0.

Choice of coefficients α,β. Here we give an intuition on why coefficients $\beta/\alpha\neq1$ can fix the teacher's errors, while $\beta/\alpha=1$ cannot. For simplicity, we assume that the minimized distance (26) actually attains minimum of 0 when

$$((1-\alpha)p_t^*(x_t)+\alpha p_t^\theta(x_t))\cdot\tilde{f}_t^*(x_t)-(1-\beta)p_t^*(x_t)\cdot f_t^*(x_t)-\beta p_t^\theta(x_t)\cdot f_t^\theta(x_t^*)=0. \quad (27)$$

- In case of $\alpha=\beta=1$, we have $\tilde{f}_t^*=f_t^\theta$, i.e., the generator learns the corrupted function.
- In case of $\alpha=\beta<1$, we have

  $$\tilde{f}_t^*(x_t)=\frac{(1-\alpha)p_t^*(x_t)}{(1-\alpha)p_t^*(x_t)+\alpha p_t^\theta(x_t)}\cdot f_t^*(x_t)+\frac{\alpha p_t^\theta(x_t)}{(1-\alpha)p_t^*(x_t)+\alpha p_t^\theta(x_t)}\cdot f_t^\theta(x_t^*).$$

  In this convex combination, the corrupted function $\tilde{f}^*$ is always between the true teacher function f^* and the optimal generator function f^θ, i.e., the generator learns even worse function.
- *In case of $\beta/\alpha\neq1$, there exist intervals of α,β which can give better generator function than the corrupted teacher.* For example, coefficients $\alpha\neq\beta$ close to 1 allow to neglect the terms $(1-\alpha)p_t^*(x_t)\cdot\tilde{f}_t^*(x_t)$ and $(1-\beta)p_t^*(x_t)\cdot f_t^*(x_t)$ in (27) to get $f_t^\theta(x_t)\approx\frac{\alpha}{\beta}\tilde{f}_t^*(x_t)$. Hence, we can steer f^θ towards the true teacher picking $\beta/\alpha<1$ or $\beta/\alpha>1$ depending on the corrupted and clean teacher's values. However, we cannot find all these intervals analytically due to complex distributions and functions.

Note that we derive the same recommendation $\beta/\alpha\neq1$ from the perspective of correcting the teacher's errors and from the perspective of the minimized distance surface from Appendix A.1.2.

Visual illustration. For visual demonstration, we consider the FM models transforming one-dimensional real data Gaussian $\mathcal{N}(\mu^*,1)$ and generated Gaussian $\mathcal{N}(\mu^\theta,1)$ to noise $\mathcal{N}(0,1)$ on the time interval $[0,1]$. In this case, the generated and real data interpolations are $p_t^\theta(x_t)=\mathcal{N}(x_t|\mu^\theta(1-$

$t), t^2 + (1-t)^2)$ and $p_t^*(x_t) = \mathcal{N}(x_t|\mu^*(1-t), t^2 + (1-t)^2)$. The unconditional vector field $u = f$ between $\mathcal{N}(0,1)$ and $\mathcal{N}(\mu, 1)$ can be calculated as

$$u_t(x_t) = \mathbb{E}_{x_0 \sim p_0(\cdot|x_t)}\left[\frac{x_t - x_0}{t}\right] = \int_{x_0}\left(\frac{x_t - x_0}{t}\right) \cdot \mathcal{N}\left(\frac{x_t - x_0(1-t)}{t}|0,1\right) \cdot \mathcal{N}(x_0|\mu, 1)dx_0$$
$$= \frac{a(2t^2 - 2t) - bt^2}{\sqrt{2\pi}(1 - 2t + 2t^2)^{\frac{3}{2}}} \exp\left(-\frac{(x_t - \mu(1-t))^2}{2(1 - 2t + 2t^2)^2}\right). \tag{28}$$

In Figure 4, we depict the optimal generator mean μ^θ and vector field u^θ satisfying (27) for various deviations $\tilde{u}^* - u^*$ and fixed time $t = 1/3$, real data $\mu^* = -2$ and point $x_t = -1$.

We can see that with $\alpha = \beta = 1$, the generator learns the corrupted vector field, and with $\alpha = \beta < 1$, the learned field and means are often even worse. In contrast, with $\beta/\alpha \neq 1$, the generator can learn vector fields and means which are closer to the real data. Although the generator cannot satisfy relation (27) under large deviations, it still produces better results with the real data.

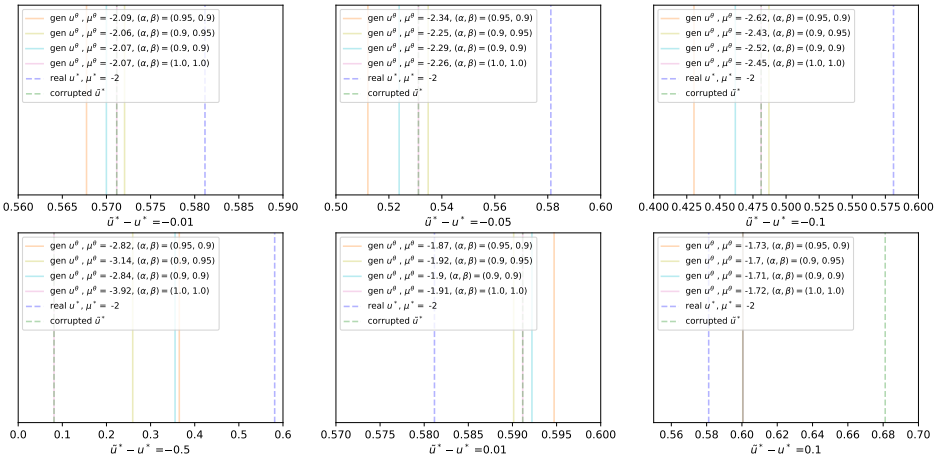

Figure 4: Learned generators for RealUID loss between 1D-Gaussians with corrupted teachers.

A.2 General RealUID loss

Expanding our real data incorporation. We recall that UID loss (Theorem 1) can be restated via linearization technique with $\delta = f^* - f$ as:

$$\mathcal{L}_{\text{UID}}(\delta, p_0^\theta) = \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta \sim p_0^\theta, \\ x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta)}}\left\{-\|\delta_t(x_t^\theta)\|^2 + 2\langle \delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\langle \delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle\right\}.$$

In turn, after real data incorporation, we obtain our RealUID loss (Theorem 2). Putting the explicit values for RealUM loss (17) in RealUID loss (18), we get the explicit formula:

$$\mathcal{L}_{\text{R-UID}}^{\alpha,\beta}(\delta, p_0^\theta) = \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta \sim p_0^\theta, \\ x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta)}}[-\alpha\|\delta_t(x_t^\theta)\|^2 + 2\alpha\langle \delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\beta\langle \delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle]$$
$$+ \mathbb{E}_{\substack{t\sim[0,T], x_0^* \sim p_0^*, \\ x_t^* \sim p_t^*(\cdot|x_0^*)}}[-(1-\alpha)\|\delta_t(x_t^*)\|^2 + 2(1-\alpha)\langle \delta_t(x_t^*), f_t^*(x_t^*)\rangle - 2(1-\beta)\langle \delta_t(x_t^*), f_t^*(x_t^*|x_0^*)\rangle].$$

These two formulas give us alternative explanation on how to add real data into arbitrary losses: we need to split each term in the linearized representation of the data-free loss between real and generated data. For example, in RealUID loss, its three terms are split with proportions α, α, β, respectively. We can go even further and split the first quadratic coefficient $-\|\delta_t(\cdot)\|^2$ using a new parameter $\gamma \in (0, 1]$ to create one more degree of freedom. Moreover, we can use other parametrization of δ, since its form does not change the proofs.

Definition 3. *We introduce* ***General RealUID loss*** $\mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}(\delta, p^\theta_0)$ *on generated data* $p^\theta_0 \in \mathcal{P}(\mathbb{R}^D)$ *with coefficients* $\alpha, \beta, \gamma \in (0, 1]$*:*

$$\mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}(\delta, p^\theta_0) := \mathbb{E}_{\substack{t\sim[0,T], x^\theta_0\sim p^\theta_0, \\ x^\theta_t\sim p^\theta_t(\cdot|x^\theta_0)}}[-\gamma\|\delta_t(x^\theta_t)\|^2 + 2\alpha\langle\delta_t(x^\theta_t), f^*_t(x^\theta_t)\rangle - 2\beta\langle\delta_t(x^\theta_t), f^\theta_t(x^\theta_t|x^\theta_0)\rangle]$$
$$+ \mathbb{E}_{\substack{t\sim[0,T], x^*_0\sim p^*_0, \\ x^*_t\sim p^*_t(\cdot|x^*_0)}}[-(1-\gamma)\|\delta_t(x^*_t)\|^2 + 2(1-\alpha)\langle\delta_t(x^*_t), f^*_t(x^*_t)\rangle - 2(1-\beta)\langle\delta_t(x^*_t), f^*_t(x^*_t|x^*_0)\rangle].$$

Optionally, one can change default reparameterization $\delta = f^* - f$ *(e.g., with* $\delta = \beta(f^* - f)$*), and substitute sampled real data term* $f^*_t(x^*_t|x^*_0)$ *with the unconditional teacher* $f^*_t(x^*_t)$ *and vice versa.*

In case of $\delta = f^* - f$ and $\gamma \neq \alpha$, the General RealUID loss cannot be expressed as inverse min-max problem (16) for simple losses, since some scalar products do not eliminate each other. Nevertheless, min-max optimization of $\mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}$ still minimizes the similar squared ℓ_2-distance between the weighted teacher and generator-induced functions, attaining minimum when $p^\theta_0 = p^*_0$.

Lemma 3 (Distance minimized by General RealUID loss). *Maximization of General RealUID loss* $\mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}$ *over* δ *represents the squared* ℓ_2*-distance between the weighted teacher* f^* *and student function* $f^\theta := \arg\min_f \mathcal{L}_{UM}(f, p^\theta_0)$*:*

$$\max_\delta \mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}(\delta, p^\theta_0) = \mathbb{E}_{\substack{t\sim[0,T], \\ x^*_t\sim p^*_t}}\left[\frac{\|\frac{\beta}{\alpha}[p^*_t(x^*_t)f^*_t(x^*_t) - p^\theta_t(x^*_t)f^\theta_t(x^*_t)] + (p^\theta_t(x^*_t) - p^*_t(x^*_t))f^*_t(x^*_t)\|^2}{p^*_t(x^*_t)((1-\gamma)p^*_t(x^*_t) + \gamma p^\theta_t(x^*_t))/\alpha^2}\right].$$

The distances being minimized for RealUID (Lemma 2) and General RealUID (Lemma 3) are almost identical except the scale factor. Thus, we keep the same recommendations for choosing coefficients α, β as we discuss in Section 3.4. The factor β/α still has the largest impact within the distance, while α and γ set the scaling. Values β/α and γ should be chosen close to 1, but not exactly 1.

Proof. First, we take math expectation over data points x^*_0. Since the expectation can be taken in a reverse order, i.e., $\mathbb{E}_{x^*_0\sim p^*_0, x^*_t\sim p^*_t(\cdot|x^*_0)} = \mathbb{E}_{x^*_t\sim p^*_t, x^*_0\sim p^*_0(\cdot|x^*_t)}$, we see that

$$\begin{aligned}\mathbb{E}_{x^*_0\sim p^*_0, x^*_t\sim p^*_t(\cdot|x^*_0)}[\langle\delta_t(x^*_t), f^*_t(x^*_t|x^*_0)\rangle] &= \mathbb{E}_{x^*_t\sim p^*_t}\langle\delta_t(x^*_t), \mathbb{E}_{x^*_0\sim p^*_0(\cdot|x^*_t)}[f^*_t(x^*_t|x^*_0)]\rangle \\ &= \mathbb{E}_{x^*_t\sim p^*_t}[\langle\delta_t(x^*_t), f^*_t(x^*_t)\rangle]. \end{aligned} \tag{29}$$

For the term $\mathbb{E}_{x^\theta_0\sim p^\theta_0, x^\theta_t\sim p^\theta_t(\cdot|x^\theta_0)}[\langle\delta_t(x^\theta_t), f^\theta_t(x^\theta_t|x^\theta_0)\rangle] = \mathbb{E}_{x^\theta_t\sim p^\theta_t}[\langle\delta_t(x^\theta_t), f^\theta_t(x^\theta_t)\rangle]$, the reasoning is similar. Thus, we write down General RealUID loss (Def. 3) in an explicit form with $\delta_t = f^*_t - f_t$

$$\mathcal{L}^{\alpha,\beta}_{\text{R-UID}}(\delta, p^\theta_0) = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x^\theta_t\sim p^\theta_t}[-\gamma\|\delta_t(x^\theta_t)\|^2 + 2\alpha\langle\delta_t(x^\theta_t), f^*_t(x^\theta_t)\rangle - 2\beta\langle\delta_t(x^\theta_t), f^\theta_t(x^\theta_t)\rangle]$$
$$+\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x^*_t\sim p^*_t}[-(1-\gamma)\|\delta_t(x^*_t)\|^2 + 2(1-\alpha)\langle\delta_t(x^*_t), f^*_t(x^*_t)\rangle - 2(1-\beta)\langle\delta_t(x^*_t), f^*_t(x^*_t)\rangle].$$

Then, we rescale the generated data terms in the General RealUID loss using the equality $p^\theta_t(x_t) = \frac{p^\theta_t(x_t)}{p^*_t(x_t)}p^*_t(x_t)$ for $x_t \in \mathbb{R}^D$ (we assume $p^*_t(x_t) > 0, \forall x_t, t$) leaving only math expectation w.r.t. the real data, i.e,

$$\mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}(\delta, p^\theta_0) = \mathbb{E}_{\substack{t\sim[0,T], \\ x^*_t\sim p^*_t}}\left[-[(1-\gamma) + \gamma\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]\|\delta_t(x^*_t)\|^2\right]$$
$$+ \mathbb{E}_{\substack{t\sim[0,T], \\ x^*_t\sim p^*_t}}\left[2[(\beta-\alpha) + \alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]\langle\delta_t(x^*_t), f^*_t(x^*_t)\rangle - 2\beta\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}\langle\delta_t(x^*_t), f^\theta_t(x^*_t)\rangle\right].$$

Then we maximize the loss w.r.t. $\delta_t(x^*_t)$ for each x^*_t and t as a quadratic function. The maximum is achieved when

$$\delta_t(x^*_t) = \frac{[(\beta-\alpha) + \alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]f^*_t(x^*_t) - \beta\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}f^\theta_t(x^*_t)}{[(1-\gamma) + \gamma\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}]}. \tag{30}$$

The maximum itself equals to

$$\max_\delta \mathcal{L}^{\alpha,\beta,\gamma}_{\text{R-UID}}(\delta, p^\theta_0) = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x^*_t\sim p^*_t}\left[\frac{\|f^*_t(x^*_t)\cdot((\beta-\alpha) + \alpha\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}) - f^\theta_t(x^*_t)\cdot\beta\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}\|^2}{(1-\gamma) + \gamma\frac{p^\theta_t(x^*_t)}{p^*_t(x^*_t)}}\right].$$

Alternative parameterization. In the proximity of the solution, when generated data approaches real one, i.e., $p_t^\theta \approx p_t^*$, the optimal δ_t (30) approaches

$$\delta_t(x_t^*) \approx \frac{[(\beta-\alpha)+\alpha\cdot 1]f_t^*(x_t^*) - \beta\cdot 1\cdot f_t^\theta(x_t^*)}{[(1-\gamma)+\gamma\cdot 1]} \approx \beta(f_t^*(x_t^*) - f_t^\theta(x_t^*)).$$

Thus, the parametrization $\delta_t = \beta(f_t^* - f_t)$ may naturally help reach the solution without making the fake model learn extra information about the teacher near the optimum. □

A.3 SiD WITH REAL DATA

Our real data incorporation. We recall that data-free UID loss (Theorem 1), which is equivalent to SiD with $\alpha_{\text{SiD}} = 1/2$, can be restated via linearization technique with $\delta = f - f^*$ as

$$\mathcal{L}_{\text{UID}}(\delta, p_0^\theta) = \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta\sim p_0^\theta,\\ x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}} \left\{-\|\delta_t(x_t^\theta)\|^2 + 2\langle\delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle\right\}. \quad (31)$$

In turn, after real data incorporation, we obtain our RealUID loss (Theorem 2). Putting the explicit values for RealUM loss (17) in RealUID loss (18), we get the explicit formula:

$$\mathcal{L}_{\text{R-UID}}^{\alpha,\beta}(\delta, p_0^\theta) = \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta\sim p_0^\theta,\\ x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}} [-\alpha\|\delta_t(x_t^\theta)\|^2 + 2\alpha\langle\delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\beta\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle]$$

$$+ \mathbb{E}_{\substack{t\sim[0,T], x_0^*\sim p_0^*,\\ x_t^*\sim p_t^*(\cdot|x_0^*)}} [-(1-\alpha)\|\delta_t(x_t^*)\|^2 + 2(1-\alpha)\langle\delta_t(x_t^*), f_t^*(x_t^*)\rangle - 2(1-\beta)\langle\delta_t(x_t^*), f_t^*(x_t^*|x_0^*)\rangle].$$

These two formulas give us alternative explanation on how to add real data into arbitrary losses: we need to split each term in the linearized representation of the data-free loss between real and generated data. For example, in RealUID loss, its three terms are split with proportions α, α, β, respectively.

Combining with SiD. In SiD framework (Zhou et al., 2024a;b), the authors notice that UID loss (31) for generator updates, with additional normalization and the first coefficient $-\|\delta_t(x_t^\theta)\|^2$ scaled by $2\alpha_{\text{SiD}}$, empirically yields better performance. Namely, the SiD loss for generator with parameter $\alpha_{\text{SiD}} \in [0.5, 1.2]$ is

$$\mathcal{L}_{\alpha_{\text{SiD}}}(p_0^\theta) = \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta\sim p_0^\theta,\\ x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}} \left\{\frac{-2\alpha_{\text{SiD}}\|\delta_t(x_t^\theta)\|^2 + 2\langle\delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle}{\omega_t}\right\},$$

where $\omega_t \propto \text{NO-GRAD}\{\|f_t^\theta(x_t^\theta|x_0^\theta) - f_t^*(x_t^\theta)\|_1\}$ are normalization weights. For more details about time sampling and practical implementation, please refer to the original papers (Zhou et al., 2024a;b).

Following the structure of generator SiD loss, we propose to scale the first coefficient in our weighted RealUID loss during generator updates. The whole **SiD pipeline with real data**, determined by coefficients $\alpha, \beta \in (0, 1]$, $\alpha_{\text{SiD}} \in [0.5, 1.2]$ and teacher f^*, is two alternating steps:

1. Make one or several fake model f update steps, minimizing the real data modified UM loss $\mathcal{L}_{\text{R-UM}}^{\alpha,\beta}(f, p_0^\theta)$ (Def. 2):

$$L_{\text{R-UM}}^{\alpha,\beta}(f, p_0^\theta) = \underbrace{\alpha\cdot\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0^\theta\sim p_0^\theta, x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}\left[\|f_t(x_t^\theta) - \frac{\beta}{\alpha}f^\theta(x_t^\theta|x_0^\theta)\|^2\right]}_{\text{generated data } p_0^\theta \text{ term}}$$

$$+ \underbrace{(1-\alpha)\cdot\mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0^*\sim p_0^*, x_t^*\sim p_t^*(\cdot|x_0^*)}\left[\|f_t(x_t^*) - \frac{1-\beta}{1-\alpha}f_t^*(x_t^*|x_0^*)\|^2\right]}_{\text{real data } p_0^* \text{ term}}.$$

2. Make a generator update step, minimizing the loss $\mathcal{L}_{\text{R-UID},\alpha_{\text{SiD}}}^{\alpha,\beta}(p_0^\theta) =$

$$\mathbb{E}_{\substack{t\sim[0,T], x_0^\theta\sim p_0^\theta,\\ x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta)}} \left\{\frac{-2\alpha_{\text{SiD}}\cdot\alpha\cdot\|\delta_t(x_t^\theta)\|^2 + 2\alpha\langle\delta_t(x_t^\theta), f_t^*(x_t^\theta)\rangle - 2\beta\langle\delta_t(x_t^\theta), f_t^\theta(x_t^\theta|x_0^\theta)\rangle}{\omega_t}\right\},$$

where $\delta_t = f_t - f_t^*$ and $\omega_t \propto \text{NO-GRAD}\{\|f_t^\theta(x_t^\theta|x_0^\theta) - f_t^*(x_t^\theta)\|_1\}$.

We keep the same recommendations for choosing coefficients α, β as we discuss in Section 3.4. The optimal choice is slightly different $\alpha \neq \beta$ which are close to 1. Following (Zhou et al., 2024a), the best choice for α_{SiD} is $\alpha_{\text{SiD}} \in [1, 1.2]$.

A.4 Normalized UID and RealUID losses for minimizing ℓ_2-distance

Using the linearization technique from Section 3.1, we can estimate the non-squared ℓ_2-distance between the teacher $f^* := \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*)$ and student $f^\theta := \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^\theta)$ functions. In this case, the connection with the inverse optimization disappears.

For a fixed point x_t^θ and time t, we derive:

$$\begin{aligned}\|f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta)\| &= \max_{\delta_t(x_t^\theta)} \left\{ \langle \frac{\delta_t(x_t^\theta)}{\|\delta_t(x_t^\theta)\|}, f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta) \rangle \right\} \\ &= \max_{\delta_t(x_t^\theta)} \mathbb{E}_{x_0^\theta \sim p_0^\theta(\cdot|x_t^\theta)} \left\{ \langle \frac{\delta_t(x_t^\theta)}{\|\delta_t(x_t^\theta)\|}, f_t^*(x_t^\theta) \rangle - \langle \frac{\delta_t(x_t^\theta)}{\|\delta_t(x_t^\theta)\|}, f_t^\theta(x_t^\theta|x_0^\theta) \rangle \right\}. \quad (32)\end{aligned}$$

With the reparameterization $\delta_t = f_t^* - f_t$, the **Normalized UID loss** $\hat{\mathcal{L}}_{\text{UID}}(f, p_0^\theta)$ for min-max optimization to solve $\min_\theta \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{x_t^\theta \sim p_t^\theta} \|f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta)\|$ is:

$$\min_\theta \max_f \left\{ \hat{\mathcal{L}}_{\text{UID}}(f, p_0^\theta) := \mathbb{E}_{\substack{t\sim[0,T], x_0^\theta \sim p_0^\theta, \\ x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta)}} \left[\langle \frac{f_t^*(x_t^\theta) - f_t(x_t^\theta)}{\|f_t^*(x_t^\theta) - f_t(x_t^\theta)\|}, f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta|x_0^\theta) \rangle \right] \right\}. \quad (33)$$

Adding real data. Following the intuition from the proof in Appendix A.1.1, we can incorporate real data in Normalized UID loss (33) as well. We need to split two summands in the linearized representation (32) into generated and real data parts with weights $\alpha, (1-\alpha)$ and $\beta, (1-\beta)$.

Definition 4. *We introduce **Normalized RealUID loss** $\hat{\mathcal{L}}^{\alpha,\beta}_{\text{R-UID}}(f, p_0^\theta)$ on generated data $p_0^\theta \in \mathcal{P}(\mathbb{R}^D)$ with coefficients $\alpha, \beta \in (0, 1]$:*

$$\begin{aligned}\hat{\mathcal{L}}^{\alpha,\beta}_{\text{R-UID}}(f, p_0^\theta) := \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{\substack{x_t^\theta \sim p_t^\theta, \\ x_0^\theta \sim p_0^\theta(\cdot|x_t^\theta)}} \left\{ \langle \frac{f_t^*(x_t^\theta) - f_t(x_t^\theta)}{\|f_t^*(x_t^\theta) - f_t(x_t^\theta)\|}, \alpha \cdot f_t^*(x_t^\theta) - \beta \cdot f_t^\theta(x_t^\theta|x_0^\theta) \rangle \right\} \\ + \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{\substack{x_t^* \sim p_t^*, \\ x_0^* \sim p_0^*(\cdot|x_t^*)}} \left\{ \langle \frac{f_t^*(x_t^*) - f_t(x_t^*)}{\|f_t^*(x_t^*) - f_t(x_t^*)\|}, (1-\alpha) \cdot f_t^*(x_t^*) - (1-\beta) \cdot f_t^*(x_t^*|x_0^*) \rangle \right\}.\end{aligned}$$

Similar to the proof of RealUID distance Lemma 2, we can show that min-max optimization of Normalized RealUID loss minimizes the non-squared ℓ_2-norm between the similar weighted student f^θ and teacher f^* functions:

$$\max_f \hat{\mathcal{L}}^{\alpha,\beta}_{\text{R-UID}}(f, p_0^\theta) = \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{x_t^* \sim p_t^*} \left[\| ((\beta - \alpha) + \alpha \frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)}) \cdot f_t^*(x_t^*) - \beta \frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)} \cdot f_t^\theta(x_t^*) \| \right].$$

This distance attains minimum when $p_0^\theta = p_0^*$, justifying the procedure.

A.5 DMD approach with real data

Distribution Matching Distillation (Luo et al., 2023; Wang et al., 2023; Yin et al., 2024b;a) (DMD) approach distills Gaussian diffusion models with forward process $x_t = x_0 + \sigma_t \epsilon, \epsilon \sim \mathcal{N}(0, I)$.

This approach minimizes KL divergence $\mathbb{E}_{t\sim[0,T]} D_{KL}(p_t^\theta || p_t^*) = \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{x_t^\theta \sim p_t^\theta} \left[\log \left(\frac{p_t^\theta(x_t^\theta)}{p_t^*(x_t^\theta)} \right) \right]$ between the generated data p_t^θ and the real data p_t^*. The authors show the true gradient of $\mathbb{E}_{t\sim[0,T]} D_{KL}(p_t^\theta || p_t^*)$ w.r.t. θ can be computed via the score functions:

$$\mathbb{E}_{t\sim[0,T]} \left[\frac{d D_{KL}(p_t^\theta || p_t^*)}{d\theta} \right] = \mathbb{E}_{z\sim p^{\mathcal{Z}}, x_0^\theta = G(z), x_t^\theta \sim p_t^\theta} \left[(\nabla_{x_t^\theta} \ln p_t^\theta(x_t^\theta) - \nabla_{x_t^\theta} \ln p_t^*(x_t^\theta)) \frac{dG_\theta(z)}{d\theta} \right].$$

Then, this true gradient is estimated with the teacher score function $s^* := \arg\min_s \mathcal{L}_{\text{DSM}}(s, p_0^*)$ and student score $s^\theta = \arg\min_s \mathcal{L}_{\text{DSM}}(s, p_0^\theta)$ at each time moment:

$$\mathbb{E}_{t\sim[0,T]} \left[\frac{d D_{KL}(p_t^\theta || p_t^*)}{d\theta} \right] = \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{z\sim p^{\mathcal{Z}}, x_0^\theta = G_\theta(z), x_t^\theta \sim p_t^\theta} \left[(s_t^\theta(x_t^\theta) - s_t^*(x_t^\theta)) \frac{dG_\theta}{d\theta} \right].$$

The final algorithm alternates updates for the fake model and the generator similar to SiD approach.

We would like to highlight that DMD does not fit our UID framework. The UID loss is uniquely determined by its input UM loss. In the case of Diffusion models and DMD, the UM loss is the $\mathcal{L}_{DSM}(s, p_0^\theta)$ loss. With this loss, the resulting UID loss becomes exactly the SiD loss, not DMD.

Adding real data. We investigated a theoretical possibility to incorporate real data into the DMD framework. We found that we can use the Modified DSM loss (17) to train the modified student score function $s_t^{\theta,\alpha} = \arg\min_s \mathcal{L}^{\alpha,\alpha}_{\text{M-DSM}}(s, p_0^\theta)$ with coefficients $\alpha = \beta$:

$$\begin{aligned}\mathcal{L}^{\alpha,\alpha}_{\text{M-DSM}}(s, p_0^\theta) \quad &:= \quad \underbrace{\alpha \cdot \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0^\theta\sim p_0^\theta, x_t^\theta\sim p_t^\theta(\cdot|x_0)}\left[\|s_t(x_t^\theta) - s^\theta(x_t^\theta|x_0^\theta)\|^2\right]}_{\text{generated data } p_0^\theta \text{ term}} \\ &+ \quad \underbrace{(1-\alpha) \cdot \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_0^*\sim p_0^*, x_t^*\sim p_t^*(\cdot|x_0^*)}\left[\|s_t(x_t^*) - s_t^*(x_t^*|x_0^*)\|^2\right]}_{\text{real data } p_0^* \text{ term}}.\end{aligned}$$

Then apply the generator parameters update based on the KL divergence between mixed distributions.

Lemma 4 (DMD with real data). *Consider real data distribution $p_0^* \in \mathcal{P}(\mathbb{R}^D)$ and generated by generator G_θ distribution $p_0^\theta \in \mathcal{P}(\mathbb{R}^D)$. Then, KL divergence between mixed and real data for $\alpha \in (0, 1]$ has the following gradients with modified student score $s_t^{\theta,\alpha} := \arg\min_s \mathcal{L}^{\alpha,\alpha}_{\text{M-DSM}}(s, p_0^\theta)$ and teacher score $s_t^* := \arg\min_s \mathcal{L}_{DSM}(s, p_0^*)$:*

$$\mathbb{E}_{t\sim[0,T]}\left[\frac{dD_{KL}(\alpha \cdot p_t^\theta + (1-\alpha)\cdot p_t^*||p_t^*)}{d\theta}\right] = \mathbb{E}_{\substack{t\sim[0,T], z\sim p^{\mathcal{Z}}, \\ x_0^\theta = G_\theta(z), x_t^\theta\sim p_t^\theta}}\left[\alpha(s_t^{\theta,\alpha}(x_t^\theta) - s_t^*(x_t^\theta))\frac{dG_\theta}{d\theta}\right].$$

Although this approach is theoretically justified, it requires coefficients $\alpha = \beta$ which work poorly for our RealUID, see Section 3.4. In the proof below, we also show that use of coefficients $\alpha \neq \beta$ in the fake model loss leads to the total collapse of a generator. The proof itself follows (Wang et al., 2023).

Proof. We aim to minimize KL divergence between generated distribution p_0^θ and the real data p_0^*

$$\min_{p_0^\theta} E(p_0^\theta) := \mathbb{E}_{t\sim[0,T]}\left[D_{KL}(\alpha \cdot p_t^\theta + (1-\alpha)\cdot p_t^*||p_t^*)\right].$$

First, we use (Wang et al., 2023, Lemma 1) which says that, for any two distributions $p, q \in \mathcal{P}(\mathbb{R}^D)$ and point $x \in \mathbb{R}^D$, we have

$$\left(\frac{\delta D_{KL}(q||p)}{\delta q}\right)[x] = \log q(x) - \log p(x) + 1.$$

Second, for the parametrization $x_0^\theta = G_\theta(z), z \sim p^{\mathcal{Z}}$ and a fixed point x_t, we have (Wang et al., 2023, Lemma 2)

$$\frac{\delta p_t^\theta(x_t)}{\delta p_0^\theta}[\theta] = \int_z p_t^\theta(x_t|x_0^\theta)p^{\mathcal{Z}}(z)dz.$$

It allows us to obtain

$$\begin{aligned}&\frac{\delta E(p_0^\theta)}{\delta p_0^\theta}[\theta] = \mathbb{E}_t \frac{\delta D_{KL}(\overbrace{\alpha \cdot p_t^\theta(\cdot) + (1-\alpha)\cdot p_t^*(\cdot)}^{=:q_t}||p_t^*(\cdot))}{\delta p_0^\theta}[\theta] \\ &= \mathbb{E}_t \int \frac{\delta D_{KL}(q_t||p_t^*)}{\delta q_t}[x_t] \cdot \frac{\delta q_t}{\delta p_t^\theta}[x_t] \cdot \frac{\delta p_t^\theta(x_t)}{\delta p_0^\theta}[\theta] \cdot dx_t \\ &= \mathbb{E}_t \int \left[\log(\alpha \cdot p_t^\theta(x_t) + (1-\alpha)\cdot p_t^*(x_t)) - \log(p_t^*(x_t)) + 1\right] \cdot \alpha \cdot \int_z p_t^\theta(x_t|x_0^\theta)p^{\mathcal{Z}}(z)dz \cdot dx_t \\ &= \mathbb{E}_{t,\epsilon,z}[\alpha \log(\alpha \cdot p_t^\theta(x_t^\theta) + (1-\alpha)\cdot p_t^*(x_t^\theta)) - \alpha \log(p_t^*(x_t^\theta)) + \alpha] \\ &= \mathbb{E}_{t,\epsilon,z}[\alpha \log\left(\alpha \cdot \frac{p_t^\theta(x_t^\theta)}{p_t^*(x_t^\theta)} + (1-\alpha)\right) + \alpha],\end{aligned} \tag{34}$$

where $x_0^\theta = G_\theta(z), x_t^\theta = x_0^\theta + \sigma_t\epsilon, \epsilon \sim \mathcal{N}(0, I)$. Finally, we take derivative w.r.t. θ from (34):

$$\begin{aligned}\nabla_\theta \frac{\delta E(p_0^\theta)}{\delta p_0^\theta}[\theta] &= \mathbb{E}_{t,\epsilon,z}\left[\alpha \cdot \nabla_{x_t^\theta} \log\left(\alpha \cdot \frac{p_t^\theta(x_t^\theta)}{p_t^*(x_t^\theta)} + (1-\alpha)\right) \cdot \frac{\partial x_t^\theta}{\partial \theta}\right] \\ &= \mathbb{E}_{t,\epsilon,z}\left[\alpha \cdot \nabla_{x_t^\theta} \log\left(\alpha \cdot \frac{p_t^\theta(x_t^\theta)}{p_t^*(x_t^\theta)} + (1-\alpha)\right) \cdot \frac{\partial G_\theta(z)}{\partial \theta}\right] \\ &= \mathbb{E}_{t,\epsilon,z}\left[\alpha^2 \frac{\nabla_{x_t^\theta} p_t^\theta(x_t^\theta)/p_t^*(x_t^\theta)}{\alpha \cdot \frac{p_t^\theta(x_t^\theta)}{p^*(x_t^\theta)} + (1-\alpha)} \cdot \frac{\partial G_\theta(z)}{\partial \theta}\right]. \end{aligned} \tag{35}$$

Now, we show how to obtain unbiased estimate of this gradient. We minimize the following loss function over the fake model s:

$$\begin{aligned}\mathcal{L}_{M-DSM}^{\alpha,\alpha}(s, p_0^\theta) &:= \alpha \cdot \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^\theta \sim p_t^\theta, x_0^\theta \sim p_0^\theta(\cdot|x_t)}\left[\|s_t(x_t^\theta) - s^\theta(x_t^\theta|x_0^\theta)\|^2\right] \\ &+ (1-\alpha) \cdot \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x_t^* \sim p_t^*, x_0^* \sim p_0^*(\cdot|x_t^*)}\left[\|s_t(x_t^*) - s_t^*(x_t^*|x_0^*)\|^2\right].\end{aligned}$$

This loss is equivalent to the following sequence

$$\min_s \left\{\alpha \mathbb{E}_{\substack{t\sim[0,T],\\ x_t^\theta \sim p_t^\theta}} \|s_t(x_t^\theta) - s_t^\theta(x_t^\theta)\|^2 + (1-\alpha)\mathbb{E}_{\substack{t\sim[0,T],\\ x_t^* \sim p_t^*}} \|s_t(x_t^*) - s_t^*(x_t^*)\|^2\right\},$$

$$\min_s \left\{\alpha \mathbb{E}_{\substack{t\sim[0,T],\\ x_t^\theta \sim p_t^\theta}} \|s_t(x_t^\theta) - \nabla_{x_t^\theta} \log p_t^\theta(x_t^\theta)\|^2 + (1-\alpha)\mathbb{E}_{\substack{t\sim[0,T],\\ x_t^* \sim p_t^*}} \|s_t(x_t^*) - \nabla_{x_t^*} \log p_t^*(x_t^*)\|^2\right\},$$

$$\min_s \mathbb{E}_{\substack{t\sim[0,T],\\ x_t^* \sim p_t^*}} \left[\alpha \|s_t(x_t^*) - \nabla \log p_t^\theta(x_t^*)\|^2 \frac{p_t^\theta(x_t^*)}{p_t^*(x_t^*)} + (1-\alpha)\|s_t(x_t^*) - \nabla \log p_t^*(x_t^*)\|^2\right].$$

The optimal solution $s^{\theta,\alpha}$ of this quadratic minimization for each point x_t and time moment t is

$$s_t^{\theta,\alpha}(x_t) = \frac{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} \nabla_{x_t} \log p_t^\theta(x_t) + (1-\alpha)\nabla_{x_t} \log p_t^*(x_t)}{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} + (1-\alpha)}.$$

Thus, we have the following estimate with modified student score $s^{\theta,\alpha}$ and teacher score $s_t^*(x_t) := \nabla_{x_t} \log p_t^*(x_t)$

$$\begin{aligned}s_t^{\theta,\alpha}(x_t) - s_t^*(x_t) &= \frac{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} \nabla_{x_t} \log p_t^\theta(x_t) + (1-\alpha)\nabla_{x_t} \log p_t^*(x_t)}{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} + (1-\alpha)} - \nabla_{x_t} \log p_t^*(x_t) \\ &= \frac{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} (\nabla_{x_t} \log p_t^\theta(x_t) - \nabla_{x_t} \log p_t^*(x_t))}{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} + (1-\alpha)} \\ &= \frac{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} \nabla_{x_t} \log \frac{p_t^\theta(x_t)}{p_t^*(x_t)}}{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} + (1-\alpha)} = \frac{\alpha \nabla_{x_t} p_t^\theta(x_t)/p_t^*(x_t)}{\alpha \frac{p_t^\theta(x_t)}{p_t^*(x_t)} + (1-\alpha)}.\end{aligned}$$

Hence, this estimate completely matches with required gradient (35):

$$(35) = \mathbb{E}_{t,\epsilon,z}\left[\alpha \cdot (s^{\theta,\alpha}(x_t^\theta) - s_t^*(x_t^\theta)) \cdot \frac{\partial G_\theta(z)}{\partial \theta}\right].$$

The use of other coefficients during student score optimization does not work. For the other student scores $s_t^{\theta,\alpha,\beta} := \arg\min_s \mathcal{L}_{\text{M-DSM}}^{\alpha,\beta}(s, p_0^\theta)$, the estimate $s_t^{\theta,\alpha,\beta}(x_t) - \nabla_{x_t} \log p_t^*(x_t)$ does not lead to the necessary difference $\nabla_{x_t} \log p_t^\theta(x_t) - \nabla_{x_t} \log p_t^*(x_t) = 0$. And the optimal generator collapses due to large bias.

□

B REALUID ALGORITHM FOR FLOW MATCHING MODELS

We provide a practical implementation of our RealUID approach for FM in Algorithm 1. In the loss functions, we retain only the terms dependent on the target parameters. For the fake model, we reformulate the maximization objective as a minimization. We use alternating optimization, updating the fake model K times per one student update for stability.

Algorithm 1 Real data modified Unified Inversion Distillation (RealUID) for Flow Matching

Input: teacher u^*, student generator G_θ, fake model u_ψ, real data p_0^*, coefficients $\alpha, \beta \in (0, 1]$, generator update steps K, number of iterations N, batch size B, fake model minimizer Opt_{st}, generator minimizer Opt_{gen}, latent distribution $p^{\mathcal{Z}}$, noise distribution p_1.

for $n = 0, \dots, N-1$ **do**
  Sample noise batch $\{x_{1,i}\}_{i=1}^B \sim p_1$ and generated batch $\{x_{0,i}^\theta = G_\theta(z_i)\}_{i=1}^B$, $z_i \sim p^{\mathcal{Z}}$;
  Sample time batch $\{t_i\}_{i=1}^B \sim U[0,1]$ and calculate $x_{t_i,i}^\theta = (1-t_i)x_{0,i}^\theta + t_i x_{1,i}$;
  if student step $(n \% K \neq 0)$ **then**
    Sample real data batch $\{x_{0,i}^*\}_{i=1}^B \sim p_0^*$ and calculate $x_{t_i,i}^* = (1-t_i)x_{0,i}^* + t_i x_{1,i}$;
    Update fake model parameters ψ via minimizer Opt_{st} step with gradients of

$$\frac{1}{B}\sum_{i=1}^{B}\left[\alpha\|u_\psi(t_i, x_{t_i,i}^{sg[\theta]}) - \frac{\beta}{\alpha}(x_{1,i} - x_{0,i}^{sg[\theta]})\|^2 + (1-\alpha)\|u_\psi(t_i, x_{t_i,i}^*) - \frac{1-\beta}{1-\alpha}(x_{1,i} - x_{0,i}^*)\|^2\right];$$

  else
    Update generator parameters θ via minimizer Opt_{gen} step with gradients of

$$\frac{1}{B}\sum_{i=1}^{B}\left[\alpha\|u^*(t_i, x_{t_i,i}^\theta) - \frac{\beta}{\alpha}(x_{1,i} - x_{0,i}^\theta)\|^2 - \alpha\|u_{sg[\psi]}(t_i, x_{t_i,i}^\theta) - \frac{\beta}{\alpha}(x_{1,i} - x_{0,i}^\theta)\|^2\right];$$

  end if
end for

C UNIFIED INVERSE DISILLATION FOR BRIDGE MATCHING AND STOCHASTIC INTERPOLANTS

C.1 BRIDGE MATCHING

Bridge Matching (Liu et al., 2022b; Peluchetti, 2023) is an extension of diffusion models specifically design to solve data-to-data, e.g. image-to-image problems. Typically, the distribution p_T is the distribution of "corrupted data" and p_0 is the distribution of clean data, furthermore, there is some coupling of clean and corrupted data $\pi(x_0, x_T)$ with marginals $p_0(x_0)$ and $p_T(x_T)$. To construct the diffusion which recovers clean data given a corrupted data, one first needs to build prior process (which often is the same forward process used in diffusions):

$$dx_t = f_t(x_t) + g_t dw_t,$$

where $f_t(\cdot)$ is a drift function and g_t is a time-dependent scalar noise scheduler. This prior process defines conditional density $p_t(x_t|x_0)$ and the posterior density $p_t(x_t|x_0, x_T)$ called "diffusion bridge". To recover p_0 from p_T, one can use reverse-time SDE

$$dx_t = \left(f_t(x_t) - g_t^2 \cdot v^\pi(x_t)\right)dt + g_t d\bar{w}_t,$$

where the drift $v_t^\pi(x_t)$ is learned via solving of the bridge matching problem:

$$\mathcal{L}_{\text{BM}}(v, \pi) = \mathbb{E}_{t\sim[0,T],(x_0,x_T)\sim\pi(x_0,x_T),x_t\sim p_t(x_t|x_0,x_T)}\left[w_t\|v_t(x_t) - \nabla_{x_t}\log p_t(x_t|x_0)\|^2\right]. \quad (36)$$

However, this reverse-time diffusion in general does not guarantee that the produced samples come from the same coupling $\pi(x_0, x_T)$ used for training. This happens only if $\pi(x_0, x_T)$ solves entropic optimal transport between p_0 and p_T. To guarantee the preservance of the coupling $\pi(x_0, x_T)$, there

exists another version of Bridge Matching called either Augmented Bridge Matching or Conditional Bridge Matching, which differs only by addition of a condition on x_T to the drift function $v_t(x_t, x_T)$:

$$\mathcal{L}_{\text{ABM}}(v, \pi) = \mathbb{E}_{t\sim[0,T],(x_0,x_T)\sim\pi(x_0,x_T),x_t\sim p(x_t|x_0,x_T)} \left[w_t \| v_t(x_t, x_T) - \nabla_{x_t} \log p_t(x_t|x_0) \|_2^2 \right].$$

The learned conditional drift is then used for sampling via the reverse-time SDE starting from a given $x_T \sim p_T$:

$$dx_t = \left(f_t(x_t) - g_t^2 \cdot v_t^{\pi}(x_t, x_T) \right) dt + g_t d\bar{w}_t.$$

C.2 Stochastic Interpolants

The Stochastic Interpolants framework generalizes Flow Matching and diffusion models, constructing a diffusion or flow between two given distributions p_0 and p_T. To do so, one needs to consider the interpolation between any pair of points (x_0, x_T) which are sampled from the coupling $\pi(x_0, x_T)$ with marginals p_0 and p_T. The interpolation itself is given by formula

$$x_t = I(t, x_0, x_T) + \gamma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad t \in [0, T],$$

where $I(0, x_0, x_T) = x_0$, $I(T, x_0, x_T) = x_T$, $\gamma_0 = \gamma_T = 0$ and $\gamma_t > 0$ for all $t \in (0, T)$. This interpolant defines a conditional Gaussian path $p_t(x_t|x_0, x_T)$. Note that in the original paper (Albergo et al., 2023), the authors consider the time interval $[0, 1]$, but those two intervals are interchangeable by using a change of variable $t' = \frac{T}{t}$. Thus, the ODE interpolation between p_0 and p_T is given by:

$$dx_t = u_t(x_t)dt, \quad x_0 \sim p_0,$$

where $u_t(x, x_T) := \mathbb{E}[\dot{x}_t | x_t = x] = \mathbb{E}[\partial_t I(t, x_0, x_T) + \dot{\gamma}\epsilon | x_t = x]$ is the unique minimizer of the quadratic objective:

$$\mathcal{L}_{\text{SI}}(v, \pi) = \mathbb{E}_{\substack{t\sim[0,T],(x_0,x_T)\sim\pi(x_0,x_T),\\(x_t,\epsilon)\sim p(x_t|x_0,x_T)}} \left[w_t \| v_t(x_t, x_T) - (\partial_t I(t, x_0, x_T) + \dot{\gamma}_t \epsilon) \|^2 \right]. \tag{37}$$

The authors also provide a way of matching the score and the SDE drift of the reverse process by solving similar MSE matching problems.

C.3 Objective for Unified Inverse Distillation for general data coupling

The essential difference of Bridge Matching and Stochastic Interpolants from diffusion models and Flow Matching with a Gaussian path is that they additionally introduce coupling $\pi(x_0, x_T)$ used to sample x_t and can work with conditional drifts.

This difference can be easily incorporated to our RealUID distillation framework just by parametrizing the generator G_θ to output not the samples from the initial distribution p_0^θ, but from the coupling π^θ. One can do it by setting $\pi^\theta(x_0, x_T) = p_T(x_T)\pi_0^\theta(x_0|x_T)$, where conditional data distribution $\pi_0^\theta(x_0|x_T)$ is parametrized by the *student generator* $G_\theta : \mathcal{Z} \times \mathbb{R}^D \to \mathbb{R}^D$ conditioned on a sample $x_T \sim p_T$. This approach is specifically used in Inverse Bridge Matching Distillation (IBMD) (Gushchin et al., 2024). Hence, our Universal Inverse Distillation objective can be written just by substituting student distribution p_0^θ by student coupling π^θ, substituting real data p_0^* by real data coupling π^* and adding extra conditions.

Definition 5. *We define* ***Universal Matching loss with real data for general coupling*** *on generated data coupling $\pi^\theta \in \mathcal{P}(\mathbb{R}^D \times \mathbb{R}^D)$ with $\alpha, \beta \in (0, 1]$:*

$$\mathcal{L}^{\alpha,\beta}_{\textit{R-UM-coup}}(f, \pi^\theta) = \underbrace{\alpha \cdot \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{\substack{x_T\sim p_T, x_0^\theta\sim\pi_0^\theta(\cdot|x_T),\\ x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta,x_T)}} \left[\| f_t(x_t^\theta, x_T) - \frac{\beta}{\alpha} f^\theta(x_t^\theta | x_0^\theta, x_T) \|^2 \right]}_{\textit{generated data } \pi^\theta \textit{ term}}$$

$$+ \underbrace{(1-\alpha) \cdot \mathbb{E}_{t\sim[0,T]} \mathbb{E}_{\substack{x_T\sim p_T, x_0^*\sim\pi_0^*(\cdot|x_T),\\ x_t^*\sim p_t^*(\cdot|x_0,x_T)}} \left[\| f_t(x_t^*, x_T) - \frac{1-\beta}{1-\alpha} f_t^*(x_t^* | x_0^*, x_T) \|^2 \right]}_{\textit{real data } \pi^* \textit{ term}}.$$

And the corresponding ***Universal Inverse Distillation loss with real data for general coupling*** *is:*

$$\min_\theta \max_f \{ \mathcal{L}^{\alpha,\beta}_{\textit{R-UID-coup}}(f, \pi^\theta) := \mathcal{L}^{\alpha,\beta}_{\textit{R-UM-coup}}(f^*, \pi^\theta) - \mathcal{L}^{\alpha,\beta}_{\textit{R-UM-coup}}(f, \pi^\theta) \}.$$

Table 6: Ablation of the fine-tuning for α_{FT} and β_{FT} for unconditional (left) and conditional (right) generation. Each cell reports the resulting FID score for the corresponding $(\alpha_{\text{FT}}, \beta_{\text{FT}})$; "–" indicates the method did not converge. Best results are **bolded**.

$\alpha_{\text{FT}}/\beta_{\text{FT}}$	0.94	0.96	0.98	1.0
0.94	-	-	2.07	**2.03**
0.96	-	-	-	2.11
0.98	2.07	-	-	-
1.0	-	-	-	-

$\alpha_{\text{FT}}/\beta_{\text{FT}}$	0.94	0.96	0.98	1.0
0.94	-	-	1.96	**1.91**
0.96	-	-	-	1.96
0.98	1.95	-	-	-
1.0	-	-	-	-

In case of coupling match $\pi^\theta = \pi^*$, the RealUID loss for couplings attains its minimum, i.e.,

$$\begin{aligned}\min_\theta \max_f \mathcal{L}^{\alpha,\beta}_{\text{R-UID-coup}}(f, \pi^\theta) &= \min_\theta \{\underbrace{\mathcal{L}^{\alpha,\beta}_{\text{R-UM-coup}}(f^*, \pi^\theta) - \min_f \{\mathcal{L}^{\alpha,\beta}_{\text{R-UM-coup}}(f, \pi^\theta)\}}_{\geq 0}\} \\ &= \mathcal{L}^{\alpha,\beta}_{\text{R-UM-coup}}(f^*, \pi^*) - \underbrace{\min_f \{\mathcal{L}^{\alpha,\beta}_{\text{R-UM-coup}}(f, \pi^*)\}}_{=\mathcal{L}^{\alpha,\beta}_{\text{R-UM-coup}}(f^*, \pi^*)} = 0.\end{aligned}$$

D Experimental details

Training hyperparameters. We train with Adam (Kingma & Ba, 2014), using $(\beta_1, \beta_2) = (0, 0.999)$. The learning rate is 3×10^{-5} for training from scratch and 1×10^{-5} for fine-tuning. A 500-step linear warm-up is applied only when training from scratch. We use a batch size of 256 and maintain an EMA of the generator parameters with decay 0.999. To regulate adaptation between the generator and the fake model, the generator is updated once for every $K = 5$ updates of the fake model, following DMD2 (Yin et al., 2024a). Additionally, at each optimization step we apply ℓ_2 gradient-norm clipping with threshold 1.0 to both the generator and the fake model.

Training time. All distillation experiments were trained for 400,000 gradient updates, corresponding to approximately 4.5 days. All finetuning experiments were conducted for 100,000 gradient updates, which took a little more than 1 day, starting from the best distillation checkpoints. All experiments were executed on a single Ascend910B NPU with 65 GB of VRAM memory. The reported results are based on the checkpoints that achieved the best Fréchet Inception Distance (FID) during training.

Codebase and Dataset. Building on the reference codebase of Tong et al. (2023), which serves as our primary experimental infrastructure, we integrate the training algorithm described in Algorithm 1. We evaluate the resulting approach on CIFAR-10 (32×32) under both conditional and unconditional settings, benchmarking against established baselines.

Models Initialization and Generator Parametrization. The generator G_θ is initialized by replicating both the architecture and parameters of the teacher model f^*, while the fake model f is initialized with random weights. We parameterize the generator using a residual formulation:

$$G_\theta(z) = z + g_\theta(0, z),$$

where the input $t = 0$ corresponds to the fixed control input used in the teacher model f^*. Empirically, we observe that this initialization strategy and parameterization lead to improved performance.

GAN details. We integrate a GAN loss into our framework in line with SiD^{2}A and DMD2 (Zhou et al., 2024a; Yin et al., 2024a). In the original setup of Zhou et al. (2024a), the adversarial loss employs a coefficient ratio of $\lambda^D_{\text{adv}}/\lambda^{G_\theta}_{\text{adv}} = 10^2$ (see Table 6 in Zhou et al. (2024a)), a choice that poses practical difficulties due to the extreme imbalance between generator and discriminator losses. To mitigate this issue, we adopt the formulation of Yin et al. (2024a), where the ratio is ≈ 3, and evaluate different coefficient scales (see result in the Table 1).

$\alpha \diagdown \beta$	0.94	0.96	0.98	1.0
0.94	2.60	1.93	2.13	2.53
0.96	**1.70**	2.77	2.00	2.47
0.98	2.16	2.04	2.62	2.42
1.0	2.96	2.48	2.23	2.62

Table 7: Ablation studies of (α, β) coefficients for CelebA (400k training steps). The baseline RealUID ($\alpha = 1.0, \beta = 1.0$) does not use real data. Configurations that outperform and significantly outperform the baseline are highlighted. All values report FID ↓, where lower is better. The best configuration is **bolded**.

Evaluation protocol. We evaluate image quality using the Fréchet Inception Distance (FID; Heusel et al., 2017), computed from 50,000 generated samples following Karras et al. (2022; 2020; 2019). In line with SiD (Zhou et al., 2024b), we periodically compute FID during distillation and select the checkpoint achieving the minimum value. To ensure statistical reliability, we repeat the evaluation over 3 independent runs, rather than 10 as in SiD, because the empirical variance of FID in our experiments was below 0.01.

E Additional Results

E.1 Fine-tuning ablation study on coefficients $\alpha_{\text{FT}}, \beta_{\text{FT}}$.

This section presents an ablation of the fine-tuning stage over the loss-balancing coefficients α_{FT} and β_{FT}. Results are summarized in Table 6, where "–" denotes non-convergence. We observe that training is highly sensitive to the choice of $(\alpha_{\text{FT}}, \beta_{\text{FT}})$: many configurations do not converge, underscoring the need for careful selection. Notably, the same set of $(\alpha_{\text{FT}}, \beta_{\text{FT}})$ exhibit stable optimization and yield improved FID for both conditional and unconditional CIFAR-10 generation.

E.2 Ablation study on CelebA dataset

In this section, we present the results of the same ablation studies from §4.2 on the CelebA dataset with higher 64×64 resolution (Liu et al., 2015). The results are summarized in Table 7.

Many pairs (α, β) demonstrate improvements relative to the baseline $(\alpha = 1.0, \beta = 1.0)$. Similar to the results from Table 1 for CIFAR10, the same pairs of coefficients with $\beta/\alpha = 1.02$ or $\beta/\alpha = 0.98$ yield a significant improvement in quality. For example, pair $(\alpha = 0.96, \beta = 0.94)$ yields FID **1.70** against FID **2.62** for the data-free baseline.

Training hyperparameters. For training from zero, we take the same architecture (Tong et al., 2023) as for the CIFAR-10 dataset with 32×32 resolution, but adapted it to a larger dimension. We train it with Adam (Kingma & Ba, 2014), using $(\beta_1, \beta_2) = (0, 0.999)$, learning rate 5×10^{-6} and a 500-step linear warm-up. We use a batch size of 64 and maintain an EMA of the generator parameters with decay 0.999. To regulate adaptation between the generator and the fake model, the generator is updated once for every $K = 5$ updates of the fake model, following DMD2 (Yin et al., 2024a). Additionally, at each optimization step we apply ℓ_2 gradient-norm clipping with threshold 1.0 to both the generator and the fake model.

E.3 Example of samples for different methods.

This section presents representative sample outputs from various studies conducted within the RealUID framework.

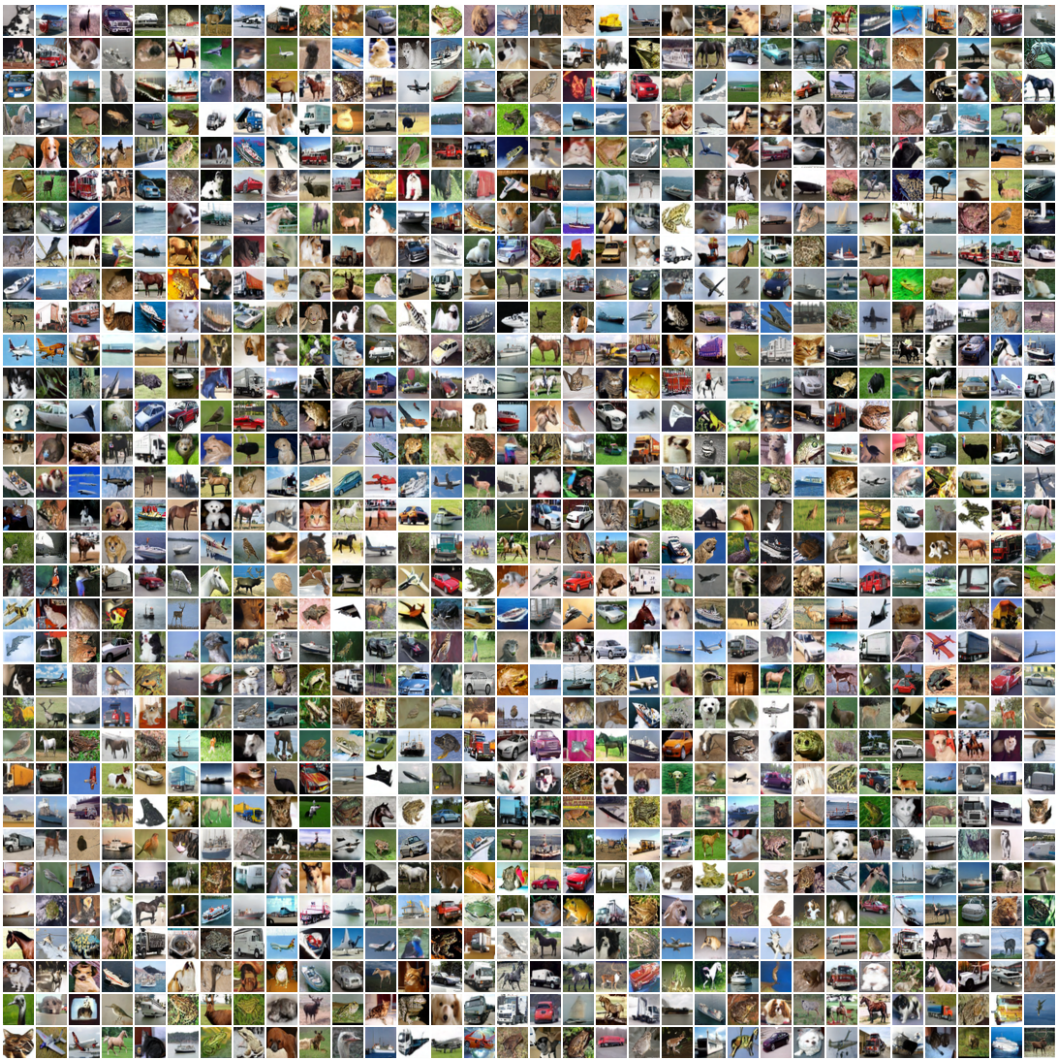

Figure 5: Uncurated samples for *unconditional* generation by the one-step RealUID ($\alpha = 1.0, \beta = 1.0$) trained on CIFAR-10. Quantitative results are reported in Table 2.

Figure 6: Uncurated samples for *unconditional* generation by the one-step RealUID ($\alpha = 1.0, \beta = 1.0$) + GAN ($\lambda_{\text{adv}}^{G_\theta} = 0.3, \lambda_{\text{adv}}^{D} = 1$) trained on CIFAR-10. Quantitative results are reported in Table 2.

Figure 7: Uncurated samples for *unconditional* generation by the one-step RealUID ($\alpha = 0.94, \beta = 0.96$) trained on CIFAR-10. Quantitative results are reported in Table 2.

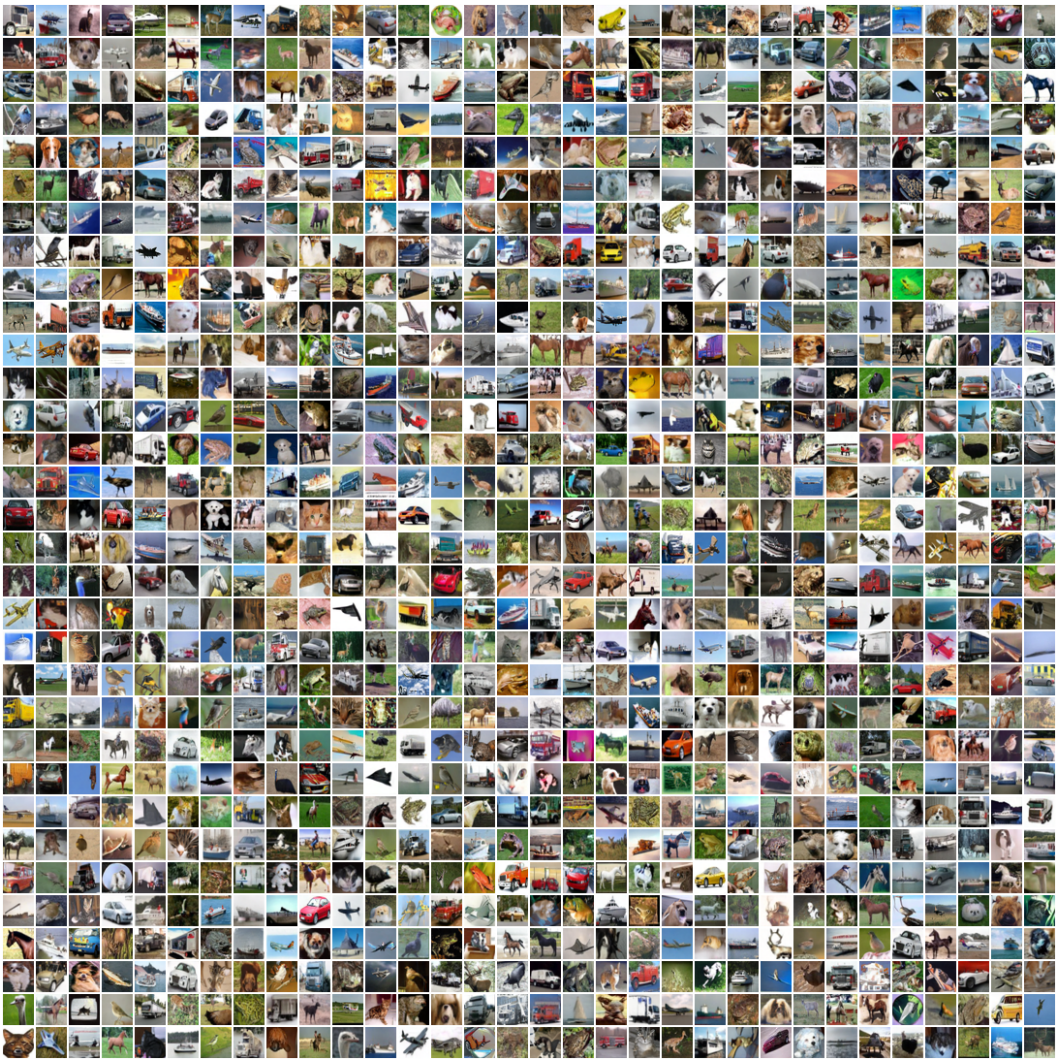

Figure 8: Uncurated samples for *unconditional* generation by the one-step RealUID ($\alpha = 0.94, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$) trained on CIFAR-10. Quantitative results are reported in Table 2.

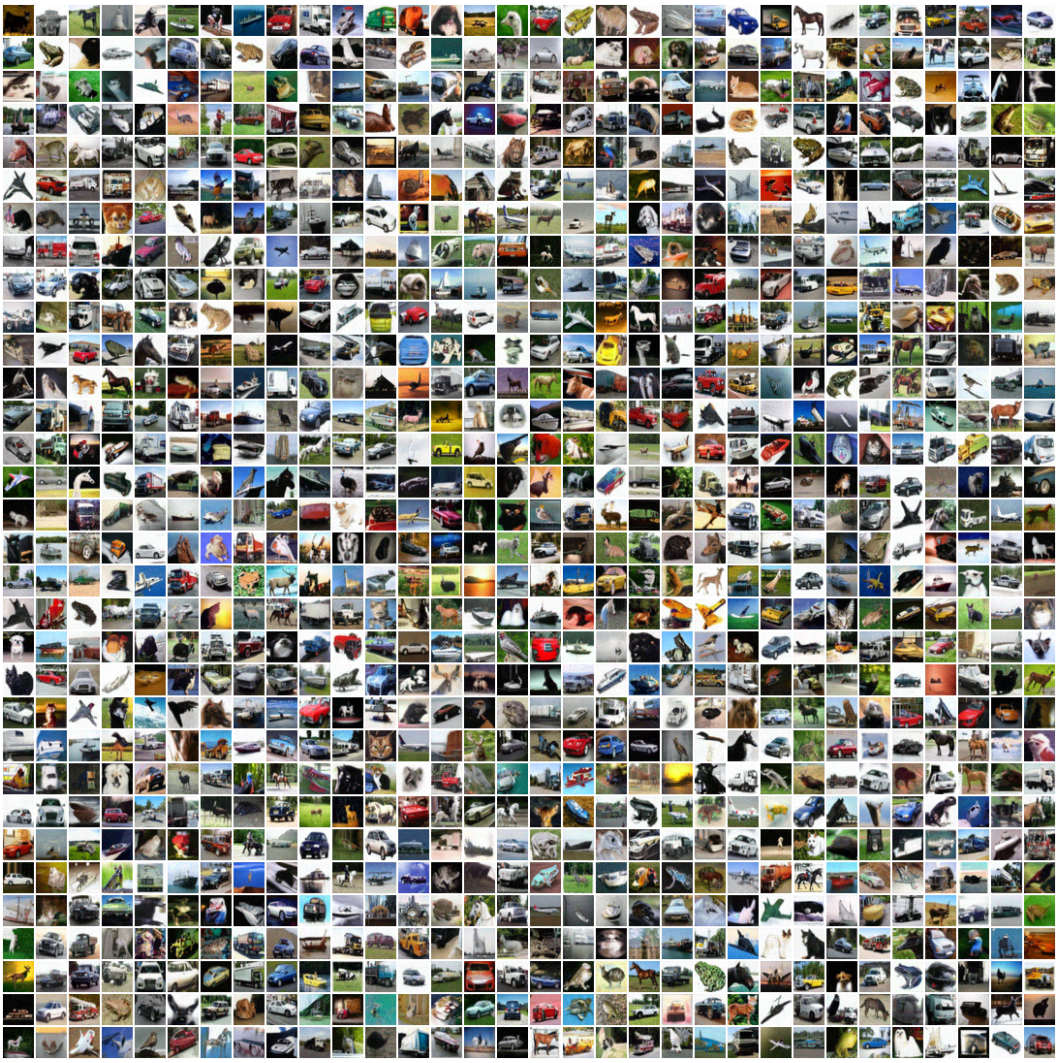

Figure 9: Uncurated samples for *conditional* generation by the one-step RealUID ($\alpha = 1.0, \beta = 1.0$) trained on CIFAR-10. Quantitative results are reported in Table 2.

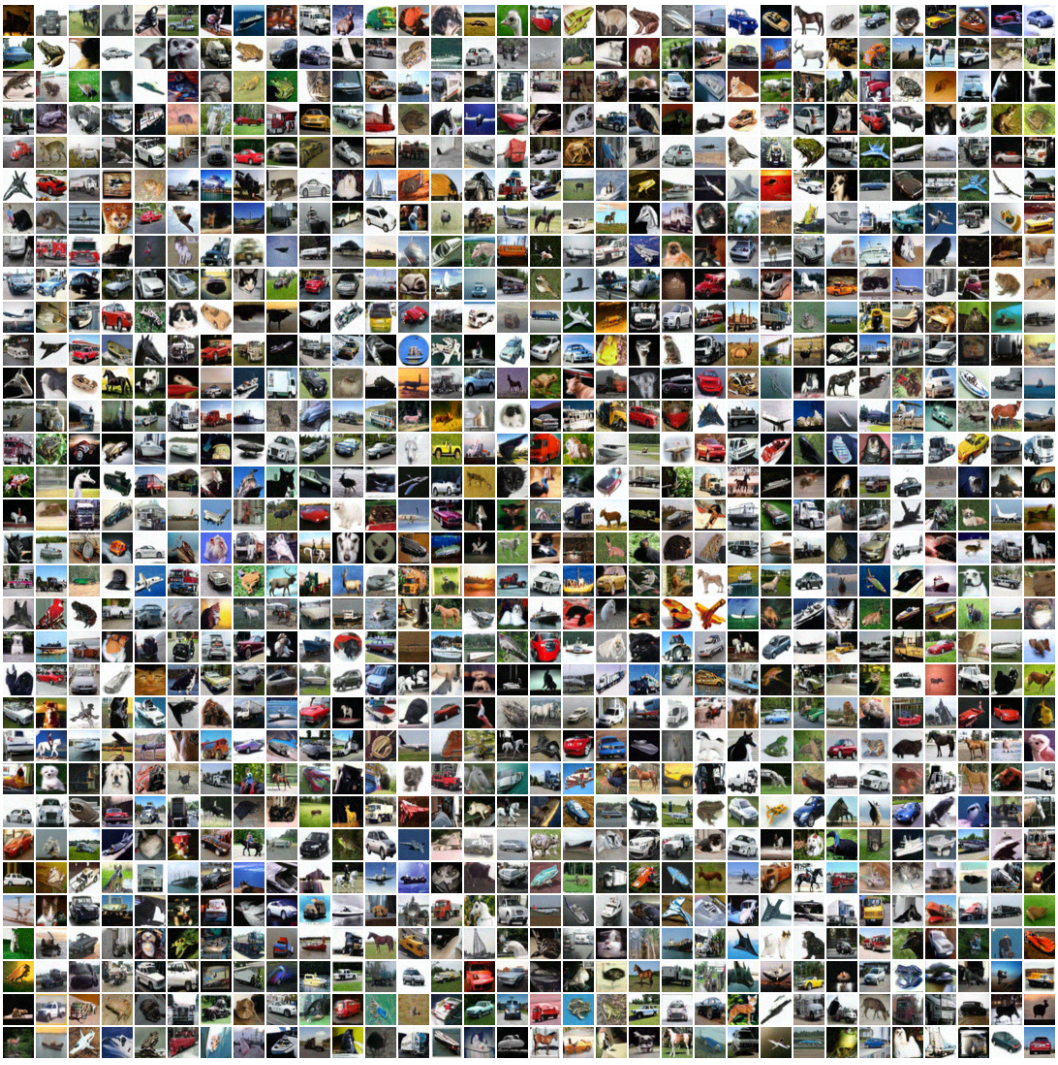

Figure 10: Uncurated samples for *conditional* generation by the one-step RealUID ($\alpha = 1.0, \beta = 1.0$) + GAN ($\lambda_{\text{adv}}^{G_\theta} = 0.3, \lambda_{\text{adv}}^{D} = 1$) trained on CIFAR-10. Quantitative results are reported in Table 2.

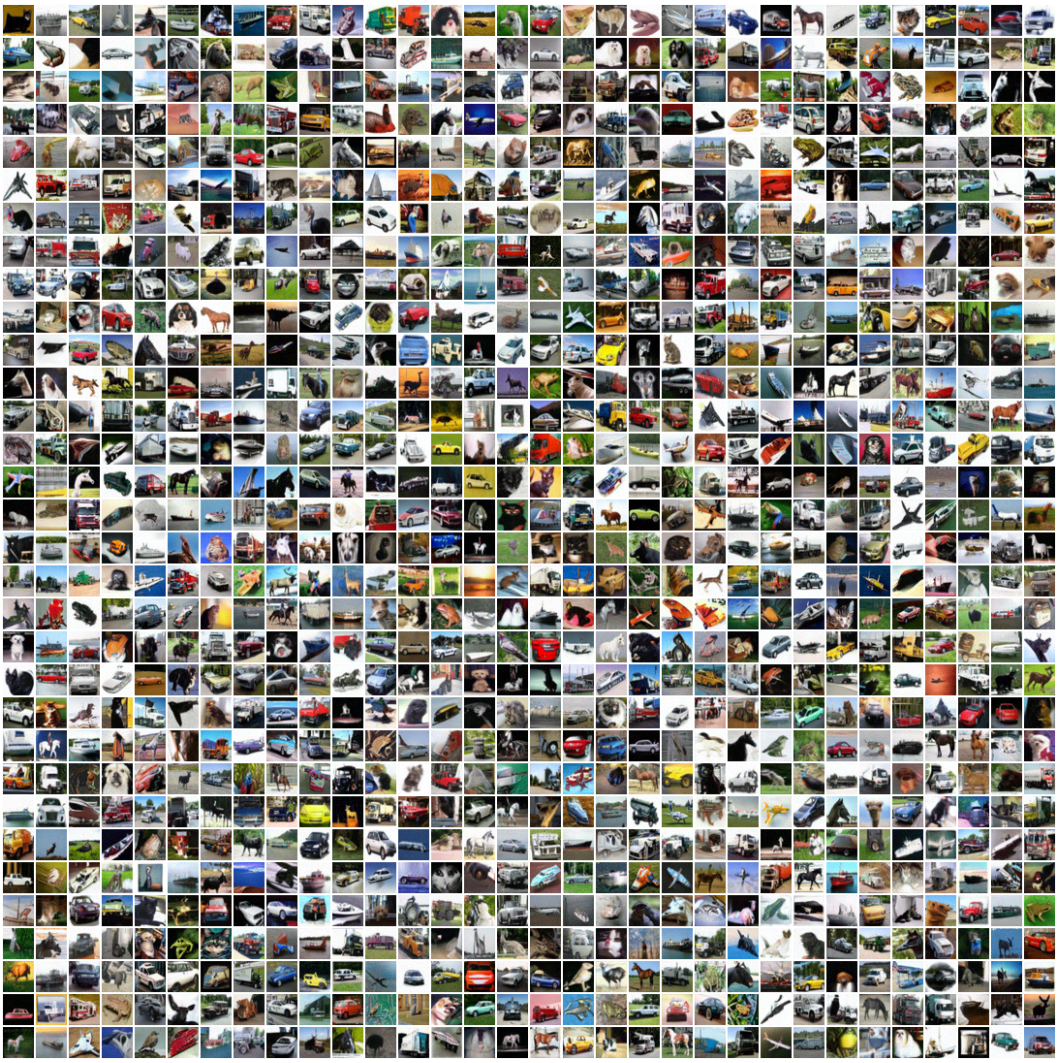

Figure 11: Uncurated samples for *conditional* generation by the one-step RealUID ($\alpha = 0.98, \beta = 0.96$) trained on CIFAR-10. Quantitative results are reported in Table 2.

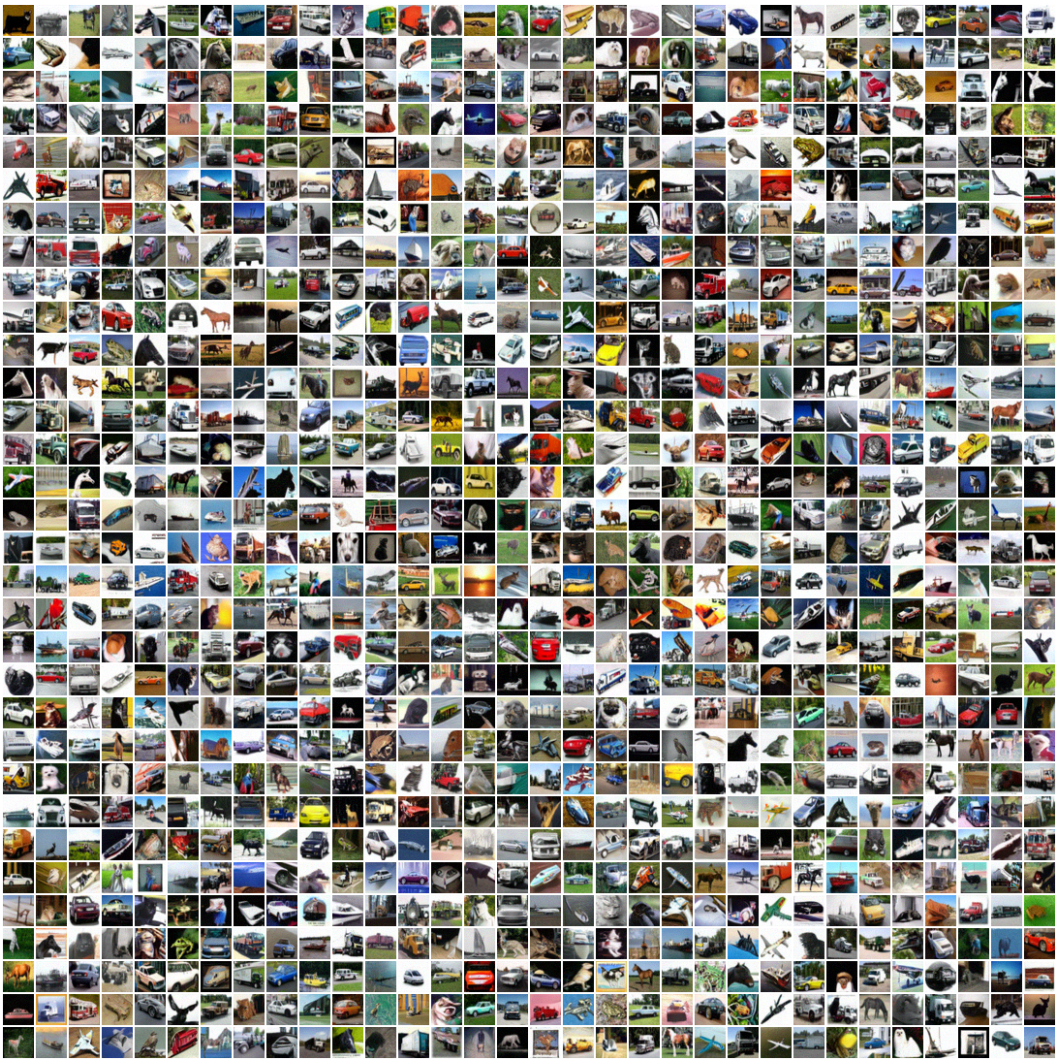

Figure 12: Uncurated samples for *conditional* generation by the one-step RealUID ($\alpha = 0.98, \beta = 0.96 \mid \alpha_{\text{FT}} = 0.94, \beta_{\text{FT}} = 1.0$) trained on CIFAR-10. Quantitative results are reported in Table 2.

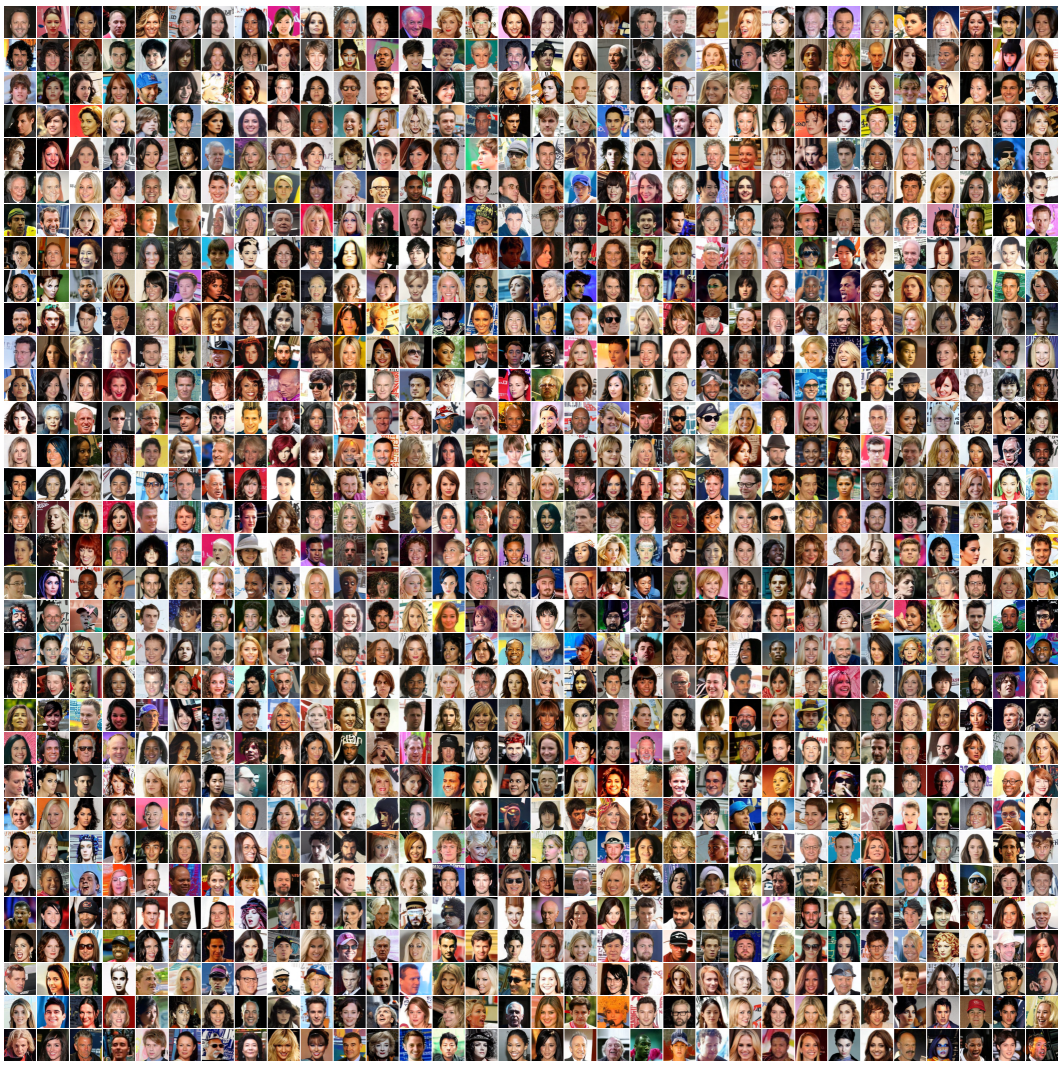

Figure 13: Uncurated samples by the one-step RealUID ($\alpha = 1.0, \beta = 1.0$) trained on CelebA. Quantitative results are reported in Table 7.

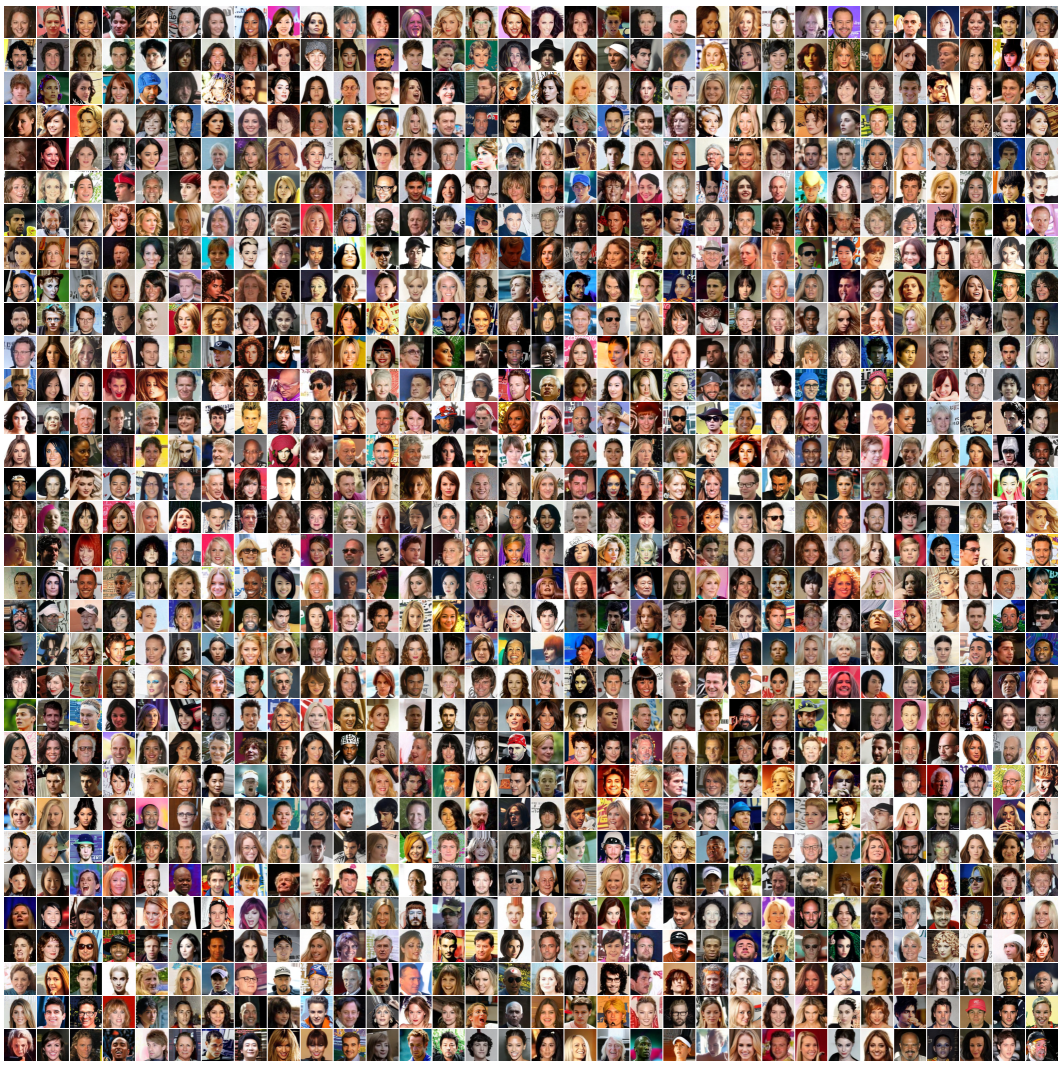

Figure 14: Uncurated samples by the one-step RealUID ($\alpha = 0.96, \beta = 0.94$) trained on CelebA. Quantitative results are reported in Table 7.