Representation Fréchet Loss for Visual Generation

Jiawei Yang[1] **Zhengyang Geng**[2] **Xuan Ju**[3] **Yonglong Tian**[4] **Yue Wang**[1]
[1]USC [2]CMU [3]CUHK [4]OpenAI



Figure 1: **One-step samples on ImageNet 256×256, before and after *FD-loss* post-training. Left**: samples from the base generators. **Right**: samples after post-training with *FD-loss*. **Top two rows**: pMF-H [34], a one-step generator. **Bottom**: JiT-H [27], a multi-step generator. All models generate in a single network evaluation (1-NFE). *FD-loss* improves existing one-step models and can repurpose multi-step models into one-step generators. See uncurated samples in Appendix E.

Abstract

We show that Fréchet Distance (FD), long considered impractical as a training objective, can in fact be effectively optimized in the representation space. Our idea is simple: decouple the *population size* for FD estimation (*e.g.*, 50k) from the *batch size* for gradient computation (*e.g.*, 1024). We term this approach *FD-loss*. Optimizing *FD-loss* reveals several surprising findings. First, post-training a base generator with *FD-loss* in different representation spaces consistently improves visual quality. Under the Inception feature space, a one-step generator achieves *0.72* FID on ImageNet 256×256. Second, the same *FD-loss* repurposes multi-step generators into strong one-step generators *without* teacher distillation, adversarial training or per-sample targets. Third, FID can misrank visual quality: modern representations can yield better samples despite worse Inception FID. This motivates FDr^k, a multi-representation metric. We hope this work will encourage further exploration of distributional distances in diverse representation spaces as both training objectives and evaluation metrics for generative models. Code and checkpoints are available at https://github.com/Jiawei-Yang/FD-loss.

1 Introduction

The Fréchet Inception Distance (FID) [19] has been the *de facto* metric for evaluating image generation, especially on academic benchmarks such as ImageNet [6]. For nearly a decade, the community has been collectively performing "gradient descent" on this single metric to push the state of the art. But this "gradient descent" has always been *indirect*: FID serves only as an evaluator, *not* as a loss. In this paper, we ask: *can FID be optimized directly as a training loss—and what does that reveal?*

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

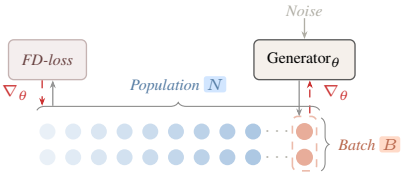

Figure 2: ***FD-loss.*** The generator produces B images per step; their features are accumulated into a large population via a queue or EMA. This decouples the population size N for reliable FD estimation from the batch size B for gradient computation.

In principle, this is straightforward: every term in Fréchet Distance (FD) is differentiable and nothing in its definition restricts it to evaluation. In practice, however, this has been widely considered impractical. Computing FID requires a large population of samples (*e.g.*, 50k) to estimate distributional statistics. Using it as a training loss would further demand gradients through all of them at every step, which is prohibitive. Prior work has therefore only explored FD as a loss under restricted settings, *e.g.*, estimating FD from a small batch [38, 9]. Our experiments confirm that optimizing batch-wise FD degrades base generators (Tab. 1a).

In this work, we show that FD can in fact be optimized directly *at scale*. Our idea is simple (Fig. 2): *decouple* the *population size* used for FD estimation from the *batch size* used for gradient computation. We call this approach *FD-loss* and realize it in two ways. The first maintains an online queue of features from recently generated samples and computes FD over the full queue (*e.g.*, 50k) while back-propagating only through the current batch. The second maintains exponential moving averages (EMA) of the first and second moments of features and restricts gradients to the current batch. Both work well in practice. This one approach unlocks several *surprising* findings.

First, *FD-loss* is a strong post-training objective for visual generators. We fine-tune a pre-trained generator with *FD-loss* using only pre-computed feature statistics—means and covariances of real data under one or more representation backbones.[1] Across pixel-space and latent-space generator families [34, 13, 27], model sizes, and image resolutions, this consistently improves visual quality. Under Inception [51], *FD-loss* drives a *one-step* generator [34] to an FID of 0.72 on ImageNet 256×256 (Tab. 4). The same recipe, applied under modern representations [31, 17, 42, 44, 52], yields even stronger perceptual quality (Fig. 4, Tab. 3).

Second, *FD-loss* serves as a simple distribution-matching objective. We show that a pre-trained *multi-step* generator can be repurposed into a *one-step* generator by post-training it with *FD-loss* (from around 300 FID to 0.72). This repurposing works without teacher distillation, adversarial training, or per-sample targets, suggesting that the representation-space FD is a useful distribution-level objective rather than only an evaluation metric. We validate this on both class-conditioned models (Fig. 1, bottom) and text-conditioned models (Fig. 7).

Third, *FD-loss* provides a diagnostic lens. Having models trained by *FD-loss* with different representations, we can test whether the best-FID model is also the perceptually best. Often, it is *not*: models optimized with modern representations achieve better visual quality while scoring *worse* under FID (Tab. 1c, Fig. 4). More broadly, this connects to a paradox in the field (Fig. 3): state-of-the-art generators already surpass *real* validation images under FID, yet their outputs remain clearly distinguishable from real images. We therefore report FDr^k, a normalized FD ratio averaged over k feature spaces (Eq. 8), as a more representation-diverse automatic metric. As shown in Figure 3, FID suggests the "ImageNet generation" problem is nearly solved, yet FDr^k reveals that significant quality gaps remain. Post-training with *FD-loss* narrows this gap, achieving an FDr^6 of 1.89.

Taken together, our findings reposition FD in generative modeling. FD has long lived on one side of the generative modeling pipeline, *i.e.*, *evaluation*. We show it is also useful on the other side, *i.e.*, *training*. The two sides turn out to be linked: making FD as a loss both improves generators and exposes the limits of any single FD as an evaluator. We hope the simplicity of *FD-loss* will encourage the community to rethink both how we train and how we evaluate generative models.

2 Related Work

Fréchet Distance as an evaluation metric. FID [19] is the dominant metric for evaluating image generation. A growing body of work has highlighted its limitations [46, 25, 24, 50, 22], motivating alternatives such as precision–recall [46, 25], CKA [57], and MMD [22]. We make FD directly optimizable as a training loss, which both probes the reliability of FID and motivates our FDr^k metric.

[1] Real data is used only once, offline, to compute these reference statistics. No real images are seen during post-training.

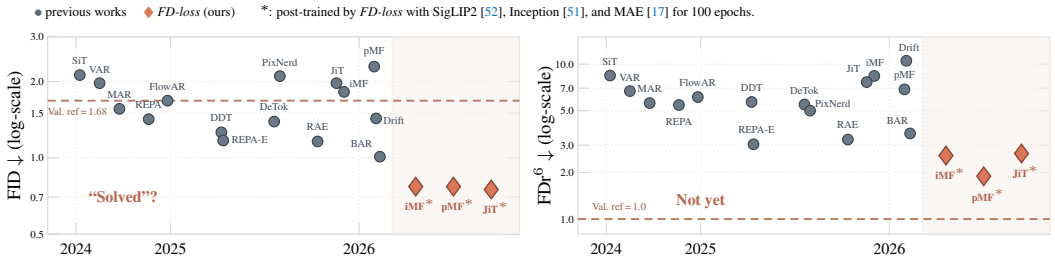

Figure 3: **Is ImageNet generation "solved"? Left**: FID over time. Recent methods surpass the *real* validation set images (red dashed) in terms of FID. **Right**: FDr^6 (Eq. 8), which averages normalized Fréchet Distance ratios across six representation spaces. Under this metric, even the strongest existing methods remain far from the validation images, indicating that FID alone masks significant quality gaps. Each method is shown at its largest publicly available model size. Post-training with *FD-loss* (♦) improves both metrics; see Table 4 for numerical results.

Fréchet Distance as a training loss. Distributional distances as training objectives date back to adversarial learning [15], MMD-based generators [29, 2, 65, 7], and sliced Wasserstein objectives [8]. More closely related, several works train generators by matching feature-space moments [40, 48], or by minimizing FD in Inception [38] or discriminator [9] feature space. The main limitation of existing FD optimization work is statistical scale: their FD estimates are computed within a single batch and become too noisy to scale up. Our work makes FD optimization practical *at scale*.

Optimizing over broader sample windows. A recurring difficulty in modern deep learning optimization is that some objectives require a much larger effective sample set than a single batch provides. In contrastive learning, this motivates memory banks [56] and feature queues [18]. Deep networks have also long used exponential moving averages (EMA) to maintain stable estimates of population statistics, *e.g.*, Batch Normalization [21]. Our work adopts a similar principle: we compute FD over a queue of recent features or EMA estimates of feature moments, while back-propagating only through the current batch.

One-step generators and post-training. Modern high-quality image generators often rely on multi-step denoising [43, 10, 28, 64, 39]. This has motivated more efficient one-step or few-step generators, including straightening ODEs [30], consistency models [49, 33], score distillation [35, 36, 61, 66], identity-based methods [12, 13, 34], and drifting models [7]. Our work is complementary: optimizing FD in capable representation spaces is a powerful and new way to improve existing one-step generators [34, 13] and to repurpose multi-step generators [27, 10] into one-step ones, in both pixel and latent space, *without* denoising teacher distillation or adversarial training. This positions FD not only as an evaluation tool, but also as a practical post-training objective.

3 Method

We study using Fréchet Distance (FD) as a training objective for post-training image generators. We begin by outlining the challenges of directly optimizing FD, and then present our method for making this practical at scale.

3.1 Preliminaries: Fréchet Distance

Let $\phi(\cdot)$ denote a feature extractor. Given real images $\mathcal{R} = \{\mathbf{x}_i\}$ and generated images $\mathcal{G} = \{\hat{\mathbf{x}}_i\}$, their feature distributions are modeled as multivariate Gaussians with means and covariances:

$$\boldsymbol{\mu}_r = \mathbb{E}[\phi(\mathbf{x})], \quad \boldsymbol{\Sigma}_r = \text{Cov}[\phi(\mathbf{x})], \qquad \boldsymbol{\mu}_g = \mathbb{E}[\phi(\hat{\mathbf{x}})], \quad \boldsymbol{\Sigma}_g = \text{Cov}[\phi(\hat{\mathbf{x}})]. \tag{1}$$

The FD between the two Gaussian distributions is:

$$\text{FD}_\phi(\mathcal{R}, \mathcal{G}) = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|_2^2 + \text{Tr}\Big(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{\frac{1}{2}}\Big). \tag{2}$$

When ϕ is Inception-v3 [51], this becomes the Fréchet Inception Distance (FID) [19].

In standard evaluation, $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ are pre-computed once from the training set, while $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ are estimated from a large population of generated samples, typically on the order of tens of thousands.

Challenges. Unlike sample-wise losses, FD is a *distributional* quantity. Directly optimizing FD is difficult because the population size needed for reliable estimation (*e.g.*, 50k) far exceeds a typical training batch (*e.g.*, 64 to 1024). Small-batch estimates are unstable, especially for high-dimensional features. For example, reliably estimating a full-rank covariance matrix ($\mathbf{\Sigma}_g \in \mathbb{R}^{2048\times2048}$) for Inception [51] requires far more than 2048 samples. At the same time, back-propagating through a full evaluation-sized population at every training step is computationally infeasible for most practical setups.[2] The central problem is therefore to estimate FD using a large effective population while keeping the cost of optimization at the scale of an affordable training batch.

3.2 *FD-loss*: Decoupling Population Scale from Optimization Scale

Our key idea is decoupling: we estimate FD using statistics aggregated over a much broader sample window than the current batch, while computing gradients with respect to the current batch alone (Fig. 2). We consider two implementations, introduced below and summarized in Algorithm 1.

Queue-based estimator. Let N denote the queue size, which determines the effective population used for statistics estimation (*e.g.*, $N = 100,000$). At each training iteration, the generator produces a batch of B images, where $B \ll N$ (*e.g.*, $B = 1024$). We extract their features using a representation model ϕ and enqueue them, while removing the oldest B features. FD is computed using the empirical mean and covariance of the full queue. During back-propagation, only the current batch features carry gradients; queued features from previous iterations are treated as constants. This is similar in spirit to the queue in MoCo [18]. The dynamically updated queue makes the feature statistics *on-policy*. This implementation already enables us to achieve **0.89** FID with 50 epochs post-training using a 118M-sized model on ImageNet 256×256 (Tab. 1a).

Algorithm 1 Post-Training with *FD-loss*.

```
# G: generator
# phi: frozen representation model
# (mu_r, sig_r): real feature statistics
# (mu_ema, M_ema): EMA mean and 2nd moment
# beta: EMA decay
# z: current batch of noise

x = G(z)
feat = phi(x)
feat = all_gather(feat) # gather across devices

# Queue version:
# gen_feats = cat([queue.detach(), feat])
# mu_g, sig_g = compute_stats(gen_feats)

# EMA version:
mu_b, M_b = batch_moments(feat)
mu_g = beta * mu_ema.detach() + (1 - beta) * mu_b
M_g = beta * M_ema.detach() + (1 - beta) * M_b
sig_g = M_g - mu_g @ mu_g.T

loss = FD((mu_g, sig_g), (mu_r, sig_r))
loss.backward()
optimizer.step()

# Queue version:
# queue.enqueue_and_dequeue(feat.detach())

# EMA version
mu_ema = mu_g.detach()
M_ema = M_g.detach()
```

EMA-based estimator. We also consider an approach that avoids storing feature queues entirely. Concretely, we maintain exponential moving averages (EMA) of the first and second feature moments. Let $\beta \in (0, 1)$ denote the EMA decay rate, and $\boldsymbol{\mu}_g^{(t)}$ and $\mathbf{M}_g^{(t)}$ denote the running estimates of the first and second moment at iteration t. Given the features of images generated in the current batch $\{\phi(\hat{\mathbf{x}}_i)\}_{i=1}^{B}$, we define the batch moments as:

$$\boldsymbol{\mu}_{\text{batch}}^{(t)} = \frac{1}{B}\sum_{i=1}^{B}\phi(\hat{\mathbf{x}}_i), \qquad \mathbf{M}_{\text{batch}}^{(t)} = \frac{1}{B}\sum_{i=1}^{B}\phi(\hat{\mathbf{x}}_i)\,\phi(\hat{\mathbf{x}}_i)^\top, \tag{3}$$

and update the running estimates as:

$$\boldsymbol{\mu}_g^{(t)} = \beta\boldsymbol{\mu}_g^{(t-1)} + (1-\beta)\boldsymbol{\mu}_{\text{batch}}^{(t)}, \qquad \mathbf{M}_g^{(t)} = \beta\mathbf{M}_g^{(t-1)} + (1-\beta)\mathbf{M}_{\text{batch}}^{(t)}, \tag{4}$$

and recover the covariance from

$$\mathbf{\Sigma}_g^{(t)} = \mathbf{M}_g^{(t)} - \boldsymbol{\mu}_g^{(t)}\boldsymbol{\mu}_g^{(t)\top}. \tag{5}$$

FD is then computed using $(\boldsymbol{\mu}_g^{(t)}, \mathbf{\Sigma}_g^{(t)})$ via Equation 2, with gradients back-propagated only through the current batch. Unlike the queue, this variant stores *no* feature buffer, making it more scalable when using multiple representation models. It also provides a more on-policy estimate since EMA naturally upweights recent samples. This enables the base model to achieve **0.81** FID under the same post-training procedure (Tab. 1b).

[2] Admittedly, this challenge would likely vanish if one could use a batch size of 100,000, which is rarely practical but not entirely impossible.

Discussion. Both variants implement the same *decoupling* principle (§3.2). The queue uses an explicit window controlled by N; the EMA uses a smoothed estimate controlled by β. Both work well in practice; each trades off population size against how on-policy the statistics remain.

Multi-representation *FD-loss*. Our *FD-loss* naturally supports minimizing FD measured in different representation spaces. In practice, FD can vary by orders of magnitude across representation models. To combine losses from multiple representations $\{\phi_i\}$, we normalize each term:

$$\mathcal{L} = \textstyle\sum_i w_i \cdot \mathcal{L}_{\phi_i}, \qquad \mathcal{L}_{\phi_i} = \dfrac{\mathrm{FD}_{\phi_i}(\mathcal{R},\, \mathcal{G})}{\mathtt{sg}(\mathrm{FD}_{\phi_i}(\mathcal{R},\, \mathcal{G})) + c}, \tag{6}$$

where $\mathtt{sg}(\cdot)$ denotes stop-gradient, c is a small constant for numerical stability, and w_i are per-representation weights. This makes each term unit-scale regardless of the feature space. For simplicity, we also do normalization even when there is only one representation model. In this work, we simply use equal weights with $w_i = 1$.

3.3 Training setup

We apply our *FD-loss* for post-training. In all cases, we start from a pre-trained generator, called the base model, and fine-tune it with the *FD-loss*.

Post-training one-step generators. Our primary setting is post-training existing one-step generators with *FD-loss*. In practice, it is desirable to improve sample quality while preserving fast generation. We therefore study both pixel-space one-step generators, such as pixel-MeanFlow (pMF) [34], and latent-space one-step generators, such as improved-MeanFlow (iMF) [13], to demonstrate the versatility of our method.

Repurposing multi-step generators. Surprisingly, we find that the *FD-loss*, when measured in capable representation spaces, can repurpose a pre-trained multi-step generator into a one-step generator. Concretely, given Gaussian noise z, we run the model only once at the terminal timestep and interpret its output as a one-step prediction of the clean image. For example, $\hat{x}_0 = z - v_\theta(z,\ t{=}1)$ for a velocity-prediction model (*e.g.*, SiT [37] and MMDiT [10]), and $\hat{x}_0 = x_\theta(z,\ t{=}1)$ for an x_0-prediction model (*e.g.*, JiT [27]), assuming $z_{t=1}$ is pure Gaussian noise. A multi-step model used this way initially produces poor samples (*e.g.*, 290 FID), since it was never trained for one-step generation. We simply treat it as a one-step generator and optimize it with the *FD-loss*. This procedure can transform multi-step generators into competitive one-step generators, without adversarial training or teacher distillation. In this sense, *FD-loss* provides a minimalistic distribution matching objective.

3.4 FDr^k: Normalized Fréchet Distance Ratios under K Models

Metric paradox. If measured only by FID [19], state-of-the-art image generators appear to have already surpassed real validation images. As shown in Figure 3, when evaluated against ImageNet training images, validation images themselves obtain FID 1.68,[3] while many recent works report FID around or below 1.5 [62, 63, 64, 28, 58, 54, 59]. Yet, generated images are still clearly distinguishable from real ones. This disconnect between metric and visual quality suggests that FID, which relies on a single, dated feature space, has saturated as a quality signal.

FD ratio. A natural remedy is to evaluate across diverse feature spaces. However, raw FD values are not comparable across representations (*e.g.*, FD-MAE and FD-DINOv2 differ by orders of magnitude; see Tab. F.7). To obtain comparable quantities, we normalize the FD of generated images by the FD of *real validation images* in the same feature space. Let $\mathcal{T}$ denote the ImageNet training set, $\mathcal{V}$ the validation set, and $\mathcal{G}$ a set of generated images. For a representation model ϕ_i, we define the *normalized FD ratio*, abbreviated as FDr:

$$\mathrm{FDr}_{\phi_i}(\mathcal{G}) = \frac{\mathrm{FD}_{\phi_i}(\mathcal{G}, \mathcal{T})}{\mathrm{FD}_{\phi_i}(\mathcal{V}, \mathcal{T})}. \tag{7}$$

This ratio is unitless and has a direct interpretation. For example, $\mathrm{FDr}_{\phi_i} = 2.0$ means that under ϕ_i, the generated images are perceptually twice as far from the training set as the validation images. By definition, validation images score exactly 1.0.

[3] For ImageNet, FID is computed against *training* set statistics. Therefore, the FID of validation images is not zero.

FDr^K. We define FDr^k by averaging these normalized ratios over K representation models:

$$\mathrm{FDr}^K(\mathcal{G}) = \frac{1}{K}\sum_{i=1}^{K}\mathrm{FDr}_{\phi_i}(\mathcal{G}). \tag{8}$$

This yields a single multi-representation metric while preserving the interpretation of the per-model ratios. Existing strong generators may beat validation images in Inception [51] feature space alone, but remain substantially inferior once evaluated across diverse representations (Tabs. 4, F.6, F.7).

In this paper, we instantiate FDr^k with 6 representative models spanning supervised, self-supervised, and vision-language objectives across both CNN and ViT architectures: Inception-v3 [51], ConvNeXt-v2 [31], DINOv2 [42], MAE [17], SigLIP2 [52], and CLIP [44]. Details are in Appendix B.1.

Scope. FDr^6 should *not* be viewed as a north star, nor as a replacement for human evaluation, but rather as a more robust *automatic* metric than FID alone. It retains the simplicity of Fréchet-style evaluation while reducing the blind spots of any single representation. Like any automatic metric, FDr^k has its own limitations: its value depends on the choice of K representations, and it still inherits the Gaussian moment-matching assumption of Fréchet Distance itself. We view it as a step forward from FID, not a final answer. We provide more discussions in Appendix A.

4 Experiments

Setup. We study class-conditional image generation on ImageNet-1k [6] at 256×256 and 512×512 resolutions. Our experiments cover both pixel-space generators (pMF [34], JiT [27]) and latent-space generators (iMF [13]). All methods are reimplemented and integrated in a unified codebase for fair comparison; all models are initialized from officially released pre-trained weights. We evaluate using FID [19] and IS [47] to facilitate comparison with prior work. We also report FDr^6 as defined in Equation 8. Following standard practice, all metrics are computed from 50,000 generated images against *training* set statistics. We also report metrics for 50,000 *validation* images as a reference.

Training. We post-train with a global batch size of 1024 using AdamW [32] with a cosine lr schedule and 5 epochs of warm-up. We set $lr{=}10^{-6}$ for pMF [34] and iMF [13], and $lr{=}10^{-5}$ for JiT [27]. For ablation experiments, we post-train for 50 epochs; for system-level results, 100 epochs.

4.1 Properties of Population Size in *FD-loss*

We first analyze the importance of population size when optimizing *FD-loss*. Unless stated otherwise, we post-train pMF-B/16 [34] for 50 epochs using Inception [51] as the only representation model.

Population size via queue size. Table 1a studies the effect of the queue size N. Without a queue ($N{=}0$), statistics are estimated from the current batch only, which degrades all metrics relative to the base model (FID: 3.31→3.84, FDr^6: 13.70→17.06). Performance improves steadily when using queue size from 5k to 50k (FID 0.89, FDr^6 10.91), but degrades beyond 100k as cached features become increasingly off-policy, and the stale statistics outweigh the benefit of a larger population. Notably, at 500k the queue becomes overly stale that FID and FDr^6 *disagree*: FID still improves over the base model (1.22 *vs*. 3.31), whereas FDr^6 degrades beyond the base model (17.67 *vs*. 13.70). This is an early sign that FID alone can be misleading.

Population size via EMA decay rate. Table 1b studies the EMA estimator, where the decay rate β implicitly controls the effective population size. The estimator is robust across a wide range ($\beta{=}0.9$ to 0.9999). The best setting, $\beta{=}0.999$, achieves 0.81 FID and 10.81 FDr^6, improving the best queue result while requiring negligible extra memory. We use $\beta{=}0.999$ in all subsequent experiments.

Both studies confirm that *FD-loss* needs a population larger than the optimization batch, but not so large that staleness dominates. We default to EMA for its simplicity and stronger results.

4.2 Properties of Representation Models in *FD-loss*

Single representation model. Table 1c studies *FD-loss* under different representations; Figure 4 shows qualitative comparisons. As expected, each model scores best in the representation space it

queue size	FID↓	IS↑	FDr^6 ↓
Base	3.31	254.6	13.70
0k‡	3.84	250.9	17.06
5k	1.05	280.0	11.89
10k	0.93	283.9	11.71
50k	**0.89**	288.3	**10.91**
100k	0.93	288.8	11.15
500k	1.22	**294.4**	17.67

(a) **Queue size**. A large queue is critical for stable FD optimization (*e.g.*, 5–100k), while an overly large queue becomes too stale to remain on-policy.

β	FID↓	IS↑	FDr^6 ↓
Base	3.31	254.6	13.70
0.0‡	3.84	250.9	17.06
0.9	0.98	283.6	11.19
0.99	0.84	291.8	**10.74**
0.999	**0.81**	**294.5**	10.81
0.9999	0.98	287.7	11.63

(b) **EMA decay rate**. The EMA estimator is stable across a wide range (β=0.9 to 0.9999) and works best at β=0.999.

	FDr (Eq. 7) ↓								
loss	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	FID↓	IS↑	FDr-CLIP†↓	FDr^6 ↓
Validation set images	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*	*1.68*	*232.2*	*1.00*	*1.00*
Base	1.98	1.93	10.13	13.81	31.03	3.31	254.6	23.30	13.70
FD-Inception	**0.48**	1.26	7.52	8.51	26.02	**0.81**	294.5	21.07	10.81
FD-ConvNeXt	0.98	**0.34**	4.93	7.48	17.38	1.64	281.0	19.66	8.46
FD-DINOv2	2.91	2.14	**2.11**	10.88	16.92	4.89	347.1	15.83	8.47
FD-MAE	3.83	1.92	5.30	**1.11**	14.44	6.42	344.0	13.19	6.63
FD-SigLIP	4.60	2.69	4.27	9.09	**3.61**	7.71	**399.4**	10.84	5.85
FD-SigLIP+Incep.	0.53	0.95	4.99	8.66	6.83	0.89	307.5	13.75	5.95
FD-SigLIP+Incep.+MAE (**SIM**)	0.56	0.85	4.65	2.36	6.94	0.94	307.8	**9.81**	**4.20**

(c) **Representation model**. Each row optimizes *FD-loss* in one or more representation spaces; each column evaluates under a different representation. Setting: EMA with $\beta = 0.999$.

Table 1: **Properties of *FD-loss*.** Ablation results on pMF-B/16 [34] post-trained for 50 epochs. (a, b) study population size; (c) studies representation model choice. Base: base model *before* post-training. ‡Statistics estimated from the current batch only (batch size 1024). †FDr-CLIP is reported separately because CLIP is never used as a training signal in any row shown here; it is included in FDr^6. Default setting is marked in Gray.

optimizes (on-diagonal). However, the off-diagonal behavior differs across model families. Optimizing Inception [51] gives the best FID (3.31→0.81) and improves FDr^6 slightly (13.70→10.81). Optimizing modern ViTs, *e.g.*, DINOv2, MAE, SigLIP2, *worsens* FID but *improves* FDr^6 more substantially. ConvNeXt [31], a modern CNN, sits in between: it improves FID less than Inception (3.31→1.64) but FDr^6 more (13.70→8.46). This reveals that CNN-based representations tend to improve FID, while ViT-based ones improve the broader FDr^6 more.

These results already indicate that lower FID and better overall quality are *not* always the same objective. Improvements captured by modern feature spaces can be invisible, or even unfavorable, under Inception. Moreover, under Inception [51] and ConvNeXt [31], *FD-loss* can even drive FDr below 1.0; in other words, the generated images become statistically closer to the training set than real validation images, suggesting that some feature spaces are easier to saturate than others.

Figure 4 further supports this. Post-trained models improve visually over the base model. Yet, the lowest-FID model (Inception, 0.81) does not produce the best samples; instead, models post-trained with modern representations show better object structure despite much higher FID.

Multiple representation models. Table 1c further studies combinations of representations (Eq. 6). Combining representations is generally more effective than using a single one. Optimizing SigLIP with Inception recovers FID to 0.89 while maintaining strong FDr^6. Adding MAE (denoted FD-SIM) further improves FDr^6 with a negligible FID trade-off. We use FD-SIM as the default from now on.

4.3 Repurposing Multi-Step Generators into One-Step Generators

We study whether *FD-loss* can repurpose a pre-trained multi-step model into a one-step generator. We use JiT-L/16 [27] as the base model and post-train it for 50 epochs following Section 3.3. Table 2 and Figure 5 report the results. As expected, the base model fails in the naive one-step setting, since it is trained for multi-step denoising. After post-training with *FD-loss*, all variants produce sensible images. FD-SIM achieves the best FDr^6 (3.29) with FID 0.85, while FD-Inception gives the best FID (0.77) but a much higher FDr^6 (12.86). Notably, repurposing is more demanding than improving an already strong one-step model: capable representations such as MAE, SigLIP2, and their combinations all yield visually compelling samples, whereas Inception alone might not be strong

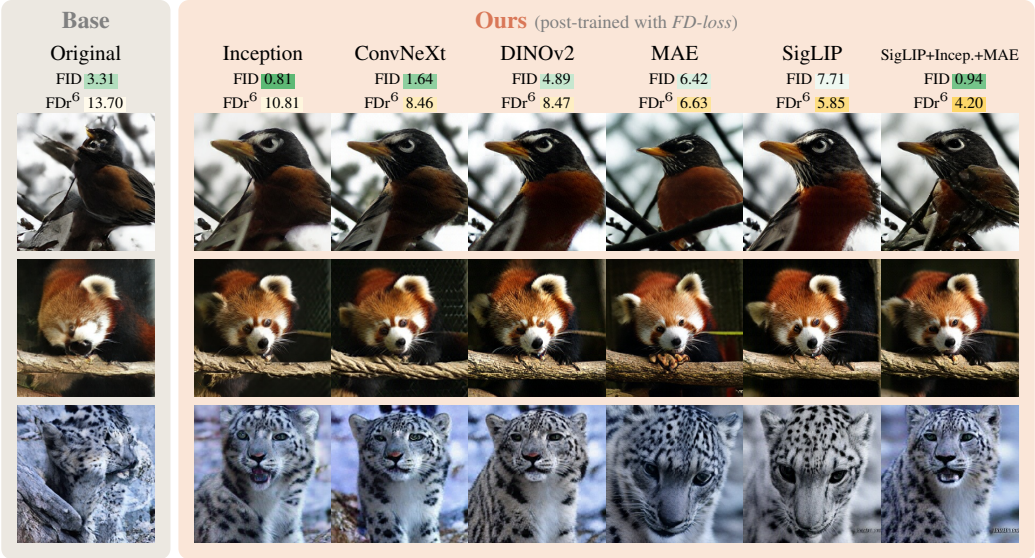

Figure 4: ***FD-loss* improves visual quality under different representations.** Samples from pMF-B/16 [34] post-trained with *FD-loss*. Darker green: lower FID; darker yellow: lower FDr^6. Post-trained models improve over the base model (left). Inception post-trained model achieves the lowest FID (0.81) yet does not produce the best samples; models post-trained with modern representations achieve lower FDr^6 and show better object structure despite higher FID.

Table 2: ***FD-loss* repurposes multi-step JiT models to generate in one step.** All post-trained models use 1 NFE. Setting: JiT-L/16 [27], post-trained for 50 epochs. †200 NFE = 50 steps × 2 (Heun) × 2 (CFG).

setting	NFE	FID↓	IS↑	FDr^6 ↓
base model				
JiT-L (50-step)	200†	2.59	288.5	10.73
JiT-L (1-step)	1	291.59	2.0	214.75
models post-trained with FD-loss				
FD-Incep.	1	**0.77**	293.7	12.86
FD-MAE	1	6.52	280.4	9.30
FD-SigLIP	1	5.10	329.6	9.04
FD-SigLIP+MAE	1	4.67	**354.0**	3.83
FD-SigLIP+Incep.+MAE (**SIM**)	1	0.85	319.5	**3.29**

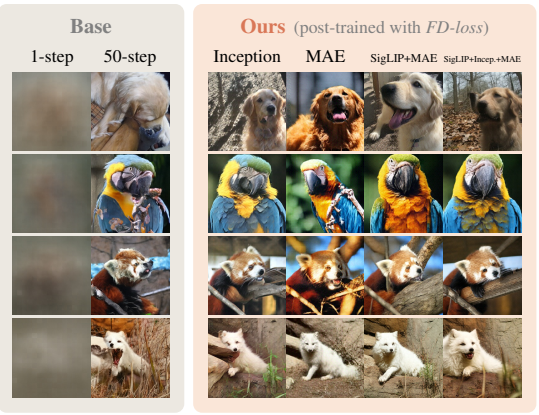

Figure 5: **Repurposing a multi-step model into a one-step generator with *FD-loss*.** Samples from the *same* noise input across the base model and different post-trained models. The naive one-step base model fails to produce sensible images. After post-training, the 1-NFE models generate sensible images, and the strongest variants are visually comparable or superior to the 50-step base model.

enough, so the repurposed models may exhibit certain artifacts. These results indicate that the same *FD-loss* recipe works both for improving one-step generators and for converting multi-step ones into one-step, with no distillation, adversarial loss, or per-sample regression target.

4.4 Comparisons

Scalability. Table 3 reports *FD-loss* post-trained models on three generator families (pMF [34], iMF [13], JiT [27]), each at three model sizes, on ImageNet 256×256, and on pMF at three sizes on ImageNet 512×512. Across all configurations, FD-SIM drives FDr^6 to between 1.81 and 5.56, and FD-Inception drives FID to between 0.72 and 0.79. These gains are obtained with the same hyperparameters. Per-model tuning could yield better results. *FD-loss* thus transfers across pixel and latent spaces, one-step and repurposed multi-step generators, model scales, and image resolutions.

System-level comparison. As a reference, we compare *FD-loss* post-trained models with prior work in Table 4. Since FDr^6 has not been reported before, we re-sample 50k images from official

Table 3: ***FD-loss* generalizes across model families, sizes, and image resolutions.** (a)–(c) cover three generator families on ImageNet at 256px; (d) covers pMF on ImageNet 512px. Each sub-table shows the base generator and its *FD-loss* post-trained variants (shaded). All post-trained models use 1 NFE. SIM: SigLIP+Inception+MAE. Setting: post-trained for 100 epochs, EMA with $\beta = 0.999$.

(a) pMF [34] (pixel, 256px)

method	FID	FDr^6
pMF-B	3.31	13.70
+ Incep.	**0.77**	10.66
+ SIM	0.85	**3.50**
pMF-L	2.72	9.09
+ Incep.	**0.73**	6.19
+ SIM	0.78	**2.09**
pMF-H	2.29	6.87
+ Incep.	**0.72**	4.86
+ SIM	0.77	**1.89**

(b) iMF [13] (latent, 256px)

method	FID	FDr^6
iMF-B	3.45	15.29
+ Incep.	**0.79**	11.34
+ SIM	0.88	**5.56**
iMF-L	1.93	9.06
+ Incep.	**0.75**	6.63
+ SIM	0.79	**2.74**
iMF-XL	1.82	8.39
+ Incep.	**0.72**	6.01
+ SIM	0.76	**2.45**

(c) JiT [27] (pixel, 256px)

method	FID	FDr^6
JiT-B	3.71	15.65
+ Incep.	**0.76**	22.48
+ SIM	1.00	**5.53**
JiT-L	2.59	10.73
+ Incep.	**0.73**	12.75
+ SIM	0.77	**3.24**
JiT-H	1.97	7.66
+ Incep.	**0.72**	10.18
+ SIM	0.75	**2.65**

(d) pMF [34] (pixel, 512px)

method	FID	FDr^6
pMF-B	3.59	15.04
+ SIM	**0.87**	**3.82**
pMF-L	2.56	9.76
+ SIM	**0.80**	**2.12**
pMF-H	2.43	7.33
+ SIM	**0.78**	**1.81**

checkpoints using official code, and re-evaluate *all* methods under the same pipeline. Our *FD-loss* post-trained models push FID below all prior systems and drive FDr^6 substantially lower. More importantly, they achieve this with only a *single* network function evaluation (1 NFE).

Human preference. Since automatic metrics are only proxies for perceptual quality, we further conduct a pairwise human preference study (Fig. 6; Appendix C). The study provides two signals. First, *FD-loss* post-trained models are preferred over their corresponding base models across all three generator families. Second, even the strongest generator tested in this study is still preferred less often than real validation images, corroborating that ImageNet generation is not yet solved (Fig. 3).

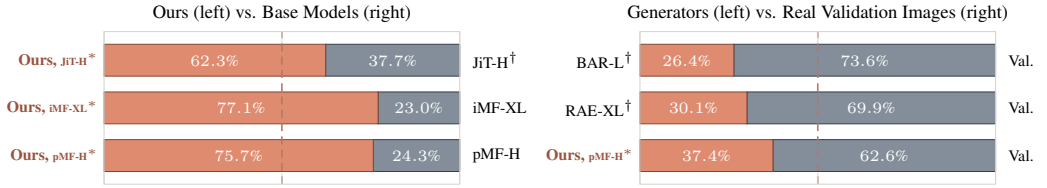

Figure 6: **Human preference study. Left**: Our post-trained 1-NFE models (**warm***) are preferred over their base models. **Right**: Our pMF-H* is the most preferred generator against real ImageNet validation images, but *still* loses to real. This is consistent with Figure 3: ImageNet generation is *not* yet solved. †: multi-step models; all post-trained models use FD-SIM. Protocol in Appendix C.

4.5 Text-Conditioned Generation

Beyond class-conditioned generation models on ImageNet, we repurpose SD3.5 Medium [10], a 2.5B-parameter MMDiT originally trained for multi-step latent-space denoising, into a 1-NFE text-to-image generator with *FD-loss* (Fig. 7). This demonstrates that *FD-loss* can extend beyond class-conditioned settings and scale to large text-conditioned models. A full comparison including a BLIP3o-Pretrain-Long-3M variant is in Figure G.1; training details are in Appendix B.4.



Figure 7: **Qualitative text-conditioned generation example.** We post-train SD3.5 Medium [10] with *FD-loss* using BLIP3o-GPT4o-60k [4], a curated 60k dataset distilled from GPT-4o with a stylized aesthetic as the reference image distribution. Despite 56× NFE reduction, the post-trained 1-NFE model can preserve recognizable prompt content while inheriting the stylized look of the reference distribution. More in Appendix G.

Table 4: **System-level comparison on ImageNet** 256×256**.** All metrics for all methods are computed by us under a unified evaluation pipeline; numbers may differ slightly from the original papers. Our *FD-loss* (shaded rows) improves already strong generators across geneartor families and model scales. †CFG is applied only in a time sub-interval; we report the full-CFG upper bound for simplicity. Uncurated qualitative samples are in Appendix E.

method	NFE	space	#params	FDr^6 ↓	FID↓	IS↑	Prec↑	Recall↑
reference (real images)								
50k validation images	N/A	N/A	N/A	1.00	1.68	232.2	0.75	0.66
discrete-space models								
VAR-d30	10×2	discrete	2B	6.70	1.97	**304.6**	0.82	0.59
BAR-L [62]	256×2×4	discrete	1.1B	**3.57**	**1.01**	281.9	0.77	0.68
latent-space models, multi-step								
without semantic distillation								
SiT-XL/2 [37]	250×2	latent	675M	8.44	2.12	256.7	0.81	0.60
MAR-L [28]	256×2×100	latent	478M	6.68	1.80	293.4	0.80	0.60
FlowAR-H [45]	50×2†	latent	1.9B	6.13	1.68	274.1	0.80	0.62
MAR-H [28]	256×2×100	latent	942M	5.61	1.56	299.5	0.80	0.62
MAR-L, DeTok [58]	256×2×100	latent	478M	**5.49**	**1.39**	**306.2**	0.81	0.62
with semantic distillation								
REG [55]	250×2†	latent	685M	4.64	1.54	302.9	0.78	0.62
SiT-XL/2-REPA [63]	250×2†	latent	675M	5.45	1.42	306.1	0.80	0.65
LightningDiT [59]	250×2	latent	675M	4.57	1.42	294.3	0.80	0.64
DDT-XL [54]	250×2	latent	675M	5.70	1.26	**309.3**	0.79	0.66
REPA-E [26]	250×2†	latent	676M	**3.04**	1.17	298.3	0.79	0.66
RAE-XL [64]	50×2†	latent	839M	3.26	**1.16**	261.0	0.77	0.67
latent-space models, one-step								
Drift-L (latent) [7]	1	latent	463M	10.92	1.53	257.2	0.79	0.63
iMF-XL [13]	1	latent	610M	8.39	1.82	278.9	0.78	0.63
iMF-XL [13]	2	latent	610M	7.48	1.61	289.1	0.79	0.63
+ *FD-loss*	1	latent	610M	**2.45**	**0.76**	301.3	0.77	0.67
pixel-space models, multi-step								
PixNerd-XL [53]	100×2	pixel	1.0B	5.01	2.10	**318.8**	0.81	0.59
JiT-L [27]	50×2×2†	pixel	459M	10.73	2.59	288.5	0.79	0.59
+ *FD-loss*	1	pixel	459M	3.24	0.77	317.3	0.77	0.66
JiT-H [27]	50×2×2†	pixel	953M	7.66	1.97	296.0	0.78	0.63
+ *FD-loss*	1	pixel	953M	**2.65**	**0.75**	313.0	0.76	0.66
pixel-space models, one-step								
Drift-L (pixel) [7]	1	pixel	465M	10.51	1.43	305.8	0.81	0.60
pMF-L [34]	1	pixel	410M	9.09	2.72	261.7	0.81	0.56
+ *FD-loss*	1	pixel	410M	2.09	0.78	309.2	0.76	0.67
pMF-H [34]	1	pixel	935M	6.87	2.29	267.2	0.80	0.59
+ *FD-loss*	1	pixel	935M	**1.89**	0.77	**310.1**	0.77	0.68

5 Conclusion

Generation is, at its core, a distributional problem. Over the years, however, the training of generative models has focused primarily on sample-level losses, *e.g.*, diffusion, flow matching, and adversarial objectives, while distributional distances, such as Fréchet Distance, have lived only as evaluators. This separation has been a matter of practicality rather than principle: FD has always been differentiable, yet reliable estimation requires populations far beyond a training batch. Under these perspectives, our findings with *FD-loss* are, in hindsight, a natural outcome: once population size and gradient size are decoupled, a distributional distance can serve directly as a training loss.

We hope our work will make distribution-level post-training broadly applicable: to other modalities, to settings where access to real data during post-training is scarce or restricted, and to paradigms beyond image generation. More generally, once a distributional distance becomes optimizable at scale, a central design question shifts from *how to optimize the distance* to *which representation space should define it*. Our work makes initial attempts by exploring several existing representations. Different representations induce different notions of visual similarity, and no single feature space should be expected to fully capture perceptual quality. We hope this perspective will encourage future work on distribution-level objectives and representation-diverse evaluation for generative models.

Acknowledgments

We are grateful to Tianhong Li, Tianyuan Zhang, Quankai Gao, Songlin Wei, and Rundi Wu for their helpful discussions and suggestions for this project.

The USC Physical Superintelligence Lab acknowledges generous supports from Toyota Research Institute, Dolby, Google DeepMind, Capital One, Nvidia, Bosch, NSF, and Qualcomm. Jiawei Yang is supported by the NVIDIA Graduate Fellowship. Yue Wang is also supported by a Powell Research Award.

Competing interests. The authors declare no competing interests.

References

[1] Anthropic. Claude code. https://www.anthropic.com/claude-code, 2025.

[2] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

[3] Black Forest Labs. FLUX.1-dev. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024.

[4] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv:2505.09568*, 2025.

[5] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

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

[7] Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv:2602.04770*, 2026.

[8] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced wasserstein distance. In *CVPR*, 2018.

[9] Khoa D Doan, Saurav Manchanda, Fengjiao Wang, Sathiya Keerthi, Avradeep Bhowmik, and Chandan K Reddy. Image generation via minimizing fréchet distance in discriminator feature space. *arXiv:2003.11774*, 2020.

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[11] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *ICML*, 2023.

[12] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *NeurIPS*, 2025.

[13] Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fast-forward generative models. In *CVPR*, 2026.

[14] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023.

[15] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[16] Charles AE Goodhart. Problems of monetary management: the uk experience. In *Monetary theory and practice: The UK experience*, 1984.

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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

[20] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv:2403.05135*, 2024.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[22] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *CVPR*, 2024.

[23] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *NeurIPS*, 2023.

[24] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of ImageNet classes in fréchet inception distance. In *ICLR*, 2023.

[25] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

[26] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *ICCV*, 2025.

[27] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *CVPR*, 2026.

[28] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024.

[29] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, 2015.

[30] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[31] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

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

[33] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *ICLR*, 2025.

[34] Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv:2601.22158*, 2026.

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

[36] Weijian Luo, Zemin Huang, Zhengyang Geng, J. Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. In *NeurIPS*, 2024.

[37] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.

[38] Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fréchet inception distance. *arXiv:2009.14075*, 2020.

[39] Sicheng Mo, Thao Nguyen, Richard Zhang, Nick Kolkin, Siddharth Srinivasan Iyer, Eli Shechtman, Krishna Kumar Singh, Yong Jae Lee, Bolei Zhou, and Yuheng Li. Group diffusion: Enhancing image generation by unlocking cross-sample collaboration. In *CVPR*, 2026.

[40] Youssef Mroueh, Tom Sercu, and Vaibhava Goel. Mcgan: Mean and covariance feature matching gan. In *ICML*, 2017.

[41] OpenAI. Codex cli. `https://github.com/openai/codex`, 2025.

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

[43] William Peebles and Saining Xie. Scalable diffusion models with Transformers. In *ICCV*, 2023.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[45] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. In *ICML*, 2025.

[46] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018.

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

[48] Cicero Nogueira dos Santos, Youssef Mroueh, Inkit Padhi, and Pierre Dognin. Learning implicit generative models by matching perceptual features. In *ICCV*, 2019.

[49] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

[50] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *NeurIPS*, 2023.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[52] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv:2502.14786*, 2025.

[53] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *ICLR*, 2026.

[54] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. In *CVPR*, 2026.

[55] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *NeurIPS*, 2025.

[56] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[57] Ceyuan Yang, Yichi Zhang, Qingyan Bai, Yujun Shen, Bo Dai, et al. Revisiting the evaluation of image synthesis with GANs. In *NeurIPS*, 2023.

[58] Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good visual tokenizers. In *ICLR*, 2026.

[59] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025.

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

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

[62] Qihang Yu, Qihao Liu, Ju He, Xinyang Zhang, Yang Liu, Liang-Chieh Chen, and Xi Chen. Autoregressive image generation with masked bit modeling. *arXiv:2602.09024*, 2026.

[63] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

[64] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026.

[65] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *ICML*, 2025.

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

Appendix for
Representation Fréchet Loss for Visual Generation

Contents

A. Additional Design Attempts and Observations pp. 15–16
B. Implementation and Evaluation Details pp. 16–18
C. Human Preference Study pp. 18–19
D. Limitations and Broader Impact pp. 19–20
E. Additional Qualitative Samples pp. 20–25
F. Detailed Results pp. 25–25
G. Text-to-Image Prompts pp. 30–31
H. Additional Experiments and Analysis pp. 32–34

A Additional Design Attempts and Observations

We study several exploratory settings that shaped the design of *FD-loss*. They expose a recurring limitation: optimizing a narrow representation can improve its score without improving the properties we care about. These observations motivate representation-diverse objectives and evaluation.

Representation-coupled reward hacking. *All* automatic objectives are proxies for the properties we ultimately care about. When the proxy is optimized directly, it can become misaligned with the underlying goal, a phenomenon often discussed through Goodhart's law [16]. A closely related issue appears in reward-model optimization for large language models: reinforcement learning from human feedback methods learn a reward model from human preferences [5], but over-optimizing that learned reward can reduce true preference quality [11]. Technically, *FD-loss* is not an RL algorithm, but it faces an analogous proxy-optimization issue: reducing FD in a chosen feature space can be viewed as optimizing a distribution-level reward supplied by that representation model.

This proxy is useful, but it is incomplete. For example, optimizing Inception FD reliably improves the base generator and can bring pMF-B/16 to 0.81 FID; the resulting model is visually better than the base model. However, it is not necessarily perceptually stronger than some other generators whose FID is around 1.0–1.5. In this regime, continuing to optimize the same narrow representation further improves the chosen score without addressing the representation's blind spots. This is exactly the distinction exposed by FDr^6: Inception-only optimization greatly improves FID, while modern representation spaces reveal remaining quality gaps.

Over-optimizing FID and IS. The same point can be made more visibly by deliberately over-optimizing Inception-based metrics. In early experiments, we found that Inception Score (IS) can also be optimized with a queue-style estimator. As a focused stress test, we post-train pMF-B with a learning rate 10^{-4}, which is $100\times$ larger than our default learning rate for pMF. Figure A.1 shows the resulting samples: the model attains 660 IS and 2.09 FID, yet its FDr^6 rises to 50.66, and the images exhibit clear artifacts and unnatural colors. The chosen metrics improve, but the samples do not. This stress test further illustrates why an Inception-based score alone is insufficient for judging visual quality. During early experimentation, we repeatedly observed similar variants that achieved exceptional FID and IS despite poor visual quality (*e.g.*, FID between 0.9 and 2.1 with IS between 500 and 900), which ultimately motivated us to explore representation-diverse evaluation metrics.

The representation problem remains open. Even with multiple modern representation models, we believe some imperfections in generated images are still not captured by any representation model we tested. In this work, FDr^6 uses six feature spaces spanning supervised, self-supervised, reconstructive, and vision-language objectives. This set is substantially more diverse than FID alone, and it reveals many gaps that Inception misses. Still, six representations are unlikely to exhaust perceptual quality. Every representation discards information, and a metric inherits the blind spots of its feature space. Finding representations that better capture perceptual quality therefore remains an open problem.



Figure A.1: **A stress test of over-optimizing Inception-based metrics.** We deliberately optimize Inception-based scores with a 100× larger learning rate. The resulting model attains 660 IS and 2.09 FID, but its samples exhibit clear artifacts and its FDr^6 degrades to 50.66. This illustrates that Inception-based metrics can be pushed without improving perceptual quality, motivating representation-diverse evaluation and human preference checks.

We view FDr^k as a practical step toward reducing single-model bias, not as a final solution. This is also why we conduct a human preference study. When one-step post-trained models score better than strong multi-step systems such as RAE under FDr^6, we need to check whether they genuinely look better to humans, or whether we have merely found a broader but still hackable automatic metric.

B Implementation and Evaluation Details

B.1 Representation Models for FDr^6

Table B.1 summarizes the representation models used to compute FDr^6. These span supervised, self-supervised, and vision-language training objectives across both CNN and ViT architectures.

Table B.1: **Representation models used in FDr^6.**

model	timm identifier	arch.	dim	objective	pooling
Inception-v3 [51]	`inception_v3` (torch-fidelity)	CNN	2048	supervised	global avg pool
ConvNeXt-v2 [31]	`convnextv2_base.fcmae_ft_in22k_in1k`	CNN	1024	self-supervised	global avg pool
MAE [17]	`vit_large_patch16_224.mae`	ViT	1024	reconstructive	CLS token
DINOv2 [42]	`vit_large_patch14_dinov2.lvd142m`	ViT	1024	contrastive	CLS token
SigLIP2 [52]	`vit_so400m_patch16_siglip_256.v2_webli`	ViT	1152	vision-language	CLS token
CLIP [44]	`vit_large_patch14_clip_224.openai`	ViT	1024	vision-language	CLS token

For CNNs, we extract features after the final spatial pooling layer. For ViTs, we use the final-layer CLS token. We freeze all representation models during training.

B.2 Configurations

Table B.2 summarizes the configurations used for ImageNet class-conditional post-training across the three generator families (pMF, iMF, JiT).

B.3 Training Details

Initialization. We initialize the statistics estimators with images generated *on-the-fly* from the base model. For the EMA variant, we generate 50k images and compute the initial feature moments $(\boldsymbol{\mu}_g^{(0)}, \mathbf{M}_g^{(0)})$ from these samples. For the queue variant, we similarly generate N images (where N is the queue size) and fill the queue with their features. This gives both estimators a meaningful FD estimate from the first training step.

	pMF [34]	iMF [13]	JiT [27]
base model			
space	pixel	latent (SD-VAE)	pixel
sizes (256px)	B, L, H	B, L, XL	B, L, H
sizes (512px)	B, L, H	not used	
patch size	16 (256px), 32 (512px)		
initialization	official pre-trained weights		
representation models for FDr			
FD-Incep.	Inception-v3 [51]		
FD-SIM	SigLIP2 [52] + Inception-v3 [51] + MAE [17]		
input resolution	224 (SigLIP2), 299 (Incep.), 224 (MAE)		
normalization c	0.01		
matrix square root	`torch.linalg.eigvalsh`		
statistics estimator			
default estimator	EMA, β=0.999		
queue size (ablation)	see Table 1a		
warm-start samples	50k generated from base model		
training			
epochs (Tables 1a, 1b, 1c, 2)	50		
epochs (Tables 3, 4)	100		
optimizer	AdamW [32], β_1, β_2=0.9, 0.95		
weight decay	0		
learning rate	1e-6	1e-6	1e-5
lr schedule	cosine		
warmup epochs	5		
global batch size	1024		
precision	bf16		
gradient clipping	none		
dropout	0		
model-weight EMA	none (online weights)		
augmentation	center crop, horizontal flip		
sampling / evaluation			
NFE	1		
CFG at inference	default (CFG-distilled)	default (CFG-distilled)	1.0

Table B.2: **Configurations for ImageNet class-conditional post-training** with *FD-loss*. All settings are shared across ablation and final runs unless noted. Base models are used as released; only post-training differs from the source papers.

Matrix square root. Computing FD (Eq. 2) requires the matrix square root $(\mathbf{\Sigma}_r\mathbf{\Sigma}_g)^{1/2}$. We precompute $\mathbf{\Sigma}_r^{1/2}$ via eigendecomposition and then compute the trace term efficiently using `torch.linalg.eigvalsh` on the symmetric product $\mathbf{\Sigma}_r^{1/2}\mathbf{\Sigma}_g\mathbf{\Sigma}_r^{1/2}$, avoiding an explicit matrix square root at each training step. In early exploration, we compared `torch.linalg.eigvals` and `torch.linalg.eigvalsh` and found them to perform similarly, with the latter being significantly faster; we therefore adopt it.

B.4 Text-to-Image Post-Training

We take the MMDiT transformer from Stable Diffusion 3.5 Medium [10] (2.5B parameters) and post-train it with *FD-loss* for one-step generation at 256×256 resolution. The SD3.5 VAE tokenizer is used unchanged. We use the SIM representation set (SigLIP2+Inception+MAE). Reference statistics $(\boldsymbol{\mu}_r, \mathbf{\Sigma}_r)$ are pre-computed from all real images in each set (3M for variant (i), 60k for variant (ii)). EMA feature statistics are warm-started with 50k images generated from the base model before training begins.

Training uses a cosine learning rate schedule with peak lr=10^{-5} and 2,500 warmup steps, for 15,000 total steps. The global batch size is 1024; each step generates one image per caption with one denoising step (no classifier-free guidance, CFG=1). EMA statistics are tracked with β=0.999 and the eigenvalue-based matrix square root (`eigvalsh`) is used.

We train two variants that differ only in the caption and image sources: **(i)** BLIP3o-Pretrain-Long-3M [4], a 3M subset randomly sampled from the original 30M caption-image web dataset, paired with

realistic photographic images; and **(ii)** BLIP3o-GPT4o-60k [4], a 60k curated set whose images are distilled from GPT-4o and exhibit a stylized, illustration-leaning aesthetic. All other hyperparameters are identical between the two runs.

	SD3.5 Medium [10]
base model	
architecture	MMDiT, 2.5B params
resolution	256×256
tokenizer	SD3.5 VAE
representation models for FDr	
FD-SIM	SigLIP2 + Inception-v3 + MAE
statistics estimator	
estimator	EMA, β=0.999
warm-start	50k samples from base
reference stats $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$	computed from all images in each set (3M / 60k)
training	
total steps	15,000
warmup steps	2,500
optimizer	AdamW, β_1, β_2=0.9, 0.95, wd = 0
peak learning rate	1e-5
lr schedule	cosine
global batch size	1024
precision	bf16
gradient clipping	none
dropout	0
model-weight EMA	none
sampling	
NFE	1
CFG	1.0
caption sets (two variants)	
(i) photographic	BLIP3o-Pretrain-Long-3M [4]
(ii) stylized	BLIP3o-GPT4o-60k [4] (GPT-4o distilled)

Table B.3: **Configurations for text-to-image post-training** of SD3.5 Medium with *FD-loss*. The two variants differ only in the caption source.

B.5 Evaluation Protocol

We sample 50,000 images per method using official checkpoints and code. We primarily use coding agents, including Claude Code [1] and Codex [41], to follow each codebase's sampling instructions. We manually inspect the resulting samples and run the evaluation ourselves. Class-conditional models sample uniformly across all 1,000 classes (50 images per class). Reference statistics $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ are computed once from the full ImageNet training set.

C Human Preference Study

We conduct a pairwise human preference study using anonymized 3×3 image grids. For each trial, voters choose the grid with higher visual fidelity, with an additional tie option. The left–right order is randomized independently for every trial, and model identities are hidden. We evaluate two settings: (i) *Post-trained vs. Base*, where each *FD-loss* post-trained model is compared with its corresponding base model using matched initial noise, so the two grids are directly comparable; and (ii) *Generator vs. Real*, where a generator is compared against ImageNet validation images.

We collect 2,929 valid votes from 17 participants after filtering incomplete or invalid responses. For reporting, ties are split evenly between the two sides: if W, T, and L denote win, tie, and loss rates for *FD-loss*, its preference score is $W + T/2$. Figure 6 reports aggregated preference scores.

Ethics review. Institutional review board (IRB) review was not sought for the human preference studies. We do not claim IRB approval or an exemption determination.

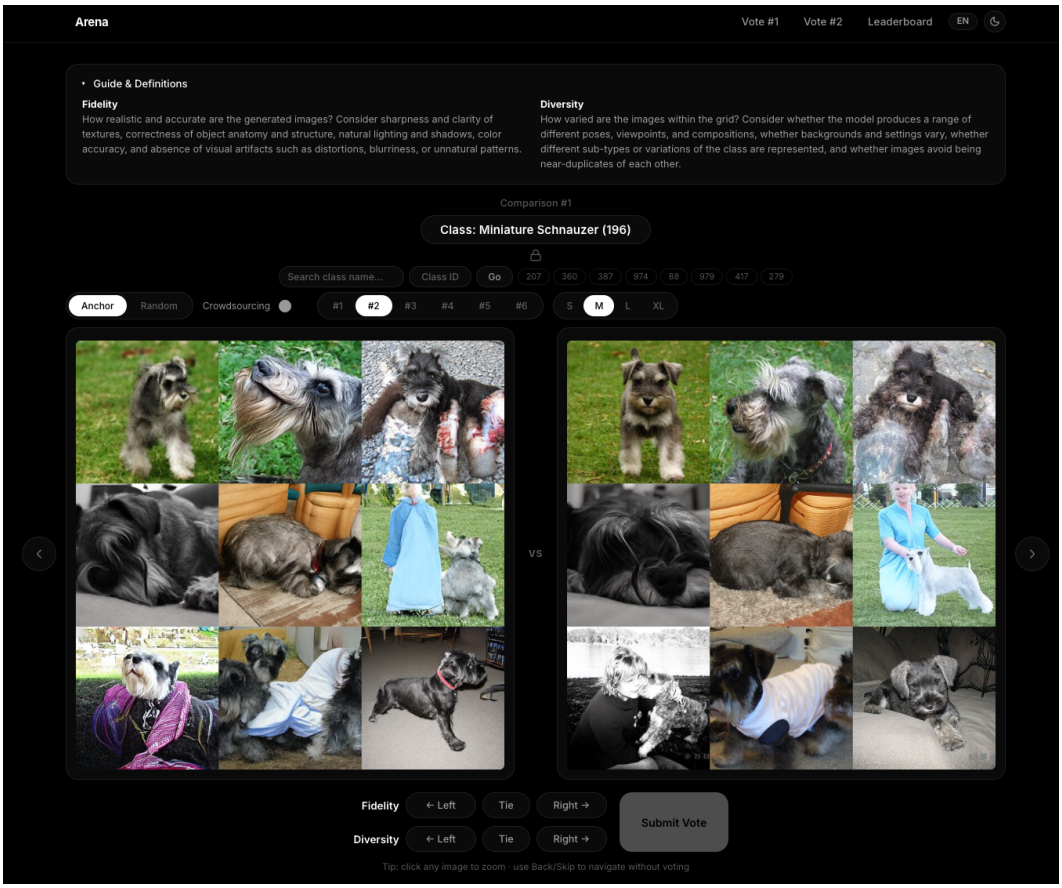

Figure C.1: Screenshot of the voting page. Voters see two anonymized 3×3 grids of samples for the same ImageNet class and pick *Left*, *Tie*, or *Right* for fidelity.

Interface. The voting interface is shown in Figure C.1. Each trial shows two side-by-side grids of uncurated samples from the same ImageNet class, sampled uniformly from the 1000 classes. The voter selects *Left*, *Tie*, or *Right*. A *Skip* arrow advances to a new pair without recording a vote, and a *Back* arrow revisits the immediately preceding pair. Images can be clicked for closer inspection.

Sampling. For each model, we sample 50,000 images in total, with 50 images per class. For each class, images are grouped into 3×3 grids; when the number of images is insufficient for an integer number of grids, we sample with replacement. All generators are sampled using their best-FID inference settings. *Post-trained vs. Base* trials are sampled uniformly among the three base/post-trained pairs, and *Generator vs. Real* trials are sampled uniformly among the three generator/validation pairs.

D Limitations and Broader Impact

Limitations. We focus on image generation, primarily on ImageNet, with text-conditioned generation as an additional demonstration. The behavior of *FD-loss* depends on the representation spaces and reference statistics. Different domains may require different feature sets or weights. Our experiments study post-training rather than replacing existing generator training recipes. Extending distribution-level optimization to broader data distributions, other modalities, higher resolutions, and adaptive representations remains an important direction.

Broader impact. Our work improves the quality of image generators, which carries dual-use risks common to all generative modeling research. Higher-quality generators could be misused to generate

disinformation or deceptive content. Our work also has diagnostic value: exposing the limits of FID can help the field better evaluate and understand generative models. We do not release any new datasets; all experiments use publicly available models and data.

E Additional Qualitative Samples

We show uncurated paired samples for two generators from Table 4: the one-step pMF-H/16 base model and its *FD-loss* (SIM) post-trained counterpart, and JiT-H/16 at 200 NFE (50 steps $\times$ 2 (Heun) $\times$ 2 (CFG)) and its post-trained counterpart at 1 NFE. Each pair uses the same initial noise for direct comparison.

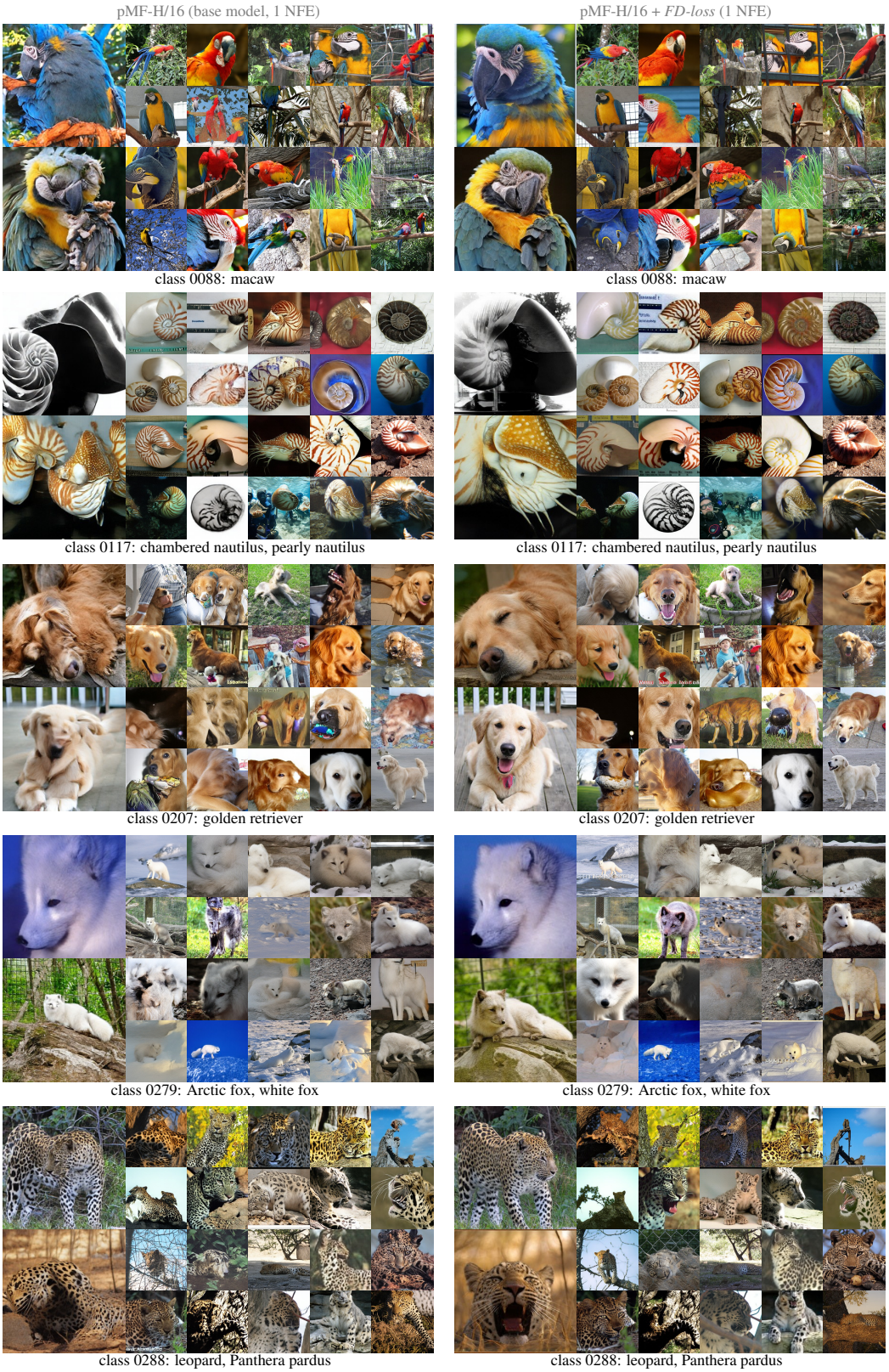

Figure E.1: *Uncurated* paired samples on ImageNet 256×256. Each class shows the **base model** pMF-H/16 [34] (left) and the **post-trained** pMF-H/16 with *FD-loss* (SIM) (right), using the *same initial noise*. SIM: SigLIP+Inception+MAE.

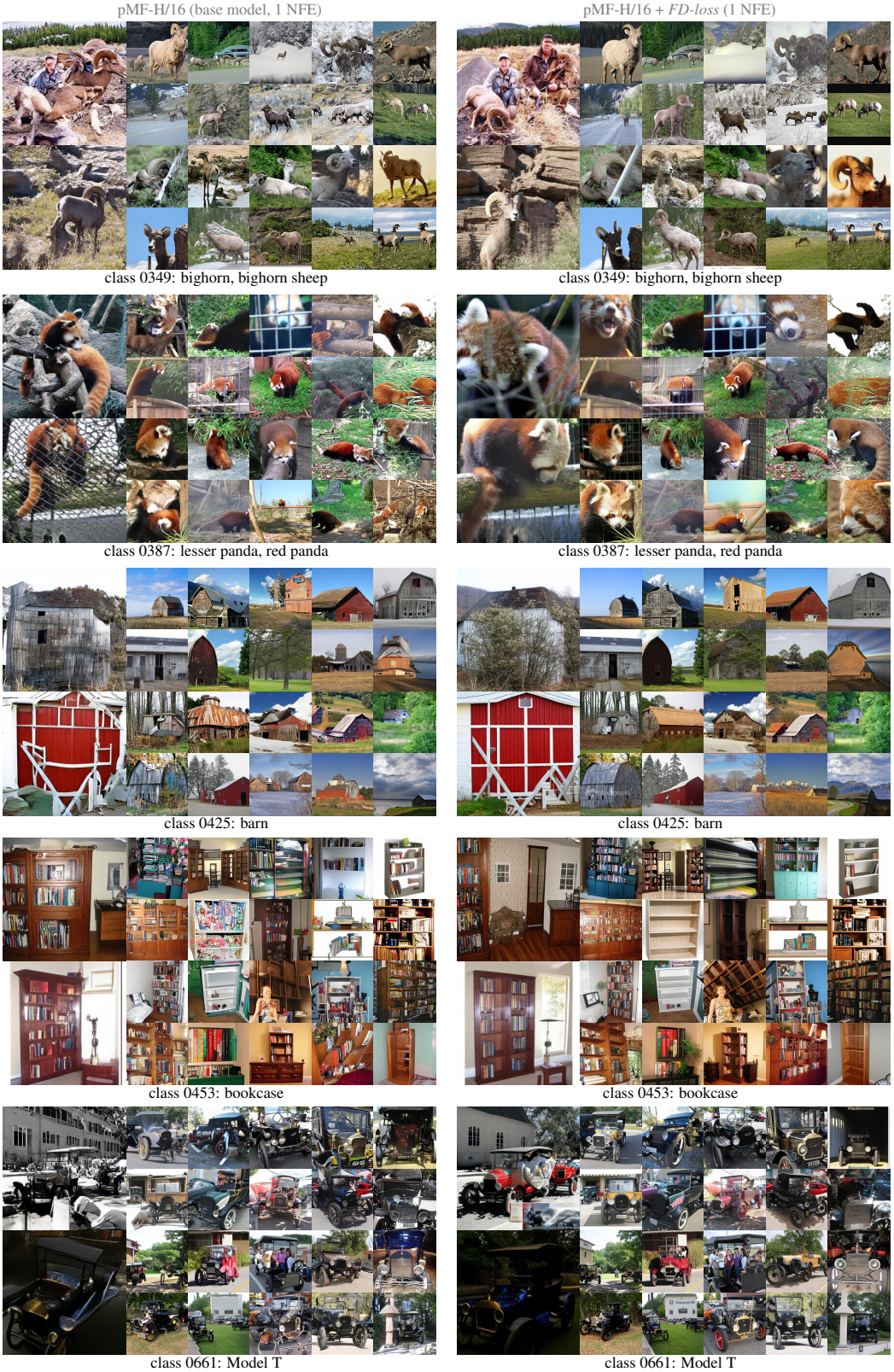

Figure E.2: *Uncurated* paired samples on ImageNet 256×256. Each class shows the **base model** pMF-H/16 [34] (left) and the **post-trained** pMF-H/16 with *FD-loss* (SIM) (right), using the *same initial noise*. SIM: SigLIP+Inception+MAE. (cont.)



Figure E.3: *Uncurated* paired samples on ImageNet 256×256. Each class shows the **base model** pMF-H/16 [34] (left) and the **post-trained** pMF-H/16 with *FD-loss* (SIM) (right), using the *same initial noise*. SIM: SigLIP+Inception+MAE. (cont.)

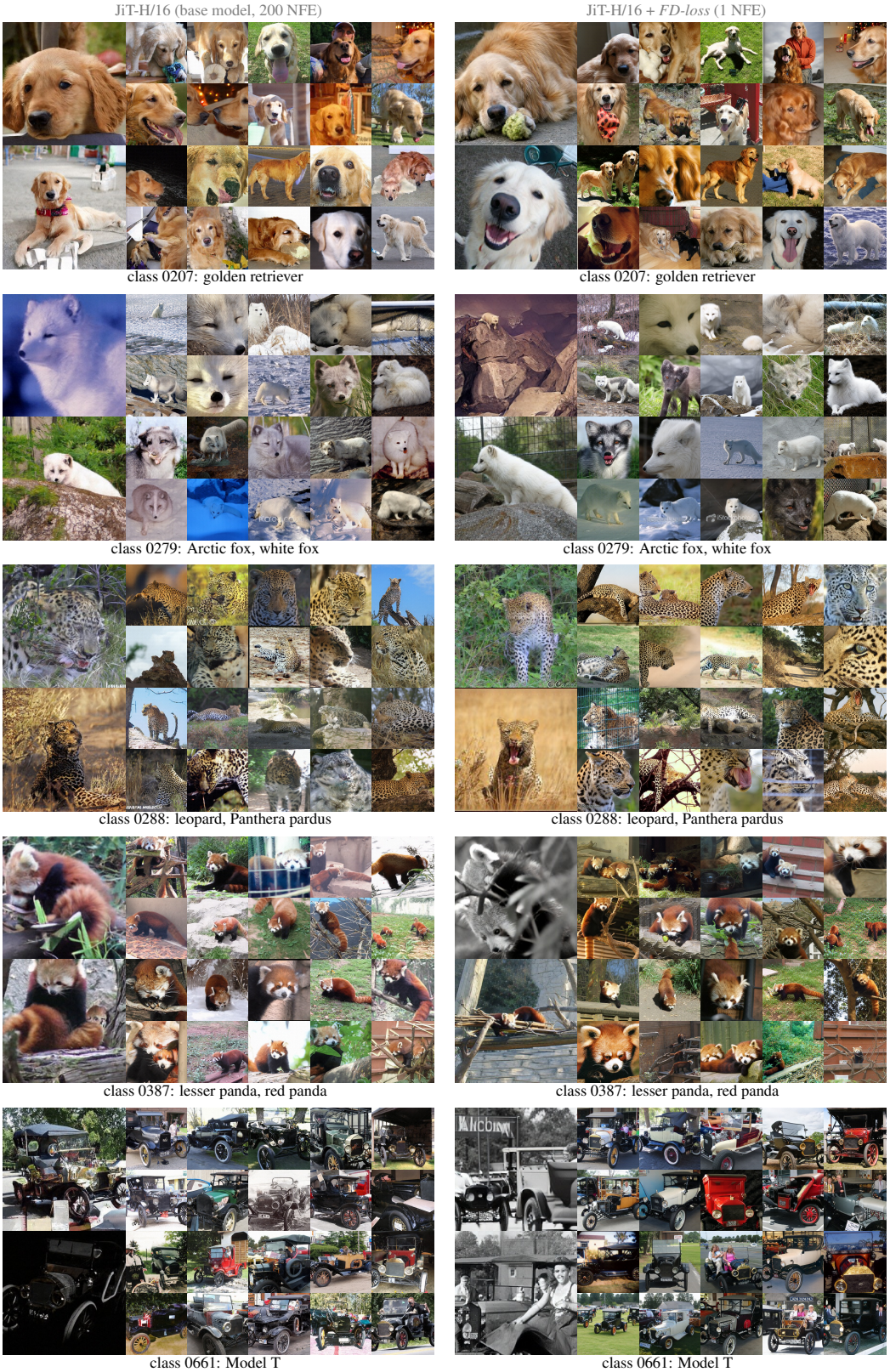

Figure E.4: *Uncurated* paired samples on ImageNet 256×256. Each class shows the **base model** JiT-H/16 [27] with 200 NFE (50 steps × 2 (Heun) × 2 (CFG)) (left) and the **post-trained** JiT-H/16 with *FD-loss* (SIM), 1 NFE (right), using the *same initial noise*. SIM: SigLIP+Inception+MAE.

F Detailed Results

We report per-representation results in both normalized FDr (Eq. 7) and raw Fréchet Distance. FDr^6 is the arithmetic mean of FDr over six representation spaces (Incep., ConvNeXt, DINOv2, MAE, SigLIP, CLIP). FDr-CLIP is additionally reported as a held-out evaluator.

Table F.1: **Per-representation FDr for population size ablation** (Tabs. 1a and 1b). pMF-B/16 post-trained for 50 epochs with FD-Inception. Default setting in Gray. †Statistics from current batch only.

setting	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	FDr-CLIP	FID↓	IS↑	FDr^6 ↓
Queue size									
Base	1.98	1.93	10.13	13.81	31.03	23.30	3.31	254.6	13.70
0k†	2.29	3.99	13.74	16.73	35.49	30.15	3.84	250.9	17.06
5k	0.62	1.41	8.73	9.54	28.15	22.89	1.05	280.0	11.89
10k	0.56	1.36	8.35	9.02	28.08	22.87	0.93	283.9	11.71
50k	0.53	1.50	7.83	8.28	26.12	21.20	0.89	288.3	10.91
100k	0.56	1.60	7.84	9.06	26.63	21.22	0.93	288.8	11.15
500k	0.72	2.17	9.49	19.57	40.11	33.92	1.22	294.4	17.67
EMA decay rate (β)									
Base	1.98	1.93	10.13	13.81	31.03	23.30	3.31	254.6	13.70
0.0†	2.29	3.99	13.74	16.73	35.49	30.15	3.84	250.9	17.06
0.9	0.58	1.50	8.28	8.74	26.21	21.84	0.98	283.6	11.19
0.99	0.50	1.34	7.57	8.51	25.50	20.98	0.84	291.8	10.74
0.999	0.48	1.26	7.52	8.51	26.02	21.07	0.81	294.5	10.81
0.9999	0.58	1.35	8.10	9.34	28.18	22.24	0.98	287.7	11.63

Table F.2: **Raw FD values for population size ablation** (Tabs. 1a and 1b). pMF-B/16 post-trained for 50 epochs with FD-Inception. Default setting in Gray. †Statistics from current batch only.

setting	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	CLIP	FID	IS
Validation set	1.68	56.87	14.19	0.04	0.60	5.60	1.68	232.2
Queue size								
Base	3.31	109.54	143.69	0.59	18.77	130.61	3.31	254.6
0k†	3.84	226.62	194.92	0.72	21.47	168.97	3.84	250.9
5k	1.05	80.02	123.78	0.41	17.03	128.32	1.05	280.0
10k	0.93	77.15	118.50	0.39	16.98	128.19	0.93	283.9
50k	0.89	85.11	111.10	0.35	15.80	118.84	0.89	288.3
100k	0.93	91.04	111.22	0.39	16.11	118.94	0.93	288.8
500k	1.22	123.48	134.67	0.84	24.26	190.12	1.22	294.4
EMA decay rate (β)								
Base	3.31	109.54	143.69	0.59	18.77	130.61	3.31	254.6
0.0†	3.84	226.62	194.92	0.72	21.47	168.97	3.84	250.9
0.9	0.98	85.05	117.40	0.37	15.85	122.42	0.98	283.6
0.99	0.84	76.44	107.43	0.36	15.43	117.59	0.84	291.8
0.999	0.81	71.80	106.67	0.36	15.74	118.08	0.81	294.5
0.9999	0.98	76.97	114.90	0.40	17.05	124.67	0.98	287.7

Table F.3: **Raw FD values for representation model ablation** (Tab. 1c). pMF-B/16 post-trained for 50 epochs. SIM: SigLIP+Inception+MAE.

loss	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	CLIP	FID	IS
Validation set	1.68	56.87	14.19	0.04	0.60	5.60	1.68	232.2
Base	3.32	109.75	143.70	0.59	18.77	130.59	3.31	254.6
FD-Inception	0.81	71.65	106.68	0.36	15.74	118.09	0.81	294.5
FD-ConvNeXt	1.64	19.33	69.94	0.32	10.51	110.19	1.64	281.0
FD-DINOv2	4.88	121.69	29.93	0.47	10.23	88.72	4.89	347.1
FD-MAE	6.42	109.18	75.18	0.05	8.73	73.93	6.42	344.0
FD-SigLIP	7.72	152.97	60.57	0.39	2.18	60.76	7.71	399.4
FD-SigLIP+Incep.	0.89	54.26	70.80	0.37	4.13	77.08	0.89	307.5
FD-SIM	0.94	48.38	65.90	0.10	4.20	55.00	0.94	307.8

Table F.4: **Per-representation FDr for JiT-L repurposing** (Tab. 2). JiT-L/16 post-trained for 50 epochs. SIM: SigLIP+Inception+MAE.

setting	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	FDr-CLIP	FID↓	IS↑	FDr^6 ↓
JiT-L (50-step)	1.54	3.49	6.10	8.07	19.37	25.82	2.59	288.5	10.73
JiT-L (1-step)	173.83	142.28	151.86	322.61	327.20	170.71	291.59	2.0	214.75
FD-Incep.	0.46	2.57	7.34	15.28	26.22	25.29	0.77	293.7	12.86
FD-MAE	3.89	3.82	7.87	2.20	18.21	19.84	6.52	280.4	9.30
FD-SigLIP	3.04	2.79	4.86	27.68	2.20	13.68	5.10	329.6	9.04
FD-SigLIP+MAE	2.78	1.91	4.14	0.96	2.23	10.97	4.67	354.0	3.83
FD-SIM	0.51	1.09	3.06	1.01	2.78	11.30	0.85	319.5	3.29

Table F.5: **Raw FD values for JiT-L repurposing** (Tab. 2). JiT-L/16 post-trained for 50 epochs. SIM: SigLIP+Inception+MAE.

setting	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	CLIP	FID	IS
Validation set	1.68	56.87	14.19	0.04	0.60	5.60	1.68	232.2
JiT-L (50-step)	2.59	198.19	86.54	0.35	11.72	144.69	2.59	288.5
JiT-L (1-step)	291.59	8091.15	2154.20	13.82	197.92	956.77	291.59	2.0
FD-Incep.	0.77	146.26	104.15	0.65	15.86	141.76	0.77	293.7
FD-MAE	6.52	217.29	111.65	0.09	11.01	111.19	6.52	280.4
FD-SigLIP	5.10	158.87	68.96	1.19	1.33	76.69	5.10	329.6
FD-SigLIP+MAE	4.67	108.46	58.79	0.04	1.35	61.48	4.67	354.0
FD-SIM	0.85	62.03	43.40	0.04	1.68	63.34	0.85	319.5

Table F.6: **Per-representation FDr for system-level comparison** (Tab. 4). Our *FD-loss* post-trained models in shaded rows. SIM: MAE+SigLIP+Incep.

method	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	FDr-CLIP	FID↓	IS↑	FDr^6 ↓
50k validation images	1.00	1.00	1.00	1.00	1.00	1.00	1.68	232.2	1.00
discrete									
VAR-d30	1.18	1.70	5.31	6.83	11.89	13.31	1.97	304.6	6.70
BAR-B [62]	0.68	0.93	4.13	5.02	8.30	6.78	1.15	273.9	4.31
BAR-L [62]	0.61	0.78	3.29	4.20	6.60	5.98	1.01	281.9	3.57
latent, multi-step									
without semantic distillation									
SiT-XL/2 [37]	1.26	2.02	7.89	5.62	16.14	17.69	2.12	256.7	8.44
MAR-L [28]	1.07	1.10	6.09	4.38	12.67	14.78	1.80	293.4	6.68
FlowAR-H [45]	1.00	1.30	4.68	4.59	12.75	12.49	1.68	274.1	6.13
MAR-H [28]	0.93	0.95	4.95	3.71	10.07	13.02	1.56	299.5	5.61
MAR-L, DeTok [58]	0.83	1.36	4.66	4.40	9.57	12.12	1.39	306.2	5.49
with semantic distillation									
REG [55]	0.92	1.14	3.45	3.02	8.42	10.86	1.54	302.9	4.64
SiT-XL/2-REPA [63]	0.85	1.22	4.27	3.85	9.87	12.65	1.42	306.1	5.45
LightningDiT [59]	0.85	1.09	3.76	3.02	8.47	10.21	1.42	294.3	4.57
DDT-XL [54]	0.75	1.02	4.26	4.11	10.16	13.86	1.26	309.3	5.70
REPA-E [26]	0.70	1.28	2.44	2.52	5.04	6.28	1.17	298.3	3.04
RAE-XL [64]	0.69	1.79	2.11	3.30	3.79	7.87	1.16	261.0	3.26
latent, one-step									
Drift-L [7]	0.91	2.03	10.35	6.51	24.12	21.59	1.53	257.2	10.92
iMF-XL [13] (1 NFE)	1.09	1.72	7.30	6.09	17.02	17.14	1.82	278.9	8.39
iMF-XL [13] (2 NFE)	0.96	1.54	6.31	5.62	14.91	15.52	1.61	289.1	7.48
pixel, multi-step									
PixNerd-XL [53]	1.25	1.21	3.57	3.56	9.12	11.36	2.10	318.8	5.01
JiT-L [27]	1.54	3.49	6.10	8.07	19.37	25.82	2.59	288.5	10.73
JiT-H [27]	1.18	2.52	4.28	5.65	11.91	20.40	1.97	296.0	7.66
pixel, one-step									
Drift-L [7]	0.85	0.73	6.33	6.51	22.13	26.49	1.43	305.8	10.51
pMF-L [34]	1.62	1.36	6.70	9.72	20.34	14.81	2.72	261.7	9.09
pMF-H [34]	1.37	1.15	5.43	6.25	15.33	11.68	2.29	267.2	6.87
+ FD-loss *(ours)*									
iMF-XL, Incep.	0.43	1.04	4.54	4.25	12.17	13.64	0.72	295.0	6.01
iMF-XL, SIM	0.45	0.69	2.38	1.02	3.64	6.54	0.76	301.3	2.45
JiT-H, Incep.	0.43	1.92	5.39	13.21	20.70	19.44	0.72	294.2	10.18
JiT-H, SIM	0.45	0.86	2.10	0.43	1.68	10.37	0.75	313.0	2.65
pMF-L, Incep.	0.44	0.94	4.60	4.24	14.34	12.55	0.73	293.9	6.19
pMF-L, SIM	0.47	0.57	2.21	0.56	3.03	5.68	0.78	309.2	2.09
pMF-H, Incep.	0.43	0.62	3.58	3.42	10.81	10.27	0.72	298.8	4.86
pMF-H, SIM	0.46	0.57	1.74	0.35	2.46	5.77	0.77	310.1	1.89

Table F.7: **Raw FD values for system-level comparison** (Tab. 4). Our *FD-loss* post-trained models in shaded rows.

method	Incep.	ConvNeXt	DINOv2	MAE	SigLIP	CLIP	FID	IS
Validation set	1.68	56.87	14.19	0.04	0.60	5.60	1.68	232.2
discrete								
VAR-d30	1.97	96.57	75.35	0.29	7.19	74.61	1.97	304.6
BAR-B [62]	1.15	52.76	58.61	0.22	5.02	38.00	1.15	273.9
BAR-L [62]	1.01	44.55	46.68	0.18	3.99	33.52	1.01	281.9
latent, multi-step								
without semantic distillation								
SiT-XL/2 [37]	2.12	114.89	111.86	0.24	9.76	99.16	2.12	256.7
MAR-L [28]	1.80	62.33	86.32	0.19	7.66	82.81	1.80	293.4
FlowAR-H [45]	1.68	73.68	66.42	0.20	7.71	69.99	1.68	274.1
MAR-H [28]	1.56	54.25	70.25	0.16	6.09	72.99	1.56	299.5
MAR-L, DeTok [58]	1.39	77.56	66.14	0.19	5.79	67.94	1.39	306.2
with semantic distillation								
REG [55]	1.54	64.86	48.93	0.13	5.09	60.87	1.54	302.9
SiT-XL/2-REPA [63]	1.42	69.36	60.62	0.17	5.97	70.90	1.42	306.1
LightningDiT [59]	1.42	62.19	53.38	0.13	5.12	57.24	1.42	294.3
DDT-XL [54]	1.26	58.25	60.39	0.18	6.15	77.70	1.26	309.3
REPA-E [26]	1.17	72.89	34.57	0.11	3.05	35.18	1.17	298.3
RAE-XL [64]	1.16	101.72	29.92	0.14	2.29	44.13	1.16	261.0
latent, one-step								
Drift-L [7]	1.53	115.37	146.88	0.28	14.59	121.01	1.53	257.2
iMF-XL [13] (1 NFE)	1.82	97.73	103.55	0.26	10.30	96.05	1.82	278.9
iMF-XL [13] (2 NFE)	1.61	87.79	89.51	0.24	9.02	86.98	1.61	289.1
pixel, multi-step								
PixNerd-XL [53]	2.10	68.78	50.69	0.15	5.52	63.67	2.10	318.8
JiT-L [27]	2.59	198.19	86.54	0.35	11.72	144.69	2.59	288.5
JiT-H [27]	1.97	143.09	60.71	0.24	7.20	114.35	1.97	296.0
pixel, one-step								
Drift-L [7]	1.43	41.35	89.84	0.28	13.39	148.46	1.43	305.8
pMF-L [34]	2.72	77.52	95.04	0.42	12.30	83.02	2.72	261.7
pMF-H [34]	2.29	65.45	76.96	0.27	9.27	65.47	2.29	267.2
+ FD-loss *(ours)*								
iMF-XL, Incep.	0.72	59.13	64.46	0.18	7.36	76.46	0.72	295.0
iMF-XL, SIM	0.76	39.35	33.71	0.04	2.20	36.63	0.76	301.3
JiT-H, Incep.	0.72	109.09	76.51	0.57	12.52	108.96	0.72	294.2
JiT-H, SIM	0.75	49.09	29.74	0.02	1.02	58.10	0.75	313.0
pMF-L, Incep.	0.73	53.41	65.27	0.18	8.67	70.36	0.73	293.9
pMF-L, SIM	0.78	32.58	31.39	0.02	1.83	31.85	0.78	309.2
pMF-H, Incep.	0.72	35.47	50.83	0.15	6.54	57.58	0.72	298.8
pMF-H, SIM	0.77	32.34	24.69	0.02	1.49	32.32	0.77	310.1

Table F.8: **Full metrics for all *FD-loss* post-trained models** (expanded version of Table 3). Each group shows the base generator and its post-trained variants using FD-Inception and FD-SIM. All post-trained models use 1 NFE. *FD-loss* post-trained models in shaded rows. SIM: SigLIP+Inception+MAE.

method	NFE	space	#params	FDr^6 ↓	FID↓	IS↑	Prec↑	Recall↑
pMF [34] (pixel-space, one-step)								
pMF-B	1	pixel	118M	13.70	3.31	254.6	0.81	0.52
+ *FD-loss* (Incep.)	1	pixel	118M	10.66	0.77	294.9	0.76	0.67
+ *FD-loss* (SIM)	1	pixel	118M	3.50	0.85	314.1	0.77	0.64
pMF-L	1	pixel	410M	9.09	2.72	261.7	0.81	0.56
+ *FD-loss* (Incep.)	1	pixel	410M	6.19	0.73	293.9	0.76	0.68
+ *FD-loss* (SIM)	1	pixel	410M	2.09	0.78	309.2	0.76	0.67
pMF-H	1	pixel	935M	6.87	2.29	267.2	0.80	0.59
+ *FD-loss* (Incep.)	1	pixel	935M	4.86	0.72	298.8	0.76	0.68
+ *FD-loss* (SIM)	1	pixel	935M	1.89	0.77	310.1	0.77	0.68
JiT [27] (pixel-space, multi-step → one-step)								
JiT-B	$50\times2\times2^{\dagger}$	pixel	131M	15.65	3.71	269.0	0.81	0.50
+ *FD-loss* (Incep.)	1	pixel	131M	22.48	0.76	296.4	0.76	0.67
+ *FD-loss* (SIM)	1	pixel	131M	5.53	1.00	325.5	0.78	0.60
JiT-L	$50\times2\times2^{\dagger}$	pixel	459M	10.73	2.59	288.5	0.79	0.59
+ *FD-loss* (Incep.)	1	pixel	459M	12.75	0.73	296.6	0.76	0.67
+ *FD-loss* (SIM)	1	pixel	459M	3.24	0.77	317.3	0.77	0.66
JiT-H	$50\times2\times2^{\dagger}$	pixel	953M	7.66	1.97	296.0	0.78	0.63
+ *FD-loss* (Incep.)	1	pixel	953M	10.18	0.72	294.2	0.75	0.68
+ *FD-loss* (SIM)	1	pixel	953M	2.65	0.75	313.0	0.76	0.66
iMF [13] (latent-space, one-step)								
iMF-B	1	latent	89M	15.29	3.45	254.2	0.79	0.53
+ *FD-loss* (Incep.)	1	latent	89M	11.34	0.79	296.8	0.76	0.67
+ *FD-loss* (SIM)	1	latent	89M	5.56	0.88	307.7	0.78	0.64
iMF-L	1	latent	409M	9.06	1.93	275.1	0.79	0.61
+ *FD-loss* (Incep.)	1	latent	409M	6.63	0.75	293.8	0.76	0.69
+ *FD-loss* (SIM)	1	latent	409M	2.74	0.79	302.8	0.77	0.67
iMF-XL	1	latent	610M	8.39	1.82	278.9	0.78	0.63
+ *FD-loss* (Incep.)	1	latent	610M	6.01	0.72	295.0	0.76	0.68
+ *FD-loss* (SIM)	1	latent	610M	2.45	0.76	301.3	0.77	0.67



Figure G.1: **Full text-to-image qualitative comparison (reproduction of Figure 7).** We post-train SD3.5 Medium [10] with *FD-loss* on two reference image distributions: BLIP3o-Pretrain-Long-3M [4], a 3M subset of a web corpus of realistic photographs, and BLIP3o-GPT4o-60k [4], a curated 60k dataset distilled from GPT-4o with a stylized aesthetic. These qualitative examples suggest that the post-trained 1-NFE models can preserve recognizable prompt content despite a $56\times$ NFE reduction. The reference image distribution also appears to shape the post-trained aesthetic: the 3M variant tends toward photorealism, while the 60k variant tends toward the stylized look. Each column shares the same prompt; full prompts in Appendix G.

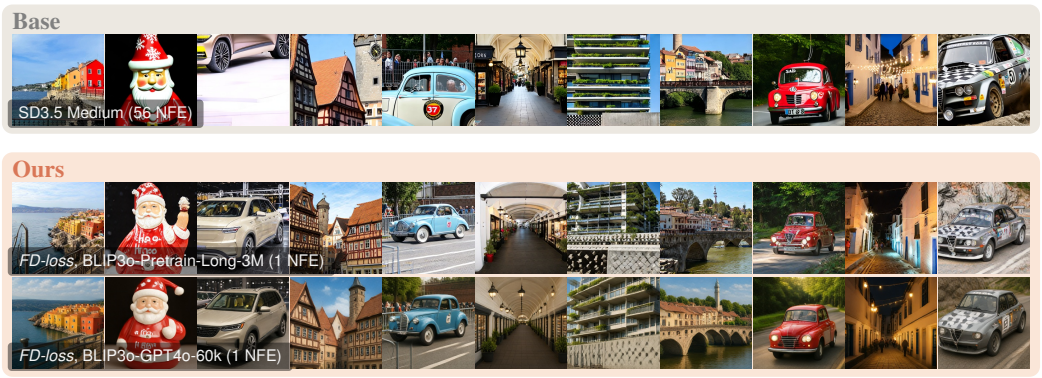

Figure G.2: **Additional text-to-image qualitative samples.** Extended version of Figure 7 with 12 additional prompts from BLIP3o-Pretrain-Long-3M. Top row: SD3.5 Medium (56 NFE). Bottom two rows: *FD-loss* post-trained models (1 NFE), trained on BLIP3o-Pretrain-Long-3M and BLIP3o-GPT4o-60k. Full prompts are listed in Appendix G.

G Text-to-Image Prompts

We list the full prompts for Figures 7 and G.2 in column order.

Main figure (Figure 7).

#1 A vibrant red hibiscus flower in full bloom, with its large, delicate petals spread wide open. The flower's center features a prominent stamen with a bright yellow tip, contrasting beautifully against the deep red of the petals.

#2 The iconic Oriental Pearl Tower in Shanghai illuminated at night, standing tall against the dark sky. The tower is adorned with colorful lights, predominantly blue and white. In the foreground, a beautifully landscaped garden with flowers in shades of pink, purple, and white.

#3 A serene coastal scene with a sandy beach in the foreground, dotted with scattered rocks. The beach leads to a rocky shoreline that meets the turquoise waters of the sea. In the background, a small, fortified structure perches atop a rocky outcrop.

#4 A corner building with a classic architectural style, featuring multiple stories and a symmetrical facade. The building is painted in a light yellow hue with white decorative elements. Each window has red awnings and is adorned with potted plants.

#5 A quaint, historic building with a weathered, light beige facade. The structure features multiple windows with wooden frames and balconies adorned with plants. A stone archway leads into the building. The street in front is paved with cobblestones.

#6 A vibrant garden scene dominated by lush green foliage and striking red flowers. The flowers appear to be some variety of red-hot poker (Kniphofia), standing out vividly against the backdrop of glossy, dark green leaves.

#7 A cozy, rustic restaurant with a warm and inviting atmosphere. The interior features wooden beams on the ceiling and polished wood flooring, giving a cabin-like feel.

#8 A fishing boat navigating through choppy waters under a clear blue sky with scattered clouds. The boat is white with blue accents and features a small cabin, a deck area with railings, and various fishing equipment visible on top.

#9 The exterior of a traditional-style building with a white facade and red shutters on the windows. Two flags, one on each side of the entrance, add a cultural touch. The entrance reveals an interior with wooden furniture and framed pictures.

Extended figure (Figure G.2).

#1 A picturesque coastal scene with a row of colorful buildings perched on a rocky outcrop overlooking the sea. The structures are painted in vibrant hues of orange, yellow, and red.

#2 A festive ceramic figurine of Santa Claus, characterized by his iconic red hat adorned with white snowflakes and a pom-pom at the tip, with a prominent white beard and mustache.

#3 A sleek, modern SUV displayed at an auto show. The vehicle is light beige or silver with a shiny exterior, reflecting the bright showroom lights.

#4 A charming European town square, likely in Germany, characterized by traditional half-timbered houses with red-tiled roofs and intricate wooden detailing.

#5 A vintage car painted in a striking combination of light blue and white, driving on a road. The car has a classic design with whitewall tires and a rounded body shape.

#6 A long, narrow indoor shopping arcade with a high, arched ceiling supported by white beams. The floor is paved with large, dark stone tiles, with rows of shops on both sides.

#7 A modern, multi-story building with a unique architectural design featuring horizontal balconies that extend outward, creating a series of platforms.

#8 A picturesque scene of a historic town nestled along the banks of a river, viewed through the arch of an old stone bridge, blending traditional and modern architecture.

#9 A vintage red Saab 96 car, number 38, driving on a winding road surrounded by lush green trees, with its headlights on.

#10 A charming, narrow cobblestone street at night, illuminated by warm, ambient lighting, flanked by traditional whitewashed buildings with blue accents on the doors.

#11 A vintage rally car, an Alfa Romeo Giulia, participating in a rally event. The car is painted metallic gray with a black and white checkered hood.

H Additional Experiments and Analysis

H.1 Comparison with Distribution-Matching Methods

We study how *FD-loss* compares with DMDv1 [61] and DMDv2 [60], which also match a generated distribution to a target distribution. The key difference is the training signal: DMD uses differences between score fields, whereas *FD-loss* directly matches first and second moments in representation spaces.

Setup. We initialize all methods from the same JiT-L source checkpoint and tune each method separately. For DMDv2, we linearly probe all 24 JiT-L blocks and select block 18 for the discriminator head. We use approximately 2,500 H100 GPU-hours for 10-epoch recipe sweeps followed by 50-epoch final training. We evaluate Coverage with the official dgm-eval protocol [50], using 100k samples per model, ImageNet training images as the reference, and $k = 5$. DINOv2 is held out from training. We also report held-out FDr^3, which averages ConvNeXt, DINOv2, and CLIP, none of which FD-SIM optimizes.

Results. Table H.1 reports the comparison.

Table H.1: **Comparison with distribution-matching methods.** JiT-L models evaluated with FD metrics and DINOv2 Coverage. † denotes retraining from EMA source weights.

Method	NFE	FID ↓	FDr^6 ↓	Held-out FDr^3 ↓	Coverage ↑
JiT-L source	200	2.62	10.80	11.74	0.93
DMDv1	1	9.82	31.02	19.07	0.83
DMDv2	1	5.31	14.91	10.51	0.90
FD-SigLIP†	1	4.58	8.19	7.73	0.92
FD-SIM†	1	0.81	3.18	5.16	0.95

FD-loss improves FD and Coverage over the tested DMD variants. These results establish its competitiveness under the configurations studied; they do not imply equal data or compute requirements across the methods.

Optimization components. We next study the auxiliary components in DMD. The two-time-scale update rule (TTUR) controls the update-frequency ratio between the fake-score network and the generator. The regression term in DMDv1 distills paired ODE teacher trajectories with LPIPS. Table H.2 shows that the update ratio, regression term, and GAN component all affect performance.

Table H.2: **Optimization components in DMD.** We compare TTUR settings, the regression term, and the GAN component.

Configuration	FID ↓	FDr^6 ↓	Coverage ↑
DMD, TTUR=1	59.70	65.99	0.34
DMD, TTUR=1 + regression (DMDv1)	9.82	31.02	0.83
DMD, TTUR=5	13.36	27.43	0.83
DMD, TTUR=5 + GAN (DMDv2)	5.31	14.91	0.90

Combining objectives. Adding DMD and GAN to FD-SigLIP improves FID from 4.58 to 3.80 and FDr^6 from 8.19 to 7.92, while preserving DINOv2 Coverage at 0.92. We do not tune the relative loss weights. This initial result suggests that the objectives can provide complementary signals; a more systematic study remains open.

H.2 High-Resolution Text-to-Image Generation

We further study *FD-loss* on FLUX.1-dev [3] at its native 1024×1024 resolution. We compare the base and post-trained models using GenEval [14], PickScore [23], and DPG-Bench [20]. Table H.3 reports results without a discriminator.

Table H.3: **High-resolution text-to-image generation.** FLUX.1-dev before and after post-training with *FD-loss*, without a discriminator.

Model	NFE	GenEval ↑	PickScore ↑	DPG-Bench ↑
FLUX.1-dev reference	50	0.67	23.01	83.77
FLUX.1-dev	1	0.04	18.00	14.09
FLUX.1-dev + FD-loss	1	0.63	22.11	82.18
FLUX.1-dev	4	0.46	21.29	69.29
FLUX.1-dev + FD-loss	4	0.67	22.88	82.56

Fine-grained detail. Scores from representation models operating at 224/256/299 pixels do not by themselves establish whether native-resolution details are preserved. We therefore also inspect the generated images. At 4 NFE, the post-trained model closely matches the 50-NFE reference in fine-grained quality. At 1 NFE, it preserves global structure and prompt fidelity, but loses some of the finest textures.

We also observe mild patch-grid artifacts at 1024×1024, but not at 512×512. This experiment uses SigLIPv2 and MAE, pretrained at resolutions of 256 and 224, respectively; we omit Inception for training throughput. We hypothesize that the artifacts are related to the patch sizes of these representation models. Higher-resolution encoders may help, but we have not tested this hypothesis. Adding a discriminator qualitatively reduces the grid artifacts.

H.3 Self-Representation

Can the generator itself provide a useful representation for *FD-loss*? We study this by using a copy of the pre-trained JiT-L denoiser as the FD encoder, which we call *self-representation*.

Setup. For partially noised inputs $x_t = (1-t)x+t\epsilon$, we probe all 24 blocks at $t \in \{0, 0.1, 0.5, 0.75\}$ with three learning rates (288 probes). We select block 18 by validation accuracy. We then compare three encoders: a frozen pre-trained model, the online generator, and an EMA copy of the generator. For the latter two, the feature space changes during training. We therefore update both real and generated moment estimates by EMA in the current feature space. Unlike our standard frozen-encoder setting, these variants require real images during post-training.

Results. Table H.4 reports results after 50 training epochs with $t \sim \mathrm{Uniform}(0, 0.75)$.

Table H.4: **Self-representation for JiT-L.** We compare frozen, online, and EMA encoders.

JiT-L self-representation	FID ↓	FDr^6 ↓
Frozen pretrained	9.42	20.25
Online	16.81	29.58
Model EMA=0.999	25.61	39.53

The frozen encoder performs best among the variants tested. Self-representation can thus repurpose JiT-L into a one-step generator, although it does not yet match the semantically richer external encoders. Our exploration of co-evolving representations remains limited; we leave a more systematic study to future work.

H.4 Numerical Stability and Seed Variation

Moment precision. We store EMA feature moments in FP64 and compute the covariance term in FP64 using the explicitly symmetrized positive-semidefinite product $\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2}$. Negative numerical eigenvalues are clamped to zero. We use neither diagonal jitter nor gradient clipping. This precision setting applies to the feature moments and FD computation, separately from the model-training precision reported in the implementation tables.

We study its effect by post-training pMF-B with FD-Inception for 10 epochs. FP32 and FP64 moment storage achieve similar final FID. In contrast, BF16 and FP16 moment storage substantially degrade

performance, even when the moments are later upcast. The practical requirement is therefore to preserve sufficient precision when storing the moments.

Training stability. We do not observe numerical training instability in completed ImageNet runs. A separate failure mode occurs when repurposing a multi-step model with a single representation and an excessively high learning rate: the model can produce adversarial examples. This reflects the limitations of the representation rather than numerical failure of the covariance computation.

Seed variation. We repeat a 10-epoch FD-SigLIP experiment with three seeds for each model. Table H.5 reports the mean $\pm$ sample standard deviation.

Table H.5: **Variation across training seeds.** Mean $\pm$ sample standard deviation for the short-horizon FD-SigLIP study.

Model	FID $\downarrow$	FDr^6 $\downarrow$
pMF-B	9.265 ± 0.151	6.495 ± 0.035
JiT-L	7.577 ± 0.255	11.705 ± 0.035

H.5 Human Preference across Generators

We conduct an additional human preference study to test whether FDr^6 agrees with human judgments across generators, especially when FID and FDr^6 disagree. This study collects 1,187 votes from 12 unique participants. It is separate from the 2,929-vote, 17-participant base-versus-post-training study in Appendix C.

Comparisons. For pairs 1–9, FID favors model A while FDr^6 favors model B. In all nine pairs, participants prefer model B. For pairs 10–11, both metrics favor model A, which also receives more preference votes. Table H.6 reports A, B, and tie votes separately, rather than splitting ties as in the original study. Rounded percentages may not sum to 100.

Table H.6: **Human preference across generators.** We report preference and tie votes separately. Pairs are chosen to test metric disagreement and comparisons with external generators.

Pair	Model A	Model B	A (%)	B (%)	Tie (%)
1	pMF-H + FD-Inception	pMF-H + FD-SIM	28.4	61.2	10.3
2	iMF-XL + FD-Inception	iMF-XL + FD-SIM	20.2	49.6	30.3
3	JiT-H + FD-Inception	JiT-H + FD-SIM	18.8	62.4	18.8
4	BAR-L	REPA-E	36.8	50.9	12.3
5	BAR-L	RAE-XL	35.5	48.2	16.4
6	JiT-L + FD-SIM	REPA-E	40.4	46.2	13.5
7	JiT-L + FD-SIM	RAE-XL	41.2	48.5	10.3
8	Drift-L (latent)	PixNerd-XL	36.4	48.5	15.2
9	Drift-L (pixel)	PixNerd-XL	40.0	50.9	9.1
10	pMF-H + FD-SIM	BAR-L	52.3	38.5	9.2
11	pMF-H + FD-SIM	RAE-XL	49.3	42.5	8.2

These results support FDr^6 as a broader quality signal than FID for the model pairs tested. The study is still limited by the selected comparisons, finite vote counts, and subjective judgments. We do not view FDr^6 as a replacement for human evaluation.

H.6 Evaluation in Held-Out Representation Spaces

FD-SIM optimizes three of the six representations in FDr^6. To test whether its gains extend beyond these training spaces, we evaluate with 11 held-out encoders across 12 base/post-trained model pairs.

Post-training improves all 132 comparisons (12 model pairs across 11 encoders), with mean reductions from 51.2% to 67.9%. The gains thus extend to representation spaces not optimized by FD-SIM. As with FDr^6, the aggregate still depends on the chosen encoders and their equal weighting.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims: scalable direct optimization of Fréchet Distance, generator post-training and repurposing, and the limitations of single-representation FID. The experimental scope is reflected in the experiments and limitations sections.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations in Appendix D and also notes in Section 3.4 that FDr^k depends on the chosen representations and inherits the Gaussian moment-matching assumption of Fréchet Distance.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results or theorem-proof statements. The mathematical expressions define the training objective, estimators, and evaluation metrics.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The method, estimators, representation models, training hyperparameters, evaluation protocol, and detailed per-representation metrics are described in Sections 3–4 and Appendices B–F.

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

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

   Answer: [Yes]

   Justification: The manuscript links the public repository and checkpoints. The repository provides installation instructions, training and released-checkpoint evaluation code, statistics utilities, and scripts for the main ImageNet ablation, JiT, and scalability experiments. This statement does not assert public code coverage of every additional study in Appendix H.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The main experimental setup is described in Section 4. Additional details, including model configurations, representation backbones, optimizer settings, training schedules, and evaluation protocol, are provided in Appendix B.

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Appendix H reports mean and sample standard deviation over three seeds for a 10-epoch FD-SigLIP study on pMF-B and JiT-L. These descriptive statistics are limited to that study; we do not report replicated uncertainty estimates for every experiment or claim statistical significance from the human preference percentages alone.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Training schedules, batch sizes, and precision are provided, but the manuscript does not give execution time and resource requirements for every experiment.

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors confirm that the research conforms to the NeurIPS Code of Ethics. The research uses public datasets and released checkpoints and discusses limitations and broader impacts. IRB review was not sought for the human preference studies; no IRB approval or exemption determination is claimed.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix D discusses broader impact. It notes both the positive diagnostic value of improving generative evaluation and the dual-use risks of higher-quality image generation, including deceptive or misleading content.

11. **Safeguards**

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

    Answer: [No]

    Justification: The paper describes dual-use risks in the broader-impact discussion but does not document specific safeguards for the released checkpoints.

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: We cite the datasets, model families, representation backbones, and baseline methods used in the paper. The current manuscript does not enumerate the licenses or terms of use for each existing asset.

13. **New assets**

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

    Answer: [Yes]

    Justification: The linked repository documents installation, training, evaluation, statistics computation, and experiment scripts alongside the released code, and links the released checkpoints. The code is released under the MIT license.

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

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

Answer: [No]

Justification: Appendix C describes the original preference protocol, voting choices, and interface screenshot. The manuscript does not provide compensation details or complete participant-facing instructions for both human studies.

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

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

Answer: [No]

Justification: IRB review was not sought for the human preference studies. No IRB approval or exemption determination is claimed. The manuscript does not provide complete details of participant risk disclosure.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed method. One external text-to-image reference set is described as GPT-4o-distilled and is cited, but the core method does not rely on LLM usage.