NORMALIZED PERCEPTUAL PULLBACKS FOR ONE-STEP GENERATIVE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

One step image generation places most distribution correction into training, making the construction of regression targets central to optimization. Feature guided drifting provides a natural way to build such targets, but the effect of transferring a feature space correction back to a pixel generator remains unclear. We show that, under a matched frozen field and scaling, feature target regression and an unnormalized pixel pullback induce the same local first order generator update. The pulled back correction is unevenly scaled across samples: with DINOv3-L, the largest 10% of samples account for 41.8% of the total correction magnitude. We therefore normalize each pulled back direction before forming stop gradient pixel targets. Normalizing in feature space leaves substantial variation after the encoder Jacobian, while normalization after the pullback directly controls pixel correction scale. Experiments show the same pattern across representation families, encoder capacities, image resolutions, and generator scales. On ImageNet, the method reaches FID 1.47, 2.04, and 3.18 at 256, 512, and 1024 resolution. On BLIP3o 60k, it reaches cFreD 3.03 and CLIPScore 23.23 under one step inference.

1 INTRODUCTION

One step image generation replaces iterative sampling with a single generator evaluation, reducing inference latency and simplifying deployment. A one step model, however, no longer has access to the sequence of intermediate corrections used by iterative generators. It must determine semantic layout, object structure, texture, and local detail in one prediction (Ho et al., 2020; Song et al., 2023; Yin et al., 2024b). Errors produced by the generator cannot be repaired by later sampling steps, which places greater weight on the correction signal used during training. This requirement becomes more demanding as resolution grows, because local errors occupy more pixels while global structure must remain coherent across a larger image.

Drifting based training addresses this setting by moving distribution evolution into the training procedure. Generated samples receive correction targets, and the generator learns to regress toward them. Frozen visual representations provide a useful source of such supervision because their geometry captures image structure that is poorly expressed by raw pixel distance. Drifting Models already support frozen feature representations and stop gradient feature targets, and relate their drifting field to maximum mean discrepancy (Deng et al., 2026). For a pixel generator, this still leaves an unresolved issue: a correction defined in representation space must pass through the representation map before it can supervise pixel outputs, and the effect of this transformation is not well characterized.

There are two natural ways to transfer a feature correction to the generator. One can construct a target directly in feature space and backpropagate the regression loss through the frozen encoder. The correction can also be pulled back explicitly through the encoder Jacobian and used to form a target in pixel space. Although the targets live in different spaces, we find that their local optimization behavior is closely related. Under a matched frozen field and matched scaling, feature target regression and an unnormalized pixel pullback induce the same local first order update on the generator parameters. The Jacobian transpose written explicitly in the pullback already appears when the feature regression objective is differentiated through the encoder. An explicit pullback alone therefore does not account for a different generator update.

The distinction appears in the scale of the pulled back directions. Let s_j denote the feature space correction for a generated sample and let J_ϕ be the Jacobian of the frozen encoder ϕ. The corresponding

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

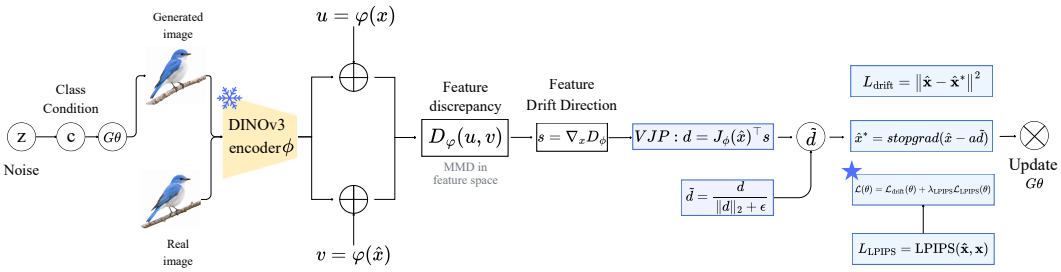

Figure 1: Training pathways considered in this work. A shared feature space discrepancy produces a sample wise correction s. The upper branch constructs a feature regression target, while the lower branch explicitly pulls the correction back to pixel space. Under matched scaling, the two objectives induce the same local first order generator update before normalization. Our method normalizes the pulled back direction before constructing a stop gradient pixel target. LPIPS serves as an auxiliary local regularizer, and inference requires only one generator forward pass.

pixel direction is $d_j = J_\phi^\top s_j$. Its norm varies substantially across samples. With DINOv3-L, the largest 10% of samples contribute 41.8% of the total pullback magnitude at a representative training checkpoint. This concentration remains visible across training stages, image resolutions, and several representation families. Since the sample corrections are combined through minibatch optimization, these scale differences change the relative contribution of individual samples to the generator update. A few large pullbacks can therefore dominate an update even when their feature space directions are not unusually informative. The imbalance also increases at higher image resolutions.

We control this variation by normalizing each direction after the pullback. The normalized direction is then used to construct a stop gradient pixel target for generator regression. The location of the normalization matters because the encoder Jacobian itself introduces sample dependent scaling. Normalizing the feature correction before applying $J_\phi^\top$ leaves considerable variation in the resulting pixel directions, whereas normalizing the final pullback directly controls the scale presented to the generator. This operation preserves the direction of each nonzero pullback while changing its relative contribution within a minibatch. We also retain LPIPS as a local auxiliary regularizer. Its main effect in our experiments is to recover coverage and reduce local artifacts after the distribution correction has been constructed.

Controlled comparisons isolate representation quality, target space, and correction scale under the same generator and training protocol. With the same DINOv3 encoder, feature target regression reaches an ImageNet FID of 1.56. An unnormalized pixel pullback reaches 1.64, batch level normalization reaches 1.55, and post pullback per sample normalization reaches 1.49. The full objective gives 1.47. We further vary the normalization strength continuously and observe that lower sample scale imbalance is accompanied by lower gradient variability, fewer training spikes, and better generation quality. Applying normalization before the pullback leaves more residual scale variation and gives weaker performance. Across representation families, configurations with greater raw pullback imbalance also tend to obtain larger gains from post pullback normalization. These controls identify post pullback scale control as the main optimization change.

The behavior extends beyond a single representation or model size. Pullback imbalance appears with pretrained VAE, supervised, CLIP, MAE, DINOv2, and DINOv3 encoders, while stronger representations generally provide better correction directions. The benefit of post pullback normalization remains when the generator is scaled from approximately 120M to 320M parameters. On class conditional ImageNet, the final method reaches FID 1.47, 2.04, and 3.18 at 256, 512, and 1024 resolution. On BLIP3o 60k text to image generation, it reaches cFreD 3.03 and CLIPScore 23.23 under one step inference.

Our results characterize a distinction that is easy to overlook in feature guided generative training. The representation determines the direction of a correction, while the pullback determines how strongly different samples enter the generator update. Controlling this scale after the pullback improves optimization across representations, resolutions, and generator sizes, while preserving the same one step inference procedure.

2 Related Work

One-step and few-step generative modeling. Diffusion models generate high quality images through iterative denoising, while latent diffusion reduces the cost of high resolution generation by operating in the latent space of a pretrained autoencoder (Ho et al., 2020; Rombach et al., 2022). A growing line of work compresses this iterative process into one or a few network evaluations. Consistency models learn predictions that remain consistent across noise levels (Song et al., 2023), while Distribution Matching Distillation and its variants train compact generators with distribution level supervision from pretrained diffusion models (Yin et al., 2024b;a). Related approaches extend distribution matching through broader divergence families and alternative objectives (Xu et al., 2025; Zhu et al., 2025a;c;b). Flow based methods reduce sampling cost through straighter probability paths or trajectory objectives (Liu et al., 2024; Lipman et al., 2023; Wu et al., 2025), and recent formulations such as MeanFlow and rCM further target very small numbers of function evaluations (Geng et al., 2025; Zheng et al., 2025). One-step text-to-image generation has also been studied through guided and high resolution distillation (Zhou et al., 2025; Chen et al., 2025; Pu et al., 2025). These methods differ in how supervision is constructed for a compressed generator. Our study focuses on the geometry and scale of the correction target once a feature-space distribution signal is used for training.

Drifting models and target construction. Drifting Models formulate generative learning as the evolution of a generator distribution during training and construct sample-level targets from a drift field that vanishes at equilibrium (Deng et al., 2026). The framework supports frozen feature representations, stop-gradient feature targets, and an interpretation of its drifting field through maximum mean discrepancy, making feature-space drifting a direct starting point for our analysis. We study how such a correction is transferred to a generator that produces pixels. Under a matched field and scaling, feature-target regression and an unnormalized pixel pullback induce the same local first-order generator update. This relation isolates the effect of the pullback itself and motivates our analysis of the sample-dependent scale introduced after the correction passes through the representation Jacobian.

Perceptual representations and distribution matching. Deep feature representations have long been used to measure visual similarity and supervise image synthesis. Perceptual losses compare generated and reference images through learned features (Zhang et al., 2018), while earlier texture synthesis methods matched multiscale image statistics to reproduce visual appearance (Heeger & Bergen, 1995). Distribution matching in learned representations has also been studied through kernel and discrepancy based objectives, including MMD-based generative models and related two-sample criteria (Li et al., 2017; Liu et al., 2016; Gorham & Mackey, 2017; Deshpande et al., 2018; Wu et al., 2019). Recent pixel-space generators further combine reconstruction or diffusion objectives with perceptual supervision from pretrained visual encoders. PixelGen, for example, uses LPIPS and DINO-based losses to complement pixel-space training (Ma et al., 2026), while self-supervised models such as DINOv3 provide strong global and dense representations for feature-based objectives (Siméoni et al., 2026). In our setting, the representation defines a distribution correction whose gradient is propagated toward pixel space. We examine the sample-wise scaling introduced by this pullback and how controlling that scale affects one-step generator training.

3 Problem Setup

3.1 Feature Space Corrections

We consider a one step generator G_θ with $\hat{x} = G_\theta(z)$, where $z \sim p(z)$, and a frozen image encoder ϕ. For a generated sample $\hat{x}_j$, let

$$v_j = \phi(\hat{x}_j), \tag{1}$$

and let s_j denote a feature space correction constructed from the current real and generated mini-batches. We keep the definition of s_j general here and specify our choice in Sec. 4.

A feature target can be formed as

$$v_j^\star = \mathrm{sg}(v_j - \eta s_j)\,, \tag{2}$$

where η is the step size and sg denotes stop gradient. The corresponding regression objective is

$$\mathcal{L}_{\text{feat}} = \frac{1}{2m} \sum_{j=1}^{m} \left\| \phi(G_\theta(z_j)) - v_j^\star \right\|_2^2 . \tag{3}$$

The same correction can be pulled back to pixel space through the encoder Jacobian,

$$d_j = J_\phi(\hat{x}_j)^\top s_j, \tag{4}$$

and used to construct a pixel target

$$\hat{x}_j^\star = \text{sg}(\hat{x}_j - \alpha d_j) . \tag{5}$$

3.2 Feature Targets and Pixel Pullbacks

Although Eqs. equation 2 and equation 5 define targets in different spaces, their local generator updates are closely related. Let θ_t denote the parameters at target construction time and $J_{G,j}$ the Jacobian of $G_\theta(z_j)$ with respect to θ at θ_t. Then

$$\nabla_\theta \mathcal{L}_{\text{feat}}|_{\theta_t} = \frac{\eta}{m} \sum_{j=1}^{m} J_{G,j}^\top J_\phi(\hat{x}_j)^\top s_j. \tag{6}$$

Using an unnormalized pixel regression target gives

$$\nabla_\theta \mathcal{L}_{\text{pix}}|_{\theta_t} = \frac{\alpha}{m} \sum_{j=1}^{m} J_{G,j}^\top J_\phi(\hat{x}_j)^\top s_j. \tag{7}$$

Proposition 1. For a fixed feature correction and matched step sizes $\alpha = \eta$, feature target regression and an unnormalized pixel pullback induce the same local first order generator update.

Thus, explicitly writing the pullback does not itself introduce a new local update direction. The practical difference must come from how the pulled back correction is modified before it enters generator optimization.

3.3 Pullback Scale

The pullback magnitude for sample j is

$$r_j = \left\| J_\phi(\hat{x}_j)^\top s_j \right\|_2 . \tag{8}$$

Because $J_\phi(\hat{x}_j)$ depends on the generated image, r_j can vary substantially across samples. This changes their relative contribution to a minibatch update even when the feature corrections themselves are comparable in scale.

We observe this imbalance across training stages, resolutions, and representation families. With DINOv3-L, the largest 10% of samples account for 41.8% of the total raw pullback magnitude at a representative checkpoint. A batch level rescaling preserves these relative differences. This motivates normalizing each correction after the pullback, which we introduce in the next section.

4 Method

4.1 Feature Space Discrepancy

We instantiate the feature correction in Sec. 3 using a frozen visual encoder ϕ. Our default encoder is DINOv3-L (Siméoni et al., 2026), while other representation families are evaluated in Sec. 6.1. For an image x, we combine its global token with the mean pooled patch features,

$$\phi(x) = \left[\text{norm}(\phi_{\text{cls}}(x)) \, ; \text{norm}(\text{pool}(\phi_{\text{patch}}(x))) \right] , \tag{9}$$

where $\text{norm}(a) = a/\|a\|_2$ and $\text{pool}(\cdot)$ denotes mean pooling over patch tokens. The encoder remains fixed throughout training.

Given real samples $\{x_i\}_{i=1}^n$ and generated samples $\{\hat{x}_j\}_{j=1}^m$, let

$$u_i = \phi(x_i), \qquad v_j = \phi(\hat{x}_j). \tag{10}$$

We measure their distribution discrepancy with the squared maximum mean discrepancy using an RBF kernel,

$$\begin{aligned} D_\phi = \frac{1}{n^2}\sum_{i,i'} k(u_i, u_{i'}) + \frac{1}{m^2}\sum_{j,j'} k(v_j, v_{j'}) \\ - \frac{2}{nm}\sum_{i,j} k(u_i, v_j), \end{aligned} \tag{11}$$

where

$$k(a,b) = \exp\left(-\frac{\|a-b\|_2^2}{2\sigma^2}\right). \tag{12}$$

The bandwidth σ is set by the median heuristic over the union of real and generated features in the minibatch. The correction for generated sample j is

$$s_j = \nabla_{v_j} D_\phi. \tag{13}$$

4.2 Normalized Pixel Targets

The feature correction is pulled back through the frozen encoder as

$$d_j = J_\phi(\hat{x}_j)^\top s_j. \tag{14}$$

We compute this vector Jacobian product with automatic differentiation and never form the Jacobian explicitly.

The analysis in Sec. 3 motivates controlling the scale after this pullback. We normalize each sample independently,

$$\tilde{d}_j = \frac{d_j}{\|d_j\|_2 + \epsilon}, \tag{15}$$

where ϵ is a small constant. This preserves the direction of every nonzero pullback while removing its sample dependent magnitude. The pixel target is then

$$\hat{x}_j^\star = \mathrm{sg}\left(\hat{x}_j - \alpha \tilde{d}_j\right), \tag{16}$$

where α controls the correction step. The target is detached before the generator update.

4.3 Training Objective

The generator regresses toward the normalized pixel targets with

$$\mathcal{L}_{\mathrm{drift}} = \frac{1}{m}\sum_{j=1}^m \left\|\hat{x}_j - \hat{x}_j^\star\right\|_2^2. \tag{17}$$

We additionally use LPIPS (Zhang et al., 2018) as a local consistency term. In class conditional training, each generated sample is paired with a real image of the same class when such a sample is available in the minibatch; otherwise we use a fixed within batch pairing rule. With pairing function $\pi(j)$,

$$\mathcal{L}_{\mathrm{LPIPS}} = \frac{1}{m}\sum_{j=1}^m \mathrm{LPIPS}\left(\hat{x}_j, x_{\pi(j)}\right). \tag{18}$$

The full training objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{drift}}(\theta) + \lambda_{\mathrm{LPIPS}}\mathcal{L}_{\mathrm{LPIPS}}(\theta). \tag{19}$$

For text conditioned generation, let $\psi(c)$ be the frozen text embedding of prompt c. We form the joint representation

$$h(x,c) = [\phi(x); \beta\ \mathrm{norm}(\psi(c))] \tag{20}$$

and compute the same MMD discrepancy in this joint space. The text features are detached, so only the image component of the resulting correction is pulled back to pixels. We use $\beta = 1$ in our experiments. The perceptual encoder, text encoder, discrepancy computation, and pullback are required only during training; inference consists of a single evaluation of G_θ.

5 EXPERIMENTAL SETUP

We evaluate all methods under a strict one step inference protocol. For class conditional generation, we train on ImageNet2012 and evaluate at resolutions of 256, 512, and 1024. Training images are randomly resized, cropped, and horizontally flipped, while evaluation uses deterministic resizing and center cropping at the target resolution. Each ImageNet evaluation uses 50,000 generated samples and fixed reference statistics. We report FID, Inception Score, Improved Precision and Recall with $K = 5$, KID, and CMMD. Since DINOv3 is used to construct the default training signal, distances measured with the same representation are retained only as diagnostics in the appendix and are not used as primary evidence. For text to image generation, we use the BLIP3o 60k prompt image pairs at 256 resolution and generate one image per prompt. We report cFreD, CLIPScore, and FID; CLIPScore is computed with OpenCLIP ViT-L/14, separate from the text encoder used for conditioning.

Our default experiments use a one step generator with approximately 120M parameters and a frozen DINOv3-L encoder (Siméoni et al., 2026). We additionally train an approximately 320M parameter generator to test whether the observed behavior persists with increased model capacity. Unless stated otherwise, optimization uses AdamW with learning rate 2×10^{-4}, cosine decay with warmup, drift step size $\alpha = 0.75$, LPIPS weight $\lambda_{\text{LPIPS}} = 0.30$, and bf16 precision. ImageNet models are trained for 400,000 optimizer updates with a global effective batch size of 256. At 256 resolution we use a per device batch size of 64; at 512 and 1024 we use batches of 16 and 4 with gradient accumulation of 4 and 16, respectively. The default experiments run on four A100 40GB GPUs. For text conditioned training, we set the joint representation weight to $\beta = 1$ and choose the RBF bandwidth with the minibatch median heuristic.

The controlled comparisons are designed to separate representation quality, target space, and correction scale. The target geometry study includes feature target regression with the same DINOv3 encoder, an unnormalized pixel pullback, batch RMS normalization, post pullback per sample normalization, and the full objective with LPIPS. We also compare against pixel drifting, pixel drifting with perceptual regularization, DMD, and Consistency using the same generator architecture, preprocessing, optimizer, training schedule, update budget, and one step evaluation protocol whenever the method permits. Representation experiments replace the frozen encoder with pretrained VAE, supervised, CLIP, MAE, DINOv2, and several DINOv3 variants while keeping the remaining training procedure fixed. Critical target geometry comparisons are repeated with five random seeds; other controlled experiments use three seeds unless stated otherwise. We report mean and standard deviation for the primary metrics and provide seed level results, runtime and memory measurements, and complete implementation details in the appendix.

6 RESULTS

6.1 MAIN GENERATION RESULTS

Table 1 reports the main ImageNet results under one step inference. At 256 resolution, our full objective reaches an FID of 1.47 ± 0.04, compared with 1.69 ± 0.06 for pixel drifting and 1.65 ± 0.05 when LPIPS is added without changing the drift construction. It also improves over the matched DMD and Consistency configurations, which obtain FID 1.60 ± 0.07 and 1.62 ± 0.06, respectively. Precision and recall remain balanced at 0.86 and 0.64. The same pattern persists at higher resolutions: our method reaches FID 2.04 ± 0.07 at 512 and 3.18 ± 0.12 at 1024, compared with 2.31 ± 0.09 and 3.74 ± 0.15 for pixel drifting. KID and CMMD follow the same overall ordering.

Figure 2(a) shows that the absolute FID improvement over pixel drifting increases with resolution, from 0.22 at 256 to 0.27 at 512 and 0.56 at 1024. The error audit in Fig. 2(c) gives a similar high resolution pattern. On a fixed set of 100 samples at 1024, the number of successful generations increases from 62 with pixel drifting to 78 with the full objective. Texture repetition decreases from 12 to 7 cases, semantic drift from 10 to 5, and edge tearing from 7 to 4. Blur decreases from 5 to 3 cases. The remaining failures indicate that one step generation still has difficulty correcting local artifacts once they are produced.

The additional computation is confined to training. Relative to pixel drifting, the full method requires approximately $1.55\times$, $1.77\times$, and $2.04\times$ training time per iteration at 256, 512, and 1024,

Table 1: One step class conditional generation on ImageNet. All controlled comparisons use the same generator architecture, training schedule, and evaluation protocol. DMD and Consistency are reported at 256 resolution, where matched controlled runs are available.

Method	Res.	FID ↓	IS ↑	Prec. ↑	Rec. ↑	KID $\times 10^3$ ↓	CMMD ↓
Pixel drifting	256	1.69 ± 0.06	301.8 ± 0.55	.84	.62	.74	.62
+ perceptual loss	256	1.65 ± 0.05	304.6 ± 0.50	.85	.61	.68	.58
DMD	256	1.60 ± 0.07	310.0 ± 0.65	.85	.62	.62	.53
Consistency	256	1.62 ± 0.06	308.0 ± 0.62	.84	.62	.65	.55
Ours	**256**	$\mathbf{1.47 \pm 0.04}$	$\mathbf{312.6 \pm 0.62}$	**.86**	**.64**	**.54**	**.47**
Pixel drifting	512	2.31 ± 0.09	296.2 ± 0.58	.83	.60	1.10	.70
+ perceptual loss	512	2.24 ± 0.08	298.9 ± 0.55	.84	.58	1.01	.67
Ours	**512**	$\mathbf{2.04 \pm 0.07}$	$\mathbf{305.4 \pm 0.61}$	**.85**	**.62**	**.87**	**.55**
Pixel drifting	1024	3.74 ± 0.15	283.7 ± 0.64	.80	.58	1.56	.81
+ perceptual loss	1024	3.59 ± 0.14	286.4 ± 0.60	.81	.56	1.44	.76
Ours	**1024**	$\mathbf{3.18 \pm 0.12}$	$\mathbf{294.1 \pm 0.67}$	**.82**	**.60**	**1.21**	**.63**

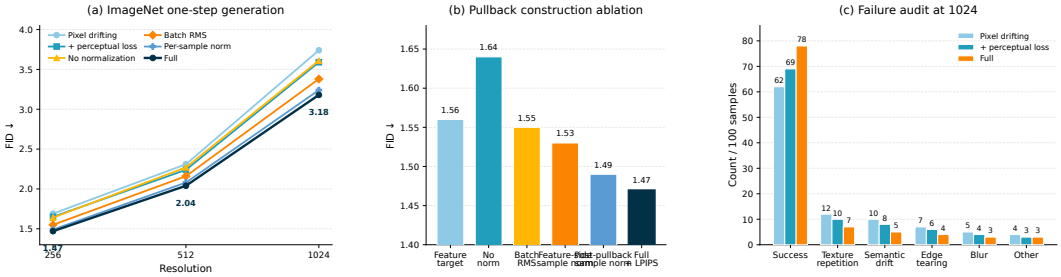

Figure 2: Main generation results and controlled target comparisons. **(a)** ImageNet FID across 256, 512, and 1024 resolution under one step inference. **(b)** ImageNet 256 comparison of feature targets, pullback targets, and normalization choices. **(c)** Failure taxonomy from a fixed 100-sample audit at 1024 resolution.

respectively. This cost comes from constructing the feature correction and propagating it through the frozen encoder, including the vector Jacobian product. These components are removed at inference, where generation remains a single forward pass of G_θ. The results above establish the generation performance and training cost under a fixed one step deployment setting. Section 6.2 next isolates which part of the target construction produces the observed improvement.

6.2 Pullback Scale and Normalization

We first isolate the effect of target space under a matched representation and training protocol. As shown in Table 2 and Fig. 2(b), feature target regression with DINOv3-L reaches FID 1.56 ± 0.05. Replacing the feature target with an explicit unnormalized pixel pullback gives 1.64 ± 0.07, while batch RMS scaling after the pullback gives 1.55 ± 0.05. This behavior agrees with the local analysis in Sec. 3.2. Numerically, the matched feature-target and unnormalized-pullback objectives produce generator gradients with cosine similarity 0.9998 ± 0.0001, norm ratio 1.003 ± 0.006, and relative ℓ_2 difference 0.0038 ± 0.0014. The explicit pullback therefore does not provide a different local update direction by itself.

The location of normalization determines how much sample-level scale variation remains. Without normalization, the pullback-norm CV is 1.31. Batch RMS scaling leaves the relative sample magnitudes almost unchanged, with CV 1.30. Per-sample normalization in feature space reduces this value to 0.91, but the subsequent encoder Jacobian introduces additional sample-dependent scaling. Normalizing after the pullback reduces the CV to 0.018 and improves FID from 1.64 to 1.49. Ap-

Table 2: Controlled target and normalization comparisons on ImageNet 256. All rows use the same generator, DINOv3-L representation, optimizer, and training schedule. CV denotes the coefficient of variation of the resulting pixel-space correction norm; for feature-target regression, it is measured from the effective pullback through the frozen encoder.

Target	Feature norm	Pixel norm	Correction	CV ↓	FID ↓	Prec. ↑	Rec. ↑
Feature target	Batch RMS	–	Feature	1.30	1.56 ± 0.05	.85	.62
Pixel target	None	None	Pullback	1.31	1.64 ± 0.07	.84	.60
Pixel target	None	Batch RMS	Pullback	1.30	1.55 ± 0.05	.85	.62
Pixel target	Per-sample	None	Pullback	.91	1.53 ± 0.05	.86	.61
Pixel target	None	Per-sample	Pullback	**.018**	$\mathbf{1.49 \pm 0.04}$	**.87**	.61
Full + LPIPS	**None**	**Per-sample**	**Pullback**	**.019**	$\mathbf{1.47 \pm 0.04}$	**.86**	**.64**

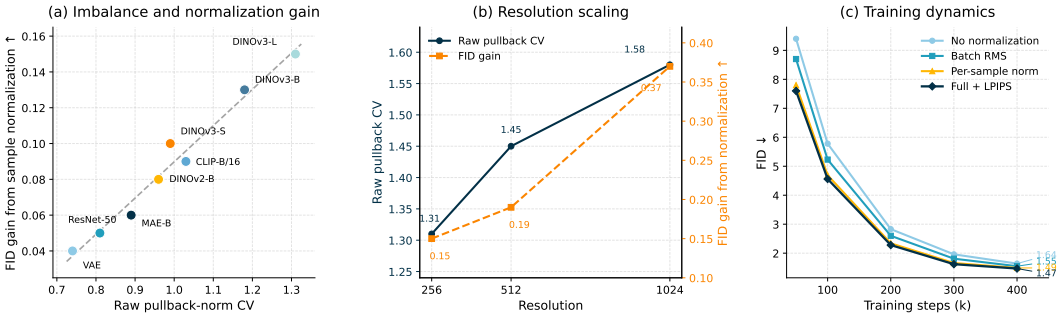

Figure 3: Pullback-scale diagnostics and generalization. **(a)** Raw pullback-norm coefficient of variation and the FID gain from post-pullback per-sample normalization across frozen representations; the dashed line is a descriptive linear fit. **(b)** Raw pullback-norm variation and normalization gain across image resolutions. **(c)** ImageNet FID during training for different normalization choices.

plying per-sample normalization both before and after the pullback gives FID 1.50, providing no further improvement over post-pullback normalization alone. These results locate the relevant scale control after the representation Jacobian rather than at the feature correction itself.

We further vary the normalization continuously using $d_j^{(\gamma)} = d_j/(\|d_j\|_2 + \epsilon)^\gamma$. Increasing γ from 0 to 1 reduces the target CV from 1.31 to 0.018 and the fraction of total correction magnitude carried by the largest 10% of samples from 41.8% to 10.4%. Over the same sweep, generator-gradient CV decreases from 1.49 to 0.76, the training-spike rate from 8.1% to 2.2%, and FID from 1.64 to 1.49. The intermediate settings follow the same trend: FID decreases from 1.59 at $\gamma = 0.25$ to 1.54 at 0.50 and 1.50 at 0.75. Figure 3(c) shows that the difference is already visible early in training and persists through 400k updates. At the final checkpoint, no normalization, batch RMS, and per-sample normalization reach FID 1.64, 1.55, and 1.49, respectively.

LPIPS provides a smaller and complementary effect. Adding it to the per-sample normalized target changes FID from 1.49 to 1.47 and recall from .61 to .64, while precision changes from .87 to .86. This pattern is consistent with its role as a local regularizer rather than the source of the distribution-level improvement. The main change in optimization is already present before LPIPS is added: controlling the sample-dependent scale introduced by the pullback reduces gradient variability and improves the one step generator throughout training.

6.3 Generalization Across Representations and Tasks

The pullback-scale effect is not specific to DINOv3-L. Across pretrained VAE, supervised, CLIP, MAE, DINOv2, and DINOv3 representations, Table 3 shows that post-pullback normalization improves the final generator in every tested setting. Figure 3(a) further shows that representations with larger raw pullback variation tend to obtain larger normalization gains. For example, VAE has a pullback-norm CV of .74 and improves by 0.04 FID, whereas DINOv3-L has CV 1.31 and im-

Table 3: Generalization across frozen representations, generator scale, and text to image generation. Encoder results use the full objective on ImageNet 256.

Setting	Raw CV	FID ↓	Prec. ↑	Rec. ↑	cFreD ↓	CLIPScore ↑
Frozen representation						
VAE	.74	1.66	.84	.62	–	–
ResNet-50	.81	1.63	.85	.61	–	–
MAE-B	.89	1.61	.84	.63	–	–
CLIP-B/16	.96	1.58	.86	.61	–	–
DINOv2-B	1.03	1.60	.85	.63	–	–
DINOv3-S	.99	1.56	.85	.62	–	–
DINOv3-B	1.18	1.51	.86	.63	–	–
DINOv3-L	**1.31**	**1.47**	**.86**	**.64**	–	–
Generator scale						
Pixel drifting, 120M	–	1.69	.84	.62	–	–
Ours, 120M	–	**1.47**	**.86**	**.64**	–	–
Pixel drifting, 320M	–	1.56	.86	.64	–	–
Ours, 320M	–	**1.42**	**.87**	**.65**	–	–
BLIP3o 60k text to image						
T2I drifting	–	35.80	–	–	3.62	22.14
+ perceptual loss	–	34.60	–	–	3.33	22.71
Ours	–	**34.10**	–	–	**3.03**	**23.23**

proves by 0.15. The same pattern strengthens with image resolution: Fig. 3(b) shows CV increasing from 1.31 at 256 to 1.58 at 1024, while the FID gain from sample normalization increases from 0.15 to 0.37.

The behavior also persists beyond the default generator and task. Scaling the generator from approximately 120M to 320M parameters improves the full method from FID 1.47 to 1.42, while retaining an advantage over the corresponding pixel-drifting model. On BLIP3o 60k, the same construction reduces cFreD from 3.62 to 3.03, increases CLIPScore from 22.14 to 23.23, and lowers FID from 35.80 to 34.10. These results indicate that the observed pullback scale behavior extends across representation families, model capacities, image resolutions, and conditional generation.

7 Discussion and Conclusion

Our analysis separates two effects that are easily conflated in feature-guided one step generation. A perceptual representation determines the direction of a distribution correction, while the pullback through the representation Jacobian determines how that correction is scaled across samples. Feature target regression and an unnormalized pixel pullback can induce the same local first order generator update, yet their behavior changes once sample-wise scale control is introduced. The experiments show that this scale variation is present across representation families and becomes more pronounced in settings where post-pullback normalization provides larger gains. This suggests that target geometry should account for both representation quality and the optimization geometry induced when feature corrections are transferred to the generator output space.

The method retains several limitations. The local equivalence result applies at the target construction point with a fixed correction field and matched scaling; it does not imply identical optimization trajectories after targets are refreshed. Training also requires a frozen encoder and a vector Jacobian product, adding cost that increases with image resolution even though inference remains one step. Performance depends on the chosen representation, and a feature encoder that is poorly aligned with a new domain may provide weak or misleading correction directions. Strict one step generation also continues to exhibit local and semantic failures at high resolution.

Within these constraints, the results identify post-pullback sample scale as a consequential part of feature-guided target construction. Per-sample normalization provides a simple way to control this variation without changing the inference procedure, and its effect persists across encoders, generator sizes, resolutions, and conditional generation settings.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work for supporting tasks including language polishing, literature retrieval, research ideation, experimental planning, and manuscript organization. They also assisted with revising the presentation of methods, experiments, figures, and tables. AI tools were not used to generate experimental data, derive reported numerical results, or establish mathematical proofs. All methodological decisions, experiments, citations, analyses, and scientific claims were independently reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide the complete training and evaluation protocol in the appendix, including generator architectures, frozen representation backbones, feature construction, MMD estimation, pullback computation, normalization variants, LPIPS settings, optimization hyperparameters, and text-conditioned training details. We also document random seeds, evaluation metrics, runtime and memory measurements, gradient-equivalence checks, representation and resolution controls, and the statistical reporting procedure used in the main experiments. Additional implementation details and configuration settings needed to reproduce the reported results are included in the supplementary material. Code and configuration files will be released where permitted by the submission policy.

REFERENCES

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Enze Xie, and Song Han. SANA-Sprint: One-step diffusion with continuous-time consistency distillation. In *IEEE/CVF International Conference on Computer Vision*, pp. 16185–16195, 2025. URL https://dblp.org/rec/conf/iccv/ChenXZYPCCHX25.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *CoRR*, abs/2602.04770, 2026. doi: 10.48550/arXiv.2602.04770. URL https://dblp.org/rec/journals/corr/abs-2602-04770.

Ishan Deshpande, Ziyu Zhang, and Alexander G. Schwing. Generative modeling using the sliced wasserstein distance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3483–3491, 2018. doi: 10.1109/CVPR.2018.00367. URL https://dblp.org/rec/conf/cvpr/DeshpandeZS18.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025. URL https://dblp.org/rec/conf/nips/GengDBKH25.

Jackson Gorham and Lester W. Mackey. Measuring sample quality with kernels. In *International Conference on Machine Learning*, pp. 1292–1301, 2017. URL https://dblp.org/rec/conf/icml/GorhamM17.

David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 229–238, 1995. doi: 10.1145/218380.218446. URL https://dblp.org/rec/conf/siggraph/HeegerB95.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://dblp.org/rec/conf/nips/HoJA20.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017. URL https://dblp.org/rec/conf/nips/LiCCYP17.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://dblp.org/rec/conf/iclr/LipmanCBNL23.

Qiang Liu, Jason D. Lee, and Michael I. Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*, pp. 276–284, 2016. URL https://dblp.org/rec/conf/icml/LiuLJ16.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024. URL https://dblp.org/rec/conf/iclr/LiuZM0024.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. PixelGen: Pixel diffusion beats latent diffusion with perceptual loss. *CoRR*, abs/2602.02493, 2026. doi: 10.48550/arXiv.2602.02493. URL https://dblp.org/rec/journals/corr/abs-2602-02493.

Yifan Pu, Yizeng Han, Zhiwei Tang, Jiasheng Tang, Fan Wang, Bohan Zhuang, and Gao Huang. Few-step distillation for text-to-image generation: A practical guide. *CoRR*, abs/2512.13006, 2025. doi: 10.48550/arXiv.2512.13006. URL https://dblp.org/rec/journals/corr/abs-2512-13006.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://dblp.org/rec/conf/cvpr/RombachBLEO22.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. URL https://dblp.org/rec/journals/tmlr/SimeoniVSBOJKSYRMHWWDMS26.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252, 2023. URL https://dblp.org/rec/conf/icml/SongD0S23.

Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3713–3722, 2019. doi: 10.1109/CVPR.2019.00383. URL https://dblp.org/rec/conf/cvpr/WuHALTPG19.

Zhangkai Wu, Xuhui Fan, Hongyu Wu, and Longbing Cao. TraFlow: Trajectory distillation on pre-trained rectified flow. *CoRR*, abs/2502.16972, 2025. doi: 10.48550/arXiv.2502.16972. URL https://dblp.org/rec/journals/corr/abs-2502-16972.

Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching. *CoRR*, abs/2502.15681, 2025. doi: 10.48550/arXiv.2502.15681. URL https://dblp.org/rec/journals/corr/abs-2502-15681.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a. URL https://dblp.org/rec/conf/nips/YinGP0SDF24.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b. doi: 10.1109/CVPR52733.2024.00632. URL https://dblp.org/rec/conf/cvpr/YinG0SDFP24.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068. URL https://dblp.org/rec/conf/cvpr/ZhangIESW18.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. *CoRR*, abs/2510.08431, 2025. doi: 10.48550/arXiv.2510.08431. URL https://dblp.org/rec/journals/corr/abs-2510-08431.

Mingyuan Zhou, Zhendong Wang, Huangjie Zheng, and Hai Huang. Guided score identity distillation for data-free one-step text-to-image generation. In *International Conference on Learning Representations*, 2025. URL https://dblp.org/rec/conf/iclr/ZhouWZH25.

Huaisheng Zhu, Teng Xiao, Shijie Zhou, Zhimeng Guo, Hangfan Zhang, Siyuan Xu, and Vasant G. Honavar. Simple distillation for one-step diffusion models. In *Advances in Neural Information Processing Systems*, 2025a. URL https://dblp.org/rec/conf/nips/ZhuXZGZXH25.

Yuanzhi Zhu, Eleftherios Tsonis, Lucas Degeorge, and Vicky Kalogeiton. One-step diffusion models with bregman density ratio matching. *CoRR*, abs/2510.16983, 2025b. doi: 10.48550/arXiv.2510.16983. URL https://dblp.org/rec/journals/corr/abs-2510-16983.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Di[M]O: Distilling masked diffusion models into one-step generator. In *IEEE/CVF International Conference on Computer Vision*, pp. 18606–18618, 2025c. doi: 10.1109/ICCV51701.2025.01729. URL https://dblp.org/rec/conf/iccv/ZhuWLK25.

APPENDIX

A LOCAL EQUIVALENCE DERIVATION

This section gives the full derivation of the local equivalence result used in Sec. 3.2. The result concerns the generator gradient at the point where the regression targets are constructed. It does not require the encoder to be linear, and it does not imply that feature-target and pixel-target training follow the same trajectory after the targets are refreshed.

Let

$$\hat{x}_j = G_{\theta_t}(z_j), \qquad v_j = \phi(\hat{x}_j), \tag{21}$$

where θ_t denotes the generator parameters at target construction time. Let s_j be a feature-space correction that is held fixed during the subsequent regression update. The stopped feature target is

$$v_j^\star = \mathrm{sg}(v_j - \eta s_j)\,. \tag{22}$$

For a single sample, define

$$\ell_{\mathrm{feat},j}(\theta) = \frac{1}{2}\left\|\phi(G_\theta(z_j)) - v_j^\star\right\|_2^2. \tag{23}$$

At $\theta = \theta_t$, the residual is

$$\phi(G_{\theta_t}(z_j)) - v_j^\star = \eta s_j. \tag{24}$$

Let $J_{G,j}$ be the Jacobian of $G_\theta(z_j)$ with respect to θ, evaluated at θ_t, and let $J_\phi(\hat{x}_j)$ be the Jacobian of the frozen representation with respect to its image input. Applying the chain rule,

$$\nabla_\theta \ell_{\mathrm{feat},j}|_{\theta_t} = \eta J_{G,j}^\top J_\phi(\hat{x}_j)^\top s_j. \tag{25}$$

Now define the unnormalized pixel pullback

$$d_j = J_\phi(\hat{x}_j)^\top s_j, \tag{26}$$

and the stopped pixel target

$$\hat{x}_j^\star = \mathrm{sg}(\hat{x}_j - \alpha d_j)\,. \tag{27}$$

The corresponding single-sample regression objective is

$$\ell_{\mathrm{pix},j}(\theta) = \frac{1}{2}\left\|G_\theta(z_j) - \hat{x}_j^\star\right\|_2^2. \tag{28}$$

At θ_t, its residual is αd_j, giving

$$\nabla_\theta \ell_{\mathrm{pix},j}|_{\theta_t} = \alpha J_{G,j}^\top J_\phi(\hat{x}_j)^\top s_j. \tag{29}$$

Averaging Eqs. equation 25 and equation 29 over a minibatch preserves the same relation. Therefore, when $\alpha = \eta$, the two objectives induce the same local first-order generator gradient at the target construction point.

Scope of the result. The equality assumes a fixed feature correction during target regression and matched scalar step sizes. Once the generator parameters change, the encoder Jacobian, minibatch discrepancy, and refreshed targets may also change. The result therefore characterizes a local update, not a global optimization trajectory. Operations applied after the pullback, including per-sample normalization, break this equality by changing the relative sample weighting before the generator update is aggregated.

B NUMERICAL GRADIENT VERIFICATION AND TARGET CONTROLS

We verify the analytical result numerically and then separate the effects of the feature field, target space, and normalization rule. All comparisons in this section use the same generator, DINOv3-L representation, minibatches, optimizer, and training schedule unless explicitly stated otherwise.

Table A1: Numerical comparison of generator gradients. Cosine similarity, norm ratio, and relative ℓ_2 error are computed on matched minibatches. The first row tests the local equivalence result directly; later rows quantify how changing the field or normalization changes the update.

Gradient pair	Cosine ↑	Norm ratio	Relative ℓ_2 ↓
Matched feature target vs. unnormalized pullback	$\mathbf{0.9998 \pm 0.0001}$	$\mathbf{1.003 \pm 0.006}$	$\mathbf{0.0038 \pm 0.0014}$
MMD feature target vs. original drifting field	0.946 ± 0.014	0.91 ± 0.05	0.118 ± 0.021
Batch RMS vs. post-pullback sample normalization	0.931 ± 0.019	1.14 ± 0.08	0.183 ± 0.032
Sample normalization vs. full objective	0.884 ± 0.028	1.09 ± 0.07	0.247 ± 0.041

Table A2: Full decomposition of the correction field, target space, and normalization rule on ImageNet 256. The table separates changes to the field from changes to the geometry of the regression target.

Field	Target	Normalization	FID ↓	Prec. ↑	Rec. ↑	Role
Original drifting	Feature	Batch RMS	1.56	.85	.62	Control
MMD	Feature	Batch RMS	1.59	.84	.62	Control
Original drifting	Pixel	Batch RMS	1.58	.85	.61	Control
MMD	Pixel	None	1.64	.84	.60	Control
MMD	Pixel	Batch RMS	1.55	.85	.62	Control
MMD	Pixel	Per-sample	1.49	.87	.61	Core
MMD + LPIPS	Pixel	Per-sample	**1.47**	**.86**	**.64**	**Full**
Direct MMD optimization	–	–	1.62	.83	.59	Control

The matched feature-target and unnormalized-pullback gradients are nearly identical numerically, with cosine similarity 0.9998 ± 0.0001. The larger differences in the remaining rows appear only after the field or normalization rule is changed, consistent with the distinction made in the main text.

The decomposition shows that neither replacing the drifting field with MMD nor writing the Jacobian-transpose pullback explicitly explains the final gain. The largest change appears when the raw pullback is normalized per sample after the representation Jacobian has been applied.

Feature-side normalization reduces part of the scale variation but does not control the variation introduced by $J_\phi(\hat{x})^\top$. Applying normalization after the pullback reduces the CV by more than an order of magnitude and gives the strongest result before the auxiliary LPIPS term is added.

C Pullback Scale Diagnostics

We measure raw pullback imbalance using four complementary statistics: coefficient of variation (CV), the ratio between the 95th percentile and median norm, the Gini coefficient, and the fraction of total correction magnitude carried by the largest 10% of samples.

The imbalance is present for every tested representation family, although its severity varies substantially. This distinction is useful because a stronger representation can provide better correction directions while simultaneously producing a more uneven pixel-space scale after the pullback.

The checkpoint sweep shows that the raw imbalance decreases during training but remains substantial at convergence. The resolution sweep shows the opposite trend: the same correction mechanism becomes increasingly uneven as spatial resolution grows.

The raw pullback imbalance is not confined to a single training checkpoint. Figure A1 tracks its evolution during training and across image resolutions. Although the imbalance gradually decreases as training progresses, it remains substantial at the final checkpoint. The opposite trend appears with spatial resolution: both the coefficient of variation and the concentration of correction magnitude increase from 256 to 1024.

Table A3: Location of normalization. Pullback CV is the coefficient of variation of the pixel-space correction norm after all operations listed in the row.

Variant	Feature-side norm	Pixel-side norm	Pullback CV ↓	FID ↓
No normalization	None	None	1.31	1.64
Batch RMS in feature space	Batch RMS	None	1.30	1.56
Per-sample feature normalization	Per-sample	None	.91	1.53
Batch RMS after pullback	None	Batch RMS	1.30	1.55
Per-sample after pullback	None	Per-sample	**.018**	**1.49**
Per-sample before and after	Per-sample	Per-sample	.019	1.50

Table A4: Raw pullback imbalance across frozen representation families on ImageNet 256. All statistics are computed before post-pullback normalization.

Encoder	CV	P95 / median	Gini	Top-10% mass (%)
VAE	.74	2.55	.31	25.1
ResNet-50	.81	2.79	.34	27.8
MAE-B	.89	3.08	.36	29.4
CLIP-B/16	.96	3.31	.38	30.7
DINOv3-S	.99	3.47	.40	32.5
DINOv2-B	1.03	3.56	.40	33.2
DINOv3-B	1.18	4.16	.45	37.9
DINOv3-L	**1.31**	**4.72**	**.48**	**41.8**

These measurements indicate that pullback-scale variation is a persistent property of the representation-to-pixel mapping rather than a transient artifact of early optimization. Its growth with resolution is also consistent with the larger normalization gains observed in the high-resolution experiments.

D Representation Controls

The main paper reports the final representation sweep. Here we provide the complete metrics, training cost, random-feature controls, and the relation between raw imbalance and normalization benefit.

The trend is not perfectly monotonic, but representations with larger raw pullback variation generally receive larger gains from post-pullback normalization. This is the pattern visualized in Fig. 3(a).

E Continuous Normalization and Optimization Dynamics

We use the continuous family

$$d_j^{(\gamma)} = \frac{d_j}{(\|d_j\|_2 + \epsilon)^\gamma}, \qquad \gamma \in [0, 1], \tag{30}$$

to interpolate between the raw pullback and full per-sample normalization. This experiment changes the strength of scale control without changing the underlying feature correction direction.

Lower correction-scale dispersion is accompanied by lower gradient variability, fewer training spikes, and lower FID throughout the interpolation.

We next examine how the normalization rule changes the optimization statistics behind the final generation metrics. Figure A2 compares correction-scale variation, generator-gradient variation, and training spike frequency under the same training protocol.

Table A5: Raw DINOv3-L pullback imbalance across training checkpoints and image resolutions. The checkpoint sweep uses ImageNet 256; the resolution sweep uses the representative checkpoint employed in the main diagnostic analysis.

Setting	CV	P95 / median	Gini	Top-10% mass (%)
Training checkpoint				
20k	1.46	5.18	.52	45.0
100k	1.37	4.91	.50	43.1
200k	1.31	4.72	.48	41.8
300k	1.25	4.44	.46	40.1
400k	1.20	4.19	.44	38.9
Image resolution				
256	1.31	4.72	.48	41.8
512	1.45	5.23	.51	44.5
1024	1.58	5.88	.54	47.3

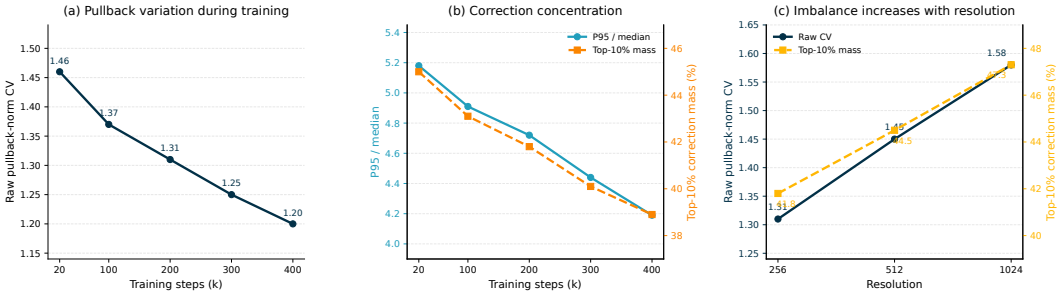

Figure A1: Raw pullback imbalance across training and resolution. **(a)** Pullback-norm coefficient of variation for DINOv3-L across training checkpoints. **(b)** P95-to-median ratio and the fraction of total correction magnitude carried by the largest 10% of samples. **(c)** Pullback-norm variation and correction concentration at 256, 512, and 1024 resolution.

Batch RMS scaling leaves the sample-level target variation almost unchanged, while post-pullback per-sample normalization sharply reduces both target and gradient variability. The training-spike rate decreases from 8.1% without normalization to 2.2% with sample normalization, and to 1.8% after the LPIPS term is added. The stability change therefore appears before the final FID difference and is consistent with the optimization mechanism studied in the main paper.

F Local Regularization

LPIPS is used as a local auxiliary term after the distribution correction has been constructed. We evaluate several pairing rules to determine whether the main result depends on a specific matching heuristic.

The differences among reasonable same-class pairing rules are small relative to the effect of post-pullback normalization. We use the simple random same-class rule in the default configuration. Removing LPIPS slightly improves precision but reduces recall and increases local failures, consistent with the auxiliary role described in the main text.

G Hyperparameter Sensitivity

We evaluate sensitivity to four quantities that affect different stages of the training procedure: the pixel-target step size α, the LPIPS weight λ_{LPIPS}, the RBF bandwidth used to construct the MMD field, and the global effective batch size. Unless the parameter under study is being varied, all remaining settings are fixed to the default configuration. The default values are $\alpha = 0.75$, $\lambda_{\text{LPIPS}} = 0.30$, an RBF bandwidth multiplier of 1.0 relative to the minibatch median heuristic, and an effective

Table A6: Full representation sweep with the final objective on ImageNet 256. Relative time is measured against pixel drifting under the same generator.

Encoder	FID ↓	Prec. ↑	Rec. ↑	KID $\times 10^3$ ↓	Rel. time ↓
VAE	1.66 ± 0.06	.84	.62	.69	1.14
ResNet-50	1.63 ± 0.05	.85	.61	.66	1.16
CLIP-B/16	1.58 ± 0.05	.86	.61	.61	1.31
MAE-B	1.61 ± 0.06	.84	.63	.64	1.32
DINOv2-B	1.60 ± 0.05	.85	.63	.63	1.36
DINOv3-S	1.56 ± 0.05	.85	.62	.60	1.24
DINOv3-B	1.51 ± 0.04	.86	.63	.56	1.37
DINOv3-L	$\mathbf{1.47 \pm 0.04}$	**.86**	**.64**	**.54**	1.55

Table A7: Negative controls for representation quality on ImageNet 256. Random features do not reproduce the gain of pretrained representations.

Representation / objective	FID ↓
Pixel drifting	1.69
Gaussian random projection	1.68
Random ConvNet	1.71
Random ViT-B	1.74
Pretrained VAE	1.66
Pretrained DINOv3-L	**1.47**

batch size of 256. Table A14 reports ImageNet 256 FID, precision, and recall for all sweeps. For α and λ_{LPIPS}, we additionally report cFreD from the corresponding text-conditioned experiment to check whether the same setting transfers beyond class-conditional generation.

The drift step α determines how far the stopped pixel target is moved along the normalized pullback direction. Small values weaken the correction: reducing α from .75 to .50 increases ImageNet FID from 1.47 to 1.54 and cFreD from 3.03 to 3.12. Increasing the step beyond the default also degrades performance. At $\alpha = 1.00$, FID rises to 1.53, recall drops from .64 to .61, and cFreD increases to 3.11. The intermediate values show a smooth trend around the default rather than a sharp threshold. This is consistent with the role of α as a target-step scale: the correction should be large enough to provide useful movement toward the current distribution field, while an overly large target displacement makes the single-step regression more aggressive.

The LPIPS sweep separates the distribution correction from the local auxiliary term. With $\lambda_{\text{LPIPS}} = 0$, the normalized pullback alone already achieves FID 1.49, confirming that LPIPS is not responsible for the main improvement over the unnormalized target. Its contribution appears more clearly in coverage and conditional alignment: adding the term increases recall from .61 to .64 and improves cFreD from 3.20 to 3.03 at the default weight. Larger values do not continue to improve the result. At $\lambda_{\text{LPIPS}} = .60$, FID increases to 1.55 and precision falls to .84. The sweep therefore supports using LPIPS as a moderate local regularizer rather than allowing it to dominate the target-regression objective.

The RBF bandwidth controls the spatial scale at which the minibatch distribution discrepancy is measured. We express the tested values as multipliers of the bandwidth returned by the median heuristic. The default multiplier of 1.0 gives FID 1.47 with precision .86 and recall .64. Narrowing the kernel to .50 increases FID to 1.55, while a multiplier of 2.0 gives FID 1.54. Moderate changes are less consequential: .75 and 1.50 obtain FID 1.50 and 1.49, respectively. This relatively shallow sensitivity around the default indicates that the result does not depend on a finely tuned kernel width, although excessively narrow or broad kernels weaken the correction field.

Finally, the batch-size sweep tests the dependence of the minibatch MMD estimate on the number of samples used to construct the field. Reducing the effective batch size from 256 to 128 worsens FID from 1.47 to 1.52 and slightly reduces both precision and recall. Increasing it to 512 gives FID 1.46, only .01 below the default, while precision and recall remain unchanged at .86 and .64. The small difference between batches 256 and 512 suggests that the discrepancy estimate is already

Table A8: Raw pullback imbalance and benefit from post-pullback sample normalization. The gain is $\mathrm{FID}_{\text{no norm}} - \mathrm{FID}_{\text{sample norm}}$.

Encoder	Raw CV	No norm FID	Sample norm FID	Gain ↑
VAE	.74	1.72	1.68	.04
ResNet-50	.81	1.70	1.65	.05
MAE-B	.89	1.69	1.63	.06
CLIP-B/16	.96	1.68	1.60	.08
DINOv3-S	.99	1.68	1.58	.10
DINOv2-B	1.03	1.71	1.62	.09
DINOv3-B	1.18	1.66	1.53	.13
DINOv3-L	1.31	1.64	1.49	**.15**

Table A9: Continuous normalization intervention on ImageNet 256. Top-10% mass is the fraction of total correction magnitude contributed by the largest 10% of samples. Spike rate is measured using the same training-stability criterion for all rows.

γ	Target CV	Top-10% mass	Grad. CV	Spike (%)	FID ↓	Prec.	Rec
0	1.31	41.8	1.49	8.1	1.64	.84	.60
.25	.92	34.6	1.24	6.2	1.59	.84	.61
.50	.56	27.0	1.02	4.3	1.54	.85	.62
.75	.24	18.1	.84	2.9	1.50	.86	.62
1.00	**.018**	**10.4**	**.76**	2.2	**1.49**	**.87**	.61
1.00 + LPIPS	.019	10.5	.79	**1.8**	**1.47**	.86	**.64**

sufficiently stable at the default batch size. More broadly, the target-construction gain is preserved throughout the tested batch range and is not tied to a single minibatch configuration.

Across the four sweeps, the default configuration lies within a broad region of stable performance. The largest degradations occur when the target step, local regularization weight, or kernel scale is moved substantially away from its default value, while nearby settings produce comparatively small changes. The sensitivity results therefore support the use of one fixed configuration across the main experiments rather than resolution- or task-specific tuning.

H Generator Scaling and Computational Cost

We test whether the target construction remains useful when the generator capacity is increased and provide a component-wise accounting of the additional training cost.

We further decompose the training-time overhead into the operations required to construct the regression target. Table A16 reports each component relative to the cost of the corresponding pixel-drifting generator forward/update, which is normalized to 1.00 at every resolution. The values therefore represent relative computation rather than absolute wall-clock time. For example, a total of 1.55 at 256 means that one training iteration of the full objective takes approximately $1.55\times$ the time of the matched pixel-drifting iteration. The decomposition includes the frozen representation forward pass, construction of the minibatch MMD discrepancy, differentiation of that discrepancy with respect to generated features, the vector-Jacobian product that transfers the feature correction to pixels, and the auxiliary LPIPS computation.

The largest method-specific cost is the vector-Jacobian product. At 256, it adds 0.21 relative units, compared with 0.17 for the frozen encoder, 0.07 for computing the feature correction, 0.04 for MMD itself, and 0.06 for LPIPS. The same ordering remains at higher resolutions. The VJP cost rises to 0.29 at 512 and 0.39 at 1024, accounting for roughly 38% of the additional training overhead at all three resolutions. This reflects the need to propagate the feature-space correction through the representation network back to a higher-dimensional image tensor.

Table A10: Optimization statistics for the principal normalization variants. Max seed spread is the largest observed FID range across the matched repeated runs during training.

Variant	Target CV	Grad. CV	Spike (%)	Max seed spread
No normalization	1.31	1.49	8.1	.17
Batch RMS	1.30	1.33	4.9	.13
Per-sample	.018	**.76**	2.2	**.10**
Full + LPIPS	.019	.79	**1.8**	**.10**

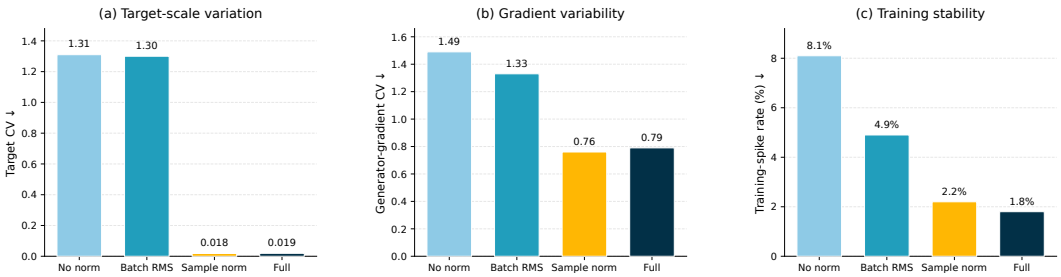

Figure A2: Optimization diagnostics for different normalization rules. **(a)** Variation in pixel-target correction scale. **(b)** Variation in the resulting generator gradients. **(c)** Training-spike rate under the same optimization protocol.

The frozen encoder is the second largest source of overhead, increasing from 0.17 at 256 to 0.27 at 1024. This increase follows the larger spatial input processed by the representation network. The cost of differentiating the feature discrepancy also grows with resolution, from 0.07 to 0.14, because the correction must be propagated through a larger representation computation. LPIPS shows a similar increase, from 0.06 to 0.19, as its local perceptual comparison operates on increasingly large images.

By comparison, the MMD computation remains inexpensive: its relative cost is 0.04, 0.04, and 0.05 across the three resolutions. The discrepancy is computed after images have been mapped to the representation space, so its cost depends primarily on minibatch and feature dimensions rather than directly on the number of image pixels. The main computational burden therefore comes from transferring information between feature and pixel spaces, rather than from evaluating the distribution discrepancy itself.

These components together increase the total training-time ratio from $1.55\times$ at 256 to $1.77\times$ at 512 and $2.04\times$ at 1024. The increase is concentrated in the encoder, VJP, feature-gradient, and LPIPS terms, all of which involve image-resolution-dependent computation. This profiling also clarifies the method's compute tradeoff: additional training cost is used to construct a normalized perceptual target, while none of these operations are retained for generation. At inference, the frozen encoder, MMD computation, feature gradient, VJP, and LPIPS branch are removed, and the model uses only a single generator forward pass.

The VJP is the largest method-specific runtime component, and its relative cost grows with resolution. Peak memory shows a different pattern: the gap is largest at 256 and narrows at higher resolutions because the pixel baseline itself becomes increasingly memory intensive. These costs do not affect inference, where only the generator is retained.

I Evaluation Robustness and Representation Blind Spots

Because the default training signal uses DINOv3-L, the primary conclusions do not rely on a DINO-based evaluation metric. We audit FID implementations and stress the frozen representation with perturbations for which invariance may be undesirable.

We additionally examine whether the frozen representation used to construct the training signal is uniformly sensitive to visually meaningful image changes. This question matters because a represen-

Table A11: FID during training. The advantage of sample normalization is visible well before convergence and remains through the final checkpoint.

Variant	50k	100k	200k	300k	400k
No normalization	9.40	5.78	2.83	1.96	1.64
Batch RMS	8.70	5.23	2.60	1.81	1.55
Per-sample	7.80	4.69	2.34	1.66	1.49
Full + LPIPS	**7.60**	**4.56**	**2.28**	**1.62**	**1.47**

Table A12: Five-seed FID results for the critical target-geometry comparison on ImageNet 256. The reported means match the values used in the main text.

Variant	S1	S2	S3	S4	S5	Mean
Feature target	1.50	1.53	1.56	1.61	1.60	1.56
No normalization	1.55	1.60	1.64	1.72	1.69	1.64
Batch RMS	1.49	1.52	1.55	1.62	1.57	1.55
Per-sample	1.44	1.47	1.49	1.54	1.51	1.49
Full + LPIPS	**1.42**	**1.45**	**1.47**	**1.52**	**1.49**	**1.47**

tation can provide useful semantic geometry while remaining comparatively invariant to particular local or low-level distortions. If the same representation were also used as the primary evaluation metric, such invariances could make improvements appear larger than they are perceptually. We therefore apply controlled perturbations to the same source images and compare the response of the default DINO diagnostic with an independent representation metric and LPIPS. The independent representation is kept separate from the training encoder and is used only as an evaluation control.

Table A19 shows that the three diagnostics respond differently to the same image perturbations. For color shifts, the DINO response is substantially smaller than the independent representation response at both severities. A mild shift produces changes of .09 and .23, respectively, while the strong shift gives .34 and .79. LPIPS is also relatively insensitive to the mild color change, with a response of .07, but increases to .19 under the stronger perturbation. This pattern indicates that the training representation is comparatively tolerant to appearance changes that preserve much of the underlying object and spatial structure.

Blur shows a similar but not identical pattern. Under mild blur, DINO changes by only .05, compared with .14 for the independent metric and .11 for LPIPS. At strong blur, the gap remains visible: the DINO response rises to .25, while the independent metric and LPIPS reach .50 and .32. This difference is relevant for one-step generation because blur can substantially affect local image quality without necessarily altering high-level semantic content. A representation optimized for semantic structure may therefore understate some degradation in fine spatial detail.

JPEG artifacts and high-frequency noise provide a related test of local distortions. JPEG compression produces only a .04 DINO change at mild severity and .20 at strong severity, compared with .08 and .30 for the independent metric. High-frequency noise produces DINO responses of .08 and .31, again below the corresponding independent responses of .12 and .46. LPIPS is comparatively sensitive to high-frequency noise, reaching .14 even at mild severity and .37 at strong severity. These measurements are consistent with the complementary role of LPIPS in our training objective: the frozen representation supplies the distribution-level correction, while the local perceptual term remains more responsive to several forms of fine-scale corruption.

Cropping behaves differently from the other perturbations. DINO changes by .48 for mild cropping and 1.37 for strong cropping, which is comparable to or larger than the independent representation response of .43 and 1.20. LPIPS remains lower at .13 and .29. This suggests that DINO is not simply less sensitive to all perturbations. Its response depends on the type of image change: spatial removal

Table A13: LPIPS pairing strategies on ImageNet 256. Local failure rate is measured on the same audit protocol for all rows.

Pairing	FID ↓	Prec.	Rec.	Local failure (%) ↓
No LPIPS	1.49	.87	.61	19.8
Fixed same class	1.48	.86	.63	16.7
Random same class	1.47	.86	**.64**	14.4
Nearest same class	**1.46**	.86	**.64**	**13.8**
Soft same-class weighting	1.47	.86	**.64**	14.0

Table A14: Hyperparameter sensitivity. Defaults are highlighted. cFreD is reported when the same parameter is evaluated in the text-conditioned setting.

Parameter	Value	FID ↓	Prec.	Rec.	cFreD ↓
Drift step α					
α	.50	1.54	.85	.63	3.12
α	.65	1.50	.86	.63	3.07
α	**.75**	**1.47**	**.86**	**.64**	**3.03**
α	.90	1.49	.86	.62	3.06
α	1.00	1.53	.85	.61	3.11
LPIPS weight λ_{LPIPS}					
λ_{LPIPS}	0	1.49	.87	.61	3.20
λ_{LPIPS}	.15	1.48	.86	.63	3.10
λ_{LPIPS}	**.30**	**1.47**	**.86**	**.64**	**3.03**
λ_{LPIPS}	.45	1.50	.85	.64	3.05
λ_{LPIPS}	.60	1.55	.84	.63	3.09
RBF bandwidth multiplier					
Bandwidth	.50	1.55	.85	.62	–
Bandwidth	.75	1.50	.86	.63	–
Bandwidth	**1.00**	**1.47**	**.86**	**.64**	–
Bandwidth	1.50	1.49	.85	.63	–
Bandwidth	2.00	1.54	.84	.62	–
Global effective batch size					
Batch	128	1.52	.85	.62	–
Batch	**256**	**1.47**	**.86**	**.64**	–
Batch	512	1.46	.86	.64	–

and composition changes produce a strong response, whereas color, blur, compression artifacts, and high-frequency noise are attenuated more strongly. The stress test therefore reveals a structured sensitivity profile rather than a uniformly weak perceptual metric.

These differences are important for interpreting the main experiments. The frozen representation is used to define the training correction, not as the sole criterion for judging final image quality. A model could reduce representation-space discrepancy while leaving distortions that are only weakly expressed by that representation. For this reason, the main evaluation uses FID, KID, CMMD, precision, recall, and independent text-image alignment metrics, with representation-dependent distances treated only as diagnostics. The consistency of the main ordering across these independent metrics reduces the risk that the reported improvement is an artifact of the particular representation used to construct the drift field.

The stress test also clarifies the scope of the method. Post-pullback normalization controls how feature-space corrections are weighted after they pass through the encoder Jacobian, but it does not remove the semantic biases or invariances of the encoder itself. If a representation assigns a small change to a visually important distortion, the resulting correction can still underemphasize that direction. Improving the representation or combining complementary feature spaces is therefore

Table A15: Generator scaling on ImageNet 256. Relative training time is normalized by pixel drifting at the corresponding generator size.

Generator / objective	FID ↓	Prec.	Rec.	KID $\times 10^3$	Rel. time
Approximately 120M parameters					
Pixel drifting	1.69	.84	.62	.74	1.00
+ perceptual loss	1.65	.85	.61	.68	–
Ours	**1.47**	**.86**	**.64**	**.54**	1.55
Approximately 320M parameters					
Pixel drifting	1.56 ± 0.05	.86	.64	.50	1.00
+ perceptual loss	1.53 ± 0.05	.86	.63	.47	–
Ours	$\mathbf{1.42 \pm 0.04}$	**.87**	**.65**	**.40**	1.39

Table A16: Training-time component breakdown relative to the matched pixel-drifting generator forward/update at each resolution. The generator cost is normalized to 1.00. Encoder denotes the frozen representation forward pass; Feature grad. denotes differentiation of the MMD discrepancy with respect to generated features; VJP denotes the vector-Jacobian product $J_\phi(\hat{x})^\top s$ used to pull the correction back to pixels. All additional components are used only during training.

Res.	Generator	Encoder	MMD	Feature grad.	VJP	LPIPS	Total
256	1.00	.17	.04	.07	.21	.06	**1.55**
512	1.00	.22	.04	.10	.29	.12	**1.77**
1024	1.00	.27	.05	.14	.39	.19	**2.04**

orthogonal to the scale-control mechanism studied in this work and remains a possible direction for extending the method.

J Additional Conditional Generation Results

We evaluate whether the target-geometry ordering persists when the discrepancy is prompt conditioned. The text branch is frozen and only the image component of the joint-space correction is pulled back to pixels.

The same ordering observed on ImageNet remains visible in the conditional setting: the raw pullback is weaker than the matched feature target, batch scaling recovers part of the difference, and post-pullback sample normalization provides the strongest target before LPIPS is added.

For BLIP3o 60k, CLIPScore is computed with OpenCLIP ViT-L/14, separate from the frozen CLIP text encoder used for conditioning. Each prompt produces one image under the same one-step inference protocol.

K Failure Analysis

We complement the aggregate generation metrics with a fixed 100-sample audit at each reported resolution. Each sample is assigned one mutually exclusive label so that the counts in every row sum to 100. A sample is marked as *success* when none of the predefined failure categories is observed. The named failure categories capture the recurring errors found in one-step generation: *texture repetition* denotes duplicated or periodically repeated local patterns; *semantic drift* denotes a visible mismatch

Table A17: Peak GPU memory (GB) for pixel drifting and the full objective.

Resolution	Pixel drifting	Full objective
256	18.8	31.1
512	31.4	37.6
1024	37.0	39.2

Table A18: Independent FID implementation audit on ImageNet 256. The ordering is stable across three implementations; KID is included as an additional distributional metric.

Method	CleanFID	torch-fidelity	Custom	KID $\times 10^3$
Pixel drifting	1.69	1.71	1.68	.74
+ perceptual loss	1.65	1.66	1.64	.68
Feature target	1.56	1.58	1.55	.60
Ours	**1.47**	**1.49**	**1.46**	**.54**

in object identity, structure, or class-consistent content; *edge tearing* covers broken, discontinuous, or locally distorted object boundaries; and *blur* denotes loss of local sharpness that is not better explained by another category. Remaining visible failures are grouped under *other*. The taxonomy is intended as a compact diagnostic of the dominant failure mode rather than a complete perceptual quality score.

Table A22 shows that the failure profile changes systematically with resolution. At 256, the normalized target without the local LPIPS term already produces mostly successful samples, and adding LPIPS raises the success count from 83 to 88. The improvement is concentrated in local appearance errors: texture repetition decreases from 4 to 2 cases, edge tearing from 3 to 2, and blur from 4 to 2, while semantic drift is unchanged at 3. This pattern is consistent with the auxiliary role assigned to LPIPS in the main method: it primarily improves local structure after the distribution-level correction has already been constructed.

The same trend becomes more visible at 512. Without LPIPS, 76 of the 100 audited samples are successful. The full objective increases this number to 83, while texture repetition decreases from 7 to 4, edge tearing from 5 to 3, and blur from 5 to 3. Semantic failures remain at 4 cases. The increasing number of local failures relative to 256 reflects the greater difficulty of producing fine spatial detail in a single generator pass. The fact that LPIPS reduces several of these local categories without changing the semantic count also separates its effect from the representation-space distribution correction.

At 1024, the audit exposes a larger gap between the target-construction variants. Pixel drifting succeeds on 62 of 100 samples, leaving texture repetition as the most common named failure with 12 cases, followed by semantic drift with 10, edge tearing with 7, and blur with 5. Adding a perceptual loss to the pixel-drifting objective improves the success count to 69, but substantial texture and semantic errors remain. Post-pullback sample normalization without LPIPS reaches 70 successful samples. Its most visible change relative to pixel drifting is semantic drift, which decreases from 10 to 6 cases, while texture repetition falls from 12 to 10. The full objective raises the success count further to 78 and reduces every named failure category relative to pixel drifting: texture repetition decreases to 7, semantic drift to 5, edge tearing to 4, and blur to 3.

The comparison between sample normalization and the full objective is also informative. Both variants use the same post-pullback scale control, but adding LPIPS increases the number of successful samples from 70 to 78. The largest additional reductions occur in texture repetition and edge tearing, which decrease from 10 to 7 and from 7 to 4, respectively. This again suggests complementary roles for the two components. Post-pullback normalization changes how samples contribute to the generator update, while the local perceptual term reduces several residual image-space artifacts.

Resolution nevertheless remains a clear limitation. Even with the full objective, the success count decreases from 88 at 256 to 83 at 512 and 78 at 1024. The degradation is distributed across several categories rather than being dominated by a single failure type. Texture repetition grows from 2 to 7

Table A19: Perturbation stress test for representation blind spots. Values report the change measured by the default DINO diagnostic, an independent representation metric, and LPIPS. Larger values indicate a stronger response to the applied perturbation. The independent metric is intentionally distinct from the training representation.

Perturbation	Severity	DINO change	Independent change	LPIPS change
Color shift	Mild	.09	.23	.07
Color shift	Strong	.34	.79	.19
Blur	Mild	.05	.14	.11
Blur	Strong	.25	.50	.32
Crop	Mild	.48	.43	.13
Crop	Strong	1.37	1.20	.29
JPEG artifacts	Mild	.04	.08	.06
JPEG artifacts	Strong	.20	.30	.20
High-frequency noise	Mild	.08	.12	.14
High-frequency noise	Strong	.31	.46	.37

Table A20: Target-geometry comparison for BLIP3o 60k text-to-image generation at 256 resolution.

Target construction	cFreD ↓	CLIPScore ↑	FID ↓
Feature target, same encoder	3.18	23.05	34.62
Pixel pullback, no norm	3.29	22.92	34.93
Pixel pullback, batch RMS	3.15	23.08	34.48
Pixel pullback, per-sample	3.08	23.18	34.23
Full + LPIPS	**3.03**	**23.23**	**34.10**

cases, semantic drift from 3 to 5, and edge tearing from 2 to 4. These remaining errors are consistent with the constraint of strict one-step generation: the model must establish global structure and fine-scale appearance in one prediction, with no later refinement stage to correct a local or semantic mistake.

We next ask whether these visible failures are related to the pullback-scale imbalance studied in the main paper. For the 1024 audit, samples are divided into quartiles according to the norm of their raw, unnormalized pullback $\|J_\phi(\hat{x})^\top s\|_2$. We then count the failures within each quartile for three target constructions. This analysis does not assume that a large pullback directly causes a failure; it tests whether the optimization diagnostic is associated with the samples on which one-step generation is least reliable.

Table A23 reveals a monotonic relationship between raw pullback magnitude and observed failure frequency. Without normalization, the number of failures increases from 5 in Q1 to 7 in Q2, 10 in Q3, and 15 in Q4. The highest-norm quartile therefore contains three times as many failures as the lowest-norm quartile. This pattern is consistent with the optimization analysis: samples receiving unusually large raw pixel-space corrections are also the samples for which the generator is more likely to produce a visible error.

Post-pullback sample normalization reduces the failure count in every quartile. The reduction is modest in Q1, from 5 to 4, but becomes larger as the raw pullback norm increases: Q3 decreases from 10 to 8, and Q4 from 15 to 12. The total number of failures falls from 37 to 30. Since the normalization leaves each nonzero pullback direction unchanged while controlling its magnitude, this result is consistent with the interpretation that unusually large corrections can receive excessive weight in the unnormalized minibatch update.

Adding LPIPS reduces the counts further to 3, 4, 6, and 9 across Q1–Q4, for a total of 22 failures. The absolute improvement relative to the unnormalized target is largest in the upper quartiles: Q4 decreases by 6 failures and Q3 by 4, compared with reductions of 2 and 3 in Q1 and Q2. The pattern suggests that local regularization is particularly useful for the more difficult samples identified by

Table A21: Additional one-step text-to-image results. All controlled rows use the same generator-side training and one-step inference protocol within each dataset.

Method	Dataset	cFreD ↓	CLIPScore ↑	FID ↓
T2I drifting	BLIP3o 60k	3.62 ± 0.28	22.14 ± 0.33	35.80 ± 0.52
+ perceptual loss	BLIP3o 60k	3.33 ± 0.24	22.71 ± 0.19	34.60 ± 0.41
Ours	BLIP3o 60k	$\mathbf{3.03 \pm 0.22}$	$\mathbf{23.23 \pm 0.18}$	$\mathbf{34.10 \pm 0.37}$
T2I drifting	COCO	3.92	22.35	37.8
+ perceptual loss	COCO	3.62	22.78	36.9
Ours	COCO	**3.38**	**23.12**	**36.2**

Table A22: Failure taxonomy across resolutions. Each row contains counts out of 100 audited samples. Labels are mutually exclusive, so each sample contributes to exactly one category.

Res.	Method	Success	Texture	Semantic	Edge	Blur	Other
256	No LPIPS	83	4	3	3	4	3
256	Full	**88**	**2**	3	**2**	**2**	3
512	No LPIPS	76	7	4	5	5	3
512	Full	**83**	**4**	4	**3**	**3**	3
1024	Pixel drifting	62	12	10	7	5	4
1024	+ perceptual loss	69	10	8	6	4	3
1024	Sample norm, no LPIPS	70	10	6	7	4	3
1024	Full	**78**	**7**	**5**	**4**	**3**	**3**

the raw-pullback diagnostic, although the effect remains complementary to normalization rather than replacing it.

The quartile analysis provides a connection between the internal optimization measurement and observable generation quality, but it should be interpreted as an association rather than a causal identification result. A large raw pullback can arise because a generated sample is already far from the current feature distribution, because the local encoder Jacobian amplifies the correction, or from a combination of both effects. Such samples may therefore be intrinsically more difficult before any normalization is applied. The controlled normalization experiments in the main paper provide the direct intervention on correction scale; the quartile analysis instead shows that the same scale statistic is concentrated on visually difficult examples.

Together, the two audits clarify the remaining failure regime. Increasing resolution makes one-step generation more prone to both local artifacts and semantic errors, while post-pullback normalization reduces failures most strongly among samples with large raw corrections. LPIPS provides an additional reduction in local artifacts and coverage failures. Neither component removes the fundamental difficulty of producing a high-resolution image in a single forward pass, but their effects are complementary and are visible both in aggregate metrics and in sample-level failure analysis.

The quartile analysis connects the optimization diagnostic to visible generation errors. Samples with the largest raw pullbacks are more failure prone, and the reduction from scale control is largest in the upper quartiles.

Figure A3 complements the aggregate metrics with representative 1024-resolution generations. We include both successful samples and remaining failure cases rather than selecting only visually favorable outputs.

Successful examples preserve global object structure while maintaining fine local detail under a single generator evaluation. The remaining errors largely match the audit categories reported above, including texture repetition, semantic drift, edge tearing, and blur. These examples also illustrate the remaining limitation of strict one step generation: once a local or semantic error is produced, there is no later refinement stage in which it can be corrected.

Table A23: Failure count at 1024 stratified by quartiles of the raw pullback norm. Each quartile contains the same number of audited samples. Larger raw corrections are associated with more failures for every target construction, while post-pullback sample normalization and LPIPS reduce the count within each quartile.

Raw-norm quartile	No normalization	Sample normalization	Full
Q1 (lowest)	5	4	**3**
Q2	7	6	**4**
Q3	10	8	**6**
Q4 (highest)	15	12	**9**
Total	37	30	**22**



Figure A3: Qualitative results at 1024 resolution under one step inference. The top row shows representative successful generations with preserved global structure and local detail. The bottom row shows remaining failure modes, including texture repetition, semantic drift, edge tearing, and blur. Enlarged crops highlight the corresponding local artifacts.

L Implementation and Reproducibility Details

Datasets and preprocessing. For class-conditional generation, we train on the ImageNet2012 training split and evaluate using the validation split. At each target resolution $R \in \{256, 512, 1024\}$, training images use random resized cropping to $R \times R$ and random horizontal flipping. Evaluation uses deterministic resize and center crop with the same image normalization for every method. For text-to-image generation, BLIP3o 60k uses a fixed train/validation split and a shared image pipeline; the default conditional experiments use 256×256.

Generator and representations. The default controlled experiments use the same approximately 120M-parameter one-step generator across methods. The scaling study additionally uses an approximately 320M-parameter generator. The default frozen image representation is DINOv3-L. Its global token and mean-pooled patch representation are individually normalized and concatenated as defined in the main method section. The representation sweep replaces this encoder while keeping the remaining training protocol fixed. For text conditioning, prompts are encoded with a frozen

CLIP text encoder and injected through the same conditioning interface for all compared methods. The joint image-text discrepancy uses $\beta = 1$.

Distribution discrepancy and pullback. We use the biased empirical squared MMD written in the main text with an RBF kernel. Unless a bandwidth sensitivity row is being evaluated, the kernel bandwidth is selected by the minibatch median heuristic. Sample corrections are obtained from the gradient of the discrepancy with respect to each generated feature. The Jacobian is never materialized explicitly; the product $J_\phi(\hat{x}_j)^\top s_j$ is computed as a vector-Jacobian product with automatic differentiation. The same fixed numerical stabilizer ϵ is used in every normalization experiment.

Optimization. Unless stated otherwise, ImageNet models are trained for 400,000 optimizer updates with AdamW, learning rate 2×10^{-4}, cosine decay with warmup, drift step size $\alpha = 0.75$, LPIPS weight $\lambda_{\mathrm{LPIPS}} = 0.30$, and bf16 precision. The global effective batch size is 256. At 256 resolution the per-device batch size is 64. At 512 we use batch size 16 with gradient accumulation 4, and at 1024 batch size 4 with gradient accumulation 16. Controlled runs use four A100 40GB GPUs. Critical target-geometry comparisons are repeated with five fixed random seeds; other controlled experiments use three seeds unless stated otherwise.

LPIPS pairing. For class-conditional ImageNet, each generated image is paired with a real image of the same class when such a sample is available in the minibatch; otherwise a fixed within-batch fallback pairing is used. The default reported configuration uses random same-class pairing. In the text-conditioned setting, LPIPS uses the paired real image for the prompt.

Evaluation. Every reported sample is generated with one network evaluation. For ImageNet, we generate 50,000 samples per method and resolution. We report FID, Inception Score, Improved Precision and Recall with $K = 5$, KID, and CMMD using fixed reference statistics. Representation-dependent distances are treated as diagnostics rather than primary evidence. For BLIP3o 60k, we report cFreD, CLIPScore, and FID; CLIPScore uses OpenCLIP ViT-L/14, separate from the conditioning text encoder. Runtime and memory comparisons use matched hardware, batching, and software settings within each resolution.

Release. Code and configuration files for the generator training loop, frozen representation interfaces, MMD construction, vector-Jacobian pullback, normalization variants, evaluation scripts, and reported random seeds will be released where permitted by the submission policy.