dREPA: Difficulty-Aware Representation Alignment Accelerates Diffusion Training

Anonymous authors
Paper under double-blind review

Abstract

Representation alignment (REPA) accelerates diffusion transformer training by aligning intermediate features with a frozen encoder, but it averages both the denoising and the alignment loss uniformly over spatial tokens, although denoising difficulty concentrates on boundaries and fine structure. We show that the per-token alignment loss, which REPA already computes and discards, is a free spatial difficulty signal: tokens that are hard to align tend to be hard to denoise, a weak but structured correspondence that is stable across timesteps and training. Turning a difficulty signal into useful weights is not automatic, however. Reweighting the denoising loss by its own per-token residual *hurts*, and a matched comparison of signals through one pipeline points to two properties a useful signal needs: it should rank tokens consistently across noise draws, and it should stay current with the model. The online residual lacks the first, a frozen signal lacks the second, and the alignment loss has both at no extra cost. We propose **dREPA**, which converts a per-sample-normalized alignment signal into bounded spatial weights for the denoising loss in five lines of code with no additional parameters. dREPA is a training-acceleration method: on SiT-XL/2 it reaches at 1M steps the FID that iREPA reaches at 4M, and with guidance tuned independently for each model it improves FID from 2.21 to 2.04. The gain is consistent across model scales, teacher encoders, latent and pixel-space backbones, resolutions, and class-conditional and text-to-image generation, and on SiT-XL/2 it exceeds measured run-to-run variation. Code: `https://anonymous.4open.science/r/dREPA-17C2/README.md`.

1 Introduction

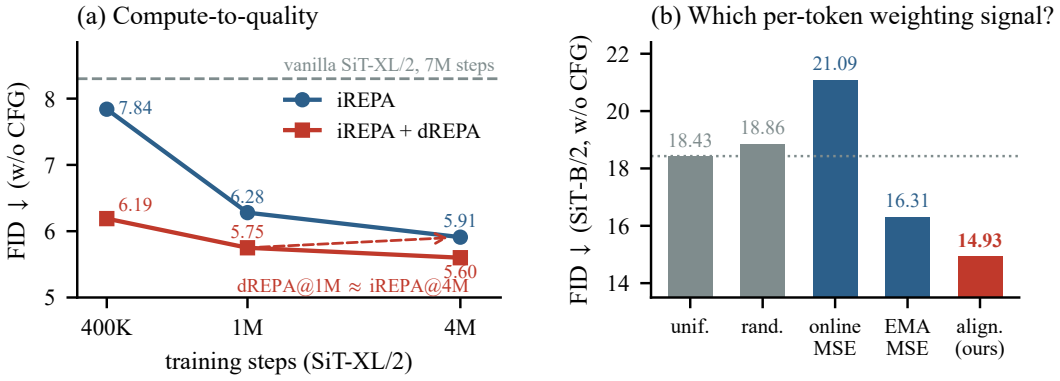

Figure 1: **dREPA in one figure.** **(a)** Compute-to-quality on SiT-XL/2 (without classifier-free guidance, CFG; 50K-sample FID): with the same recipe and only the per-token weighting changed, dREPA at 1M steps reaches the FID of iREPA at 4M (5.75 vs. 5.91; one run per point), and dREPA is ahead at each of the three horizons. The dashed line is vanilla SiT-XL/2 after 7M steps (Ma et al., 2024). **(b)** The same weighting pipeline driven by different per-token signals (SiT-B/2, 400K steps): the denoising loss reweighted by its own online residual is *worse* than uniform, an EMA-model residual helps, and the alignment loss helps most while costing nothing extra; only the signal differs across bars, so the comparison isolates the effect of the signal's source.

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.

Diffusion transformers dominate image generation (Peebles & Xie, 2023; Ma et al., 2024; Wu et al., 2025a; 2026a; Deng et al., 2025; Xia et al., 2025; Liu et al., 2025b), and reducing their training cost is an active problem. Representation alignment methods such as REPA (Yu et al., 2025) and its variants (Leng et al., 2025; Wang et al., 2026; Yao et al., 2025; Singh et al., 2026) accelerate training by matching intermediate features to a frozen pretrained encoder such as DINOv2 (Oquab et al., 2024). The alignment loss is computed *per spatial token*: it measures, for every location, how well the intermediate representation matches the teacher feature. Yet this spatially resolved signal is averaged uniformly over tokens before it enters the objective, as is the denoising loss itself.

Uniform averaging ignores a well-known property of images: denoising difficulty is spatially non-uniform, concentrated on object boundaries, thin structures and fine detail. On SiT-B/2 the hardest 10% of tokens carry over 30% of the total denoising error (Sec. 3). This suggests spatially adaptive supervision, i.e. larger denoising weights on harder tokens.

The natural implementation, reweighting each token's denoising loss by its own residual, is counterproductive: Fréchet inception distance (FID) degrades from 18.43 to 21.09 on SiT-B/2 at every weighting strength we test (Tab. 1). A matched comparison of weighting signals through an identical pipeline (Fig. 1b) points to two properties that a useful signal needs. **(i) Persistence:** it should rank tokens consistently across noise draws. The online residual is computed from the freshly drawn noise and reranks tokens from step to step (mean cross-draw rank correlation 0.569), so upweighting it largely amplifies the current draw. **(ii) Currency:** it should track the current model; a signal frozen early in training stops helping (FID 19.60). Reading the residual from an exponential-moving-average (EMA) model is only somewhat more persistent (0.657) and already recovers part of the gain (16.31).

The per-token alignment loss has both properties at no extra cost. Its target $f(\mathbf{x}_0)$ is fixed per image, so its token ranking is far more persistent (0.853); it is recomputed from the current backbone at every step; and it tracks denoising difficulty in aggregate, a weak but structured correspondence that is stable across timesteps and training. Used as the weighting signal it improves FID from 18.43 to 14.93, and the component of it that a 16-bin control on current residual and timestep cannot explain still recovers most of the gain (15.86), so it is not a restatement of the residual.

We therefore propose **dREPA** (**d**ifficulty-aware **REPA**), which converts the per-token alignment loss into bounded, detached spatial weights for the denoising objective (Sec. 4). The weights are normalized within each image, so they encode *relative* difficulty: before clamping, every sample and timestep receives the same total coefficient $1 + \lambda$, and a control that fixes it to exactly 1 retains the gain (Sec. 4.3). Only the denoising loss is reweighted; the alignment loss stays uniform so that it keeps serving as the sensor. dREPA adds five lines of code and no parameters to any REPA variant.

dREPA is a training-acceleration method, and we evaluate it as one. On SiT-XL/2 with an identical recipe, dREPA at 1M steps (FID 5.75) reaches what iREPA (Singh et al., 2026), the improved REPA recipe we build on, reaches at 4M (5.91), and dREPA is ahead at each of the three evaluated horizons (Fig. 1a); with guidance tuned independently for each model it improves $2.21 \rightarrow 2.04$ on XL/2 and $5.71 \rightarrow 4.38$ on B/2. The gain is consistent across scales (B/2 to XL/2), teachers (DINOv2 (Oquab et al., 2024), DINOv3 (Siméoni et al., 2025), PE-Core-G (Bolya et al., 2025)), backbones (SiT, pixel-space JiT, MMDiT), resolutions (256^2, 512^2) and tasks (class-conditional and text-to-image), and on SiT-XL/2 it exceeds the run-to-run variation measured in a matched-seed study (Sec. 5). Our contributions are:

- **Analysis.** The contribution is not hard-token mining but identifying what makes a token-difficulty signal usable. The alignment loss tracks denoising difficulty, and matched controls show that two properties distinguish it from the model's own residual: it ranks tokens persistently across noise draws and stays current with the model (Secs. 3, 5.4).
- **Method.** dREPA turns this free signal into bounded spatial weights for the denoising loss through within-sample normalization, parameter-free and in five lines of code (Sec. 4).
- **Evaluation.** dREPA shifts the learning curve of REPA-style training horizontally, by up to roughly $4\times$ fewer steps at matched FID on SiT-XL/2 over the evaluated range, and the gain generalizes across scales, encoders, architectures, resolutions and tasks (Sec. 5).

2 Related Work

Diffusion transformers and representation alignment. Diffusion and flow-matching models (Song et al., 2021; Lipman et al., 2023; Liu et al., 2023; Rombach et al., 2022; Liu et al., 2025a) and transformers (Peebles & Xie, 2023; Ma et al., 2024; Ho et al., 2020; Liu et al., 2024) have become prevalent in image generation (Cai et al., 2025; Xiao et al., 2025; Wu et al., 2025b), but remain expensive to train. Representation alignment (Yu et al., 2025; Kouzelis et al., 2025) accelerates training by aligning intermediate features with a frozen pretrained encoder. Extensions have improved the alignment target (Leng et al., 2025; Wang et al., 2026; Yao et al., 2025; Wang & He, 2025), the alignment recipe (Singh et al., 2026), or expand to video and 3D generation (Zhang et al., 2025; Wu et al., 2026b), but all average both losses uniformly over tokens. dREPA is orthogonal: it operates on the *spatial distribution of optimization effort* and can be combined with REPA variants.

Non-uniform training strategies for diffusion. P2 (Choi et al., 2022), Min-SNR (Hang et al., 2023), and EDM (Karras et al., 2022) reweight along the *temporal* axis (across noise levels), treating all tokens identically within each timestep. MDTv2 (Gao et al., 2023) and MaskDiT (Zheng et al., 2024) operate on the spatial axis but via binary *masking*—dropping tokens to reduce cost. dREPA addresses a different point in this design space: *soft* spatial reweighting that preserves all tokens while reallocating gradient budget toward harder regions.

3 Observation: Alignment Loss as a Free Spatial Difficulty Signal

REPA trains a denoising loss jointly with a per-token alignment loss ℓ_i^{repa}:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell_i^{\text{denoise}} + \alpha \frac{1}{N}\sum_{i=1}^{N} \ell_i^{\text{repa}}, \tag{1}$$

where N is the number of spatial tokens and both objectives are averaged uniformly over tokens. Uniform averaging ignores that spatial locations differ widely in denoising difficulty, and it discards the spatial structure that the per-token alignment objective already exposes. Three observations motivate our method. All analyses use SiT-B/2 on ImageNet 256×256 (SD-VAE latents, $N{=}1024$ tokens per image, 2048 held-out validation images).

3.1 Token difficulty is highly non-uniform

The per-token denoising error $d_{h,w} = \frac{1}{C}\sum_c (\hat{y}_{c,h,w} - y_{c,h,w})^2$ concentrates on boundaries and fine structures (Fig. 2, top). Three concentration metrics confirm substantial inequality: coefficient of variation CV$=0.93$, Gini$=0.44$, and the hardest 10% of tokens carry over 30% of the total error (Fig. 2a), stable across 50K–400K steps. Uniform averaging assigns every token the same explicit coefficient $1/N$. Larger-residual tokens already contribute larger gradient magnitudes through $\nabla_{\hat{v}_i}\mathcal{L} = \frac{2}{N}(\hat{v}_i - v_i)$, so uniform weighting does not give all tokens equal optimization influence; what the objective lacks is any auxiliary token-importance multiplier beyond this inherent MSE scaling. The concentration widens further at the top-1% level (Fig. 2b), and edge tokens are harder than non-edge tokens by a ratio that varies with noise level (Appendix C).

3.2 Alignment loss tracks denoising difficulty

The per-token alignment loss ℓ_i^{repa} exhibits a *monotonic* correspondence with denoising difficulty: partitioning tokens into five difficulty bins shows that hard-to-denoise tokens are consistently harder to align, across timesteps and training stages (Fig. 2c, curves). This constrains bin means only. It does not imply $\ell_i^{\text{repa}} = g(m_i)$, nor that the two signals induce the same per-sample token ranking, and the reweighting scheme we propose relies on the latter rather than on the former. Bin means are not the only evidence for the correspondence: the direct top-k overlap between the alignment-hard and denoising-hard token sets rises from 10.9% at $t{=}0.1$ to 16.6% at $t{=}0.8$ against a 10% chance level (top-20%: 21.8% to 27.7% against 20%), and at $t{=}0.6$ it stays within 15.8–16.3% from 50K to 400K steps, so the correspondence is above chance and stable across training rather than an artifact of one checkpoint. The overlap is far from complete: the alignment loss is a weak but structured ranking signal for denoising difficulty, and what makes it usable (Sec. 3.3) is its persistence and

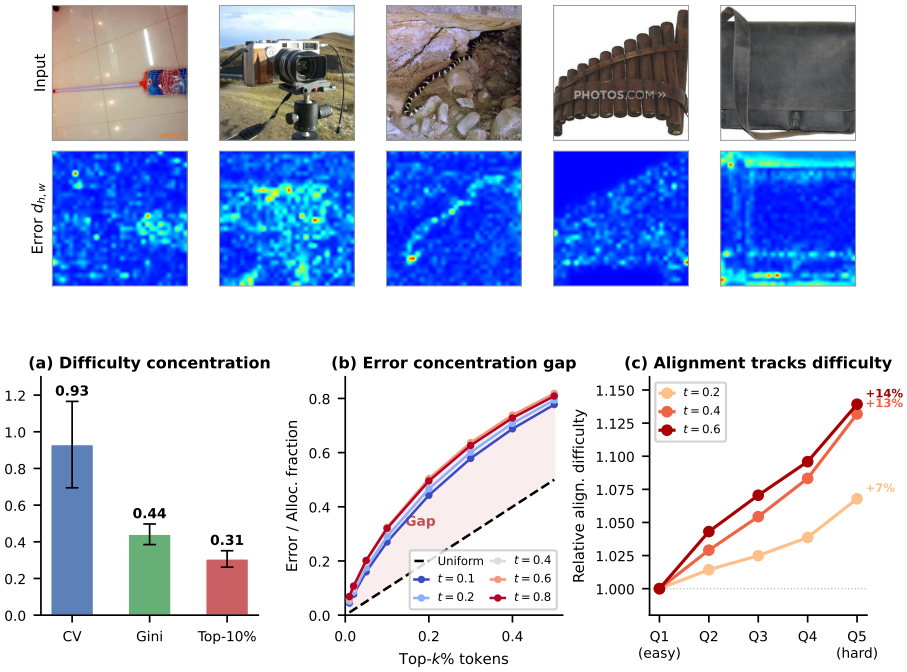

Figure 2: **Token difficulty is non-uniform and tracked by alignment loss. Top:** Per-token error maps concentrate on boundaries. **(a)** Distribution of per-token denoising error over 2048 validation images (N=1024 tokens each): error is strongly non-uniform and spatially concentrated (CV=0.93, Gini=0.44). **(b)** Cumulative error share $C(k) = \sum_{j \le \lceil kN \rceil} m_{(j)} / \sum_j m_j$ for tokens ranked by denoising MSE $m_{(1)} \ge \cdots \ge m_{(N)}$; x: top-k fraction of tokens ranked by denoising MSE, y: cumulative share of denoising MSE. One curve per timestep; dashed line is $C(k) = k$ (spatially uniform error). **(c)** Per-image normalized alignment loss $\mathbb{E}[\ell_i^{\text{repa}} / (\frac{1}{N} \sum_j \ell_j^{\text{repa}}) \mid i \in Q_q, t]$, where Q_1–Q_5 are quintiles of the *same* denoising-MSE ranking and 1 denotes the per-image average; one curve per timestep. SiT-B/2, ImageNet 256×256, 2048 images, N=1024 tokens.

currency rather than its accuracy. A plausible explanation is that the alignment objective and the denoising head share the same backbone features. We do not claim that ℓ_i^{repa} measures intrinsic denoising difficulty: a high alignment loss may also reflect semantic richness, teacher sensitivity, or features the backbone has not yet learned. For our purpose it suffices that it is a proxy for *relative* token difficulty with the two properties of Sec. 3.3. The alignment loss is therefore a *free* spatial importance signal—already computed at REPA training step, yet discarded by uniform averaging. Exploiting this signal requires no additional forward passes and no auxiliary networks.

3.3 The obvious fix fails

Given a per-token difficulty signal, the natural next step is to reweight the denoising loss, assigning larger weights to harder tokens. Tab. 1 drives one and the same reweighting pipeline (Sec. 4: detached signal, per-sample mean normalization, clamping, warmup) with different signals, so that only the *source* of the signal differs across rows.

Self-reference fails. Random per-token weights do not help (18.86 vs. 18.43), and reweighting the denoising loss by its own per-token mean-squared error (MSE) *hurts* (21.09), although this is the most direct estimate of the current denoising difficulty. The failure is not a matter of

Table 1: **What works and what does not** (SiT-B/2, DINOv2-B, 400K steps, w/o CFG). **Top:** one pipeline (Sec. 4), different signals at best strength (Tab. 2). **Bottom:** scope checks that change the objective, hence the baseline.

	Configuration	FID↓	ΔFID
Signal source (same framework, diff. signals)			
1	Uniform (iREPA baseline)	18.43	—
2	Random weights	18.86	+0.43
3	Denoising MSE (self-ref.)	21.09	+2.66
4	Align. difficulty (dREPA, ours)	14.93	−3.50
5	EMA-model MSE	16.31	−2.12
Scope checks			
(i)	No REPA, MSE only	39.06	+20.63
(ii)	Weak align. (α=0.01)	30.50	+12.07
(iii)	Freeze proj. at 50K	19.60	+1.17

Table 2: **Strength sweep and rank persistence** (setting of Tab. 1; uniform baseline 18.43). **Left:** FID with λ swept independently for each signal. **Right:** mean cross-draw rank correlation of each signal's token ranking at fixed images and timesteps, with the FID it yields.

λ	Online MSE	Alignment (ours)
0.1	18.92	17.57
0.25	19.74	16.34
0.5	21.09	**14.93**
1.0	23.16	15.08
1.25	24.38	15.67

Signal	Persistence↑	FID↓
Online MSE	0.569	21.09
EMA-model MSE	0.657	16.31
Alignment (ours)	**0.853**	**14.93**

strength: sweeping λ independently for each signal (Tab. 2), online-MSE weighting degrades monotonically and never improves on uniform training, while alignment weighting improves at every strength. Nor is it a dynamical instability: all runs complete normally and quality degrades smoothly. Algebraically, with detached $w_i = 1 + \lambda m_i/\bar{m}$ the pre-clamp update is locally equivalent to differentiating $m_i + \frac{\lambda}{2\bar{m}} m_i^2$, whereas uniform MSE already gives $\nabla_{\hat{v}_i} m_i = 2(\hat{v}_i - v_i)$: we read this as the rule amplifying the instantaneous residual that already sets the gradient magnitude, rather than adding information about where gradient should be spent.

Two properties. The remaining rows point to two properties that such a signal needs. *(i) Persistence.* The per-token residual is computed from $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{x}_0$ with freshly drawn noise, so it reranks tokens from step to step; at fixed images and timesteps its token ranking has cross-draw rank correlation only 0.569 (Tab. 2). Reading the same statistic from an EMA model makes the ranking somewhat more persistent (0.657) and already turns the sign of the effect (16.31, better than uniform). *(ii) Currency.* A persistent signal that no longer tracks the model does not help either: freezing the alignment projector at 50K steps and weighting with its stale output removes the gain entirely (19.60, worse than uniform). The alignment loss has both properties at no extra cost. Its target $f(\mathbf{x}_0)$ is fixed per image, so its ranking is far more persistent (0.853); it is recomputed from the current backbone every step; and it tracks denoising difficulty (Sec. 3.2). It is the strongest signal we evaluate (14.93, -19%) and, unlike the EMA control, needs no second model. Sec. 5.4 shows that, under a bin-based control, the component of it not explained by the current residual and timestep carries most of the gain. These controls are single-seed, so we treat the two properties as the distinction the evidence supports, not as a proof of necessity.

We do not claim that the alignment loss is statistically exogenous from the denoising loss (both depend on the shared backbone); the distinction is narrower: the online rule sets $w_i = f(m_i)$ for the exact residual being reweighted, whereas the alignment signal is correlated with m_i without being a deterministic transform of that residual.

Appendix D adds a stylized linearized analysis of self-referencing weights; it is a local-sensitivity comparison under strong assumptions, not verified by the results above.

4 METHOD: DREPA

Sec. 3 motivates a cross-objective strategy: use the alignment-derived difficulty signal, rather than the residual being reweighted, to assign spatial weights to the denoising objective. We describe the formulation and the key design choices below.

4.1 PRELIMINARIES

Flow matching. Given clean data $\mathbf{x}_0$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noisy input at time $t \in [0, 1]$ is $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$ (Lipman et al., 2023). The model predicts the conditional velocity field $\mathbf{v}_i = \boldsymbol{\epsilon}_i - (\mathbf{x}_0)_i$ at each spatial token i, yielding the per-token denoising loss:

$$\ell_i^{\text{denoise}} = \left\|\hat{\mathbf{v}}_i - \mathbf{v}_i\right\|^2, \qquad \mathcal{L}_{\text{denoise}} = \frac{1}{N}\sum_i \ell_i^{\text{denoise}}. \tag{2}$$

REPA and iREPA. Let $\mathbf{h}_i^{(l)}$ denote the hidden state of the spatial token i at the backbone layer l. A projector ϕ maps this representation to $\tilde{\mathbf{z}}_i{=}\phi(\mathbf{h}_i^{(l)})$, which is aligned with the corresponding frozen

teacher token $\mathbf{z}_i{=}f(\mathbf{x}_0)_i$ from a pretrained encoder f. We denote the per-token alignment loss by ℓ_i^{repa} and define the alignment objective as its uniform average:

$$\mathcal{L}_{\text{repa}} = \frac{1}{N}\sum_i \ell_i^{\text{repa}}. \tag{3}$$

In practice, $\ell_i^{\text{repa}} = 1 - \cos(\tilde{\mathbf{z}}_i, \mathbf{z}_i)$ measures the cosine distance between the projected backbone feature and the frozen teacher feature on each token. The layer index l is a hyperparameter (default $l{=}4$; swept in Sec. 5.4). The combined baseline objective is

$$\mathcal{L} = \mathcal{L}_{\text{denoise}} + \alpha\,\mathcal{L}_{\text{repa}}. \tag{4}$$

iREPA (Singh et al., 2026) improves REPA with a convolutional projector and spatial normalization. When applying dREPA to REPA or other representation-alignment variants, we inherit the baseline alignment recipe and only replace the uniform denoising average with our spatially weighted one.

4.2 Difficulty-adaptive reweighting

The per-token alignment loss ℓ_i^{repa} is converted into bounded spatial weights in three steps: a power transform with exponent $\gamma \geq 0$ controls the difficulty contrast ($\gamma{=}0$ yields uniform weights); mean normalization makes the signal reflect *relative* difficulty within each sample rather than the absolute alignment magnitude, which may drift during training; and clamping to $[w_{\min}, w_{\max}]$ prevents any token from dominating the denoising objective. With the default positive form $1 + \lambda\hat{d}_i$ and $\hat{d}_i \geq 0$, the lower bound is never active: only $w_{\max}$ binds under the default configuration, and $w_{\min}$ becomes relevant only for the $c{=}0$ variant ablated in Sec. 5. Formally,

$$\hat{d}_i = \frac{(\ell_i^{\text{repa}} + \varepsilon)^\gamma}{\frac{1}{N}\sum_j (\ell_j^{\text{repa}} + \varepsilon)^\gamma}, \qquad w_i = \text{clamp}\big(1 + \lambda\,\hat{d}_i,\ w_{\min},\ w_{\max}\big), \tag{5}$$

where ε is a constant and λ controls the strength of the difficulty bonus. Unless stated otherwise we use $\gamma{=}1$, $\lambda{=}0.5$, $[w_{\min}, w_{\max}]{=}[0.5, 2.0]$, $\varepsilon{=}10^{-8}$ and $T_{\text{warm}}{=}10\text{K}$. Normalization is performed within each sample over its spatial tokens, not across the batch. All methods, baselines included, sample $t \sim \mathcal{U}(0,1)$; no method oversamples any noise regime.

The additive form $1 + \lambda\hat{d}_i$ keeps a baseline coefficient for every token, so no token is discarded. Since $\frac{1}{N}\sum_i \hat{d}_i = 1$, every sample and timestep receives the same pre-clamp mean weight $1 + \lambda$, and $w_i/(1+\lambda) < 1$ exactly when $\hat{d}_i < 1$: up to this sample-independent constant factor, which only rescales the denoising term relative to $\alpha\mathcal{L}_{\text{repa}}$, the default form changes the relative within-sample allocation of denoising weight from easier to harder tokens. The exact sum-preserving control of Sec. 5.4 removes this factor and retains the gain. The final objective replaces the uniform denoising loss with the weighted version while keeping the alignment loss unchanged:

$$\mathcal{L} = \underbrace{\frac{1}{N}\sum_{i=1}^{N} w_i \left\|\hat{\mathbf{v}}_i - \mathbf{v}_i\right\|^2}_{\text{alignment-weighted denoising}} + \alpha\,\mathcal{L}_{\text{repa}}. \tag{6}$$

Algorithm 1 (Appendix A) gives the five-line implementation.

4.3 Design choices

Asymmetric reweighting. Only the denoising loss is reweighted; the alignment loss remains uniform. This asymmetry is important because the alignment objective serves as dREPA's difficulty *sensor*. Reweighting the alignment loss itself would bias the signal toward already hard tokens and distort the spatial difficulty used to guide denoising reweighting. This is a design motivation, not a demonstrated mechanism; Sec. 5.4 reports the directional ablation.

Detach and warmup. The normalized difficulty signal $\hat{d}_i$ is `detach`ed from the computational graph before being used as a weight, so gradients from the reweighted denoising objective do not flow back into the projector through the weighting path. This preserves the structural decoupling between

Table 3: **Model size scaling** (w/o CFG, 400K steps). dREPA improves FID, IS, sFID and Precision at every scale, at a modest cost in Recall (Sec. 5.3).

Model	Method	IS↑	FID↓	sFID↓	Prec.↑	Rec.↑
B/2	iREPA	77.33	18.43	6.161	0.629	0.638
	+ dREPA	88.58	14.93	5.664	0.659	0.626
L/2	iREPA	118.08	8.94	4.961	0.699	0.639
	+ dREPA	132.07	6.93	4.780	0.730	0.623
XL/2	iREPA	124.52	7.84	4.912	0.712	0.644
	+ dREPA	140.45	6.19	4.701	0.735	0.618

Table 4: **Cross-architecture transfer to JiT-B/16** (pixel-space diffusion without a VAE; w/o CFG, ImageNet 256×256). dREPA improves both REPA and iREPA with each of the three teacher encoders.

	DINOv2		*DINOv3*		*PE-Core*	
Method	**FID↓**	**IS↑**	**FID↓**	**IS↑**	**FID↓**	**IS↑**
REPA	54.48	25.42	59.18	21.29	61.24	21.25
+ dREPA	50.51	28.06	55.96	23.43	59.66	21.73
iREPA	48.18	29.59	52.80	26.32	56.38	25.35
+ dREPA	**44.85**	**33.21**	**48.88**	**29.04**	**50.44**	**28.90**

the signal source and the weighted objective. Three quantities are detached: the raw signal $(\ell_i^{\text{repa}}+\varepsilon)^\gamma$, the per-sample normalization statistic, and the final weight w_i. Consequently $\nabla_{\hat{v}_i}\mathcal{L} = \frac{1}{N} w_i \, 2(\hat{v}_i - v_i)$ with no $\nabla_{\hat{v}_i} w_i$ term, while the denoising and alignment objectives backpropagate normally. We also apply a linear warmup of λ from 0 to its target value as a precaution against the early, untrained projector; results are insensitive to it (Sec. 5.4).

Timestep-dependent spatial contrast. Because the spread of ℓ_i^{repa} across difficulty quintiles grows with the noise level (Fig. 2c), the weights show larger spatial contrast at high noise levels and are closer to uniform at low ones, without a timestep-dependent schedule. The factor $1 + \lambda$ carries no sample or timestep dependence, and the exact sum-preserving control ($w_i \leftarrow \tilde{w}_i / \frac{1}{N}\sum_j \tilde{w}_j$ after clamping) retains the gain (FID 14.58; Sec. 5.4), so the effect is within-sample spatial redistribution rather than the global scale.

5 Experiments

Our experiments ask (1) whether dREPA improves generation quality consistently across scales, encoders, architectures, resolutions and tasks, (2) where the gains concentrate, (3) which design choices and which property of the signal carry the gain, and (4) whether the benefit persists under long training and tuned guidance, on class-conditional ImageNet (256×256, 512×512) and text-to-image generation on MS-COCO at 256×256.

5.1 Experiments setup

Backbones and encoders. Primary backbone: SiT (Ma et al., 2024) at B/2, L/2, XL/2 on 256×256 (32×32 latent grids, SD-VAE (Rombach et al., 2022)) and B/4 on 512×512 (64×64 latents). Additional architectures: JiT-B/16 (Li & He, 2025) and MMDiT (Esser et al., 2024). Three teachers: DINOv2-B/14 (Oquab et al., 2024) (default), DINOv3-B/16 (Siméoni et al., 2025), PE-Core-G (Bolya et al., 2025), all with z-score normalization.

Training and evaluation. AdamW (lr=2×10^{-4}), batch 256, bf16, grad clip 2.0. Alignment at layer 4. dREPA defaults: λ=0.5, γ=1.0, $[w_{\min}, w_{\max}]$=$[0.5, 2.0]$, T_{warm}=10K. Results at 400K steps unless noted; baselines are re-run with the same settings (full hyperparameter table in Appendix J). Evaluation: 50K samples, SDE sampler, 250 steps; FID (Heusel et al., 2017), sFID, Inception score (IS) (Salimans et al., 2016), Precision, and Recall. CFG uses scale 1.5 with interval $[0.0, 0.7]$ (Kynkäänniemi et al., 2024). All experiments are run on 8×H200 GPUs; text-to-image evaluation details are in Appendix A.

5.2 Main results: scales, encoders, architectures, resolutions and tasks

Model size. Tab. 3 reports five-metric results across three SiT scales without CFG. dREPA improves iREPA at every scale: FID drops by -3.50 on B/2, -2.01 on L/2, and -1.65 on XL/2, i.e. a stable 19–22% relative gain. With CFG (Fig. 8, Appendix I) dREPA improves FID and IS at every scale and guidance strength, with the largest margin without guidance (Sec. 5.5).

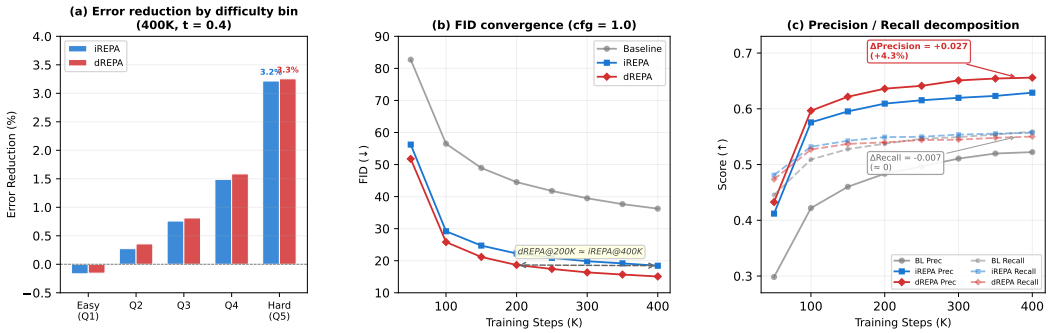

Figure 3: **Mechanistic analysis. (a)** Error reduction by difficulty bin: both methods improve hard bins most; dREPA's extra gain sits in the middle-to-hard bins. **(b)** FID convergence: dREPA at 200K $\approx$ iREPA at 400K. **(c)** Precision/Recall: FID gain driven by Precision. SiT-B/2, DINOv2-B.

Encoders. Tab. 14 (Appendix K) reports five-metric results across three teachers on SiT-B/2. dREPA improves both REPA and iREPA with every encoder; on iREPA, FID drops by -3.50 (DINOv2-B), -1.27 (DINOv3-B/16), and -2.46 (PE-Core-G). The gain holds across teachers trained by self-supervision, distillation and multimodal objectives.

Architectures and tasks. Tab. 4 extends the evaluation to JiT-B/16, a pixel-space diffusion architecture without a VAE: dREPA improves REPA and iREPA with all three encoders, with FID reductions from -1.58 to -5.94, and training curves (Appendix E) show that the gap opens early and persists. On text-to-image MS-COCO (Lin et al., 2014) 256×256 with MMDiT (150K steps; Tab. 16, Appendix K), iREPA+dREPA improves over iREPA at every CFG scale $w \in \{1.0, 1.2, 1.5, 2.0\}$, most at w=1.0 ($8.43 \rightarrow 7.61$, -9.7%), so dREPA transfers to text-conditional generation.

Resolution. On ImageNet 512×512 with SiT-B/4 (Tab. 15, Appendix K), dREPA improves every baseline at both CFG settings; on iREPA, FID drops from $28.92 \rightarrow 25.03$ without CFG (-13.4%) and from $17.06 \rightarrow 14.54$ with CFG (-14.8%), absolute gains comparable to those at 256×256.

5.3 Mechanistic Analysis

Fig. 3 analyzes where the FID improvement originates (additional analysis in Appendix H). **(a) Difficulty-bin analysis** partitions tokens into five bins by baseline denoising error. Error reduction increases monotonically from easy to hard bins under *both* methods (at t=0.4: -0.5% on the easiest bin, $+2.2\%$ on the hardest), so this ordering is a property of the training setup rather than of dREPA. The difference attributable to dREPA is that it matches or exceeds iREPA in every bin, with the gains concentrated in the middle-to-hard bins (Appendix H). **(b) Convergence**: dREPA at 200K steps reaches FID 18.67, comparable to iREPA at 400K (18.43), i.e. about $2\times$ fewer steps at that quality. **(c) Precision/Recall**: the FID gain is driven by Precision, at a modest cost in Recall. Without CFG, Precision rises and Recall falls at every scale (SiT-B/2: $0.629\rightarrow0.659$ and $0.638\rightarrow0.626$; Tab. 3), so we describe the effect as a precision–recall trade-off rather than as fidelity at no cost in diversity. One reading, which we offer as an interpretation, is that difficulty-aware spatial optimization mainly improves the high-density, high-precision region of the learned distribution, which would also explain the shrinking margin under strong CFG (Sec. 5.5).

5.4 Weighting Design, Signal Source, and Mechanism

Design. One-factor ablations (SiT-B/2, DINOv2-B, 400K, w/o CFG; default γ=1, c=1, λ=0.5, $[0.5, 2.0]$, 10K warmup) give: uniform 18.43, **default 14.93**, γ=0.5 14.95, γ=2.0 14.78, no per-sample mean normalization 18.96, no clamping 16.42, c=0 15.87 (15.21 renormalized), exact post-clamp normalization 14.58. Mean normalization is the load-bearing component, converting absolute alignment magnitude into relative within-sample allocation; it is necessary but not sufficient, since the identical pipeline gives 21.09 with an online-MSE signal and 18.86 with random weights. The transform is insensitive (≤ 0.15 FID over $\gamma \in [0.5, 2.0]$), clamping adds robustness, and c=1 preserves ordinary supervision without creating the gain. The last variant is an exact sum-preserving control with $\frac{1}{N}\sum_i w_i$=1 for every sample (the default carries $1+\lambda$ before clamping): the gain

survives with the global scale removed, so the effect is within-sample redistribution. Warmup and alignment-layer sweeps are in Appendix J: warmup is insensitive, and the layer sweep changes both the alignment location and the weighting source, so we do not attribute it to one mechanism.

Beyond the current residual. The persistence ordering of Sec. 3.3 ($0.569/0.657/0.853$) matches the FID ordering ($21.09/16.31/14.93$), and it has a structural reason: an MSE-derived signal is redrawn with $\boldsymbol{\epsilon}$ at every step, whereas the alignment target $f(\mathbf{x}_0)$ is fixed per image. To test whether alignment adds information beyond the residual, we condition on current MSE and timestep through 16 quantile bins: the captured component gives 17.94, the remainder gives 15.86, i.e. 2.57 of the 3.50-point gain, and shuffling the remainder within bins removes it (18.62). Under this control, the usable content is organized token assignment beyond the current residual. These are single-seed controls, and the shuffle perturbs several spatial properties at once.

Direction. Writing $w^{A\to D}$ for alignment-derived weights on the denoising loss and $w^{D\to A}$ for the reverse, $D \to A$ alone helps slightly ($18.43 \to 17.89$), $A \to D$ helps far more (-3.50), and applying both returns nothing ($+0.21$, $+0.60$ relative to dREPA). The experiment does not isolate the cause (Appendix F).

5.5 Long-horizon training and the inference operating point

Longer training. We extend matched SiT-XL/2 runs to 4M steps under an identical recipe; only the weighting differs (Fig. 1a; Tab. 17 in Appendix K). dREPA is ahead at each of the three evaluated horizons. As a training-acceleration method the quantity we report is compute-to-quality: dREPA at 400K (6.19) reaches what iREPA reaches at 1M (6.28), and dREPA at 1M (5.75) reaches what iREPA reaches at 4M (5.91), roughly a $4\times$ reduction in steps at that quality. Single-run FIDs vary by about 0.1–0.2 (run-to-run variation below), so we read these as matches, not wins. We claim nothing beyond the evaluated 400K–4M range.

Guidance strength. Under a *shared* CFG setting the SiT-XL/2 margin compresses monotonically (1.65 at w=1.0, 0.41 at w=1.35, 0.02 at w=1.5), so we report the w=1.5 pair (2.43 vs. 2.41) as a tie and do not use strong-guidance comparisons as evidence (full sweep in Appendix I). Tuning scale and interval per checkpoint on the same grid gives $5.71 \to \mathbf{4.38}$ on B/2 (w=2.5, interval $[0, 0.8] \to [0, 0.6]$) and $2.21 \to \mathbf{2.04}$ on XL/2, while guiding 60% rather than 80% of the trajectory ($\approx 11\%$ fewer network function evaluations).

Run-to-run variation. Forking three independently seeded continuations of a shared SiT-XL/2 checkpoint at 300K steps and training both arms to 400K, the paired no-CFG difference is -1.63 ± 0.10 FID (dREPA better in every branch); three sampling seeds on the original checkpoints give -1.69 ± 0.04. At w=1.5 both give -0.03 ± 0.03, within the variation (Appendix K.1). Since the branches share their first 300K steps, this understates full training-seed variability. Two further readings follow. The no-CFG margin is about $16\times$ the branch-to-branch standard deviation, far outside the noise; and training stochasticity (± 0.10) exceeds evaluation noise (± 0.04), so single-run differences of a few tenths of FID, such as the w=1.5 pair, should not be read as effects.

6 Conclusion

The per-token alignment loss that REPA-style training already computes is a free spatial difficulty signal for denoising, and a *usable* one: it ranks tokens persistently across noise draws while staying current with the model, whereas the model's own residual reranks tokens with every draw and makes training worse. dREPA turns this signal into bounded spatial denoising weights through within-sample normalization, with five lines of code and no parameters, and shifts the learning curve of REPA-style training horizontally, by up to roughly $4\times$ fewer steps at matched FID on SiT-XL/2 over the evaluated range, improves FID under independently tuned guidance, and does so consistently across scales, teachers, backbones, resolutions and tasks, beyond the run-to-run variation we measured.

Limitations. As an acceleration method, dREPA's margin is largest early in training and under weak guidance, and narrows toward convergence and under strong shared CFG; we claim the horizontal shift, not the asymptotic gap. Without guidance the gain is precision-driven, with a small recall cost. Higher resolutions than 512×512, video, and fixed structural proxies such as edge maps are untested.

AI USE STATEMENT

In this work, we used generative AI tools for editing and polishing the manuscript text and for assisting with the writing of plotting and evaluation scripts. We have not used generative AI tools for generating research ideas, designing experiments, or producing experimental results. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the training of image generative models on the publicly available ImageNet and MS-COCO datasets and does not involve human subjects or personally identifiable data. The proposed reweighting changes only how existing training losses are aggregated over spatial tokens; it does not introduce new data or new generation capabilities beyond those of the underlying diffusion models. dREPA reduces the training cost required to reach a given generation quality, contributing to more efficient generative model development; as with other improvements to image generation, higher fidelity may also increase the risk of synthetic-media misuse, and we encourage watermarking, provenance tracking and responsible release practices. We are not aware of conflicts of interest related to this work.

REPRODUCIBILITY STATEMENT

The weighting rule is fully specified in Sec. 4, including all constants, detach locations and the timestep sampler. Training and evaluation protocols (backbones, teachers, optimizers, batch sizes, sampler and step counts, CFG search grids, and the long-run configurations) are given in Sec. 5.1 and Appendix A. An anonymized code package containing the weighting implementation, configurations and evaluation scripts is linked at the end of the abstract.

REFERENCES

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning (ICML)*, pp. 794–803. PMLR, 2018.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11472–11481, 2022.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1851–1860, 2017.

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 *Forty-first international conference on machine learning*, 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 23164–23173, 2023.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via Min-SNR weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7441–7451, 2023.

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

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, 1988.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

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

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18262–18272, 2025.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Yexin Liu, Wen-Jie Shu, Zile Huang, Haoze Zheng, Yueze Wang, Jingjin Zhu, Manyuan Zhang, Ser-Nam Lim, and Harry Yang. AlignVid: Training-free attention scaling for semantic fidelity in text-guided image-to-video generation. *arXiv preprint arXiv:2512.01334*, 2025a.

Yexin Liu, Manyuan Zhang, Yueze Wang, Hongyu Li, Dian Zheng, Weiming Zhang, Changsheng Lu, Xunliang Cai, Yan Feng, Peng Pei, et al. Opensubject: Leveraging video-derived identity and diversity priors for subject-driven image generation and manipulation. *arXiv preprint arXiv:2512.08294*, 2025b.

Zhijun Liu, Shuai Wang, Sho Inoue, Qibing Bai, and Haizhou Li. Autoregressive diffusion transformer for text-to-speech synthesis. *arXiv preprint arXiv:2406.05551*, 2024.

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 *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. 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*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations (ICLR)*, 2021.

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 (CVPR)*, pp. 10684–10695, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in neural information processing systems*, 29, 2016.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization. *arXiv preprint arXiv:2506.09027*, 2025.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. OmniGen2: Towards instruction-aligned multimodal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3D representation for consistent world modeling. In *International Conference on Learning Representations (ICLR)*, 2026b.

Xianfeng Wu, Yajing Bai, Haoze Zheng, Harold Haodong Chen, Yexin Liu, Zihao Wang, Xuran Ma, Wen-Jie Shu, Xianzu Wu, Harry Yang, et al. LightGen: Efficient image generation through knowledge distillation and direct preference optimization. *arXiv preprint arXiv:2503.08619*, 2025b.

Bin Xia, Yuechen Zhang, Jingyao Li, Chengyao Wang, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. DreamOmni: Unified image generation and editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28533–28543, 2025.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. OmniGen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13294–13304, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15703–15712, 2025.

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 *The Thirteenth International Conference on Learning Representations*, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5824–5836, 2020.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. VideoREPA: Learning physics for video generation through relational alignment with foundation models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

APPENDIX TABLE OF CONTENTS

A. Implementation details 14
B. Curriculum learning and importance sampling 15
C. Edge vs. non-edge difficulty analysis 16
D. Stability analysis of spatial reweighting 17
E. Training curves 20
F. Asymmetric design ablation 20
G. Alignment loss as a denoising difficulty proxy 21
H. Error reduction by difficulty bin 22
I. CFG scale sweep: full results 22
J. Training hyperparameter selection 23
K. Additional main results 26
L. Qualitative comparison 28

A IMPLEMENTATION DETAILS

Algorithm 1 gives the per-step weighting rule. The signal-source rows of Table 1 are obtained by replacing $\boldsymbol{\ell}^{\text{repa}}$ in line 2 with the alternative signal (the detached per-token denoising MSE of the online model, the same statistic read from an EMA model, or random per-token values) and leaving every other line unchanged. Table 5 summarizes the hyperparameter setup for all experimental configurations.

Algorithm 1 dREPA: difficulty-adaptive reweighting for REPA training.

Require: Per-token alignment loss $\boldsymbol{\ell}^{\text{repa}} \in \mathbb{R}^{B\times N}$, per-token denoising loss $\boldsymbol{\ell}^{\text{denoise}} \in \mathbb{R}^{B\times N}$
Require: Training step s; hyperparameters γ, λ, T_{warm}, $w_{\min}$, $w_{\max}$, ε
1: $\lambda_s \leftarrow \lambda \cdot \min(1, s/T_{\text{warm}})$ ▷ Linear warmup
2: $\mathbf{d} \leftarrow \text{stopgrad}((\boldsymbol{\ell}^{\text{repa}} + \varepsilon)^{\gamma})$ ▷ Detached difficulty signal
3: $\hat{\mathbf{d}} \leftarrow \mathbf{d} \,/\, \text{mean}(\mathbf{d}, \text{dim=tokens})$ ▷ Mean-normalize per sample
4: $\mathbf{w} \leftarrow \text{clamp}(1 + \lambda_s \hat{\mathbf{d}},\ w_{\min},\ w_{\max})$ ▷ Bounded difficulty bonus
5: $\mathcal{L}_{\text{denoise}} \leftarrow \text{mean}(\mathbf{w} \odot \boldsymbol{\ell}^{\text{denoise}})$ ▷ Alignment-weighted denoising
6: **return** $\mathcal{L}_{\text{denoise}} + \alpha\, \mathcal{L}_{\text{repa}}$

Grid resolution mismatch. When the alignment and denoising grids differ in resolution (e.g., JiT operates on 16×16 pixel patches while the encoder produces 14×14 or 16×16 feature grids), the weight map is upsampled via nearest-neighbor interpolation to match the denoising token grid before applying per-token reweighting.

Projector design. The alignment projector follows the iREPA (Singh et al., 2026) convolutional design with z-score normalization on teacher features. The alignment loss uses cosine similarity at layer 4 of the backbone (layer sweep in Sec. 5.4).

Text-to-image. For MMDiT text-to-image experiments, we train for 150K steps on MS-COCO 256×256 with 10% caption dropout for classifier-free guidance (Ho & Salimans, 2022).

Evaluation protocol.

Class-conditional ImageNet. We generate 50K samples using the SDE sampler with 250 steps and evaluate FID (Heusel et al., 2017), IS (Salimans et al., 2016), sFID, Precision, and Recall against the ImageNet 256×256 reference batch. For CFG experiments, we apply guidance during the first 70% of sampling steps (interval $[0.0, 0.7]$) and disable it for the remainder. Default CFG scale is $w{=}1.5$ unless otherwise noted; the ablations in Sec. 5.4 are reported w/o CFG. All FID values are computed with the ADM (Dhariwal & Nichol, 2021) reference statistics for consistency with prior work.

Text-to-image MS-COCO. We generate 40K samples and compute FID against the MS-COCO 256×256 validation set (FID-40K). To characterize the interaction between dREPA and classifier-

Table 5: **Hyperparameter setup.** α=0.5 for REPA, α=1.0 for iREPA; dREPA inherits α from its baseline. *Default CFG scale for "w/ CFG" results. Table 15 uses w=1.5; the design and layer ablations in Sec. 5.4 are reported w/o CFG; Fig. 8 sweeps $\{1.0, 1.35, 1.5\}$; Table 16 sweeps $\{1.0, 1.2, 1.5, 2.0\}$. Tables 3, 4, 14 report w/o CFG (w=1.0) only.

	SiT-B/2 (Fig. 3, Tab. 3, Sec. 5.4)	**SiT-L/2** (Tab. 3)	**SiT-XL/2** (Tab. 3)	**SiT-B/4** (Tab. 15)	**JiT-B/16** (Tab. 4)	**MMDiT** (Tab. 16)
Architecture						
Input dim.	32×32×4	32×32×4	32×32×4	64×64×4	256×256×3	32×32×4
Num. layers	12	24	28	12	12	12
Hidden dim.	768	1024	1152	768	768	768
Num. heads	12	16	16	12	12	12
Representation Alignment						
Baseline	REPA, iREPA	iREPA	iREPA	REPA, iREPA	REPA, iREPA	iREPA
α	0.5 / 1.0	1.0	1.0	0.5 / 1.0	0.5 / 1.0	1.0
Alignment depth	4	4	4	4	8	4
$\mathrm{sim}(\cdot,\cdot)$	cos. sim.	cos. sim.	cos. sim.	cos. sim.	cos. sim.	cos. sim.
Encoder $f(\mathbf{x})$	DINOv2-B	DINOv2-B	DINOv2-B	DINOv2-B	DINOv2-B / DINOv3 / PE-Core	DINOv2-B
Projector	conv	conv	conv	conv	conv	conv
dREPA						
λ	0.5	0.5	0.5	0.5	0.5	0.5
γ	1.0	1.0	1.0	1.0	1.0	1.0
$[w_{\min}, w_{\max}]$	$[0.5, 2.0]$	$[0.5, 2.0]$	$[0.5, 2.0]$	$[0.5, 2.0]$	$[0.5, 2.0]$	$[0.5, 2.0]$
T_{warm}	10K	10K	10K	10K	10K	10K
Optimization						
Training iteration	400K	400K	400K	400K	80 epochs	150K
Batch size	256	256	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
lr	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Grad clip	2.0	2.0	2.0	2.0	2.0	2.0
Precision	bf16	bf16	bf16	bf16	bf16	fp16
Sampling						
Sampler	SDE	SDE	SDE	SDE	Euler–Maruyama	SDE
Sampling steps	250	250	250	250	250	250
CFG scale w^*	1.5	1.5	1.5	1.5	—	{1.0, 1.2, 1.5, 2.0}
CFG interval	$[0.0, 0.7]$	$[0.0, 0.7]$	$[0.0, 0.7]$	$[0.0, 0.7]$	—	$[0.0, 0.7]$
Dataset	ImageNet	ImageNet	ImageNet	ImageNet	ImageNet	MS-COCO
Resolution	256×256	256×256	256×256	512×512	256×256	256×256

free guidance, we sweep CFG scale $w \in \{1.0, 1.2, 1.5, 2.0\}$ with the same guidance interval $[0.0, 0.7]$. Results are reported in Table 16.

Computational overhead. dREPA adds negligible computational cost to standard REPA training. The per-token weight computation involves only elementwise operations on the alignment loss tensor (power transform, normalization, clamping), which takes $<0.1\%$ of the total step time on a single H200 GPU. No additional forward or backward passes are required, and the weight computation is fully parallelizable across the batch dimension. Memory overhead is also minimal: the only additional tensor is the per-token weight map $w \in \mathbb{R}^{B\times N}$ (e.g., 256×1024 for batch size 256 and 32×32 latent grid), which is negligible relative to the model activations and gradient buffers. In practice, we observe no measurable difference in wall-clock time per training step between iREPA and iREPA+dREPA.

Computational resources. All experiments are conducted on NVIDIA H200 GPUs (141 GB HBM3 each). Table 6 summarizes the hardware configuration and estimated training time for each architecture. The default setup uses 8 GPUs on a single node with data-parallel training. A subset of the SiT-B/2 ablation runs (Sec. 5.4) use 2 GPUs due to their smaller per-run cost. Since dREPA adds no extra forward or backward passes, the wall-clock time per step is identical between baseline and dREPA runs for each configuration; the total GPU-hours for a dREPA run therefore match those of its corresponding baseline.

B Curriculum learning and importance sampling

Curriculum learning (Bengio et al., 2009) schedules training examples from easy to hard, while importance sampling (Katharopoulos & Fleuret, 2018), meta-learned sample weighting (Shu et al.,

Table 6: **Computational resources.** All runs use AdamW with gradient clipping at 2.0. ‡MMDiT uses fp16 mixed precision; all other runs use bf16.

Architecture	Model	Params	GPUs	Batch	Steps	Resolution	Time
SiT	B/2	131M	8×H200	256	400K	256×256	∼4.5 h
	L/2	458M	8×H200	256	400K	256×256	∼8.3 h
	XL/2	675M	8×H200	256	400K	256×256	∼12.7 h
SiT	B/4	131M	8×H200	256	400K	512×512	∼14.2 h
JiT	B/16	131M	8×H200	256	80 ep	256×256	∼3 h
MMDiT	—	506M	8×H200	256	150K‡	256×256	∼15.5 h

2019), and hard example mining (Dong et al., 2017; Robinson et al., 2021) assign per-example weights to accelerate convergence. These methods operate at the *sample* level—each image receives a scalar weight—and do not distinguish between spatial locations within an image. Token-level efficiency has been explored in vision transformers through dynamic pruning (Rao et al., 2021) and merging (Bolya et al., 2023), but these reduce computation at inference time, not during training. In the diffusion setting, MaskDiT (Zheng et al., 2024) and MDTv2 (Gao et al., 2023) drop a subset of tokens during training to reduce cost, but these are *binary* masking strategies: each token is either fully retained or fully discarded, and the selection criterion is random rather than difficulty-aware.

dREPA occupies a different point in this design space: it performs *soft* spatial reweighting during training, preserving all tokens while reallocating gradient budget within each image toward regions that the alignment signal identifies as harder. This is conceptually related to adaptive loss balancing in multi-task learning (Chen et al., 2018), but operates at the token level rather than the task level. Unlike sample-level methods, dREPA redistributes effort at the sub-sample (token) level, so two images with identical global difficulty can receive different spatial weight maps depending on their content structure. Unlike token masking, dREPA does not reduce computation or discard information—it only changes the relative gradient contribution of each token. Additionally, dREPA derives its importance signal from a source that is not the residual being reweighted in that update (the alignment loss), rather than from the instantaneous denoising residual itself (Section 3). These properties make dREPA orthogonal to both sample-level curriculum methods and token-level masking strategies, and in principle compatible with either.

C Edge vs. non-edge difficulty analysis

To examine whether alignment difficulty concentrates on semantically meaningful spatial locations, we partition tokens into *edge* and *non-edge* sets using a Sobel gradient operator (Kanopoulos et al., 1988) applied to the clean image $\mathbf{x}_0$. We compute the Sobel magnitude at each spatial location and threshold at the 75th percentile to obtain a binary edge mask. Table 7 reports the mean alignment difficulty $\bar{d}$ for each group across different noise levels t.

Edge tokens are harder than non-edge tokens at every noise level measured, indicating that the alignment loss captures structural complexity in addition to semantic content. The *ratio* between edge and non-edge difficulty increases with t: from $1.05\times$ at $t{=}0.05$ to $1.85\times$ at $t{=}0.8$. This widening gap is consistent with dREPA's implicit timestep adaptivity (Section 4.3): at high t, the spatial difficulty landscape is more non-uniform, so the alignment-derived weights produce larger spatial contrast; at low t, difficulty is more uniform across tokens, and the weights collapse toward 1.

This noise-level dependence also illustrates why uniform weighting may be suboptimal: at high noise levels, the hardest 10% of tokens carry $\sim 2\times$ the difficulty of the average token, yet under uniform weighting they receive the same gradient signal as easy interior tokens. dREPA addresses this by directing proportionally more denoising gradient toward these boundary regions, without requiring an explicit timestep-dependent schedule.

Table 7: Edge vs. non-edge difficulty. The ratio widens at larger t. SiT-B/2, ImageNet 256×256.

t	$\bar{d}_{\text{edge}}$	$\bar{d}_{\text{non-edge}}$	Ratio
0.05	1.031	0.981	1.05×
0.1	1.017	0.915	1.11×
0.2	0.999	0.826	1.21×
0.4	1.100	0.756	1.46×
0.6	1.199	0.671	1.79×
0.8	1.080	0.584	1.85×

D STABILITY ANALYSIS OF SPATIAL REWEIGHTING

We provide a stylized analysis of why self-referencing reweighting is less stable than cross-objective reweighting. The analysis uses a two-group model under the Polyak–Łojasiewicz (PL) condition (Karimi et al., 2016) and aims to provide *qualitative insight* into the failure mechanism rather than quantitative guarantees for the full system.

Modeling assumptions. The following derivation relies on several simplifying assumptions: (i) tokens are partitioned into two equal-sized groups (easy and hard), collapsing the continuous difficulty distribution; (ii) the Polyak–Łojasiewicz condition is taken at equality ($\|\nabla L_i\|^2 = 2\mu L_i$), which holds exactly only for quadratic losses; (iii) the inter-group gradient conflict angle β and effective step size ρ are treated as constants, whereas in practice they vary across parameters, layers, and training stages; (iv) the stability analysis is local (linearized around the equal-difficulty equilibrium). These assumptions make the model tractable at the cost of direct quantitative applicability; the value lies in the qualitative predictions, which we verify empirically in Section 3.

Notation. $\rho = 2\eta\mu$ is the effective step size of the two-group dynamics, with η the learning rate and μ the local PL/curvature parameter; it is distinct from the alignment loss weight α of the training objective. $\beta = |\cos\phi_{EH}| \in [0, 1)$ measures the misalignment of the easy- and hard-group gradients through $\cos\phi_{EH} = -\beta$. Both are properties of the stylized model, not measured quantities of SiT training.

D.1 DERIVATION FROM GRADIENT DESCENT

Consider a shared-parameter model θ trained on two equal-sized token groups: easy (E) and hard (H), with per-group losses $L_E(\theta), L_H(\theta) > 0$. The weighted gradient update is:

$$\theta_{t+1} = \theta_t - \eta\big(w_E^{(t)}\nabla_\theta L_E(\theta_t) + w_H^{(t)}\nabla_\theta L_H(\theta_t)\big). \tag{7}$$

Loss change via Taylor expansion. The first-order change in L_E after the update is:

$$\begin{aligned} L_E(\theta_{t+1}) &\approx L_E(\theta_t) + \nabla L_E^T(\theta_{t+1} - \theta_t) \\ &= L_E - \eta\, w_E\|\nabla L_E\|^2 - \eta\, w_H\langle\nabla L_E, \nabla L_H\rangle. \end{aligned} \tag{8}$$

Polyak–Łojasiewicz (PL) condition. We assume $\|\nabla L_i\|^2 = 2\mu\, L_i$ (equality case of the PL condition $\|\nabla L_i\|^2 \geq 2\mu\, L_i$ (Karimi et al., 2016)), which holds exactly for quadratic losses and approximately for smooth losses near critical points. This gives $\|\nabla L_i\| = \sqrt{2\mu\, L_i}$.

Gradient angle decomposition. Decompose the cross term using the inter-group gradient angle ϕ_{EH}:

$$\langle\nabla L_E, \nabla L_H\rangle = \|\nabla L_E\|\,\|\nabla L_H\|\,\cos\phi_{EH} = 2\mu\sqrt{L_E L_H}\,\cos\phi_{EH}. \tag{9}$$

When the two groups compete for shared model capacity, their gradients conflict ($\cos\phi_{EH} < 0$): optimizing for hard tokens can increase the loss for easy tokens (Yu et al., 2020; Liu et al., 2021). We write $\cos\phi_{EH} = -\beta$ with $\beta = |\cos\phi_{EH}| \in [0, 1)$.

Two-group dynamics. Substituting into Eq. (8) with $\rho = 2\eta\mu$:

$$L_E^{(t+1)} = L_E^{(t)}(1 - \rho\, w_E^{(t)}) + \beta\rho\, w_H^{(t)}\sqrt{L_E^{(t)}\, L_H^{(t)}}, \tag{10}$$

$$L_H^{(t+1)} = L_H^{(t)}(1 - \rho\, w_H^{(t)}) + \beta\rho\, w_E^{(t)}\sqrt{L_E^{(t)}\, L_H^{(t)}}. \tag{11}$$

The first term captures self-improvement (larger weight → faster loss reduction); the second captures cross-group interference, whose magnitude scales with $\sqrt{L_E L_H}$ (the geometric mean of gradient norms under PL) and the conflict angle β. The condition $\rho w_i < 1$ ensures well-posedness.

Following the paper's reweighting formula with γ=1, the normalized self-referencing weights are:

$$w_i^{(t)} = 1 + \lambda \frac{L_i^{(t)}}{\bar{L}^{(t)}}, \quad \bar{L}^{(t)} = \frac{L_E^{(t)} + L_H^{(t)}}{2}. \tag{12}$$

D.2 Stability threshold derivation

Ratio dynamics. Define the difficulty ratio $r_t = L_H^{(t)}/L_E^{(t)}$. Using $\sqrt{L_E L_H} = L_E\sqrt{r_t}$, dividing Eq. (11) by Eq. (10):

$$r_{t+1} = \frac{r_t(1 - \rho\, w_H) + \beta\rho\, w_E\sqrt{r_t}}{(1 - \rho\, w_E) + \beta\rho\, w_H\sqrt{r_t}}. \tag{13}$$

At $r^* = 1$, $r_{t+1} = 1$ for any w_E, w_H—the equal-difficulty ratio is always a fixed point.

(a) Self-referencing instability.

The self-referencing weights in terms of r_t are $w_E = 1 + 2\lambda/(1{+}r_t)$ and $w_H = 1 + 2\lambda r_t/(1{+}r_t)$.

Linearization. Substitute $r_t = 1 + \delta_t$ with $|\delta_t| \ll 1$ and expand to first order:

$$w_E \approx (1{+}\lambda) - \tfrac{\lambda}{2}\delta_t, \quad w_H \approx (1{+}\lambda) + \tfrac{\lambda}{2}\delta_t, \quad \sqrt{r_t} \approx 1 + \tfrac{1}{2}\delta_t. \tag{14}$$

Let $w^* = 1{+}\lambda$, $A = 1 - \rho w^*$, and $B = \beta\rho w^*$. Expanding the numerator of Eq. (13) to first order in δ_t:

$$\begin{aligned} &r_t(1 - \rho w_H) + \beta\rho\, w_E\sqrt{r_t} \\ &= (1{+}\delta_t)\big(A - \tfrac{\rho\lambda}{2}\delta_t\big) + \beta\rho\big(w^* - \tfrac{\lambda}{2}\delta_t\big)\big(1 + \tfrac{1}{2}\delta_t\big) \\ &\approx (A{+}B) + \delta_t\big(A + \tfrac{B}{2} - \tfrac{\rho\lambda(1+\beta)}{2}\big). \end{aligned} \tag{15}$$

Similarly, the denominator:

$$\begin{aligned} &(1 - \rho w_E) + \beta\rho\, w_H\sqrt{r_t} \\ &= \big(A + \tfrac{\rho\lambda}{2}\delta_t\big) + \beta\rho\big(w^* + \tfrac{\lambda}{2}\delta_t\big)\big(1 + \tfrac{1}{2}\delta_t\big) \\ &\approx (A{+}B) + \delta_t\big(\tfrac{B}{2} + \tfrac{\rho\lambda(1+\beta)}{2}\big). \end{aligned} \tag{16}$$

Dividing and expanding to first order:

$$\delta_{t+1} = m_{\text{self}} \cdot \delta_t, \quad \text{where} \quad m_{\text{self}} = \frac{A - \rho\lambda(1{+}\beta)}{A + B} = \frac{1 - \rho\big(1 + \lambda(2{+}\beta)\big)}{1 - \rho(1{-}\beta)(1{+}\lambda)}. \tag{17}$$

The denominator is positive under the well-posedness condition $\rho(1{-}\beta)(1{+}\lambda) < 1$. The equilibrium is stable if and only if $|m_{\text{self}}| < 1$.

Stability boundary. Since m_{self} decreases monotonically in λ and $m_{\text{self}} < 1$ for all $\lambda > 0$, the binding constraint is $m_{\text{self}} > -1$. Setting $m_{\text{self}} = -1$:

$$\begin{aligned} 1 - \rho(1 + \lambda^*(2{+}\beta)) &= -(1 - \rho(1{-}\beta)(1{+}\lambda^*)), \\ 2 &= \rho\big[1 + \lambda^*(2{+}\beta) + (1{-}\beta)(1{+}\lambda^*)\big], \\ 2 &= \rho\big[2 + 3\lambda^* - \beta\big], \\ \lambda^*_{\text{self}} &= \frac{2(1-\rho) + \rho\beta}{3\rho}. \end{aligned} \tag{18}$$

For $\lambda > \lambda^*_{\text{self}}$, $|m_{\text{self}}| > 1$ and perturbations from the equilibrium are amplified, yielding oscillatory divergence. □

Numerical examples.

- $\rho = 0.3,\ \beta = 0.2$: $\lambda^*_{\text{self}} = (1.4 + 0.06)/0.9 \approx 1.62$. With $\lambda = 1.0$, $m_{\text{self}} \approx 0.08$ (stable, monotone). With $\lambda = 2.0$, $m_{\text{self}} \approx -2.21$ (unstable).
- $\rho = 0.5,\ \beta = 0.3$: $\lambda^*_{\text{self}} = (1.0 + 0.15)/1.5 \approx 0.77$. The paper's default $\lambda = 0.5$ is already close to this threshold. In practice, ρ varies across parameters and training stages, making self-referencing unreliable even at moderate λ.

(b) Cross-objective stability.

When weights derive from alignment losses that are structurally decoupled from the denoising losses (passing through a separate projector and frozen teacher), w_E and w_H do not depend on (L_E, L_H). At the equilibrium $r^* = 1$ with $w_E = w_H = w = 1+\lambda$, the linearization of Eq. (13) simplifies (no weight perturbation terms):

$$\delta_{t+1} = m_{\text{cross}} \cdot \delta_t, \quad \text{where} \quad m_{\text{cross}} = \frac{A}{A+B} = \frac{1-\rho(1+\lambda)}{1-\rho(1+\lambda)(1-\beta)}. \tag{19}$$

Setting $m_{\text{cross}} = -1$ and solving:

$$\begin{aligned} 1-\rho(1+\lambda^*) &= -(1-\rho(1+\lambda^*)(1-\beta)), \\ 2 &= \rho(1+\lambda^*)(2-\beta), \\ \lambda^*_{\text{cross}} &= \frac{2(1-\rho)+\rho\beta}{\rho(2-\beta)}. \end{aligned} \tag{20}$$

Multiplier decomposition: the structural gap. Comparing Eqs. (17) and (19) reveals the precise effect of state-dependent weights:

$$m_{\text{self}} = m_{\text{cross}} - \frac{\rho\lambda(1+\beta)}{A+B}. \tag{21}$$

The second term is always positive (for $\lambda, \beta > 0$), so $m_{\text{self}} < m_{\text{cross}}$ at every λ—self-referencing always reaches the $m = -1$ instability boundary sooner. The ratio of thresholds is:

$$\frac{\lambda^*_{\text{cross}}}{\lambda^*_{\text{self}}} = \frac{3}{2-\beta} > 1 \quad \text{for all } \beta \in [0, 1). \tag{22}$$

This factor is independent of ρ—the advantage of cross-objective weighting is a structural property of the decoupling, not a function of the training configuration. For $\beta = 0$ (no interference): ratio $= 1.5$; for $\beta = 0.3$: ratio ≈ 1.76; for $\beta = 0.5$: ratio $= 2.0$. □

Numerical comparison. For $\rho = 0.5,\ \beta = 0.3$:

- Self-referencing: $\lambda^*_{\text{self}} \approx 0.77$. At $\lambda = 0.5$: $m_{\text{self}} \approx 0.33$ (stable). At $\lambda = 1.0$: $m_{\text{self}} \approx -0.82$ (oscillatory, near-unstable).
- Cross-objective: $\lambda^*_{\text{cross}} \approx 1.35$. At $\lambda = 1.0$: $m_{\text{cross}} \approx 0.36$ (stable, monotone convergence—*no oscillation*).

At $\lambda = 1.0$, the qualitative difference is notable: self-referencing oscillates with $|m| \approx 0.82$, while cross-objective converges monotonically with $m \approx 0.36$.

D.3 Discussion

The analysis reveals three insights beyond the binary stable/unstable classification:

1. **Oscillation precedes divergence.** Even below λ^*_{self}, the self-referencing multiplier becomes negative for moderate λ (when $\rho(1 + \lambda(2+\beta)) > 1$, i.e., $\lambda > (1-\rho)/(\rho(2+\beta))$). This causes the difficulty ratio to overshoot and oscillate at every step. Each oscillation involves large gradient swings between token groups, which in real nonlinear networks causes cumulative feature degradation beyond what the linearized analysis captures. Cross-objective's multiplier, by contrast, remains positive over a wider range (until $\rho(1+\lambda) > 1$), producing monotone convergence.

2. **Effective ρ varies across the network.** The stability threshold depends on $\rho = 2\eta\mu$, where μ (loss curvature) varies across parameters, layers, and training stages (Li et al., 2018). Within the surrogate, a λ setting that is stable for one value of ρ may exceed the threshold for another, and the factor-of-$3/(2-\beta)$ margin of cross-objective weighting widens the range of ρ over which a given λ stays below it. We do not observe such behavior in SiT training (all runs complete without divergence), so this is a property of the surrogate, not a claim about the experiments.
3. **The $3/(2-\beta)$ gap is a structural invariant.** The threshold ratio $\lambda^*_{\text{cross}}/\lambda^*_{\text{self}} = 3/(2-\beta)$ does not depend on the step size ρ. This means the relative advantage of cross-objective weighting holds regardless of training configuration—it is a consequence of removing the state-dependent feedback, not of any specific hyperparameter regime.

Scope of this analysis. The comparison above concerns a two-group linearized surrogate, not SiT training. A wider stable range does not imply better FID, and worse FID does not imply instability: in our experiments all self-referencing runs complete without numerical divergence and quality degrades smoothly with λ. We therefore do not treat Table 1 as an empirical verification of this analysis, and the empirical claims of the paper rest on the matched sweep, the EMA-source control, cross-draw rank persistence and the MSE/timestep binning control instead. The analysis also assumes that the cross-objective weights do not depend on (L_E, L_H), which is an idealization: the paper's own premise is that alignment and denoising difficulty are correlated.

E Training curves

We provide FID training curves to characterize dREPA's convergence behavior. The loss landscape of deep networks is typically non-convex with many local minima (Li et al., 2018); training curves help verify that reweighting does not introduce instabilities.

SiT-B/2. On SiT-B/2 with DINOv2-B, a gap between iREPA+dREPA and the iREPA baseline appears early in training. By 100K steps, iREPA+dREPA reaches a lower FID than iREPA at the same checkpoint, and the separation grows throughout training. By 200K steps, iREPA+dREPA achieves FID 18.67, comparable to the iREPA baseline at 400K steps (FID 18.43), corresponding to a $\sim 2\times$ reduction in steps to reach the same quality. The gap persists rather than saturating: between 200K and 400K steps, dREPA continues to improve (from 18.67 to 14.93), whereas the baseline improves more slowly over the same interval. One possible explanation is that the difficulty landscape evolves during training and the alignment-derived weights adapt accordingly, though we do not isolate this mechanism here. The reweighting takes effect shortly after the T_{warm}=10K warmup period, once the alignment signal has become informative.

JiT-B/16. Figure 4 shows curves for JiT-B/16 (pixel-space diffusion, 80 epochs). The trends are similar to SiT: early gap opening, persistent improvement, and consistency across all three encoders (DINOv2-B, DINOv3, PE-Core). The absolute FID values are higher because JiT-B/16 operates in pixel space without a VAE, but the relative improvement direction and magnitude are comparable. For both REPA and iREPA baselines, the dREPA variant tracks below the corresponding baseline curve from early training onward, suggesting that the reweighting mechanism is not specific to latent-space diffusion.

F Asymmetric design ablation

A natural question is whether the alignment loss should *also* be reweighted by the difficulty signal, rather than only reweighting the denoising loss. Table 8 systematically ablates this choice by independently toggling reweighting on the denoising objective (Rev) and the alignment objective (DA).

The results show a clear asymmetry. Rev-only (dREPA, λ_{Rev}=0.5) reduces FID by 3.50 points (from 18.43 to 14.93), accounting for the majority of the improvement. DA-only (λ_{DA}=0.5) achieves a 0.54-point improvement (18.43 to 17.89)—positive but much smaller. Adding alignment reweighting on top of dREPA ("Both" rows) degrades performance: +0.21 FID at λ_{DA}=0.5 and +0.60 FID at λ_{DA}=1.0.

We attribute this asymmetry to the dual role of the alignment objective. The alignment loss serves as dREPA's difficulty sensor: it needs to supervise all tokens uniformly to produce a stable estimate of per-token difficulty. Reweighting the alignment loss biases it toward hard tokens, which can distort

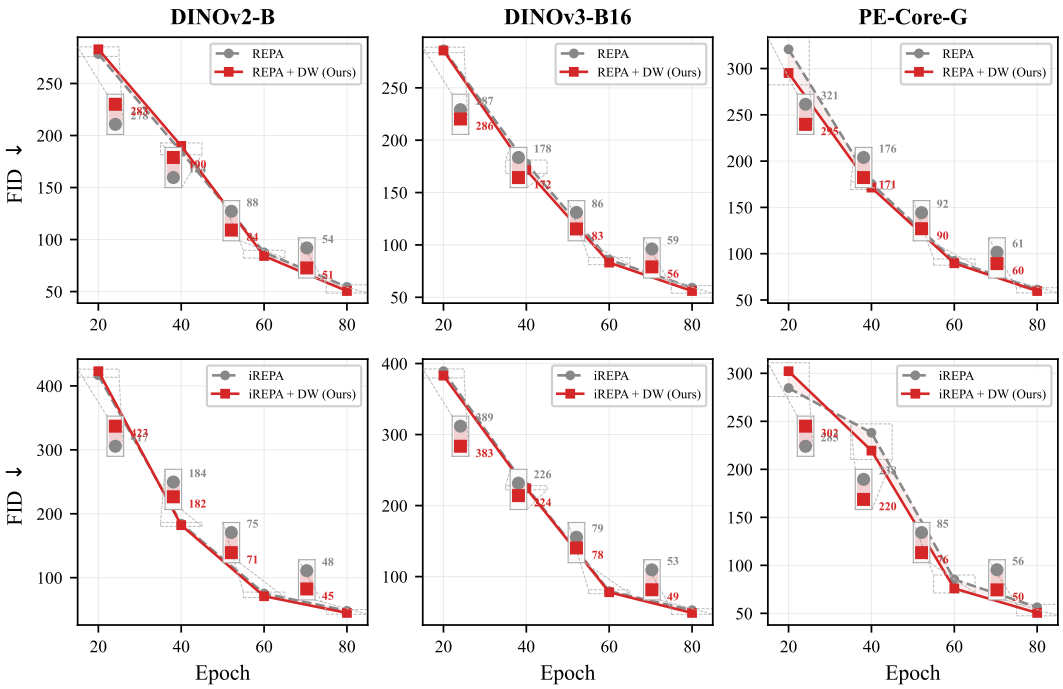

Figure 4: **FID training curves on JiT-B/16 across three teacher encoders.** Top: REPA vs. REPA + dREPA. Bottom: iREPA vs. iREPA + dREPA. w/o CFG.

the difficulty signal that the denoising reweighting relies on—creating a feedback loop between signal and response. In contrast, reweighting only the denoising loss preserves the alignment signal as an independent cross-objective proxy. This structural decoupling is a key factor in the stability of cross-objective signal transfer, and motivates the asymmetric Rev-only design as the default.

The small positive effect of DA-only (ΔFID$=-0.54$) is not surprising: even mild difficulty-aware alignment can help the projector attend more to hard tokens, slightly improving representation quality. However, this benefit is far smaller than Rev-only (ΔFID$=-3.50$) and becomes counterproductive when combined with Rev, because the compounding reweighting introduces the very state-dependent feedback that cross-objective decoupling is designed to avoid.

Table 8: **Asymmetric design ablation.** Rev = reweight denoising loss; DA = reweight alignment loss. SiT-B/2, DINOv2-B, 400K, w/o CFG.

Configuration	λ_{DA}	λ_{Rev}	**IS**↑	**FID**↓	**sFID**↓	**Prec.**↑	**Rec.**↑
Baseline (uniform)	0	0	77.33	18.43	6.161	0.6289	0.6379
DA-only (λ=0.5)	0.5	0	79.63	17.89	6.066	0.6321	0.6421
Rev-only (λ=0.5, dREPA)	0	0.5	88.58	14.93	5.664	0.6594	0.6257
Both (λ_{DA}=0.5)	0.5	0.5	88.41	15.14	5.743	0.6587	0.6290
Both (λ_{DA}=1.0)	1.0	0.5	87.68	15.53	5.736	0.6536	0.6360

G ALIGNMENT LOSS AS A DENOISING DIFFICULTY PROXY

A core premise of dREPA is that the per-token alignment loss ℓ_i^{repa} is a useful proxy for denoising difficulty. We validate this quantitatively by measuring the *top-k overlap*: the fraction of the top-k% hardest tokens by alignment loss that are also top-k% hardest by denoising error. For each image, we obtain two spatial maps: $a_i = 1 - \cos(\text{proj}(h_i), \text{teacher}(\mathbf{x}_0)_i)$ (alignment loss) and $d_i = \frac{1}{C}\sum_c (\hat{v}_{c,i} - v_{c,i})^2$ (denoising error), and compute overlap across 256 validation images.

Table 9 reports results at 400K steps. The top-10% overlap reaches 16.6% at t=0.8 (vs. 10% expected under independence), and top-20% overlap reaches 27.7% (vs. 20%), indicating that alignment-hard tokens are more likely to be denoising-hard than chance would predict. The overlap increases with noise level: at low noise, the denoising task is nearly trivial everywhere and the difficulty landscape is flat; at high noise, structural difficulty dominates and the two signals become more aligned.

The overlap is relatively stable across training: from 50K to 400K steps, top-10% overlap at t=0.6 varies between 15.8% and 16.3%. This suggests that the correspondence between alignment and denoising difficulty does not degrade substantially as the model improves over this training range. Note that dREPA does not require perfect rank prediction; a monotone correspondence sufficient to upweight hard tokens on average is enough for the reweighting to be beneficial.

Table 9: **Alignment–denoising overlap across noise levels.** Top-k overlap between per-token alignment loss and denoising error. SiT-B/2, DINOv2-B, 400K steps, 256 val images.

t	Top-10% overlap	Top-20% overlap
0.1	10.9%	21.8%
0.2	12.8%	23.7%
0.4	15.8%	26.5%
0.6	16.1%	27.3%
0.8	16.6%	27.7%

H Error reduction by difficulty bin

To directly verify that dREPA concentrates gradient on hard tokens and that this translates to preferential error reduction, we partition tokens into five equal-sized difficulty bins (based on the *baseline* model's denoising error, ensuring an unbiased reference) and measure mean error reduction for each bin under iREPA and iREPA + dREPA.

Table 10 reports results at 400K steps for three noise levels. Two patterns hold across timesteps. First, error reduction increases monotonically from easy to hard bins *under both methods*: at t=0.4 the hardest bin shows 2.2% reduction under iREPA and under dREPA alike, while the easiest bin shows a slight increase (-0.5%). This ordering is therefore a property of the training setup and is not evidence for dREPA on its own.

Second, dREPA consistently matches or slightly exceeds iREPA's reduction across all bins, with the largest gains concentrated in the middle-to-hard bins (mid1–mid3). For example, at t=0.4, dREPA achieves $+0.7\%$ on mid1 vs. iREPA's $+0.5\%$, and $+1.7\%$ on mid3 vs. $+1.6\%$. The improvement over iREPA is modest in absolute terms but systematic, which is consistent with the overall FID improvement (14.93 for iREPA + dREPA vs. 18.43 for the baseline).

The rightmost columns report the negative cosine similarity $-\cos(\phi_i)$ between the projected feature and the teacher target, not the alignment loss $\ell_i^{\text{repa}} = 1 - \cos(\phi_i)$ itself, which is non-negative by definition. They show a monotone trend: harder denoising bins have lower cosine similarity, i.e. higher alignment loss, supporting the alignment-denoising difficulty correspondence that dREPA relies on.

I CFG scale sweep: full results

Classifier-free guidance (CFG) (Ho & Salimans, 2022) trades diversity for fidelity at sampling time. Figure 7 shows the FID curve across CFG scales on SiT-B/2. Table 11 reports the complete metrics for the CFG scale sweep. SiT-B/2 is swept at eight scales; L/2 and XL/2 at three representative scales ($w \in \{1.0, 1.35, 1.5\}$). dREPA improves or matches FID at every tested scale and model size, with the largest gains at low CFG.

dREPA and CFG partially overlap. On SiT-B/2, the dREPA improvement is largest at w=1.0 (no guidance), where FID drops from 18.43 to 14.93 (Δ=-3.50). As the CFG scale increases, the dREPA gap shrinks: Δ=-1.79 at w=1.35 and Δ=-1.32 at w=1.5. A plausible explanation is that CFG biases sampling toward higher-likelihood regions, which disproportionately benefits boundary and detail tokens—the same tokens that dREPA upweights during training. The

Table 10: **Error reduction by difficulty bin.** Tokens binned by baseline denoising error (5 equal bins). Reduction = baseline error − method error. SiT-B/2, DINOv2-B, 400K steps, n=512.

		iREPA		iREPA + dREPA	
t	**Bin**	Reduction	− cos	Reduction	− cos
0.2	easy	−1.5%	−0.670	−1.2%	−0.666
	mid1	+0.0%	−0.665	+0.2%	−0.661
	mid2	+0.7%	−0.661	+0.8%	−0.657
	mid3	+1.2%	−0.657	+1.3%	−0.653
	hard	+1.7%	−0.647	+1.9%	−0.643
0.4	easy	−0.5%	−0.631	−0.5%	−0.627
	mid1	+0.5%	−0.620	+0.7%	−0.617
	mid2	+1.1%	−0.611	+1.2%	−0.608
	mid3	+1.6%	−0.600	+1.7%	−0.597
	hard	+2.2%	−0.582	+2.2%	−0.580
0.6	easy	−0.3%	−0.566	−0.1%	−0.565
	mid1	+0.3%	−0.548	+0.4%	−0.546
	mid2	+0.7%	−0.536	+0.8%	−0.534
	mid3	+0.9%	−0.525	+0.9%	−0.523
	hard	+1.4%	−0.506	+1.4%	−0.504

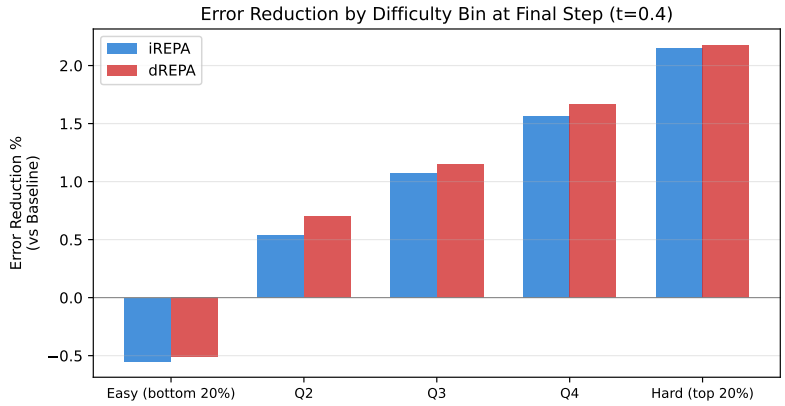

Figure 5: **Error reduction by difficulty bin at t=0.4 (400K steps).** Harder bins receive monotonically larger error reduction under both iREPA and dREPA, with dREPA matching or exceeding iREPA in every bin. The easy bin shows a slight error *increase*, consistent with gradient reallocation away from already-easy tokens.

two mechanisms thus address partially overlapping error modes, with dREPA providing larger gains when guidance is absent or weak.

Scale dependence. The same trend is more pronounced at larger model scale. On SiT-L/2, dREPA reduces FID by 2.01 at w=1.0 but by only 0.14 at w=1.5 (2.65 → 2.51). On SiT-XL/2, the gap is 1.65 at w=1.0 and narrows to 0.02 at w=1.5 (2.43 vs. 2.41), suggesting that at the quality frontier, strong classifier-free guidance at large scale already addresses most of the error modes that dREPA targets. The unconditional (w/o CFG) improvements remain positive across all scales, indicating that dREPA's training-time effect is not solely an artifact of the sampling procedure.

J Training Hyperparameter Selection

dREPA warmup and alignment layer. With SiT-B/2, DINOv2-B, 400K steps and no CFG (default FID 14.93), the warmup length is insensitive: FID 15.20/14.93/14.94/15.09 for $T_{\text{warm}} \in$

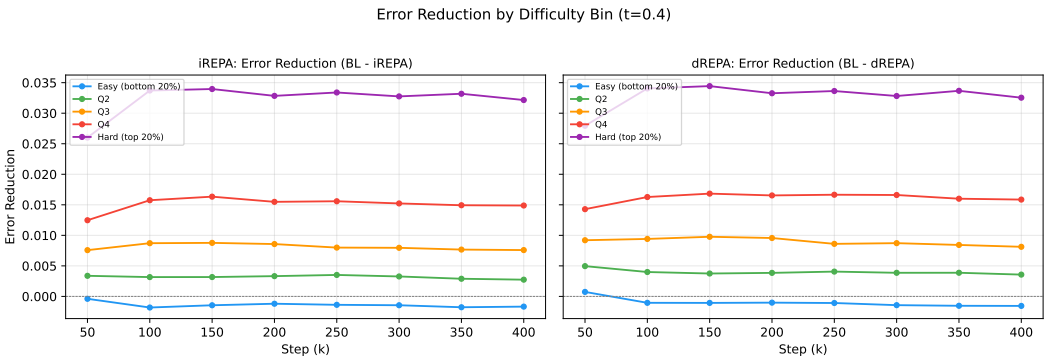

Figure 6: **Training dynamics of per-bin error reduction at t=0.4.** Left: iREPA; Right: iREPA + dREPA. Both methods show a clear monotone ordering (hard > mid > easy) that emerges early and persists throughout training. dREPA provides slightly larger reduction in mid-to-hard bins across all checkpoints (50K–400K).

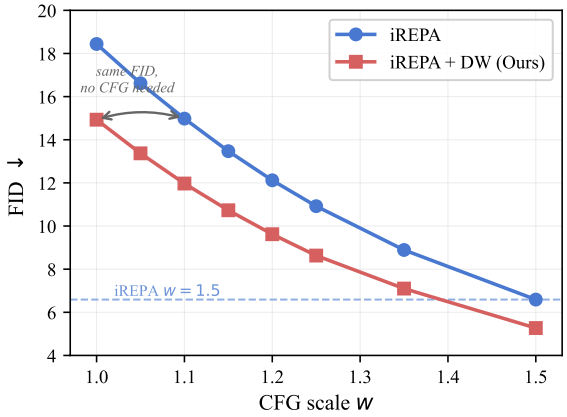

Figure 7: **CFG scale sweep on SiT-B/2.** FID vs. CFG scale w for iREPA and iREPA+dREPA. DINOv2-B, 400K steps. Default: w=1.5.

$\{0, 10\text{K}, 20\text{K}, 50\text{K}\}$. Sweeping the alignment layer gives 19.34/14.93/16.29/19.48/28.29 for layers 2/4/6/8/10; since moving the layer changes both where the alignment regularizer acts and where the weighting signal is read, this trend is not attributable to a single mechanism.

Baseline recipe. The iREPA codebase defaults to fp16 mixed precision with a learning rate of 10^{-4} and gradient clipping at 1.0. We find that fp16 produces NaN losses during extended training ($\sim$400K steps), likely due to its limited dynamic range (5-bit exponent). Switching to bf16 (8-bit exponent, matching fp32's range) resolves this instability.

Starting from this bf16 baseline, we conduct a systematic learning rate sweep with gradient clipping raised to 2.0. Table 12 reports early-stage results at 50K steps for rapid selection. Learning rates in the range $[2\times10^{-4},\ 5\times10^{-4}]$ outperform the default 10^{-4} (which, at 8×10^{-5} under clip 2.0, reaches only FID 78.56). However, 5×10^{-4} produces NaN later in training, and 10^{-3} diverges within 50K steps (omitted from the table). We select 2×10^{-4} as the best trade-off between convergence speed and long-term stability. Model weights are updated using iterate averaging (Polyak & Juditsky, 1992) (EMA with decay 0.9999) for sampling.

Table 13 confirms the advantage at full training length. At 400K steps, the optimized configuration (lr $= 2\times10^{-4}$, bf16, clip $= 2.0$) achieves FID 18.43, a 33.6% reduction compared to the bf16-adapted iREPA default (lr $= 10^{-4}$, clip $= 1.0$, FID 27.74). Convergence is also faster: the optimized setting reaches FID 29.19 at 100K steps, surpassing the default's FID 31.90 at 300K steps—roughly

Table 11: **CFG scale sweep** with DINOv2-B (400K steps, SDE sampler, 250 steps, 50K samples). CFG interval $= [0.0, 0.7]$.

Scale	Method	w	FID↓	IS↑	Prec.↑	Rec.↑
B/2	iREPA	1.0	18.43	77.3	0.629	0.638
		1.05	16.62	84.4	0.642	0.638
		1.1	14.98	91.9	0.654	0.634
		1.15	13.47	99.6	0.665	0.629
		1.2	12.12	107.3	0.676	0.622
		1.25	10.92	115.1	0.689	0.619
		1.35	8.89	131.0	0.707	0.612
		1.5	6.59	153.7	0.736	0.602
	iREPA + dREPA	1.0	14.93	88.6	0.659	0.626
		1.05	13.37	96.5	0.673	0.624
		1.1	11.97	104.6	0.686	0.616
		1.15	10.73	112.5	0.698	0.607
		1.2	9.62	120.7	0.709	0.605
		1.25	8.63	129.0	0.719	0.599
		1.35	7.10	144.3	0.733	0.593
		1.5	5.27	169.2	0.767	0.579
L/2	iREPA	1.0	8.94	118.1	0.699	0.639
		1.35	3.46	191.2	0.775	0.610
		1.5	2.65	217.2	0.795	0.594
	iREPA + dREPA	1.0	6.93	132.1	0.730	0.623
		1.35	2.95	205.5	0.797	0.595
		1.5	2.51	234.0	0.818	0.575
XL/2	iREPA	1.0	7.84	124.5	0.712	0.644
		1.35	3.09	195.3	0.782	0.605
		1.5	2.43	225.3	0.804	0.585
	iREPA + dREPA	1.0	6.19	140.4	0.735	0.618
		1.35	2.68	214.2	0.802	0.589
		1.5	2.41	241.7	0.824	0.577

Table 12: **Learning rate sweep (50K steps).** SiT-B/2, DINOv2-B, bf16, gradient clipping at 2.0, w/o CFG, SDE sampler, 250 steps. †: produces NaN after 50K steps.

Learning rate	FID↓	IS↑	Prec.↑	Rec.↑	
8×10^{-5}	78.56	17.81	0.325	0.452	
2×10^{-4}	**56.03**	24.33	0.426	0.542	
3×10^{-4}	50.35	27.73	0.459	0.555	
5×10^{-4}	43.75	**32.40**	**0.506**	**0.557**	†

$3\times$ fewer steps to reach the same quality. Based on these results, all experiments in this paper adopt lr $= 2\times10^{-4}$, gradient clipping at 2.0, and bf16 mixed precision (fp16 for MMDiT).

Table 13: **Full training comparison (400K steps).** SiT-B/2, DINOv2-B, w/o CFG, SDE sampler, 250 steps, 50K samples. Both use bf16.

Configuration	lr	clip	FID↓	IS↑	Prec.↑	Rec.↑
iREPA default + bf16	10^{-4}	1.0	27.74	55.90	0.564	**0.648**
Ours	2×10^{-4}	2.0	**18.43**	**77.33**	**0.629**	0.638

K ADDITIONAL MAIN RESULTS

This section collects the full-metric tables and the CFG-scale figure summarized in Sec. 5.2: encoder generality on SiT-B/2 (Table 14), resolution scaling on ImageNet 512×512 (Table 15), text-to-image generation on MS-COCO (Table 16), the long-horizon SiT-XL/2 runs (Table 17), and FID/IS across CFG scales for the three SiT sizes (Figure 8).

Table 14: **Encoder generality on SiT-B/2** (w/o CFG, 400K steps). Full five-metric evaluation across three teacher encoders. Δ rows show the improvement from adding dREPA. Training curves in Appendix E.

	DINOv2-B					*DINOv3-B/16*					*PE-Core-G*				
Method	**IS↑**	**FID↓**	**sFID↓**	**Prec.↑**	**Rec.↑**	**IS↑**	**FID↓**	**sFID↓**	**Prec.↑**	**Rec.↑**	**IS↑**	**FID↓**	**sFID↓**	**Prec.↑**	**Rec.↑**
REPA	72.95	19.34	6.092	0.6195	0.6394	70.83	20.41	6.344	0.6143	0.6504	61.95	23.35	6.034	0.5997	0.6389
REPA + dREPA (Ours)	77.08	17.79	5.842	0.6318	0.6307	71.24	19.92	6.081	0.6183	0.6425	63.06	22.98	6.035	0.5982	0.6406
Δ	+4.13	−1.55	−0.250	+.012	−.009	+0.41	−0.49	−0.263	+.004	−.008	+1.11	−0.37	+0.001	−.002	+.002
iREPA	77.33	18.43	6.161	0.6289	0.6379	79.43	17.74	6.401	0.6232	0.6497	76.43	19.40	6.496	0.6132	0.6503
iREPA + dREPA (Ours)	**88.58**	**14.93**	**5.664**	**0.6594**	0.6257	82.52	16.47	**6.061**	0.6339	0.6428	82.91	16.94	6.204	0.6383	0.6327
Δ	+11.25	−3.50	−0.497	+.031	−.012	+3.09	−1.27	−0.340	+.011	−.007	+6.48	−2.46	−0.292	+.025	−.018

Table 15: **Resolution scaling: ImageNet** 512×512 (SiT-B/4, DINOv2-B, 400K steps). w/ CFG uses w=1.5. dREPA improves every baseline at both CFG settings; the absolute gains on iREPA (−3.89 w/o CFG, −2.52 w/ CFG) are comparable to the 256×256 results (Table 3).

	w/o CFG				**w/ CFG** (w=1.5)			
Method	**IS↑**	**FID↓**	**sFID↓**	**Prec.↑**	**IS↑**	**FID↓**	**sFID↓**	**Prec.↑**
Vanilla SiT	23.43	61.18	7.711	0.541	32.42	47.24	7.026	0.626
REPA	43.29	36.05	7.543	0.670	67.66	24.10	6.849	0.740
+ dREPA	45.48	33.81	6.629	0.684	69.93	22.50	6.077	0.754
iREPA	55.62	28.92	7.528	0.706	91.03	17.06	6.758	0.772
+ dREPA	**61.69**	**25.03**	**6.761**	**0.728**	**100.12**	**14.54**	**6.157**	**0.793**

Table 16: **Text-to-image generation** on MS-COCO 256×256 with MMDiT (150K steps). FID across CFG scales.

	FID↓		
CFG	MMDiT	+ iREPA	+ dREPA
1.0	13.12	8.43	**7.61**
1.2	10.78	6.89	**6.36**
1.5	8.35	5.51	**5.27**
2.0	6.15	4.46	**4.35**

K.1 MATCHED-SEED STUDY ON SIT-XL/2

Table 18 reports a paired continuation-seed study: three independently seeded continuations are forked from a shared SiT-XL/2 checkpoint at 300K steps and trained to 400K under both methods, the two arms within a branch differing only in the denoising-loss weighting. Table 19 re-evaluates the original 400K checkpoints with three independent sampling seeds. All numbers are 50K-sample FID. Without CFG, dREPA is better in every branch and under every sampling seed; at w=1.5 the difference is of the same order as the branch-to-branch and seed-to-seed variation, so we report that setting as a tie. Because the branches share their first 300K steps, the continuation study measures late-training stochasticity and understates full training-seed variability.

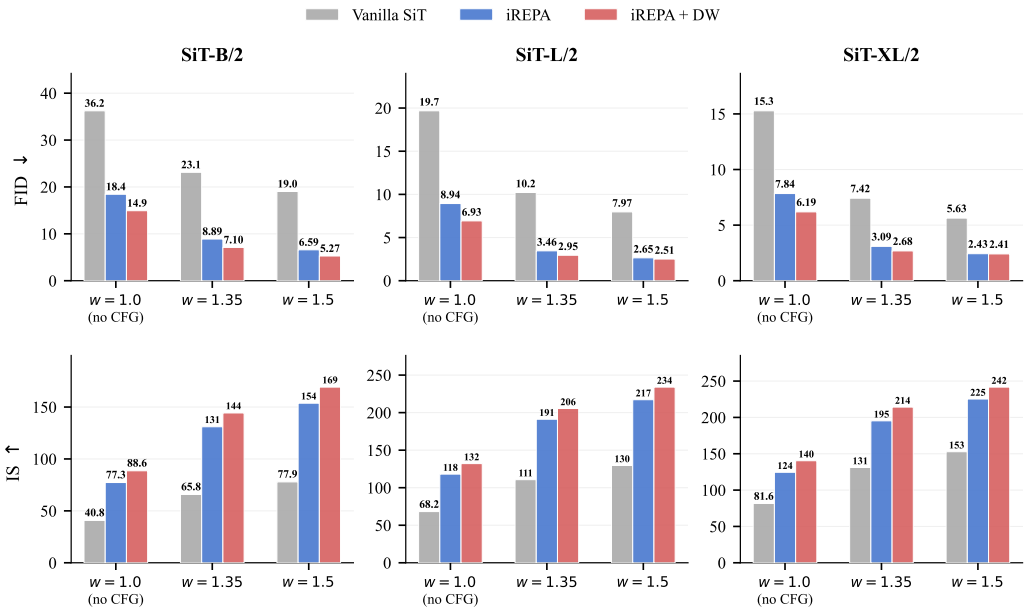

Figure 8: **FID and IS across CFG scales. Top:** FID↓; **Bottom:** IS↑. dREPA (red) improves both metrics at every scale and CFG strength. The FID gap narrows with stronger CFG, indicating partial overlap between CFG and dREPA. 400K steps, DINOv2-B.

Table 17: **Long-horizon training on SiT-XL/2** (w/o CFG, 50K-sample FID; plotted in Fig. 1a). The 7M-step vanilla SiT-XL/2 entry is the value reported by Ma et al. (2024).

Method	**Iters**	**FID↓**	vs. SiT-XL/2	vs. iREPA
SiT-XL/2	7M	8.3	—	—
iREPA	400K	7.84	−5.5%	—
+ dREPA	400K	**6.19**	−25.4%	−21.0%
iREPA	1M	6.28	−24.3%	—
+ dREPA	1M	**5.75**	−30.7%	−8.4%
iREPA	4M	5.91	−28.8%	—
+ dREPA	4M	**5.60**	−32.5%	−5.2%

Table 18: **Paired continuation-seed study** (SiT-XL/2, fork at 300K, evaluated at 400K, 50K-sample FID).

	w/o CFG			w=1.5		
Branch	iREPA	dREPA	Δ	iREPA	dREPA	Δ
1	7.84	6.19	−1.65	2.43	2.42	−0.01
2	8.07	6.35	−1.72	2.55	2.49	−0.06
3	7.69	6.16	−1.53	2.40	2.38	−0.02
Mean ± std	7.87 ± 0.19	6.23 ± 0.10	**−1.63 ± 0.10**	2.46 ± 0.08	2.43 ± 0.06	**−0.03 ± 0.03**

Table 19: **Sampling-seed variation** (original SiT-XL/2 400K checkpoints, three independent sampling seeds, 50K-sample FID).

	w/o CFG			w=1.5		
Seed	iREPA	dREPA	Δ	iREPA	dREPA	Δ
1	7.84	6.19	−1.65	2.43	2.42	−0.01
2	7.89	6.16	−1.73	2.46	2.40	−0.06
3	7.87	6.18	−1.69	2.42	2.41	−0.01
Mean ± std	7.87 ± 0.03	6.18 ± 0.02	**−1.69 ± 0.04**	2.44 ± 0.02	2.41 ± 0.01	**−0.03 ± 0.03**

L QUALITATIVE COMPARISON

Figures 9–14 show per-class qualitative comparisons on SiT-B/2, and Figures 15–29 on SiT-XL/2. Each figure shows eight samples (columns) generated by three methods (rows): Vanilla SiT, iREPA, and iREPA+dREPA. All use DINOv2-B, 400K steps, CFG $w{=}1.5$ with interval $[0.0, 0.7]$. Each column shares the same random seed across methods. Best viewed zoomed in.

SiT-B/2 observations. Across the six B/2 classes (Figures 9–14), iREPA+dREPA tends to produce sharper object boundaries and more coherent fine structures compared to both Vanilla SiT and iREPA. The differences are most apparent in classes with complex textures and edges: feather detail in *Ostrich* (Figure 9), fur texture in *Bearskin* (Figure 10), and handle geometry in *Coffee mug* (Figure 11). Vanilla SiT samples often exhibit blurred boundaries and structural artifacts (e.g., distorted body proportions), while iREPA reduces these artifacts and dREPA further improves boundary regions in many cases.

SiT-XL/2 observations. At the XL/2 scale (Figures 15–29), all three methods produce high-quality samples, and the differences between iREPA and iREPA+dREPA are subtler—consistent with the smaller FID gap at larger scale (Table 3). When visible, dREPA's improvements appear in fine-grained detail: web structure in *Garden spider* (Figure 15), feather patterns in *Bee eater* (Figure 17), and hair texture in *Baboon* (Figure 20). The improvements appear across semantically diverse classes spanning animals, objects, and scenes, suggesting that the spatial reweighting mechanism does not favor specific object categories.



Figure 9: **Ostrich** (class 9). SiT-B/2.



Figure 10: **Bearskin** (class 439). SiT-B/2.



Figure 11: **Coffee mug** (class 502). SiT-B/2.



Figure 12: **Hamper** (class 587). SiT-B/2.



Figure 13: **Lipstick** (class 643). SiT-B/2.



Figure 14: **Soccer ball** (class 806). SiT-B/2.

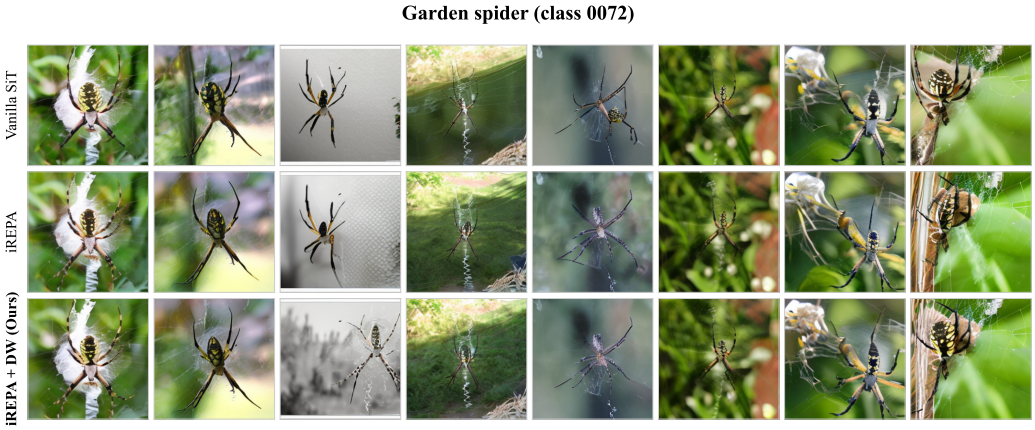

Figure 15: **Garden spider** (class 72). SiT-XL/2.

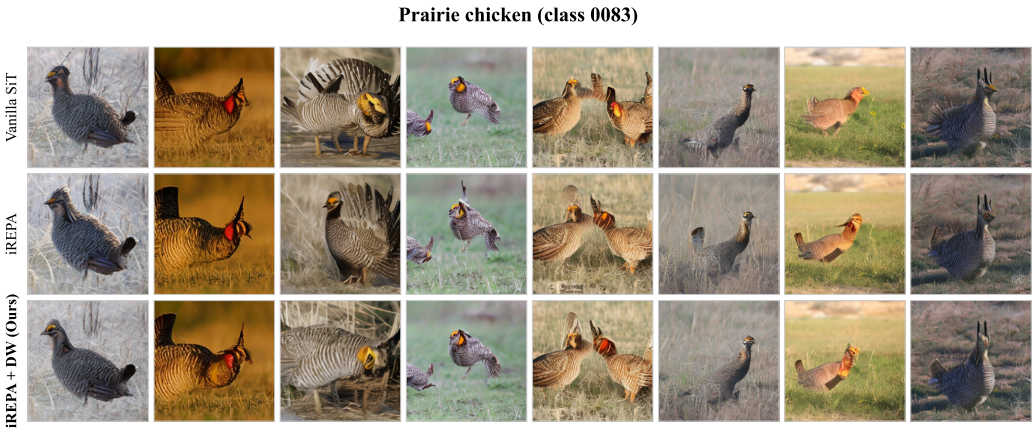

Figure 16: **Prairie chicken** (class 83). SiT-XL/2.

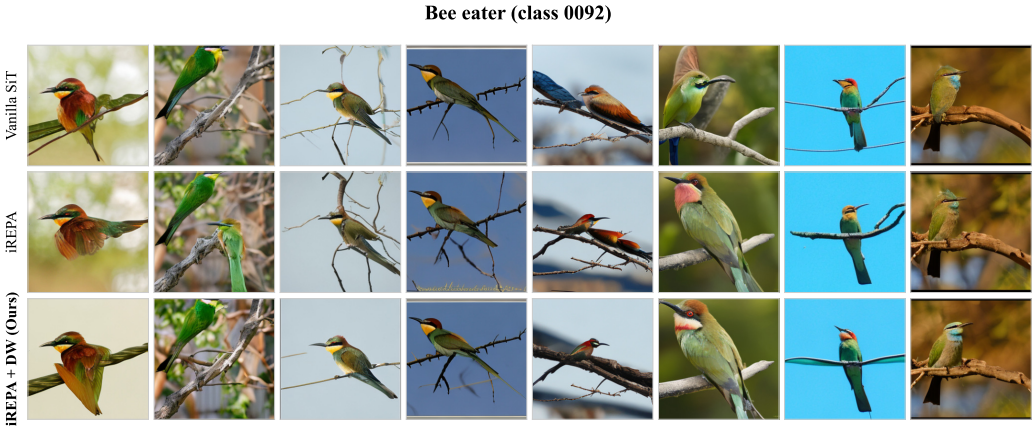

Figure 17: **Bee eater** (class 92). SiT-XL/2.

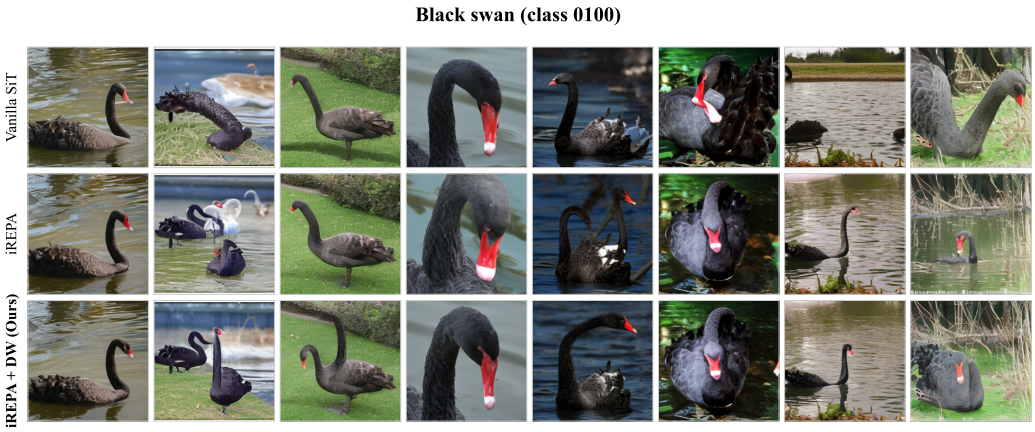

Figure 18: **Black swan** (class 100). SiT-XL/2.

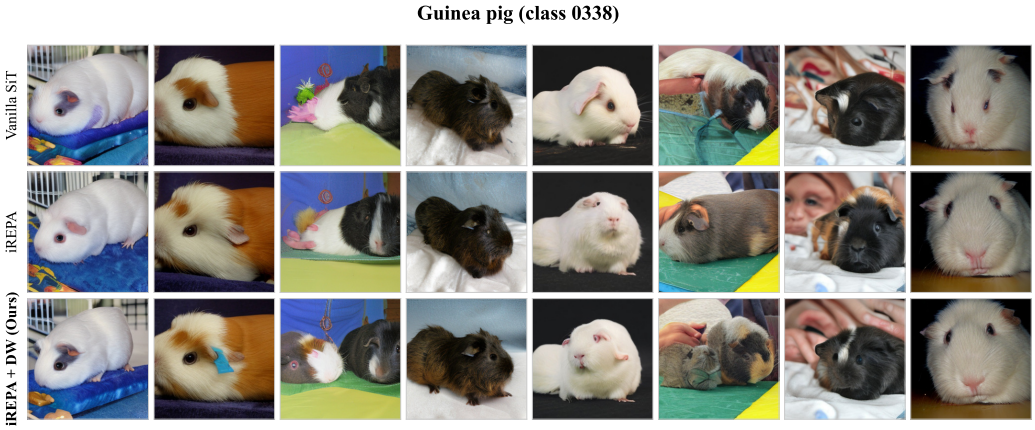

Figure 19: **Guinea pig** (class 338). SiT-XL/2.

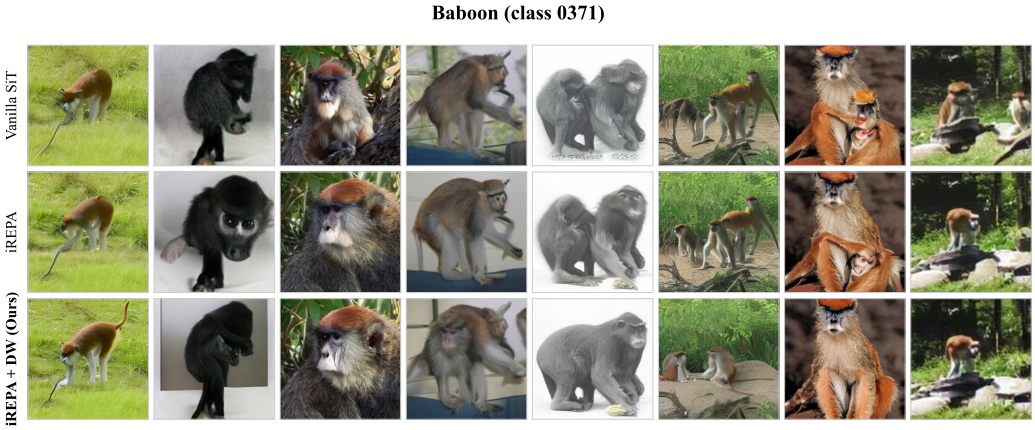

Figure 20: **Baboon** (class 371). SiT-XL/2.



Figure 21: **Broom** (class 462). SiT-XL/2.



Figure 22: **Cowboy hat** (class 515). SiT-XL/2.



Figure 23: **Flute** (class 558). SiT-XL/2.

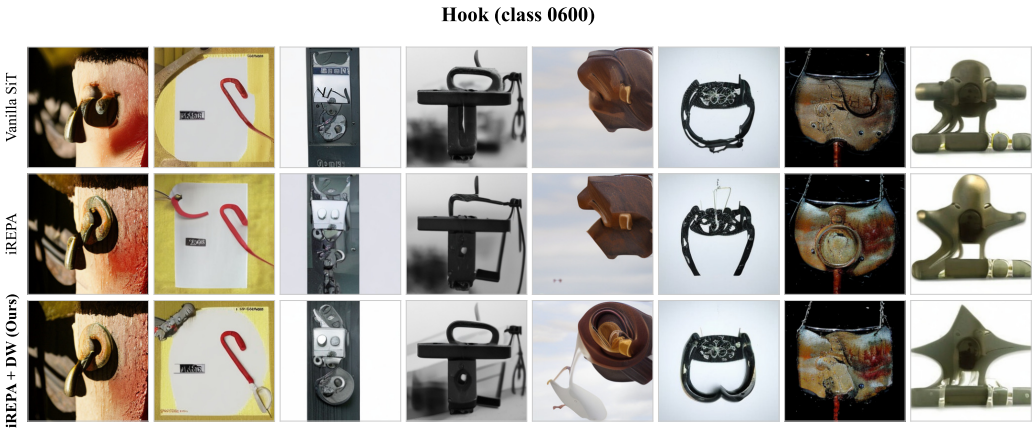

Figure 24: **Hook** (class 600). SiT-XL/2.

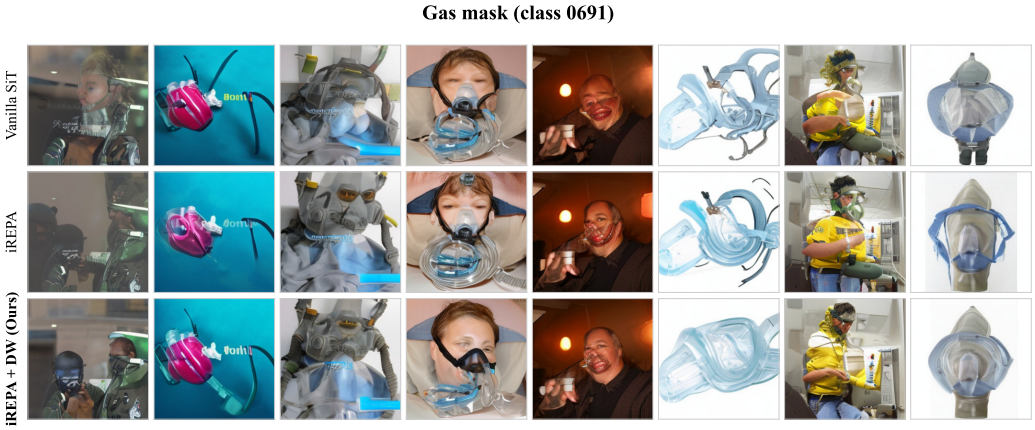

Figure 25: **Gas mask** (class 691). SiT-XL/2.

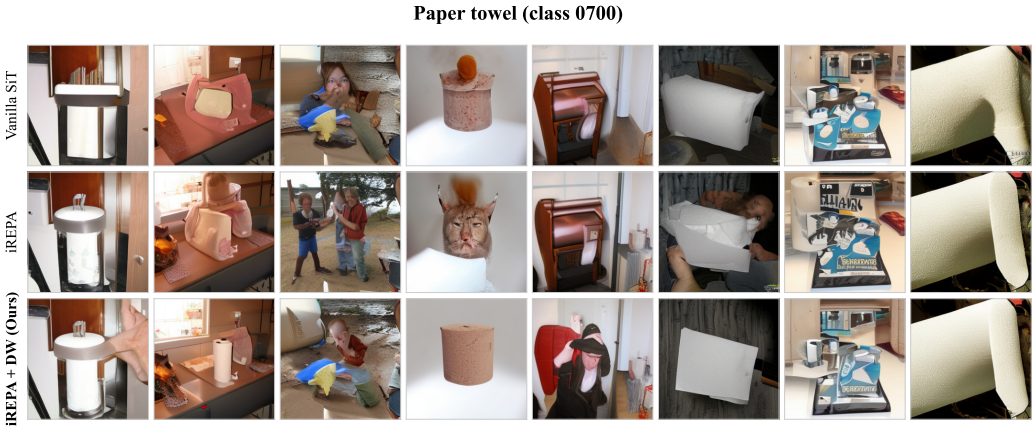

Figure 26: **Paper towel** (class 700). SiT-XL/2.



Figure 27: **Spatula** (class 813). SiT-XL/2.



Figure 28: **Wooden spoon** (class 910). SiT-XL/2.



Figure 29: **Dough** (class 961). SiT-XL/2.