Analyzing Learning-Dynamics Across Loss and Prediction Choices for Diffusion Models

Dongwoo Kim
CSE & GSAI, POSTECH, South Korea

Abstract

Denoising generative models predict clean data (x), noise (ϵ), or velocity (v), while training losses can be measured in any of these spaces. Although the resulting nine combinations are algebraically related, they can yield substantially different denoising accuracy and sample quality. We disentangle the effects of prediction and loss spaces. For a fixed prediction target, we show that changing the loss space is exactly equivalent to timestep reweighting and provide a heuristic analysis of how this reweighting affects generation quality. We then show that a rank-m linear bottleneck compressing a D-dimensional noisy input imposes an irreducible squared-error floor of $D - m$ on noise and velocity prediction, whereas clean-data prediction can exploit low-dimensional data structure. Combining this bound with prediction-dependent error conversion yields distinct necessary conditions on bottleneck capacity. Experiments with U-Net and JiT across CIFAR-10, Imagenette, and ImageNet-256 support these theoretical predictions. Under compression, prediction choice strongly affects both denoising accuracy and sample quality, with noise-prediction errors closely matching the theoretical lower bound. Our results provide a quantitative understanding of how architectural bottlenecks interact with prediction and loss choices in denoising generative models.

1 Introduction

Diffusion and flow-based generative models train a neural network to reverse a corruption process: given a noisy observation z_t at noise level t, the network predicts a target quantity (clean data x, noise ϵ, or velocity v) from which the original signal can be recovered [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song and Ermon, 2019, Lipman et al., 2023, Liu et al., 2023]. Since (x, ϵ, v) are related by affine transforms conditioned on (z_t, t), these parameterizations are algebraically equivalent in the oracle regime, and prior work often treats them as interchangeable up to loss reweighting [Salimans and Ho, 2022, Karras et al., 2022, Kingma and Gao, 2023].

Yet algebraic equivalence does not guarantee comparable performance in finite-capacity networks. Li and He [2026] introduce JiT, a pixel-space Transformer with low-rank patch embeddings, and report substantially better FID with x-prediction than with ϵ- or v-prediction. They motivate x-prediction through the low-dimensional structure of natural images, suggesting that architectural compression may affect prediction targets differently. However, how such compression affects the relative difficulty of different prediction targets remains insufficiently understood.

We revisit this question by separating two design choices: the *prediction space* (what the network predicts) and the *loss space* (where errors are measured), considering all nine combinations over $\{x, \epsilon, v\}$. Although prior work has examined prediction parameterizations and loss weightings jointly [Li and He, 2026, Gagneux et al., 2026], the effects of architectural information loss and prediction-dependent error conversion remain less understood. We analyze loss-space choices through timestep reweighting and prediction-space choices through bottleneck-induced error bounds and conversion amplification. Together, these analyses yield testable predictions about how architectural compression affects denoising accuracy across prediction targets.

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

Contributions. We summarize our contributions as follows.

1. **A joint 3×3 analysis of prediction and loss spaces.** We show that changing the loss space for a fixed prediction target is exactly equivalent to deterministic timestep reweighting (Proposition 1), collecting known affine identities into a unified formulation. We further provide a heuristic analysis of how this reweighting affects generation quality, suggesting that ϵ- and v-loss can outperform x-loss under high-t sensitivity assumptions (Proposition 2).
2. **Irreducible prediction errors under architectural bottlenecks.** For a rank-m linear bottleneck compressing a D-dimensional noisy input, we establish an error floor of $D - m$ for ϵ- and v-prediction, whereas x-prediction can exploit the low-dimensional structure of the data (Theorem 4 and Corollary 5). This result quantifies the information-loss mechanism underlying the prediction-space asymmetry observed in JiT [Li and He, 2026].
3. **Prediction-dependent error amplification.** We characterize how converting different prediction targets to clean-data estimates scales their errors, with factors $(1-t)^2/t^2$ for ϵ-prediction and $(1-t)^2$ for v-prediction (Proposition 6). Combining these identities with the bottleneck error floor yields prediction-dependent necessary conditions on bottleneck capacity (Corollary 7).
4. **Empirical validation across architectures and scales.** We evaluate prediction and loss choices using synthetic data, U-Net, and JiT across CIFAR-10, Imagenette, and ImageNet-256. Our experiments support the predicted effects of loss-space reweighting and quantitatively validate the bottleneck error bounds. In particular, the ϵ-prediction error closely matches the theoretical floor under strong compression, while bottleneck-width sweeps reveal distinct capacity requirements across prediction targets.

2 Related work

Prediction parameterizations and loss weighting. Common diffusion parameterizations predict noise [Ho et al., 2020], the score [Song et al., 2021], or a velocity-related quantity. Salimans and Ho [2022] study prediction parameterizations and loss weightings jointly. EDM [Karras et al., 2022] introduces preconditioning that normalizes the scales of network inputs and regression targets across noise levels. Kingma and Gao [2023] express common diffusion objectives as weighted losses and connect monotonic weightings to ELBO maximization with noisy data augmentation. Explicit weighting strategies include Min-SNR [Hang et al., 2023] and P2 [Choi et al., 2022], while Chen [2023] study noise-schedule design. For our linear interpolation, Proposition 1 collects the established loss-conversion identities into a common statement for a fixed prediction space.

Joint analyses of prediction and loss. JiT [Li and He, 2026], a pixel-space Transformer using x-prediction with v-loss, reports substantial performance differences across prediction targets, motivating a systematic analysis of prediction and loss choices. Their ablation evaluates all nine combinations of prediction and loss spaces. Kumar et al. [2025] compare native-target squared-error objectives and their ELBO-derived counterparts across data, noise, velocity, and score parameterizations, and analyze the rescalings relating these objectives. Gagneux et al. [2026] systematically study the interplay between output parameterization and loss weighting, comparing denoising PSNR across noise levels and FID under different architectures and data regimes. Their inverse-variance analysis further provides a statistical motivation for emphasizing low-noise timesteps.

Dimensionality, bottlenecks, and error conversion. Jin and Wang [2026] analyze the learning dynamics of linear diffusion predictors by decomposing their behavior into components parallel and orthogonal to the data subspace. They characterize how data dimensionality affects the optimal prediction target and propose k-Diff, which learns a data-dependent prediction parameter. We study predictions formed from a rank-m linear compression of a D-dimensional noisy input. Under the stated assumptions, we derive an expected squared ℓ_2 error floor of $D - m$ for noise and velocity prediction (Theorem 4 and Corollary 5). Prior work has discussed numerical instability and error amplification associated with converting noise predictions to clean-data estimates [Salimans and Ho, 2022, Karras et al., 2022, Hoogeboom et al., 2023]. Under linear interpolation, we combine the corresponding error-scaling identities with our bottleneck bounds to obtain prediction-dependent necessary conditions on bottleneck rank (Proposition 6 and Corollary 7).

3 Setting and notation

We consider training data $x \sim p_{\text{data}}$ in $\mathbb{R}^D$ and noise $\epsilon \sim \mathcal{N}(0, \mathrm{I}_D)$. For $t \in (0,1)$, define the noisy observation

$$z_t = tx + (1-t)\epsilon, \tag{1}$$

and the velocity

$$v = x - \epsilon. \tag{2}$$

A model is parameterized by a neural network whose *direct output* (prediction space) is one of

$$u \in \{x, \epsilon, v\}, \qquad u_\theta(z_t, t) = \mathrm{net}_\theta(z_t, t),$$

where $\mathrm{net}_\theta(z_t, t)$ is modeled as a trainable neural network parameterized by θ. Given (z_t, t) and one predicted quantity, the other two are obtained by solving the linear system induced by (1) and (2). This yields deterministic affine transforms among $\{x_\theta, \epsilon_\theta, v_\theta\}$.

Training uses a squared loss in a chosen *loss space* $s \in \{x, \epsilon, v\}$:

$$\mathcal{L}_{s|u}(\theta) \;=\; \mathbb{E}\Big[\, \|s_\theta(z_t, t) - s\|^2 \,\Big], \tag{3}$$

where s_θ is computed from u_θ by the appropriate transform. Note that v-prediction determines the score: by Tweedie's formula, $\nabla_{z_t} \log p(z_t) = \big(t\, v^\star(z_t, t) - z_t\big)/(1-t)$ [Song et al., 2021], an affine map whose time-dependent scaling corresponds to a specific loss weighting within our framework.

Goal. In the oracle regime, all (s, u) are equivalent up to deterministic transformations. The scientific question is to formalize how *learning dynamics* differ for finite capacity and gradient-based optimization, and to design experiments that disentangle (i) timestep reweighting effects from (ii) geometric and capacity effects.

4 Learning dynamics of diffusion

We analyze how prediction and loss choices affect learning through two distinct mechanisms. We first establish that changing the loss space induces deterministic timestep reweighting and examine how this weighting affects generation quality. We then turn to the prediction-space axis, studying how architectural bottlenecks introduce irreducible errors and how prediction-dependent conversion amplifies these errors. Proofs for the theoretical results are collected in Appendix A.

Proposition 1 (Equivalence of loss spaces as timestep reweighting)**.** *Assume the linear interpolation and velocity definition*

$$z_t = tx + (1-t)\epsilon, \qquad v = x - \epsilon, \qquad t \in (0,1),$$

with (x, ϵ) sampled from the data and noise distributions. Fix a prediction space $u \in \{x, \epsilon, v\}$*, meaning the network directly outputs*

$$u_\theta(z_t, t) = \mathrm{net}_\theta(z_t, t).$$

For any loss space $s \in \{x, \epsilon, v\}$*, define $s_\theta(z_t, t)$ as the quantity obtained by solving the linear system induced by (z_t, t) together with the constraints defining the pair (x, ϵ, v), with u replaced by u_θ.*

Then for every pair $(s, u) \in \{x, \epsilon, v\}^2$ there exist:

- *a scalar slope $a_{s|u}(t) \neq 0$ that depends* only *on t, and*
- *an intercept term $b_{s|u}(z_t, t)$ that depends on (z_t, t) but is* independent of *u,*

such that the true *target and the* model *prediction satisfy the aligned affine relations*

$$s \;=\; a_{s|u}(t)\, u + b_{s|u}(z_t, t),$$

and

$$s_\theta(z_t, t) \;=\; a_{s|u}(t)\, u_\theta(z_t, t) + b_{s|u}(z_t, t).$$

Consequently, the residuals obey the pointwise scaling identity

$$s_\theta(z_t, t) - s \;=\; a_{s|u}(t)\, \big(u_\theta(z_t, t) - u\big), \tag{4}$$

Table 1: The implicit timestep weights $w_{s|u}(t)$ induced by different loss-space choices, derived from the linear-schedule relations between (x, ϵ, v) collected from prior work [Salimans and Ho, 2022, Kingma and Gao, 2023, Karras et al., 2022]. Diagonal entries (u=s) are unity. Off-diagonal entries involving $(1-t)^{-1}$ or t^{-1} diverge at endpoints and require clamping in practice.

Weight $w_{s\|u}(t)$	Loss $s = x$	Loss $s = \epsilon$	Loss $s = v$
Pred $u = x$	1	$\left(\frac{t}{1-t}\right)^2$	$\left(\frac{1}{1-t}\right)^2$
Pred $u = \epsilon$	$\left(\frac{1-t}{t}\right)^2$	1	$\left(\frac{1}{t}\right)^2$
Pred $u = v$	$(1-t)^2$	t^2	1

and hence the squared loss in s-space is exactly *a weighted squared loss in the direct output u-space:*

$$\begin{aligned}\mathcal{L}_{s|u}(\theta) &:= \mathbb{E}\big[\|s_\theta(z_t, t) - s\|^2\big] \\ &= \mathbb{E}\Big[w_{s|u}(t)\,\|u_\theta(z_t, t) - u\|^2\Big],\end{aligned} \tag{5}$$

where $w_{s|u}(t) := a_{s|u}(t)^2 > 0$. Since $w_{s|u}$ does not depend on θ, the Bayes-optimal predictor $u^\star(z_t, t) = \mathbb{E}[u \mid z_t, t]$ is the same for all loss spaces s. For the linear schedule, the weights are listed in Table 1; the individual identities appear in prior work [Salimans and Ho, 2022, Kingma and Gao, 2023, Karras et al., 2022], and we collect them here for reference.

Proposition 1 states that changing loss space s only reweights timesteps; the Bayes-optimal predictor $u^\star$ is unchanged. Empirical differences must arise from finite capacity and optimization dynamics, not from a different oracle. In expectation, applying a weight $w(t)$ in the loss is equivalent to sampling t from a tilted distribution $\propto w(t)\,p(t)$. In practice, off-diagonal weights can diverge (e.g., $w_{v|x}(t) = (1-t)^{-2}$); following [Li and He, 2026], we clamp $(1-t) \geq 0.05$. By contrast, changing prediction space u changes $u^\star$ itself, so is *not* covered by this equivalence.

The weights connect to the SNR framework [Kingma and Gao, 2023]: for x-prediction, $w_{\epsilon|x}(t) = \mathrm{SNR}(t) = t^2/(1-t)^2$, so ϵ-loss concentrates optimization on high-SNR timesteps while x-loss distributes it uniformly. This subsumes EDM's weighting [Karras et al., 2022] and explains why ϵ/v-loss emphasize the generation-critical high-t regime.

4.1 Loss-space ordering: Why some weightings are better

Proposition 1 establishes that changing the loss space s (with fixed prediction space u) is equivalent to changing the timestep weighting $w_{s|u}(t)$. At the population level, all weightings yield the same Bayes-optimal predictor $u^\star(z_t, t)$. However, with finite model capacity, the weighting determines which timesteps receive more optimization pressure. We now provide a heuristic analysis of why this matters for generation quality, then validate the key prediction empirically.

Setup. Consider an ODE-based sampler that integrates from t=0 (noise) to t=1 (data) using the learned velocity field $\hat{v}(z_t, t)$. Let $\delta v(t) := \hat{v}(z_t, t) - v^\star(z_t, t)$ denote the velocity error at time t, and $\mathrm{MSE}_v(t) := \mathbb{E}\|\delta v(t)\|^2$. By a first-order perturbation of the ODE $dz/dt = v(z_t, t)$, the final-sample error satisfies

$$\delta z_1 \;\approx\; \int_0^1 \delta v(t)\,dt. \tag{6}$$

This linearization is accurate when velocity errors are small relative to the signal. For the heuristic surrogate below, we restrict the integration to an interior timestep interval $\mathcal{T} = [0.05, 0.95]$, neglecting contributions outside this interval. By Proposition 1 with u=v, a loss in space s weights the squared velocity error by the effective weight $w_{\mathrm{eff}}(t) := w_{s|v}(t)$, which is $(1-t)^2$, t^2, and 1 for x-, ϵ-, and v-loss (Table 1). These effective weights do not depend on the prediction space u.

Proposition 2 (Loss-Space Ordering (heuristic))**.** *Consider the following modeling assumptions:*

(i) ***First-order approximation:*** *The sample error is dominated by (6).*

(ii) ***Cross-time correlations:*** *By (i), $\mathbb{E}\|\delta z_1\|^2 \approx \int_0^1\int_0^1 \mathbb{E}[\delta v(t)^\top \delta v(t')]\,dt\,dt'$. We summarize the cross-time correlations by a sensitivity $\kappa(t) \geq 0$, so that $\mathbb{E}\|\delta z_1\|^2 \approx \int_{\mathcal{T}} \kappa(t)\,\mathrm{MSE}_v(t)\,dt$, and assume that κ does not depend on the training weighting.*

(iii) ***Capacity allocation:*** *With finite model capacity trained under effective weight $w_{\text{eff}}(t)$ and time distribution $p(t)$, the per-timestep velocity MSE scales as*

$$\mathrm{MSE}_v(t) \;\propto\; \frac{1}{p(t)\,w_{\text{eff}}(t)}. \tag{7}$$

Under these assumptions, the expected sample error satisfies

$$\mathbb{E}\big[\|\delta z_1\|^2\big] \;\propto\; J_\kappa(w_{\text{eff}}) := \int_{\mathcal{T}} \frac{\kappa(t)}{p(t)\,w_{\text{eff}}(t)}\,dt. \tag{8}$$

If in addition

(iv) ***High-t sensitivity:*** *for natural images, $\kappa(t)$ is concentrated at high t,*

then

$$\textit{generation quality}(\epsilon\textit{-loss}) \;\approx\; \textit{generation quality}(v\textit{-loss}) \;\gg\; \textit{generation quality}(x\textit{-loss}). \tag{9}$$

Derivation. By (i) and (ii), $\mathbb{E}\|\delta z_1\|^2 \approx \int_0^1 \kappa(t)\,\mathrm{MSE}_v(t)\,dt$, and substituting (iii) gives (8). Under (iv), J_κ is dominated by high t. At high t, the ϵ-loss weight t^2 approaches the v-loss weight 1, whereas the x-loss weight $(1-t)^2$ approaches zero, thereby placing diminishing emphasis on velocity-space errors. The ϵ- and v-loss integrals are therefore close, and the x-loss integral is much larger, which is (9).

Discussion. Assumption (i) approximates the accumulated sampling error by integrating local velocity errors, neglecting their propagation through the evolving trajectory. Assumptions (ii) and (iii) are heuristics: real networks share parameters across timesteps and the actual allocation depends on architecture, optimizer, and data. Assumption (iv) is a property of the data rather than of the model. We state the result as a proposition to make each premise falsifiable; the ordering is heuristic, not universal. Section B.6 evaluates J_κ under the time distributions used in training. With constant κ, the x-loss integral already exceeds the other two by one to two orders of magnitude, so (iv) is needed only to place ϵ-loss next to v-loss. Empirically, the ordering holds for x- and v-prediction on the U-Net and, more weakly, for x-prediction on JiT-B/8 (Section 5). It fails on simple 2D distributions, where (iv) does not hold (Appendix B), and under strong compression on ImageNet-256, where ϵ-loss gives the worst FID for x-prediction (Table 4).

Remark 1 (Connection to existing strategies). Our framework subsumes existing weighting strategies: EDM [Karras et al., 2022] weights the x-space error by our ϵ-loss weight plus a constant (Appendix C), min-SNR [Hang et al., 2023] interpolates between uniform and ϵ-loss, and P2 [Choi et al., 2022] modulates by perceptual priority. All of these upweight high-t relative to x-loss, consistent with (9). DDPM [Ho et al., 2020] uses (ϵ, ϵ), SD3 [Esser et al., 2024] uses (v, v), JiT [Li and He, 2026] uses (x, v).

4.2 Bottleneck lower bounds from information loss: why x is special under manifold structure

We now turn from loss-space reweighting to the effect of prediction-space choices under architectural bottlenecks. Clean data x lies on or near a low-dimensional manifold of natural images, whereas the Gaussian noise ϵ and the velocity $v = x - \epsilon$ fill the ambient $\mathbb{R}^D$. We will show that combining this asymmetry with an architectural information bottleneck yields a sharp consequence: ϵ- and v-prediction incur an irreducible error floor, while x-prediction remains feasible.

Definition 3 (Bottleneck model)**.** We use the following assumptions.

(A1) **Subspace data:** $x = P\hat{x}$, where $P \in \mathbb{R}^{D\times d}$ satisfies $P^\top P = \mathrm{I}_d$ with $d \ll D$, and $\hat{x} \sim p_{\hat{x}}$ on $\mathbb{R}^d$.

(A2) **Isotropic noise:** $\epsilon \sim \mathcal{N}(0, \mathrm{I}_D)$, independent of x.

(A3) **Linear bottleneck:** the model observes only $\rho = Ez_t \in \mathbb{R}^m$, where $E \in \mathbb{R}^{m \times D}$ and $\mathrm{rank}(E) = m < D$.

(A1) is the linear idealization of the manifold hypothesis. (A3) matches JiT's patch embedding, which applies two linear convolutions with no nonlinearity in between.

For a Vision Transformer, D corresponds to the flattened patch dimension $d_{\mathrm{patch}} = p^2C$ for $p \times p$ patches with C channels, and m corresponds to the patch-embedding dimension d_{bn} to which the architecture compresses each patch before processing. We use d_{patch} and d_{bn} throughout when referring to these quantities for concrete architectures, and write $r := d_{\mathrm{bn}}/d_{\mathrm{patch}}$ for the bottleneck ratio, where $r < 1$ indicates patch-level compression and $r \geq 1$ no compression.

Theorem 4 (Irreducible error from null-space components)**.** *Assume (A3) and let* Π_{null} *denote the orthogonal projector onto* $\mathrm{null}(E)$*. For any measurable estimator* $\hat{u}(\rho, t)$ *of a target* $u \in \mathbb{R}^D$*,*

$$\begin{aligned} \mathbb{E}\big[\, \|\hat{u}(\rho,t) - u\|^2 \,\big] &\geq \mathbb{E}\big[\, \|\mathbb{E}[u \mid \rho, t] - u\|^2 \,\big] \\ &\geq \mathbb{E}\big[\, \|\Pi_{\mathrm{null}} u - \mathbb{E}[\Pi_{\mathrm{null}} u \mid \rho, t]\|^2 \,\big]. \end{aligned} \tag{10}$$

If in addition (A2) holds, then $\Pi_{\mathrm{null}}\epsilon$ *is independent of* (ρ, t)*, and for* $u = \epsilon$

$$\mathbb{E}\big[\, \|\hat{\epsilon}(\rho,t) - \epsilon\|^2 \,\big] \;\geq\; \mathbb{E}\big[\, \|\Pi_{\mathrm{null}}\epsilon\|^2 \,\big] = (D - m).$$

Intuitively, the observation ρ carries no information about the component of ϵ in $\mathrm{null}(E)$, and that component has expected squared norm exactly $D-m$. No estimator that sees only ρ can recover it, so the null-space residual $\mathbb{E}\|\Pi_{\mathrm{null}} u - \mathbb{E}[\Pi_{\mathrm{null}} u \mid \rho, t]\|^2$ is an *irreducible error floor* from information loss through E. Data on a low-dimensional subspace can pass through the projection intact, while full-dimensional noise cannot. The next corollary makes this precise.

Corollary 5 (Bottleneck separation between x and ϵ/v)**.** *Assume (A1) to (A3) and*

(A4) ***Injectivity on the data subspace:*** $\mathrm{rank}(EP) = d$*, which requires* $m \geq d$*.*

Then the bottleneck preserves all directions in the d*-dimensional data subspace, since* $\mathrm{rank}(EP) = d$*. In contrast, every estimator of* $u \in \{\epsilon, v\}$ *based on* (ρ, t) *satisfies*

$$\mathbb{E}\|\hat{u} - u\|^2 \geq D - m.$$

Thus, the bottleneck can preserve the low-dimensional data subspace while discarding noise components in its $(D - m)$-dimensional null space. This creates an irreducible prediction-error floor for ϵ- and v-prediction that does not arise from the same null-space mechanism for x-prediction.

When does the bottleneck apply? U-Nets bypass this via *skip connections* (effective rank D), and EDM's c_{skip} [Karras et al., 2022] provides a direct input-to-output path; Appendix C places EDM in our framework. Architectures with patch-level compression and *no* skip path (JiT [Li and He, 2026], any ViT with $d_{\mathrm{bn}} < d_{\mathrm{patch}}$) satisfy the assumption directly. Since E is learned end-to-end, the gradient incentivizes it to preserve the data subspace for x-prediction, confirmed by the near-identical MSE across loss types in Table 3.

Remark 2 (Nonlinearity after linear compression). The lower bound of Theorem 4 applies to any predictor that observes the noisy input only through a rank-m linear projection $\rho = Ez_t$, regardless of nonlinear processing applied after this projection. In particular, inserting a GELU between JiT's two bottleneck convolutions preserves the initial linear compression and therefore does not violate the assumptions of the theorem. Our ablation in Table 10 shows similar prediction-space asymmetry with and without this additional nonlinearity. This demonstrates robustness to nonlinear processing after compression, rather than extending the theoretical guarantee to arbitrary nonlinear compression maps $f : \mathbb{R}^D \to \mathbb{R}^m$.

4.3 Conversion amplification: Why ϵ-prediction requires more capacity

Corollary 5 gives the same lower bound, $D - m$, on the expected squared prediction errors for ϵ and v under a bottleneck. We show that conversion to x-space yields different lower bounds and, consequently, different necessary conditions on bottleneck rank.

Proposition 6 (Conversion amplification). *Let $\hat{u}(z_t, t)$ be a model's prediction in space $u \in \{x, \epsilon, v\}$, and let $\hat{x}_u(z_t, t)$ denote the clean-data estimate obtained by algebraic conversion. If the model incurs pointwise prediction error $\delta_u := \hat{u} - u$ against the true target, then the induced x-space error satisfies:*

$$\|\hat{x}_x - x\|^2 = \|\delta_x\|^2, \tag{11}$$

$$\|\hat{x}_v - x\|^2 = (1-t)^2\|\delta_v\|^2, \tag{12}$$

$$\|\hat{x}_\epsilon - x\|^2 = \frac{(1-t)^2}{t^2}\|\delta_\epsilon\|^2. \tag{13}$$

The proof follows by direct algebraic conversion: $\hat{x}_v - x = (1-t)\delta_v$ and $\hat{x}_\epsilon - x = \frac{-(1-t)}{t}\delta_\epsilon$.

Under the bottleneck assumptions, both ϵ- and v-prediction incur an expected squared error of at least $D - m$ (Corollary 5). Combining this error floor with the conversion factors in Proposition 6 gives prediction-dependent lower bounds on the x-space error. The following corollary states the resulting necessary conditions on bottleneck rank.

Corollary 7 (Prediction-type-dependent threshold). *Under (A1) – (A4), for any ϵ- or v-predictor based on (ρ, t), the converted x-space MSE satisfies*

$$\mathrm{MSE}_{v\text{-}pred} \geq \mathbb{E}_t[(1-t)^2] \cdot (D-m)_+, \tag{14}$$

$$\mathrm{MSE}_{\epsilon\text{-}pred} \geq \mathbb{E}_t\left[\tfrac{(1-t)^2}{t^2}\right] (D-m)_+. \tag{15}$$

Under logit-normal sampling with $\mu = -0.5$ and $\sigma = 1$, the expected conversion weights are approximately 0.40 for v-prediction and 20 for ϵ-prediction. For the same target x-space MSE, the necessary upper bound on $D - m$ is therefore about 50 times smaller for ϵ-prediction than for v-prediction. These lower bounds become trivial at full rank ($m = D$).

We empirically examine the predicted difference in bottleneck sensitivity in Section 5. The bounds in Corollary 7 become trivial without compression and therefore do not explain the FID differences in that regime. Nevertheless, the per-timestep analysis in Section B.8 reveals error patterns consistent with conversion amplification, although a complete explanation of the FID gap remains open.

5 Experiments

We systematically evaluate all nine combinations of prediction target $u \in \{x, \epsilon, v\}$ and loss space $s \in \{x, \epsilon, v\}$ across two architectures: a standard U-Net with skip connections (∼97M params, no information bottleneck) and the JiT-B architecture [Li and He, 2026] (∼130M params, with a configurable bottleneck in the patch embedding). Experiments span CIFAR-10 (32×32), Imagenette (64×64) and ImageNet-256, with synthetic data validating dimensional scaling.

Setup. The U-Net is trained unconditionally with Adam (β_1=0.9, β_2=0.999, lr $= 2\times10^{-4}$), logit-normal time sampling (μ=−0.5, σ=1.0), gradient clipping at norm 1.0, and EMA with decay 0.9999. JiT-B is trained class-conditionally with AdamW (β_1=0.9, β_2=0.95), logit-normal time (μ=−0.8, σ=0.8), label dropout 0.1 for classifier-free guidance (CFG) [Ho and Salimans, 2022], and EMA with decay 0.9999. The CIFAR-10 and Imagenette models are trained for 500 epochs. For ImageNet-256, we train the compressed 3×3 configurations for 100 epochs and the uncompressed controls for 600 epochs, as detailed in Section B.8. JiT uses the full architecture from [Li and He, 2026] including RMSNorm, RoPE, QK-norm, SwiGLU, and in-context class tokens. We report validation MSE in x-space (averaged over 3 seeds for CIFAR-10 U-Net, 2 seeds for JiT CIFAR-10) and FID (computed with clean-fid [Parmar et al., 2022]; U-Net: Heun, 100 steps, unconditional; JiT: Heun, 50 steps, CFG). JiT's `BottleneckPatchEmbed` applies two linear convolutions *without* an intermediate nonlinearity, so the bottleneck is a rank-d_{bn} linear projection: precisely the setting of Theorem 4. We characterize each configuration by the bottleneck ratio $r := d_{\mathrm{bn}}/d_{\mathrm{patch}}$, where $r < 1$ indicates patch-level compression and $r \geq 1$ no compression.

Loss-space effects without bottleneck: generation quality. Proposition 2 predicts that ϵ-loss and v-loss should outperform x-loss in generation quality by placing greater optimization weight on high-t timesteps. We focus on x- and v-prediction, which achieve substantially better FID than

Table 2: U-Net results across datasets. Validation MSE in x-space and FID (↓).

Dataset	Pred\Loss	MSE ($\times 10^{-2}$)			FID		
		x	ϵ	v	x	ϵ	v
CIFAR-10	x	5.395	5.360	5.361	11.79	2.20	2.73
	ϵ	5.401	5.372	5.367	16.45	25.49	5.88
	v	5.392	**5.356**	5.359	12.51	**1.12**	1.68
Imagenette	x	4.810	4.664	4.632	53.40	12.76	21.05
	ϵ	n/a[†]	4.709	4.670	85.77	95.41	100.98
	v	4.763	4.649	**4.630**	43.11	**7.06**	14.32

[†]One of two runs diverged (NaN), due to unstable gradients caused by the timestep reweighting.

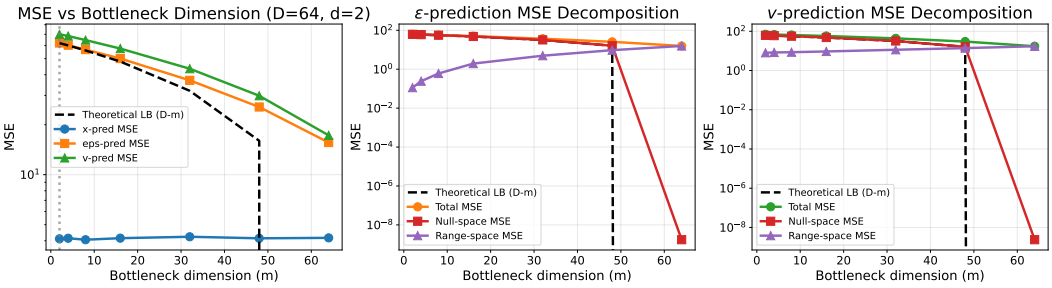


Figure 1: Synthetic validation of the bottleneck lower bound (Theorem 4) with $D = 64$ and $d = 2$. **Left:** Prediction MSE in each target's native space as a function of bottleneck dimension m. The ϵ- and v-prediction errors are bounded below by $D - m$ (dashed), while x-prediction maintains low error for $m \geq d$. **Middle, Right:** Decomposition of the prediction MSE into null-space and range-space components for ϵ- and v-prediction, respectively.

ϵ-prediction under the U-Net setting. The U-Net results in Table 2 support the predicted loss ordering. On CIFAR-10, switching from x-loss to ϵ-loss reduces FID from 11.79 to 2.20 for x-prediction and from 12.51 to 1.12 for v-prediction, with similar improvements on Imagenette. v-loss also consistently outperforms x-loss in these settings.

Prediction-space effects under bottleneck. With an architectural information bottleneck, the prediction target becomes the dominant factor in denoising accuracy. Theorem 4 and Corollary 5 establish an irreducible error floor of $D - m$ for ϵ- and v-prediction under linear compression. Our synthetic experiments in Figure 1 directly validate this bound: the null-space MSE of ϵ-prediction closely matches $D - m$, while that of v-prediction remains above the bound. On Imagenette, Table 3 shows that JiT-B with bottleneck ratio $r = 0.67$ achieves an x-space MSE of approximately 0.048 for x-prediction, compared with 0.152 for v-prediction and 5.85 for ϵ-prediction. The FID results exhibit the same ordering, while changing the loss space within each prediction type has little effect on MSE. When the bottleneck is removed ($r = 1.33$), the MSE gap narrows substantially: x- and v-prediction achieve MSE around 0.049, while ϵ-prediction reaches 0.051. Together, these results show that loss-space reweighting has a relatively modest effect on denoising error, whereas representational information loss can make the prediction target decisive. The same separation is observed on ImageNet-256 under strong compression ($r = 0.17$), where x-prediction achieves an x-space MSE of 0.023, compared with 0.294 for v-prediction and 14.61 for ϵ-prediction (Table 4). An additional ablation inserting a GELU after the initial linear compression exhibits a similar prediction-space asymmetry, indicating robustness to post-bottleneck nonlinear processing (Table 10).

Bottleneck capacity threshold. Corollary 7 shows that, under compression ($r < 1$), the irreducible error floor imposes a stronger constraint on ϵ-prediction than on v-prediction, suggesting that ϵ-prediction requires greater bottleneck capacity to achieve comparable denoising accuracy. Since the bound becomes trivial at $r \geq 1$, we empirically investigate recovery by varying the bottleneck width of BottleneckViT, a lightweight ViT that follows JiT's patch-embedding design but adds a GELU after the initial linear projection (see Appendix B). On CIFAR-10, the gap between v- and x-prediction

Table 3: JiT-B (∼130M params) on Imagenette 64×64 with class conditioning and CFG. **Top:** d_{bn}=128 ($r = 0.67$). **Bottom:** d_{bn}=256 ($r = 1.33$). Under compression, x-prediction achieves threefold better FID than v-prediction and 120-fold better MSE than ϵ-prediction.

		MSE (x-space)			FID		
Config	Pred\Loss	x	ϵ	v	x	ϵ	v
d_{bn}=128 (ratio 0.67)	x	0.050	**0.048**	0.048	41.0	37.4	**36.0**
	ϵ	5.855	5.848	5.852	146.6	147.6	151.0
	v	0.152	0.152	0.152	113.2	108.3	113.0
d_{bn}=256 (ratio 1.33)	x	0.049	**0.048**	0.048	42.1	38.9	**37.1**
	ϵ	0.051	0.051	0.051	114.1	100.1	123.6
	v	0.049	0.050	0.049	58.9	60.3	58.1

Table 4: JiT-B/16 on class-conditional ImageNet 256×256. Validation x-MSE and FID (↓, 50K samples). **Top:** d_{bn}=128 (r=0.17). **Bottom:** d_{bn}=768 (r=1.0). Under compression, prediction type dominates both metrics (ϵ-space MSE $0.834 \approx (D-m)/D = 0.833$; see Section B.8). Without compression and at full budget, MSE converges across prediction types (0.022–0.043) but FID does not, so aggregate denoising MSE does not determine FID.

		x-MSE			FID		
Config	Pred\Loss	x	ϵ	v	x	ϵ	v
d_{bn}=128 (r=0.17)	x	0.023	0.023	**0.023**	23.9	36.8	**20.1**
	ϵ	14.61	14.61	14.61	421.2	419.0	420.1
	v	0.294	0.294	0.294	201.5	202.3	199.0
d_{bn}=768 (r=1.0)	x	—	—	**0.022**	—	—	**5.7**
	ϵ	—	—	0.043	—	—	330.6
	v	—	—	0.026	—	—	140.6

nearly disappears once compression is removed (Table 7). In contrast, on Imagenette, ϵ-prediction remains less accurate even without compression and approaches x-prediction only with a substantially wider embedding ($r \approx 4$; Table 8). This recovery threshold is empirical and is not predicted by Corollary 7. Additional synthetic sweeps over D, d, and m (Section B.1) and a patch-size sweep at fixed embedding width (Section B.2) further support the predicted dependence of prediction errors on $D - m$.

Convergence behavior without compression. The bottleneck lower bound in Theorem 4 applies to rank-deficient embeddings ($r < 1$) and becomes trivial when $r \geq 1$. On JiT-B/4 CIFAR-10 ($r = 2.67$), the initially large denoising MSE gap between ϵ- and x-prediction closes with longer training (Figure 4), suggesting that prediction-type differences without compression can partly reflect optimization difficulty rather than irreversible information loss. However, comparable denoising accuracy does not necessarily imply comparable generation quality. On ImageNet-256 without compression ($r = 1.0$), the models trained for 600 epochs achieve relatively similar x-space MSE, yet their FID scores differ substantially (Table 4).

To examine the role of error conversion, Figure 2 compares per-timestep errors in velocity space, which is used by ODE sampling, and in x-space after conversion. The ϵ-prediction model exhibits sharply increasing velocity-space MSE at low t, consistent with the $1/t^2$ amplification of squared errors during ϵ-to-velocity conversion. In contrast, x- and v-prediction maintain substantially lower errors in this region. These observations suggest that conversion amplification contributes to the observed FID gap, although it does not fully explain the differences, particularly the poor FID of v-prediction. Section B.8 provides further analyses in a common velocity space and compares ODE and SDE sampling.

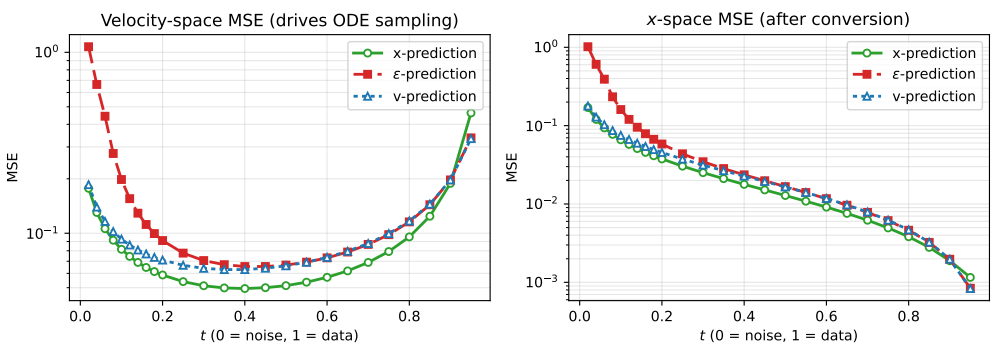


Figure 2: Per-timestep prediction errors on ImageNet-256 using JiT-B/16 without compression ($r = 1.0$, 600 epochs). **Left:** Velocity-space MSE against sampled targets, used to assess errors relevant to ODE sampling. **Right:** x-space MSE after conversion. The velocity-space error of ϵ-prediction increases sharply at low t, consistent with the $1/t^2$ amplification factor, while x- and v-prediction remain bounded over the evaluated timesteps. Results below $t = 0.05$ also reflect denominator clamping.

6 Conclusion

We have presented a joint 3×3 analysis of prediction and loss spaces in denoising generative models. For a fixed prediction target, changing the loss space is equivalent to timestep reweighting, which can affect generation quality under finite model capacity (Propositions 1 and 2). For prediction-space choices, we show that a rank-deficient linear bottleneck introduces an irreducible error floor of $D - m$ for ϵ- and v-prediction, whereas x-prediction can exploit low-dimensional data structure (Theorem 4 and Corollary 5). Combining this bound with prediction-dependent error conversion yields necessary conditions on bottleneck capacity that are more restrictive for ϵ-prediction than for v-prediction (Proposition 6 and Corollary 7).

Experiments with U-Net and JiT across CIFAR-10, Imagenette, and ImageNet-256 support these analyses. Under compression, x-prediction maintains low denoising error, while ϵ- and v-prediction are more sensitive to bottleneck capacity. In particular, the observed ϵ-prediction error closely matches the theoretical lower bound under strong compression, and bottleneck-width sweeps reveal distinct recovery patterns across prediction targets. Together, our findings clarify how prediction and loss choices interact with architectural information constraints and provide guidance for selecting prediction targets in compressed generative models.

Limitations. Our theoretical bounds rely on a linear bottleneck and a low-dimensional subspace model of the data. Our GeLU ablation retains the initial linear bottleneck and therefore does not establish corresponding guarantees for architectures with general nonlinear compression maps. The loss-ordering analysis is heuristic and depends on assumptions about sampling sensitivity and the allocation of model capacity across timesteps.

Our experiments span different architectures, conditioning schemes, and datasets, so absolute FID values are not directly comparable across settings. Moreover, the compressed ImageNet-256 configurations were trained for a shorter budget than the uncompressed controls, and we did not evaluate their convergence under extended training. Without compression, our error bounds become trivial. Although prediction-dependent conversion can amplify errors during sampling, particularly for ϵ-prediction, it does not fully explain the observed FID differences. Our stochastic sampling ablation is also limited to samplers that share the same velocity conversion. Extending the analysis to latent diffusion, text conditioning, and architectures with explicit skip connections or EDM-style preconditioning remains an important direction for future work.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00337955); by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (RS-2024-00457882, Artificial Intelligence Research Hub Project; RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH)); and by IITP grants funded by the Korea government (MSIT) under the Leading Generative AI Human Resources Development Program (IITP-2026-RS-2026-25546560) and the AI Star Fellowship Support Program (IITP-2026-RS-2026-25616313).

References

Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

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*, pages 11472–11481, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12606–12633, 2024.

Anne Gagneux, Ségolène Martin, Rémi Gribonval, and Mathurin Massias. Training flow matching: The role of weighting and parameterization. In *ICLR 2026 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026. arXiv:2603.06454.

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*, pages 7441–7451, 2023.

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.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232, 2023.

Qing Jin and Chaoyang Wang. Revisiting diffusion model predictions through dimensionality. In *Advances in Neural Information Processing Systems*, 2026.

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.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Dibyanshu Kumar, Philipp Väth, and Magda Gregorová. Loss functions in diffusion models: A comparative study. In *Machine Learning and Knowledge Discovery in Databases. Research Track (ECML PKDD 2025)*, Lecture Notes in Computer Science, pages 190–205. Springer, 2025. doi: 10.1007/978-3-032-05981-9_12.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 36115–36125, 2026.

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*, 2023.

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

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

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

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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.

A Proofs

A.1 Proof of Proposition 1

Proposition 1 (Equivalence of loss spaces as timestep reweighting)**.** *Assume the linear interpolation and velocity definition*

$$z_t = tx + (1-t)\epsilon, \qquad v = x - \epsilon, \qquad t \in (0,1),$$

with (x, ϵ) *sampled from the data and noise distributions. Fix a* prediction space $u \in \{x, \epsilon, v\}$*, meaning the network directly outputs*

$$u_\theta(z_t, t) = \mathrm{net}_\theta(z_t, t).$$

For any loss space $s \in \{x, \epsilon, v\}$*, define* $s_\theta(z_t, t)$ *as the quantity obtained by solving the linear system induced by* (z_t, t) *together with the constraints defining the pair* (x, ϵ, v)*, with* u *replaced by* u_θ*.*

Then for every pair $(s, u) \in \{x, \epsilon, v\}^2$ *there exist:*

- *a scalar slope* $a_{s|u}(t) \neq 0$ *that depends* only *on* t*, and*
- *an intercept term* $b_{s|u}(z_t, t)$ *that depends on* (z_t, t) *but is* independent of u*,*

such that the true *target and the* model *prediction satisfy the aligned affine relations*

$$s \;=\; a_{s|u}(t)\, u + b_{s|u}(z_t, t),$$

and

$$s_\theta(z_t, t) \;=\; a_{s|u}(t)\, u_\theta(z_t, t) + b_{s|u}(z_t, t).$$

Consequently, the residuals obey the pointwise scaling identity

$$s_\theta(z_t, t) - s \;=\; a_{s|u}(t)\,\big(u_\theta(z_t, t) - u\big), \tag{4}$$

and hence the squared loss in s*-space is* exactly *a weighted squared loss in the direct output* u*-space:*

$$\begin{aligned}\mathcal{L}_{s|u}(\theta) \;&:=\; \mathbb{E}\big[\|s_\theta(z_t, t) - s\|^2\big] \\ &=\; \mathbb{E}\Big[w_{s|u}(t)\,\|u_\theta(z_t, t) - u\|^2\Big],\end{aligned} \tag{5}$$

where $w_{s|u}(t) := a_{s|u}(t)^2 > 0$*. Since* $w_{s|u}$ *does not depend on* θ*, the Bayes-optimal predictor* $u^\star(z_t, t) = \mathbb{E}[u \mid z_t, t]$ *is the same for all loss spaces* s*. For the linear schedule, the weights are listed in Table 1; the individual identities appear in prior work [Salimans and Ho, 2022, Kingma and Gao, 2023, Karras et al., 2022], and we collect them here for reference.*

Proof. The result follows from the affine conversion formulas among (x, ϵ, v) given (z_t, t). From $z_t = tx + (1-t)\epsilon$ and $v = x - \epsilon$, any two of (x, ϵ, v) determine the third:

$$\begin{aligned}x &= \frac{z_t - (1-t)\epsilon}{t} = z_t + (1-t)v, \\ \epsilon &= \frac{z_t - tx}{1-t} = z_t - tv, \\ v &= x - \epsilon = \frac{x - z_t}{1-t} = \frac{z_t - \epsilon}{t}.\end{aligned}$$

For any prediction space u and loss space s, substituting u_θ for u in these formulas yields s_θ, and the *same* substitution of the true u yields the true s. Explicitly, the slopes $a_{s|u}(t)$ and intercepts $b_{s|u}(z_t, t)$ are:

Pred $u{=}x$: From the linear interpolation, we have

$$\epsilon = -\frac{t}{1-t}x + \frac{z_t}{1-t}, \qquad v = \frac{x - z_t}{1-t}.$$

Therefore,

$$a_{\epsilon|x}(t) = -\frac{t}{1-t}, \qquad b_{\epsilon|x}(z_t, t) = \frac{z_t}{1-t},$$
$$a_{v|x}(t) = \frac{1}{1-t}, \qquad b_{v|x}(z_t, t) = -\frac{z_t}{1-t}.$$

Pred u=ϵ: $x = (z_t - (1-t)\epsilon)/t$, so $a_{x|\epsilon} = -(1-t)/t$. $v = x - \epsilon = (z_t - \epsilon)/t$, so $a_{v|\epsilon} = -1/t$.

Pred u=v: $x = z_t + (1-t)v$, so $a_{x|v} = 1-t$. $\epsilon = z_t - tv$, so $a_{\epsilon|v} = -t$.

In all cases, $a_{s|u}(t)$ depends only on t, and the intercept $b_{s|u}(z_t, t)$ is independent of u_θ (it involves only z_t and t). Since $s_\theta - s = a_{s|u}(t)(u_\theta - u)$ by linearity, squaring and taking expectations gives $\mathcal{L}_{s|u} = \mathbb{E}[a_{s|u}(t)^2 \|u_\theta - u\|^2] = \mathbb{E}[w_{s|u}(t)\|u_\theta - u\|^2]$, where $w_{s|u}(t) = a_{s|u}(t)^2$. The diagonal entries are $w_{u|u} = 1$, and the off-diagonal entries are:

$$w_{\epsilon|x} = \left(\tfrac{t}{1-t}\right)^2, \quad w_{v|x} = \left(\tfrac{1}{1-t}\right)^2, \quad w_{x|\epsilon} = \left(\tfrac{1-t}{t}\right)^2,$$
$$w_{v|\epsilon} = \left(\tfrac{1}{t}\right)^2, \quad w_{x|v} = (1-t)^2, \quad w_{\epsilon|v} = t^2,$$

which are exactly the entries in Table 1. Since $w_{s|u}(t) > 0$ depends only on t, the population minimizer over all measurable predictors is $u^\star(z_t, t) = \mathbb{E}[u \mid z_t, t]$ for every loss space s. □

A.2 Proof of Theorem 4

Theorem 4 (Irreducible error from null-space components)**.** *Assume (A3) and let Π_{null} denote the orthogonal projector onto $\text{null}(E)$. For any measurable estimator $\hat{u}(\rho, t)$ of a target $u \in \mathbb{R}^D$,*

$$\begin{aligned} \mathbb{E}\big[\, \|\hat{u}(\rho,t) - u\|^2 \,\big] &\geq \mathbb{E}\big[\, \|\mathbb{E}[u \mid \rho, t] - u\|^2 \,\big] \\ &\geq \mathbb{E}\big[\, \|\Pi_{\text{null}} u - \mathbb{E}[\Pi_{\text{null}} u \mid \rho, t]\|^2 \,\big]. \end{aligned} \tag{10}$$

If in addition (A2) holds, then $\Pi_{\text{null}}\epsilon$ is independent of (ρ, t), and for $u = \epsilon$

$$\mathbb{E}\big[\, \|\hat{\epsilon}(\rho,t) - \epsilon\|^2 \,\big] \geq \mathbb{E}\big[\, \|\Pi_{\text{null}}\epsilon\|^2 \,\big] = (D - m).$$

Proof. Let $\hat{u}(\rho, t)$ be *any* estimator that depends only on the inputs (ρ, t). We define the ideal Bayes estimator $u^\star = \mathbb{E}[u \mid \rho, t]$. We can decompose the error of $\hat{u}$ as:

$$\hat{u} - u \;=\; (\hat{u} - u^\star) + (u^\star - u).$$

Squaring the norm and taking expectations:

$$\mathbb{E}\|\hat{u} - u\|^2 = \mathbb{E}\|\hat{u} - u^\star\|^2 + \mathbb{E}\|u^\star - u\|^2 + 2\mathbb{E}\big[(\hat{u} - u^\star)^\top (u^\star - u)\big].$$

The cross-term vanishes by the tower property of conditional expectation: since $(\hat{u} - u^\star)$ is measurable with respect to (ρ, t), it can be pulled out of the inner expectation:

$$\mathbb{E}\big[(\hat{u} - u^\star)^\top (u^\star - u)\big] = \mathbb{E}_{\rho,t}\Big[(\hat{u} - u^\star)^\top \underbrace{\mathbb{E}[u^\star - u \mid \rho, t]}_{=0}\Big] = 0.$$

Thus, we obtain the standard decomposition:

$$\mathbb{E}\|\hat{u} - u\|^2 \;=\; \underbrace{\mathbb{E}\|\hat{u} - u^\star\|^2}_{\geq 0} + \mathbb{E}\|u^\star - u\|^2 \;\geq\; \mathbb{E}\|u^\star - u\|^2.$$

This proves the first inequality.

The term $\mathbb{E}\|u^\star - u\|^2$ represents the total variance of the unrecoverable residual. For any vector z, and any orthogonal projector Π, we have $\|z\|^2 \geq \|\Pi z\|^2$. Applying this to the residual vector $z = u - u^\star$ with the null-space projector Π_{null}:

$$\|u - u^\star\|^2 \;\geq\; \|\Pi_{\text{null}}(u - u^\star)\|^2.$$

Since Π_{null} is a deterministic linear operator, it commutes with the conditional expectation by linearity of expectation:

$$\Pi_{\text{null}}(u - u^\star) = \Pi_{\text{null}}u - \Pi_{\text{null}}\mathbb{E}[u \mid \rho, t] = \Pi_{\text{null}}u - \mathbb{E}[\Pi_{\text{null}}u \mid \rho, t].$$

Taking expectations proves the second inequality:

$$\mathbb{E}\|u^\star - u\|^2 \;\geq\; \mathbb{E}\|\Pi_{\text{null}}u - \mathbb{E}[\Pi_{\text{null}}u \mid \rho, t]\|^2.$$

Consider the case $u = \epsilon$ with isotropic noise $\epsilon \sim \mathcal{N}(0, \mathrm{I})$. Write $\epsilon = \Pi_{\text{col}}\epsilon + \Pi_{\text{null}}\epsilon$, where $\Pi_{\text{col}} = E^\dagger E$ projects onto the column space of E. Since ϵ is isotropic Gaussian and $\Pi_{\text{col}}, \Pi_{\text{null}}$ are orthogonal projectors onto complementary subspaces, the components $\Pi_{\text{col}}\epsilon$ and $\Pi_{\text{null}}\epsilon$ are independent Gaussian vectors (a standard property of orthogonal projections of Gaussian vectors). Since $\rho = Ez_t = E(tx + (1{-}t)\epsilon)$ depends on ϵ only through $E\epsilon = E\Pi_{\text{col}}\epsilon$ (as $E\Pi_{\text{null}} = 0$ by definition), we have $\Pi_{\text{null}}\epsilon \perp\!\!\!\perp (\rho, t)$. Therefore:

$$\mathbb{E}[\Pi_{\text{null}}\epsilon \mid \rho, t] = \mathbb{E}[\Pi_{\text{null}}\epsilon] = 0.$$

Substituting this into our bound:

$$\mathbb{E}\|\hat{\epsilon} - \epsilon\|^2 \;\geq\; \mathbb{E}\|\Pi_{\text{null}}\epsilon - 0\|^2 \;=\; \mathbb{E}\|\Pi_{\text{null}}\epsilon\|^2.$$

For isotropic Gaussian noise, the expected squared norm of its projection is simply the dimension of the subspace:

$$\mathbb{E}\|\Pi_{\text{null}}\epsilon\|^2 = \mathrm{tr}(\mathrm{Cov}(\Pi_{\text{null}}\epsilon)) = \mathrm{tr}(\Pi_{\text{null}}\Pi_{\text{null}}^\top) = \mathrm{rank}(\Pi_{\text{null}}).$$

Since $E \in \mathbb{R}^{m \times D}$ has rank m, the dimension of its null space is $D - m$. Thus, the error is at least $D - m$. □

A.3 Proof of Corollary 5

Proof. Under (A1), $x = P\hat{x}$ with $\hat{x} \in \mathbb{R}^d$. By (A4), $EP \in \mathbb{R}^{m \times d}$ has full column rank, so the bottleneck mapping is injective on the data subspace. Thus, unlike the noise components in the null space of E, the data-subspace components are not irreversibly lost through the linear projection.

For ϵ-prediction, Theorem 4 directly gives

$$\mathbb{E}\|\hat{\epsilon} - \epsilon\|^2 \geq D - m.$$

For $v = x - \epsilon$, we have $\Pi_{\text{null}}v = \Pi_{\text{null}}x - \Pi_{\text{null}}\epsilon$. By (A2), $\Pi_{\text{null}}\epsilon$ is independent of (x, ρ, t) and has zero mean. Therefore,

$$\mathbb{E}[\Pi_{\text{null}}v \mid \rho, t] = \mathbb{E}[\Pi_{\text{null}}x \mid \rho, t].$$

Applying Theorem 4 and using this independence, we obtain

$$\begin{aligned}\mathbb{E}\|\hat{v} - v\|^2 &\geq \mathbb{E}\left\|\Pi_{\text{null}}v - \mathbb{E}[\Pi_{\text{null}}v \mid \rho, t]\right\|^2 \\ &= \mathbb{E}\left\|\Pi_{\text{null}}x - \mathbb{E}[\Pi_{\text{null}}x \mid \rho, t]\right\|^2 + \mathbb{E}\|\Pi_{\text{null}}\epsilon\|^2 \\ &\geq D - m.\end{aligned}$$

□

A.4 Proof of Proposition 6

The conversions from prediction space to x-space are: $\hat{x}_x = \hat{x}$, so $\hat{x}_x - x = \delta_x$; $\hat{x}_v = z_t + (1{-}t)\hat{v}$, so $\hat{x}_v - x = (1{-}t)\delta_v$; $\hat{x}_\epsilon = (z_t - (1{-}t)\hat{\epsilon})/t$, so $\hat{x}_\epsilon - x = -(1{-}t)\delta_\epsilon/t$. Taking squared norms yields (11) through (13).

A.5 Proof of Corollary 7

Fix $t \in (0, 1)$. Under (A2) and (A3), $\Pi_{\text{null}}\epsilon$ is independent of (ρ, t). Hence, the arguments of Theorem 4 and Corollary 5 apply conditionally on t, giving

$$\mathbb{E}[\|\hat{\epsilon} - \epsilon\|^2 \mid t] \geq D - m, \qquad \mathbb{E}[\|\hat{v} - v\|^2 \mid t] \geq D - m.$$

Table 5: Comparison of validation x-space MSE across prediction and loss choices on Imagenette. For the U-Net without an information bottleneck, MSE varies primarily with the loss space. For JiT-B with a bottleneck ($r = 0.67$), MSE varies primarily with the prediction space. The full results, including FID, are reported in Tables 2 and 3.

	U-Net (no bottleneck)			**JiT-B** (r=0.67, bottleneck)		
Pred\Loss	x	ϵ	v	x	ϵ	v
x	0.0481	0.0466	**0.0463**	0.050	**0.048**	0.048
ϵ	0.0486	0.0471	0.0467	**5.9**	**5.8**	**5.9**
v	0.0476	0.0465	0.0463	0.152	0.152	0.152

By Proposition 6, converting these prediction errors to x-space multiplies them by $(1-t)^2/t^2$ and $(1-t)^2$, respectively. Taking expectations over t yields the stated lower bounds.

For the numerical values, let $Z = \mathrm{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu = -0.5$ and $\sigma = 1$. Since $(1-t)/t = e^{-Z}$,

$$\mathbb{E}_t\left[\frac{(1-t)^2}{t^2}\right] = \mathbb{E}[e^{-2Z}] = e^{-2\mu+2\sigma^2} = e^3 \approx 20.1,$$

while numerical quadrature gives

$$\mathbb{E}_t[(1-t)^2] \approx 0.403.$$

The ratio of these conversion weights is approximately 50.

On the discrete evaluation grid $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ used in Tables 8 and 9, the corresponding averages are 17.53 and 0.33, respectively. The tabulated lower bounds additionally divide by D to match the reported per-dimension MSE.

B Additional experiments and details

B.1 Bottleneck lower bound: synthetic validation

We validate the irreducible error lower bound (Theorem 4) using the synthetic manifold setup with a controlled linear bottleneck. Fix ambient dimension D and intrinsic dimension $d \ll D$. We sample a column-orthonormal matrix $P \in \mathbb{R}^{D\times d}$, draw $\hat{x} \sim p_{\hat{x}}$ from a mixture of Gaussians on $\mathbb{R}^d$, and embed as $x = P\hat{x}$. With $\epsilon \sim \mathcal{N}(0, I_D)$ and $z_t = tx + (1-t)\epsilon$, the model accesses only $\rho = Ez_t \in \mathbb{R}^m$ through a trainable embedding $E \in \mathbb{R}^{m\times D}$. We train all nine (s, u) combinations and sweep m from 1 to D, decomposing the MSE into null-space (Π_{null}) and range-space (Π_{range}) components. We report prediction MSE in each target's native space, together with its null-space and range-space components and convergence diagnostics. As predicted in Figure 1, ϵ- and v-prediction exhibit an irreducible error floor, while x-prediction achieves low error for all $m \geq d$. For ϵ, the null-space MSE matches $D-m$ precisely, confirming the theorem; for v, it lies on or above this floor, as Corollary 5 requires.

Sweep over D, d, m, and the embedding E. We extend this setup along all three dimensions of Theorem 4 and Corollary 5: $D \in \{16, 64, 128\}$, $d \in \{2, 8\}$, and m from below d up to D, for all three prediction types. Each configuration is trained with v-loss for 2,000 epochs, once with a learned E and once with a random E frozen at initialization, over two seeds, for 444 runs in total. We report squared error per vector with $t \sim \mathcal{U}(0, 1)$, averaged over seeds; Table 6 lists every configuration. Three findings stand out. First, the floor of Theorem 4 is exact in all three variables. Across the 62 configurations with $m < D$, the null-space error of ϵ-prediction lies between 0.997 and 1.012 times $D-m$. v-prediction lies above the same floor in every configuration, as Corollary 5 predicts. Its proof adds the posterior error of $\Pi_{\mathrm{null}}x$ to the ϵ floor, and with a learned E and $m \geq d$ this excess stays between 12.8 and 18.7 for every D and m. Second, x-prediction depends on whether $m \geq d$ rather than on D or m. With a learned E, its error stays between 11.4 and 12.5 whenever $m \geq d$ and rises to between 15.9 and 17.1 when $m < d$, so the transition sits at the injectivity condition (A4). Third, a random E separates the two mechanisms. The ϵ floor is the same for random and learned E to within 1.5%, so it depends only on $\mathrm{rank}(E)$ and is information-theoretic rather than an artifact of training the embedding. For x-prediction, a random E is much worse near $m = d$, for example

Table 6: Synthetic sweep of Theorem 4 and Corollary 5 over ambient dimension D, intrinsic dimension d, and bottleneck rank m, with a learned or a frozen random embedding E (BottleneckMLP, v-loss, 2 seeds). The ϵ and v columns give the null-space error of ϵ- and v-prediction divided by the floor $D-m$; the x columns give the total error of x-prediction. Errors are squared norms per vector with $t \sim \mathcal{U}(0,1)$. Rows with $m{=}D$ have no floor.

			ϵ null / $(D-m)$		v null / $(D-m)$	x error	
D	d	m	learned	random	learned	learned	random
16	2	1	1.001	1.001	1.96	17.0	23.5
16	2	2	1.002	1.003	2.04	11.5	22.7
16	2	4	1.003	1.003	2.19	11.4	16.0
16	2	8	1.006	1.005	2.80	11.6	13.0
16	2	12	0.997	1.012	4.64	11.6	12.2
16	2	16	—	—	—	11.7	11.6
16	8	4	1.000	1.003	2.22	16.0	18.2
16	8	8	1.007	1.001	2.80	12.2	16.1
16	8	12	1.006	1.000	4.21	11.6	12.8
16	8	16	—	—	—	11.5	11.5
64	2	1	1.002	1.003	1.23	17.1	21.9
64	2	2	1.003	1.003	1.24	11.7	18.8
64	2	4	1.002	1.003	1.25	11.7	16.7
64	2	16	1.004	1.004	1.30	11.6	14.8
64	2	32	1.006	1.005	1.46	11.8	13.0
64	2	48	1.007	1.006	1.90	11.8	12.2
64	2	64	—	—	—	11.9	11.3
64	8	4	1.003	1.003	1.25	15.9	18.8
64	8	8	1.004	1.003	1.27	12.2	17.5
64	8	16	1.004	1.005	1.37	11.7	15.6
64	8	32	1.007	1.007	1.58	11.6	13.5
64	8	48	1.007	1.006	2.13	11.6	12.3
64	8	64	—	—	—	11.8	11.4
128	2	1	1.004	1.004	1.12	16.8	24.6
128	2	2	1.004	1.004	1.12	11.4	24.9
128	2	4	1.004	1.004	1.12	11.6	22.9
128	2	32	1.006	1.006	1.15	11.6	14.0
128	2	64	1.007	1.008	1.23	11.7	12.7
128	2	96	1.012	1.010	1.43	11.6	12.0
128	2	128	—	—	—	11.5	11.6
128	8	4	1.004	1.004	1.13	15.9	19.6
128	8	8	1.005	1.004	1.13	12.5	18.1
128	8	16	1.005	1.005	1.17	11.7	17.2
128	8	32	1.005	1.006	1.19	11.7	15.0
128	8	64	1.006	1.008	1.28	11.6	13.2
128	8	96	1.009	1.011	1.56	11.7	12.4
128	8	128	—	—	—	11.6	11.7

24.9 against 11.4 at $D{=}128$ and $m{=}d{=}2$, and approaches the learned value as $m \to D$. A learned E thus finds the subspace-aligned solution already at the minimal rank $m = d$, consistent with the discussion after Corollary 5 that end-to-end training pushes E to preserve the data subspace.

B.2 Bottleneck threshold sweep

BottleneckViT and BottleneckMLP. BottleneckViT is a simplified Vision Transformer used as a computationally cheaper proxy for JiT-B in bottleneck-width sweeps.

Its patch embedding follows JiT's `BottleneckPatchEmbed`, but inserts a GELU between the two convolutions. Importantly, the first convolution is a linear mapping from the noisy patch to a d_{bn}-dimensional representation. All subsequent computations, including the GELU, depend on the input only through this representation. Thus, under compression ($d_{\mathrm{bn}} < d_{\mathrm{patch}}$), the architecture satisfies assumption (A3) of Theorem 4, provided the first mapping has full row rank.

BottleneckViT uses fewer transformer blocks and a smaller hidden width than JiT-B, giving approximately 22M parameters versus JiT-B's 130M. This enables systematic bottleneck-width sweeps at substantially lower computational cost.

BottleneckMLP is the analogous synthetic-data variant, using an MLP backbone with a linear input bottleneck.

Tables 7 and 8 show the full bottleneck-ratio sweep on BottleneckViT.

Table 7: Bottleneck dimension sweep on CIFAR-10 (BottleneckViT, $d_{\text{patch}} = 48$).

d_{bn}	Ratio r	x-pred MSE	v-pred MSE	Gap
32	0.67	0.054	0.158	$+191\%$
64	1.33	0.054	0.054	$+0.2\%$
128	2.67	0.054	0.054	$+0.0\%$
256	5.33	0.054	0.054	$+0.0\%$

Table 8: ϵ-prediction recovery with increasing bottleneck width on Imagenette ($d_{\text{patch}} = 192$). Results use BottleneckViT with ϵ-loss, 500 training epochs, and seed 42. Validation x-space MSE is averaged over $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The floor is $17.53(D-m)_+/D$, obtained from Corollary 7 on this evaluation grid.

d_{bn}	Ratio r	x-pred MSE	ϵ-pred MSE	Gap	Floor
32	0.17	0.049	14.61	$297\times$	14.61
256	1.33	0.048	0.095	$+98\%$	—
384	2.00	0.048	0.056	$+17\%$	—
512	2.67	0.048	0.053	$+9\%$	—
768	4.00	0.048	0.050	$+2.6\%$	—

Patch size at fixed embedding width. Theorem 4 predicts that the floor depends on the patch through $D-m = p^2C - d_{\text{bn}}$. Increasing the patch size p at fixed d_{bn} should therefore degrade ϵ- and v-prediction monotonically, while leaving x-prediction unaffected as long as d_{bn} exceeds the intrinsic dimension of a patch. We test this with BottleneckViT on CIFAR-10 at $p \in \{2, 4, 8\}$ and fixed $d_{\text{bn}} = 32$, so that $D = 3p^2$ ranges from 12 to 192 under the same dataset and training recipe: v-loss, 200 epochs, seed 42. Table 9 reports validation MSE averaged over $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. At p=2 the embedding is wider than the patch, so there is no bottleneck and the three prediction types match. As p grows, x-prediction stays flat at 0.054 to 0.056, while ϵ- and v-prediction degrade monotonically. The ϵ-space error of ϵ-prediction lands on the floor at both bottlenecked sizes. Averaged over t it is 0.353 against 0.333 and 0.842 against 0.833, and for $t \leq 0.5$ it matches the floor to within 1%. Through Proposition 6, the floor then predicts the x-space error of ϵ-prediction, as in Corollary 7. The mean amplification $(1-t)^2/t^2$ over the evaluation grid is 17.53, so the prediction $17.53\,(D-m)/D$ is 5.84 at p=4 and 14.61 at p=8, against measured values of 5.85 and 14.62. v-prediction lies above its corresponding bound $0.33\,(D-m)/D$, which is 0.11 and 0.28, because its floor also contains the x component. The same values recur across setups. At r=0.67 the ϵ-prediction error of 5.85 equals that of JiT-B/8 on Imagenette (Table 3), and at r=0.17 the value 14.62 matches JiT-B/16 on ImageNet-256 (Table 4). Since the error sits on the floor and the amplification depends only on the evaluation grid, the bottleneck ratio governs this error, not the resolution or the architecture.

B.3 Compression vs. post-bottleneck nonlinearity

We examine whether introducing nonlinear processing after the initial linear bottleneck affects the prediction-space asymmetry. Specifically, we insert a GeLU between JiT's two patch-embedding convolutions. In both variants, the first convolution compresses each noisy patch into a d_{bn}-dimensional representation, and all subsequent computations depend only on this representation. Thus, the linear bottleneck assumption of Theorem 4 remains satisfied under compression.

As shown in Table 10, ϵ-prediction exhibits similarly large errors under compression in both variants, while its error decreases substantially when compression is removed. These results demonstrate that

Table 9: Patch-size sweep on CIFAR-10 (BottleneckViT, d_{bn}=32, v-loss, 200 epochs, seed 42). Validation MSE per dimension, averaged over $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The first three error columns are in x-space; the last is the ϵ-space error of ϵ-prediction, to be compared with the floor $(D-m)/D$.

				x-space error			ϵ-space error
p	D=$3p^2$	r	$(D-m)/D$	x-pred	ϵ-pred	v-pred	ϵ-pred
2	12	2.67	—	0.054	0.055	0.054	0.060
4	48	0.67	0.333	0.054	5.85	0.158	0.353
8	192	0.17	0.833	0.056	14.62	0.321	0.842

the prediction-space asymmetry persists despite the additional nonlinearity after the initial linear compression.

Table 10: Effect of post-bottleneck nonlinearity on Imagenette. A 2×2 factorial design compares JiT's original linear patch embedding with a variant inserting a GeLU between its two convolutions, under compressed and uncompressed configurations. Cells show test MSE; numbers in parentheses give the ϵ-prediction gap relative to x-prediction.

	d_{bn}=128 (compression)		d_{bn}=256 (no compr.)	
	x-pred	ϵ-pred	x-pred	ϵ-pred
Linear	0.048	5.85 (120×)	0.048	0.051 (6%)
GeLU	0.049	5.9 (120×)	0.049	0.059 (21%)

B.4 Generated sample comparison

To complement the quantitative MSE and FID gaps reported in Table 3, we visualize the qualitative effect of prediction-space choice under a fixed bottleneck. Figure 3 shows samples generated by JiT-B/8 with d_{bn}=128 at ratio 0.67 on Imagenette using classifier-free guidance, with 100 Heun steps. All three rows use the same architecture and training recipe. x- and ϵ-prediction are trained with ϵ-loss and v-prediction with v-loss; under this bottleneck the loss choice changes MSE by at most 3.3% within a prediction type (Table 3), so the rows differ mainly in the prediction target. The visual ordering tracks the MSE ordering precisely: x-prediction yields recognizable class-conditional images, v-prediction yields heavily degraded but still structured samples, and ϵ-prediction collapses to pure noise. This corroborates Theorem 4: the irreducible null-space error of $D - m$ for ϵ propagates through the ODE sampler to produce samples with no learnable structure, while x-prediction, free of the null-space penalty under the manifold assumption, recovers usable samples even at the same bottleneck width.

B.5 JiT-B/4 convergence behavior

Figure 4 reports per-epoch validation MSE for JiT-B/4 on CIFAR-10 at bottleneck ratio r=2.67, trained for 500 epochs with ϵ-loss, alongside an x-prediction baseline trained under the same setup. Panel (a) plots the absolute loss curves; panel (b) shows their ratio. This single training run is the evidence for the convergence-penalty framing in Section 5: it shows that, on this configuration, the prediction-type gap closes with continued training rather than persisting as an information-theoretic floor. We have not measured this systematically across r, datasets, or architectures. The ImageNet-256 study (Section B.8) adds one more point at the regime boundary: at r=1.0 and 600 epochs, the ϵ-prediction denoising gap narrows to a factor of two without fully closing, and the FID gap from conversion amplification persists regardless.

B.6 Numerical evaluation of the loss-ordering integral

We evaluate the integral $J_\kappa(w_{\text{eff}})$ of Proposition 2 for the three effective weights under the logit-normal time distributions used in training. A logit-normal density vanishes at both endpoints, so J_κ diverges on $(0, 1)$. We therefore integrate over $t \in [0.05, 0.95]$. This cutoff is a modeling choice:

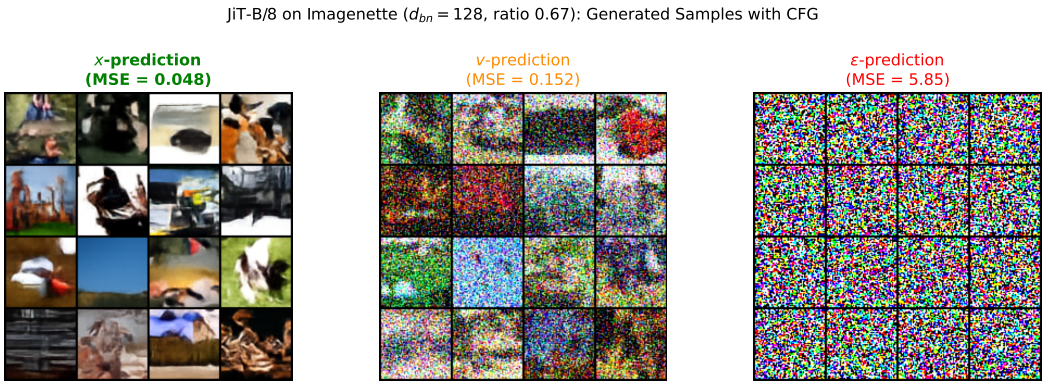


Figure 3: Generated samples from JiT-B/8 (d_{bn}=128, ratio 0.67) on Imagenette with classifier-free guidance. x-prediction produces recognizable images; v-prediction is heavily degraded; ϵ-prediction produces pure noise. All three use the same 130M-parameter architecture and 100 Heun steps; x- and ϵ-prediction use ϵ-loss and v-prediction uses v-loss.

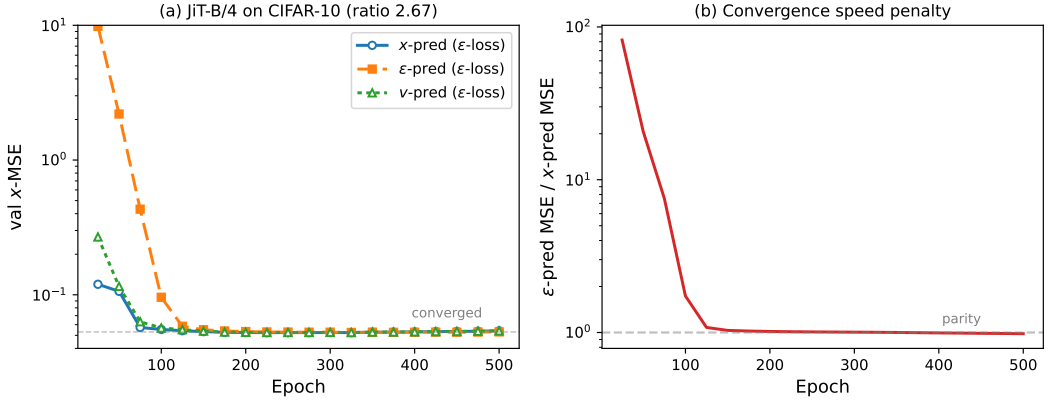


Figure 4: Loss curves on JiT-B/4 CIFAR-10 (r=2.67, ϵ-loss). **(a)** ϵ-prediction starts roughly 80-fold worse than x-prediction but reaches parity by epoch 200. **(b)** The MSE ratio decays from 80-fold to roughly unity, suggesting that the bottleneck effect at $r \geq 1$ is a convergence penalty rather than an irreducible error.

it equals the conversion clamp $t_\epsilon = 0.05$ of the JiT recipe, but the training time distribution itself covers $(0, 1)$. Table 11 normalizes each integral by its v-loss value. With constant κ, which ignores assumption (iv), the x-loss integral is 126 to 256 times the v-loss integral, while the ϵ-loss integral is 1.5 to 11.4 times. The excess of ϵ-loss comes from low t, where its weight t^2 vanishes. Restricting the sensitivity to high t, $\kappa(t) = \mathbf{1}[t > 1/2]$, brings the ϵ-loss integral within a factor of 1.43 of the v-loss integral and leaves the x-loss integral more than two orders of magnitude larger, which is the ordering (9). For integration cutoffs between 0.02 and 0.1, the x-loss integral remains the largest, by a factor of at least 28.

Table 11: The loss-ordering integral $J_\kappa(w_{\text{eff}})$ of Proposition 2 over $t \in [0.05, 0.95]$, normalized by its v-loss value. Effective weights are 1, t^2, and $(1-t)^2$ for v-, ϵ-, and x-loss.

Time distribution	Sensitivity $\kappa(t)$	v-loss	ϵ-loss	x-loss
logit-normal$(-0.5, 1.0)$, U-Net	1	1	11.4	125.9
	$\mathbf{1}[t > 1/2]$	1	1.43	148.5
logit-normal$(-0.8, 0.8)$, JiT	1	1	1.49	255.9
	$\mathbf{1}[t > 1/2]$	1	1.17	257.0

B.7 Loss ordering on simple distributions

The ordering (9) relies on assumption (iv) of Proposition 2, that generation quality depends disproportionately on high-t accuracy. This holds for natural images, where fine-grained details such as textures and edges are resolved near t=1 and dominate FID. However, for simple distributions such as 2D synthetic datasets, the denoising task is approximately equally difficult at all timesteps. In this regime, our synthetic experiments show that the loss ordering partially *reverses*: diagonal losses (u=s, uniform weight) become near-optimal, and ϵ-loss can underperform x-loss. This is consistent with the analysis: when denoising difficulty is uniform across t, concentrating capacity on any particular timestep range cannot help.

B.8 ImageNet-256 scale validation

To confirm that both regimes persist at the resolution and dataset scale of modern generative models, we train the full JiT-B/16 architecture on class-conditional ImageNet-1k at 256×256 (1.28M images, 1000 classes), where each patch has $d_{\text{patch}} = 16^2 \cdot 3 = 768$. We run the complete 3×3 design at d_{bn}=128 (ratio r=0.17, strong compression) for 100 epochs each, and a three-configuration control at d_{bn}=768 (ratio r=1.0, no compression) trained to 600 epochs. All runs use batch size 1024, AdamW (β_1=0.9, β_2=0.95, lr 10^{-4}, constant after a 5-epoch warmup), logit-normal time sampling (μ=−0.8, σ=0.8), label dropout 0.1, velocity clamp t_ϵ=0.05, and EMA decay 0.9999, matching the JiT recipe of the Imagenette experiments. FID is computed with clean-fid [Parmar et al., 2022] over 50K samples (Heun, 50 steps, CFG 2.9) against the full training set. Table 4 reports the results.

Both phenomena from the smaller-scale experiments reproduce. Under compression (r=0.17), the prediction-space dichotomy is stark: x-prediction attains val x-MSE 0.023 and FID 20–37 across all loss types, v-prediction degrades to MSE 0.29 and FID ≈200, and ϵ-prediction collapses to MSE 14.6 and FID ≈420. The ϵ-prediction error in ϵ-space sits at 0.834–0.837 across the three loss types, within 0.5% of the closed-form floor $(D-m)/D = 640/768 = 0.833$ of Theorem 4, a direct quantitative confirmation of the bottleneck lower bound at scale.

Removing the bottleneck (r=1.0) sharpens the central message of the paper. We train all three control configurations to the full 600-epoch budget so that the comparison is at convergence rather than confounded by training time. The val x-MSE spread collapses from 600-fold under compression to within a factor of two (0.022–0.043), confirming that information is no longer lost; yet the FID gap remains enormous (5.7 for x-pred, 140.6 for v-pred, 330.6 for ϵ-pred). This is the dual dichotomy: once the architecture can represent the target, denoising quality (MSE) nearly equalizes, while prediction type still dominates generation quality (FID). For ϵ-prediction, the per-timestep analysis below locates the gap at low t, where the $\epsilon \to v$ conversion of Proposition 6 is ill-conditioned. v-prediction needs no conversion for ODE sampling, so Proposition 6 does not explain its gap. Its velocity error stays within 30% of the x-model's at every t we measure and is lower at t=0.95 (Table 12), and we leave the cause of its FID gap open. Crucially, the gap is *not* an undertraining artifact: from 100 to 600 epochs the x-prediction FID falls sharply (24.8 → 5.7, reaching the range of published ImageNet-256 models [Li and He, 2026]), while ϵ-prediction barely moves (340.2 → 330.6) and v-prediction improves only modestly (173.5 → 140.6). At convergence, v-prediction remains ≈25× worse than x-prediction in FID despite a near-identical x-MSE (0.026 vs. 0.022), so the FID-vs-MSE decoupling is not a transient of finite training or an artifact of the bottleneck.

Figure 2 makes the mechanism explicit. Measuring per-timestep error on the converged r=1.0 checkpoints, ϵ-prediction's *velocity*-space error, the quantity the ODE integrates, grows sharply as $t \to 0$, reaching 0.44 at t=0.06 and 1.07 at t=0.02 versus ≈0.065 at mid-range. Above t_ϵ this is the low-t blow-up predicted by Proposition 6, and below t_ϵ the conversion clamp described next also contributes; x- and v-prediction stay bounded. Following the JiT recipe, every conversion clamps its denominator t or $1-t$ at $t_\epsilon = 0.05$. The sampler starts at t=0, so the first steps of ϵ-prediction, at $t = 0$, 0.02, and 0.04, fall in the clamped region, as do the last steps of x-prediction at $t = 0.96$ and 0.98. There the clamp caps the amplification at $1/t_\epsilon$, but even an exact network output then yields a velocity shrunk by the factor t/t_ϵ for ϵ-prediction, or $(1-t)/t_\epsilon$ for x-prediction. The ϵ-prediction FID gap therefore reflects the $\epsilon \to v$ conversion together with the clamp that regularizes it, and we do not separate the two contributions.

Comparing separately trained models. Different models may have different prediction errors before conversion. We therefore compare their predictions in velocity space, using the same target $v = x - \epsilon$ for all three models. This is also the quantity used by the ODE sampler.

Remark 3 (Comparing prediction errors in velocity space). For a model predicting $u \in \{x, \epsilon, v\}$, let $u^\star = \mathbb{E}[u \mid z_t, t]$ be the Bayes predictor and $e_u = \hat{u} - u^\star$ the error relative to it. The class label is implicit in the conditioning of the Bayes predictors. These predictors map exactly onto one another under the affine conversions in Proposition 1.

Let $\hat{v}_u$ denote the velocity estimate obtained from $\hat{u}$. At each fixed t where denominator clamping is inactive,

$$\mathbb{E}\big[\|\hat{v}_u - v\|^2 \mid t\big] = \underbrace{\mathbb{E}\big[\|v^\star - v\|^2 \mid t\big]}_{\text{common Bayes risk}} + w_{v|u}(t)\, \mathbb{E}\big[\|e_u\|^2 \mid t\big]\,,$$

where $w_{v|u}(t)$ is given in Table 1. The cross term vanishes because $\mathbb{E}[v - v^\star \mid z_t, t] = 0$ and $\hat{v}_u - v^\star$ is determined by the model's inputs.

All three prediction types share the same Bayes risk in velocity space. Differences in their expected velocity MSEs therefore equal differences in their excess MSEs. The converted excess MSE depends on both the conversion weight and the excess MSE before conversion. The same decomposition holds for MSE per dimension after normalizing all terms by the number of output coordinates.

Table 12 reports empirical estimates of these velocity-space MSEs. At $t = 0.06$, the ϵ-model's MSE is 0.0016 before conversion and 0.443 in velocity space, where the conversion weight is $w_{v|\epsilon}(t) = 1/t^2 \approx 278$. The x-model's velocity-space MSE is 0.106. The observed difference of 0.337 estimates the corresponding difference in excess MSE.

The ordering between these two models reverses at $t = 0.95$: the ϵ-model has a lower velocity-space MSE than the x-model (0.337 versus 0.462). Thus, the comparison depends on the learned prediction errors as well as the conversion weights. The table excludes timesteps where denominator clamping changes the conversion formulas.

Table 12: Per-timestep error of the three separately trained JiT-B/16 models on ImageNet-256 (r=1.0, 600 epochs, v-loss) in the common velocity space, together with the ϵ-model's native error $\|\delta_\epsilon\|^2 = t^2\|\delta v\|^2$. Mean squared error per dimension against sampled targets, which includes the Bayes risk, over 2,560 validation images.

	Velocity-space error			Native error
t	x-model	ϵ-model	v-model	ϵ-model
0.06	0.106	0.443	0.116	0.0016
0.10	0.081	0.198	0.093	0.0020
0.50	0.051	0.067	0.066	0.017
0.90	0.189	0.197	0.197	0.160
0.95	0.462	0.337	0.333	0.304

Stochastic sampling. Proposition 6 concerns the drift field, not the integrator. Consider the reverse samplers that share the marginals of the probability-flow ODE: for any schedule $w(t) \geq 0$,

$$dz = \big[\hat{v}(z,t) + w(t)\, \hat{s}(z,t)\big]\, dt + \sqrt{2w(t)}\, dW_t, \tag{16}$$

where $\hat{s}$ estimates the score $\nabla \log p_t$ and $w \equiv 0$ recovers the ODE. Both drift terms are affine conversions of the same network output. For an ϵ-prediction model, $\hat{v} = (z_t - \hat{\epsilon})/t$ and $\hat{s} = -\hat{\epsilon}/(1-t)$, so a native error δ_ϵ gives

$$\delta v = -\frac{\delta_\epsilon}{t}, \qquad \delta s = -\frac{\delta_\epsilon}{1-t}.$$

The low-t amplification therefore stays in the drift for every choice of w. Injected noise can change how a drift error affects the sample distribution, but it does not remove a systematic error in the field. We test this on the 600-epoch r=1.0 checkpoints with Euler–Maruyama sampling of (16), $w(t) = 1-t$, and 50 steps, against the Heun ODE with the same step count. Heun evaluates the network twice per step, so the ODE uses 99 network evaluations against 50 for the SDE. Both samplers use a matched 4K-sample FID protocol, which is upward-biased relative to the 50K

numbers in Table 4 but comparable across samplers. Table 13 shows that ϵ-prediction does not recover under stochastic sampling, while x-prediction changes only modestly. Scaling the schedule to $w(t) = 4(1-t)$ and $16(1-t)$ keeps the ϵ-prediction FID between 328 and 372. The FID gap thus originates in a drift-field error that the ODE and this SDE share. This conclusion is limited to samplers that use the same conversion. Samplers that avoid it, such as DDPM ancestral sampling in native ϵ-space, also avoid the amplification (Section 5).

Table 13: ODE versus SDE sampling on the 600-epoch ImageNet-256 checkpoints (r=1.0, v-loss). FID over 4K samples, 50 steps, CFG 2.9. The SDE is Euler–Maruyama on (16) with $w(t) = 1-t$.

FID (4K samples)	ODE (Heun)	SDE
x-prediction	14.5	17.1
ϵ-prediction	333.4	330.2

C Relation to EDM preconditioning

EDM [Karras et al., 2022] writes the denoiser as

$$D_\theta(\tilde{z};\sigma) = c_{\text{skip}}(\sigma)\,\tilde{z} + c_{\text{out}}(\sigma)\,F_\theta\big(c_{\text{in}}(\sigma)\,\tilde{z};\,\sigma\big),$$

$$c_{\text{skip}} = \frac{\sigma_{\text{data}}^2}{\sigma^2+\sigma_{\text{data}}^2}, \qquad c_{\text{out}} = \frac{\sigma\,\sigma_{\text{data}}}{\sqrt{\sigma^2+\sigma_{\text{data}}^2}},$$

where $\tilde{z} = x + \sigma n$ with $n \sim \mathcal{N}(0, \mathrm{I}_D)$. Our interpolant takes this form with $\tilde{z} = z_t/t$ and $\sigma = (1-t)/t$.

EDM lies outside our static 3×3 grid for two reasons. First, its network target changes continuously with the noise level. As $\sigma \to \infty$, $c_{\text{skip}} \to 0$ and F_θ targets the rescaled data x/σ_{data}. As $\sigma \to 0$, $c_{\text{skip}} \to 1$ and F_θ targets a residual close to $-n$, an ϵ-like quantity. Second, the denoiser includes the full-rank skip path $c_{\text{skip}}(\sigma)\,\tilde{z}$.

Our framework still applies to each ingredient. An error δ_F in the network output enters the x-space estimate as $c_{\text{out}}(\sigma)\,\delta_F$, so c_{out}^2 plays the role of the amplification factor in Proposition 6. In the same coordinates, that proposition gives the factor σ^2 for ϵ-prediction, which is unbounded at high noise, and 1 for x-prediction, which does not vanish at low noise. EDM's factor satisfies $c_{\text{out}}^2 \le \min(\sigma^2, \sigma_{\text{data}}^2)$ and is bounded at both ends. EDM's loss weight $\lambda(\sigma) = 1/c_{\text{out}}(\sigma)^2$ makes the effective weight on δ_F uniform over noise levels, which is a loss-space reweighting in the sense of Proposition 1. On the x-space error, $\lambda(\sigma) = \sigma^{-2}+\sigma_{\text{data}}^{-2}$, our ϵ-loss weight $w_{\epsilon|x} = t^2/(1-t)^2 = \sigma^{-2}$ plus a constant. The skip term bypasses any compressed representation, so the final denoiser is not a function of $\rho = Ez_t$ alone. Assumption (A3) then fails, and the $D-m$ floor of Theorem 4 does not apply.

We therefore expect EDM preconditioning to mitigate the bottleneck failure of ϵ- and v-prediction, but we have neither a formal result nor a controlled experiment that establishes this. A direct c_{skip} ablation on a bottlenecked ViT is an important next step.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the four contributions: a joint 3×3 analysis of prediction and loss spaces (Propositions 1 and 2), a quantitative bottleneck lower bound (Theorem 4), a conversion amplification mechanism (Proposition 6 and Corollary 7), and a dual dichotomy between MSE and FID. Each claim is supported by theory in Section 5 and experiments on CIFAR-10, Imagenette, and class-conditional ImageNet-256 across U-Net and JiT-B.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: An explicit `Limitations` paragraph in the conclusion discusses the U-Net vs. JiT-B asymmetry, the effect of dataset size on absolute FID, the restriction of Theorem 4 to linear bottlenecks (with empirical-only nonlinear extension in Remark 2), and the lack of evaluation on latent diffusion; an SDE-sampler ablation is provided in Section B.8.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The four formal results state all assumptions inline and are proved in Appendix B's preceding Appendix A: Proposition 1 (proof in A.1), Theorem 4 (proof in A.2), Corollary 5 (proof in A.3), Proposition 6 (proof in A.4), and Corollary 7 (proof in A.5). Proposition 2 is explicitly framed as a heuristic proposition with assumptions stated as numbered items (i) to (iv) and a derivation given inline.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The Setup paragraph in Section 5 specifies architectures (U-Net $\sim$97M and JiT-B $\sim$130M), optimizers and hyperparameters (Adam/AdamW with full β values and learning rates), time-sampling distributions (logit-normal with stated μ, σ), gradient clipping, EMA decay, training duration, classifier-free guidance dropout, and FID evaluation protocol (clean-fid with stated solver and step counts). Architectural specifics for JiT (RMSNorm, RoPE, QK-norm, SwiGLU, in-context class tokens) follow [Li and He, 2026]. Code and the per-run result files behind every table will be released with the camera-ready version.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: All datasets used (CIFAR-10, Imagenette, ImageNet-1k) are publicly available, and the architectures (U-Net, JiT-B [Li and He, 2026]) are described in detail with hyperparameters in Section 5 and Section B.8. Code and the per-run result files behind every table will be released with the camera-ready version. Trained checkpoints are not part of the release.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The Setup paragraph in Section 5 reports optimizer choice, full β parameters, learning rate, time-sampling distribution, gradient clipping, EMA decay, training duration, and FID evaluation protocol. Additional ablation-specific details (time-resampling sweep, BottleneckViT and BottleneckMLP architectures, bottleneck-ratio sweep configurations) appear in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The U-Net main results in Table 2 are averaged over 3 seeds with reported standard deviation ≤ 0.002 in x-MSE; JiT CIFAR-10 results are averaged over 2 seeds. Single-seed cells are noted explicitly (Imagenette JiT runs, the BottleneckViT bottleneck-ratio sweeps, and the ImageNet-256 runs in Section B.8, all seed 42). The paper's claims rest on relative orderings across $(\text{pred}, \text{loss})$ pairs, and the observed gaps (often 10- to 120-fold) substantially exceed seed-level noise.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: The CIFAR-10, Imagenette, and synthetic experiments run on modest GPUs (NVIDIA RTX 3090 and A5000). The ImageNet-256 experiments (Section B.8) run on a single 8×B200 node: each 100-epoch run takes roughly 6.7 hours on 8 GPUs, each 600-epoch run roughly 40 hours, and each 50K-sample FID evaluation roughly 15 minutes, for a total of approximately 1,500 B200 GPU-hours. The smaller-scale experiments were deliberately kept modest to permit exhaustive sweeps over the 3×3 prediction/loss grid and over bottleneck ratios, rather than to maximize absolute performance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research is foundational methodology on diffusion-model training that uses publicly available image datasets (CIFAR-10, Imagenette, and ImageNet-1k), involves no human subjects or private data, and conforms with the NeurIPS Code of Ethics in all respects.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This work studies optimization-level design choices (prediction space, loss space, and architectural bottlenecks) for diffusion generative models. The positive impact is improved understanding of training dynamics, leading to more compute-efficient and better-performing models, which benefits research and applications that already use diffusion models. The negative impact is that improvements in generative-model quality can be misused for synthetic media that misleads, such as deepfakes; this risk is shared with the broader diffusion-model literature and is not specific to our analysis.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper releases no new pretrained models or datasets that pose a high risk of misuse. We release code and result files only.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All existing assets are cited: CIFAR-10, Imagenette, and ImageNet-1k as benchmark datasets, JiT [Li and He, 2026] as the source architecture, the U-Net family of diffusion architectures via the standard literature, and clean-fid [Parmar et al., 2022] for FID evaluation. All assets are publicly released for research use under their respective open licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not release new datasets or pretrained models. The BottleneckViT and BottleneckMLP variants used for the bottleneck-ratio sweep are research artifacts described in Appendix B and will be available in the released code, but they are not standalone published assets.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or human-subjects research.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no human subjects and no IRB review is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology, theoretical analysis, or experimental design.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.