THE NOISE-TRANSMISSION BURDEN OF DIFFUSION NETWORK: WHY IMAGE PREDICTION CAN OUTPERFORM NOISE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Clean-image prediction (x-prediction) and noise prediction yield equivalent population-optimal denoisers, yet can perform very differently in finite diffusion networks. We relate this gap to a noise-transmission burden: forming a clean-image estimate from a noise predictor requires cancellation of corruption carried by the input. Under Gaussian corruption, a rank-$r < d$ linear encoder imposes a denoising-error floor $\sigma_t^2(d-r)$ for bare noise prediction even with unrestricted downstream computation, while image-prediction risk remains bounded. Analytic output preconditioning cancels this contribution and yields an image-prediction-equivalent learned branch; EDM suppresses it at high noise, with its learned target approaching a scaled clean image. Without input compression, and assuming exact affine recovery of the noisy patch, we characterize the minimum hidden width needed to approximate the population denoiser to a chosen accuracy. This width is determined by the patch dimension and the covariance spectrum of the denoiser's residual beyond its best affine approximation from that patch. Targeted JiT experiments on ImageNet examine these mechanisms. Learned output shortcuts substantially improve generation under the tested objective. Across two resolution/patch settings, widening improves generation quality and simultaneous affine access to noisy-input and clean-image information. Together, these results identify input compression and simultaneous input/denoiser accessibility as two concrete architectural mechanisms behind prediction-target differences.

1 INTRODUCTION

Why should predicting the clean image be easier than predicting the noise added to it? At the population optimum, image, noise, and path-velocity predictors are related by simple identities and yield the same denoiser. Yet their performance can differ substantially in finite networks. JiT demonstrates this gap in pixel-space diffusion Transformers: image prediction succeeds with coarse patches and narrow input embeddings in settings where noise or velocity prediction performs poorly (Li & He, 2025). Understanding this mismatch matters for choosing prediction targets and allocating model capacity: parameterizations that are equivalent in principle can place very different demands on a practical architecture.

JiT proposes a manifold-based explanation: image prediction can exploit the low-dimensional structure of clean images, whereas predicting noise or velocity requires representing full-dimensional Gaussian components (Li & He, 2025). Other work examines how this preference depends on geometry and architecture. k-Diff derives a dimensionality-dependent optimal target for a linear predictor shared across time and uses this analysis to motivate learning the target mixture (Jin & Wang, 2026). Gagneux et al. (2026) identify architectural locality as another factor: at fixed resolution, small patches favor velocity prediction, whereas large patches favor image prediction. These accounts do not quantify explicit input-rank and hidden-width constraints. Output-rank bounds restrict the directions a prediction can occupy (Zheng et al., 2025); we instead restrict the noisy information available to an unrestricted decoder. Appendix D discusses these distinctions.

We address this gap through the conversion from noise prediction to clean-image estimation. For a clean image $X_1 \in \mathbb{R}^d$ and normalized noisy observation $Y_t = X_1 + \sigma_t \varepsilon$, a noise predictor

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

gives $\widehat{X}_1 = Y_t - \sigma_t f_{\varepsilon,t}(Y_t)$. The explicit input term carries the entire corruption, so the learned branch must supply the noise contribution needed to cancel it. Clean images may have structure concentrated near a low-dimensional manifold, whereas Gaussian noise has nonzero variance in every ambient direction. This creates a *noise-transmission burden*: the architecture must make the information needed for cancellation available to its output. This distinction between population equivalence and architectural equivalence becomes especially concrete when an input embedding discards some of that information or a hidden representation has limited width.

Contributions. We address three architectural aspects of the noise-transmission burden.

1. **Input compression imposes an unavoidable error floor.** For any clean distribution with finite second moment, a rank-r linear encoder imposes an induced denoising-risk floor $\sigma_t^2(d-r)$ for bare noise prediction, even with unrestricted downstream computation. We derive the exact optimal risk, which grows as $\sigma_t^2(d-r) + O(1)$ at high noise (Thm. 1 and Cor. 2). The floor holds at every noise level without a manifold assumption. Optimal image-prediction risk remains bounded under the same compression. Affine JiT probes are consistent with the resulting $1/6$ noise-recovery ceiling.
2. **Output preconditioning can alleviate the discarded-noise burden.** We characterize how an explicit input path (a scaled copy of the intact noisy input added to the learned output) changes the optimal risk (Thm. 3). For a bounded effective skip coefficient $\tilde{c}_{\text{skip}}$ (defined in Sec. 3.2), the optimal induced denoising error is $\tilde{c}_{\text{skip}}^2\sigma_t^2(d-r) + O(1)$ at high noise (Cor. 4). Analytic cancellation yields an image-prediction-equivalent control, while EDM approaches scaled image prediction at high noise (Karras et al., 2022). Learned output shortcuts substantially improve generation with a compressed JiT backbone.
3. **We quantify the width needed for input recovery and denoising.** Assuming exact affine recovery of the noisy patch, we show that hidden width must be at least d_p, with additional coordinates needed to capture denoising information beyond the best affine estimate from that patch. The covariance spectrum of this residual determines the minimum width required for a chosen accuracy (Thm. 5 and Cor. 6). Width sweeps and frozen readouts at two resolution/patch settings show that modest widening improves both generation quality and simultaneous access to input and clean-image information.

We connect this risk analysis to practice through generation experiments and readouts of the information retained in learned representations.

2 Problem setup

Let $X_1 \in \mathbb{R}^d$ be a vectorized clean image drawn from a fixed distribution with $\mathbb{E}\|X_1\|^2 < \infty$. At an interior time t, affine Gaussian corruption and its normalized observation are

$$X_t = a_t X_1 + b_t \varepsilon, \qquad Y_t := X_t/a_t = X_1 + \sigma_t \varepsilon, \qquad \sigma_t := b_t/a_t, \tag{1}$$

where $a_t, b_t > 0$ are known and $\varepsilon \sim \mathcal{N}(0, I_d)$ is independent of X_1. All population risks are evaluated at a fixed interior time, and high noise means $\sigma_t \to \infty$. For any square-integrable target random variable T_t, write $f_{T,t}$ for a network estimator and $f^*_{T,t}(y) := \mathbb{E}[T_t \mid Y_t = y, t]$ for its population-optimal squared-loss counterpart. For path-velocity prediction, we assume additionally that a_t, b_t are differentiable and write $\dot{X}_t = \dot{a}_t X_1 + \dot{b}_t \varepsilon$ for its target. The targets X_1, ε, and $\dot{X}_t$ give the population image, noise, and path-velocity predictors $f^*_{X_1,t}$, $f^*_{\varepsilon,t}$, and $f^*_{\dot{X},t}$, related by

$$\begin{aligned} f^*_{\varepsilon,t}(y) &= (y - f^*_{X_1,t}(y))/\sigma_t, \\ f^*_{\dot{X},t}(y) &= \dot{a}_t f^*_{X_1,t}(y) + \dot{b}_t f^*_{\varepsilon,t}(y) = (\dot{b}_t/\sigma_t)y + (\dot{a}_t - \dot{b}_t/\sigma_t) f^*_{X_1,t}(y). \end{aligned} \tag{2}$$

Squared risk separates into irreducible target uncertainty and squared approximation error to $f^*_{T,t}$, also called Bayes excess risk; Appendix A.1.1 gives the identity and its derivation.

At first sight, these identities suggest that the prediction target should not matter: whenever the conversion is invertible, the population-optimal predictors give the same clean estimate. For a finite network, however, this equivalence depends on whether its architecture can represent each predictor equally well. Namely, a noise predictor $f_{\varepsilon,t}(y)$ induces the clean image estimate $\widehat{X}_1(y) = y -$

$\sigma_t f_{\varepsilon,t}(y)$, so $f_{\varepsilon,t}$ must supply the contribution needed to cancel unwanted noise carried by the explicit input term y. Cancellation can require less compressible information than the clean image because Gaussian noise has nonzero variance in every ambient direction, while clean images may concentrate near low-dimensional structure (Li & He, 2025).

This creates a *noise-transmission burden* for a network with limited capacity. Specifically, an input embedding of rank $r < d$ can discard directions needed for cancellation before they reach the predictor. Even without input compression, the model may need sufficient width to carry both the noisy input and denoising features. We study this burden through three questions. First, how much does an input bottleneck imposed by $r < d$ hurt noise recovery and the clean image estimate induced by it? Second, how much can output preconditioning, where a scaled input is added to the network output, alleviate that obstruction by supplying an explicit input contribution? Finally, without an input bottleneck, how much hidden width d_h is needed to recover the noisy input and approximate the population denoiser through affine readouts?

3 THEORY

3.1 HOW INPUT COMPRESSION LIMITS NOISE RECOVERY AND DENOISING

For noise prediction, an input bottleneck of rank $r < d$ imposes a noise-recovery floor of $d - r$ and a denoising-error floor of $\sigma_t^2(d - r)$.

If the input embedding discards some directions of the noisy image, how accurately can any subsequent computation recover the corruption noise? To isolate this question from the particular choice of network, we consider a general setup

$$Y_t = X_1 + \sigma_t \varepsilon \xrightarrow{A} AY_t \xrightarrow{f_{\varepsilon,t}} f_{\varepsilon,t}(AY_t),$$

where $A \in \mathbb{R}^{r\times d}$ is a deterministic encoder of rank $r < d$, shared across all times, and $f_{\varepsilon,t} \in \mathcal{F}$ may be any measurable decoder from $\mathbb{R}^r$ to $\mathbb{R}^d$ with finite squared risk. Thus the input bottleneck is explicit, while hidden width, attention, and the output projection are all included in the unrestricted decoder. JiT's $768 \to 128 \to 768$ bottleneck embedding provides a concrete example (Li & He, 2025, Sec. 4.2, Fig. 4, and Tab. 9). Here A collects the initial $768 \to 128$ patch projections, and the decoder includes the expansion back to 768 coordinates and all subsequent Transformer blocks and output layers. For M nonoverlapping patches of dimension d_p and a shared patch embedding of rank r_p, the whole-image encoder has $d = Md_p$ and $r = Mr_p$. The decoder can combine all encoded tokens.

To characterize the impact of this input bottleneck, compare the best direct clean-image estimates with and without compression for clean-image predictor $f_{X_1,t}$:

$$\inf_g \mathbb{E}\|X_1 - g(Y_t)\|^2 = \mathbb{E}\|X_1 - f^*_{X_1,t}(Y_t)\|^2, \tag{3}$$

$$\inf_{g\in\mathcal{F}} \mathbb{E}\|X_1 - g(AY_t)\|^2 = \mathbb{E}\|X_1 - \mathbb{E}[X_1 \mid AY_t, t]\|^2. \tag{4}$$

The first is attained by the full-observation population denoiser, the second by the conditional mean given the compressed input. A noise predictor $f_{\varepsilon,t}$ faces a different cancellation problem. Its clean estimate is $\widehat{X}_1 = Y_t - \sigma_t f_{\varepsilon,t}(AY_t)$: the explicit identity term contains the intact noisy image, but the learned branch can use only AY_t to cancel its noise.

A rank-r encoder retains r directions and discards $d - r$. At high noise, the compressed input AY_t reveals the noise components that survive the encoder more clearly: after rescaling, $AY_t/\sigma_t = AX_1/\sigma_t + A\varepsilon$ approaches $A\varepsilon$. Even so, the discarded Gaussian components remain independent of everything the decoder observes, which suggests a persistent loss of $d - r$. To quantify it, let $P_A = A^\top(AA^\top)^{-1}A$ project onto the encoder's row space. Thm. 1 gives the minimum noise-recovery risk achievable by any decoder from the compressed input.

Theorem 1 (Optimal noise recovery after input compression)**.** *Under the preceding assumptions, the optimal sampled-noise risk decomposes as*

$$\inf_{f_{\varepsilon,t}\in\mathcal{F}} \mathbb{E}\|\varepsilon - f_{\varepsilon,t}(AY_t)\|^2 = d - r + \frac{1}{\sigma_t^2}\mathbb{E}\big\|P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\big\|^2. \tag{5}$$

The minimum is attained by $f_{\varepsilon,t}(AY_t) = \mathbb{E}[\varepsilon \mid AY_t, t] = (P_A Y_t - P_A \mathbb{E}[X_1 \mid AY_t, t])/\sigma_t$.

The optimal decoder therefore pays for two sources of error: the Gaussian noise discarded at the input and uncertainty about clean signal in the retained directions. As the noise level increases, the second contribution vanishes, leaving the floor $d-r$. The noise-recovery floor translates directly into denoising error: converting the predicted noise into a clean-image estimate multiplies its squared error by σ_t^2. Indeed, since $\mathbb{E}\|\widehat{X}_1 - X_1\|^2 = \sigma_t^2 \mathbb{E}\|\varepsilon - f_{\varepsilon,t}(AY_t)\|^2$, multiplying (5) by σ_t^2 therefore gives the following growth of the best induced clean-image risk.

Corollary 2 (High-noise denoising error induced by input compression)**.** *Under the assumptions of Thm. 1, the induced clean estimate* $\widehat{X}_1 = Y_t - \sigma_t f_{\varepsilon,t}(AY_t)$ *satisfies*

$$\inf_{f_{\varepsilon,t} \in \mathcal{F}} \mathbb{E}\|\widehat{X}_1 - X_1\|^2 = \sigma_t^2 (d - r) + O(1), \qquad \sigma_t \to \infty. \tag{6}$$

As noise increases, recovering the noise components visible through A becomes easier. However, the $d - r$ discarded components remain unrecoverable, and their contribution to clean-image error grows as $\sigma_t^2(d - r)$. Increasing downstream width cannot remove this error: Thm. 1 already allows every measurable decoder. By comparison, the optimal image-prediction risk under the same input compression in Eq. (4) is bounded above by $\mathbb{E}\|X_1 - \mathbb{E}[X_1]\|^2$ at every noise level.

The same result predicts a ceiling for population noise probes. The recovery score $1 - \mathbb{E}\|\widehat{\varepsilon} - \varepsilon\|^2/d$ is at most r/d. For the shared patch embedding, this is r_p/d_p, and the whole-image floor becomes $d_p - r_p$ per patch. JiT's $768 \to 128 \to 768$ embedding therefore gives a ceiling of $128/768 = 1/6$. Affine probes form a subset of the allowed decoders and inherit this bound.

3.2 Output preconditioning under an input bottleneck

Output preconditioning can reduce the noise-transmission burden imposed by an input bottleneck: exact cancellation turns the learned branch into scaled image prediction, while EDM approaches the same behavior at high noise.

Output preconditioning combines the learned output with a scaled copy of the intact noisy input (Karras et al., 2022; Song et al., 2023; Li & He, 2025). Bare noise prediction already uses the intact input when converted to a clean estimate; the question is which coefficients leave discarded noise uncancelled. At the fixed time, write $T = T_t$ for a target mixing clean signal and noise, and suppress the time argument of the raw network f. Consider the general preconditioning

$$T = \hat{a}_t X_1 + \hat{b}_t \varepsilon, \qquad \widehat{T} = c_{\text{skip}} Y_t + c_{\text{out}} f(c_{\text{in}} AY_t). \tag{7}$$

Here $f(c_{\text{in}} AY_t)$ is the raw network output, while $\widehat{T}$ is the complete target estimate. Use the encoder A and projector P_A from Sec. 3.1; all coefficients are deterministic at the fixed time, $c_{\text{in}}, c_{\text{out}} \neq 0$, and $f \in \mathcal{F}$ is otherwise unrestricted. The skip term brings in Y_t even though f sees only AY_t. Because the network function is unrestricted, nonzero c_{in} and c_{out} do not change the best achievable risk; only c_{skip} changes how much of the intact input is added directly to the output.

The target choices $(\hat{a}_t, \hat{b}_t) = (1, 0)$, $(0, 1)$, and $(\dot{a}_t, \dot{b}_t)$ cover image, noise, and path-velocity prediction. For each target, the skip term already supplies $c_{\text{skip}} Y_t$, leaving the scaled network output $c_{\text{out}} f(c_{\text{in}} AY_t)$ to predict $T - c_{\text{skip}} Y_t = (\hat{a}_t - c_{\text{skip}}) X_1 + (\hat{b}_t - c_{\text{skip}} \sigma_t)\varepsilon$. Thus $\hat{b}_t - c_{\text{skip}} \sigma_t$ determines how much noise the network still needs to recover. The $d - r$ noise components discarded by A remain inaccessible, so their contribution to the risk is multiplied by $(\hat{b}_t - c_{\text{skip}} \sigma_t)^2$, as the following theorem makes explicit.

Theorem 3 (Target risk under output preconditioning)**.** *Under the preceding assumptions,*

$$\begin{aligned} \inf_{f \in \mathcal{F}} \mathbb{E}\|T - \widehat{T}\|^2 = (\hat{b}_t - c_{\text{skip}} \sigma_t)^2 (d - r) + \left(\hat{a}_t - \frac{\hat{b}_t}{\sigma_t} \right)^2 \mathbb{E}\big\| P_A \{ X_1 - \mathbb{E}[X_1 \mid AY_t, t] \} \big\|^2 \\ + (\hat{a}_t - c_{\text{skip}})^2 \mathbb{E}\big\| (I - P_A) \{ X_1 - \mathbb{E}[X_1 \mid AY_t, t] \} \big\|^2. \end{aligned} \tag{8}$$

The minimum is attained when the scaled learned output equals $\mathbb{E}[T - c_{\text{skip}} Y_t \mid AY_t, t]$.

The first term shows how c_{skip} controls the noise-transmission burden: the $d-r$ discarded noise components contribute $(\hat{b}_t - c_{\text{skip}}\sigma_t)^2(d-r)$ to the risk. The analytic choice $c_{\text{skip}} = \hat{b}_t/\sigma_t$ eliminates this term by making the target for the scaled network output a scaled clean image, $T - c_{\text{skip}}Y_t = (\hat{a}_t - \hat{b}_t/\sigma_t)X_1$. When $\hat{a}_t - \hat{b}_t/\sigma_t \neq 0$, the learned branch therefore performs image prediction up to a known scaling. Note that the remaining two terms can still contribute error because the clean image must be estimated from the compressed input AY_t.

EDM reduces the same burden without exact cancellation at every noise level. It uses the image target, $(\hat{a}_t, \hat{b}_t) = (1, 0)$, with coefficients

$$c_{\text{skip}} = \frac{\sigma_{\text{data}}^2}{\sigma_t^2 + \sigma_{\text{data}}^2}, \qquad c_{\text{out}} = \frac{\sigma_t \sigma_{\text{data}}}{\sqrt{\sigma_t^2 + \sigma_{\text{data}}^2}}, \qquad c_{\text{in}} = \frac{1}{\sqrt{\sigma_t^2 + \sigma_{\text{data}}^2}},$$

with fixed data scale $\sigma_{\text{data}} > 0$ (Karras et al., 2022, Tab. 1 and Appendix B.6, Eqs. (117), (131), and (138)). Substituting these coefficients into Thm. 3 gives the discarded-noise contribution

$$c_{\text{skip}}^2 \sigma_t^2 (d-r) = \frac{\sigma_{\text{data}}^4 \sigma_t^2}{(\sigma_t^2 + \sigma_{\text{data}}^2)^2}(d-r) = O(\sigma_t^{-2}).$$

It vanishes at high noise, while the remaining clean uncertainty stays bounded. Moreover, the raw network target $(X_1 - c_{\text{skip}}Y_t)/c_{\text{out}}$ converges in mean square to X_1/σ_{data}. Thus EDM's learned branch approaches scaled image prediction as the noise level grows. Appendix A.2.2 derives these consequences of the coefficients under our input restriction.

To summarize the denoising consequences for these parameterizations, we convert the target prediction into a clean-image estimate. Rewriting the same target using $\varepsilon = (Y_t - X_1)/\sigma_t$ gives $T = (\hat{a}_t - \hat{b}_t/\sigma_t)X_1 + (\hat{b}_t/\sigma_t)Y_t$. Whenever $\hat{a}_t - \hat{b}_t/\sigma_t \neq 0$,

$$\widehat{X}_1 = \frac{\widehat{T} - (\hat{b}_t/\sigma_t)Y_t}{\hat{a}_t - \hat{b}_t/\sigma_t} = \underbrace{\frac{c_{\text{skip}} - \hat{b}_t/\sigma_t}{\hat{a}_t - \hat{b}_t/\sigma_t}}_{\tilde{c}_{\text{skip}}} Y_t + \underbrace{\frac{c_{\text{out}}}{\hat{a}_t - \hat{b}_t/\sigma_t}}_{\tilde{c}_{\text{out}}} f(c_{\text{in}} A Y_t). \tag{9}$$

The converted coefficients $\tilde{c}_{\text{skip}}$ and $\tilde{c}_{\text{out}}$ determine the clean-image estimate, whose risk follows from Thm. 3.

Corollary 4 (Denoising risk under output preconditioning)**.** *For $\widehat{X}_1 = \tilde{c}_{\text{skip}}Y_t + \tilde{c}_{\text{out}} f(c_{\text{in}}AY_t)$ under the preceding assumptions, every decoder satisfies $\mathbb{E}\|\widehat{X}_1 - X_1\|^2 \geq \tilde{c}_{\text{skip}}^2 \sigma_t^2 (d-r)$. If $\tilde{c}_{\text{skip}}(t)$ stays bounded as $\sigma_t \to \infty$, then*

$$\inf_{f \in \mathcal{F}} \mathbb{E}\|\widehat{X}_1 - X_1\|^2 = \tilde{c}_{\text{skip}}(t)^2 \sigma_t^2 (d-r) + O(1), \qquad \sigma_t \to \infty. \tag{10}$$

Cor. 4 puts the three cases in a common form. Exact cancellation gives $\tilde{c}_{\text{skip}} = 0$, eliminating the discarded-noise contribution, while bare noise prediction has $\tilde{c}_{\text{skip}} = 1$ and incurs the $\sigma_t^2(d-r)$ error growth. For EDM, $\tilde{c}_{\text{skip}} = c_{\text{skip}} = \sigma_{\text{data}}^2/(\sigma_t^2 + \sigma_{\text{data}}^2)$, so this contribution vanishes at high noise and the optimal denoising risk remains bounded.

3.3 The width cost of simultaneous input recovery and denoising

Assuming exact affine recovery of the noisy patch, the extra width needed for denoising is determined by the residual beyond its best affine patch estimate.

We now take $A = I_d$ and ask how hidden width limits simultaneous input recovery and denoising. For a Transformer, d_h is the width of the hidden vector used to predict one output patch. Consider a patch m of dimension d_p, with corruption $X_{t,m} = a_t X_{1,m} + b_t \varepsilon_m$ where $X_{1,m}, X_{t,m}, \varepsilon_m \in \mathbb{R}^{d_p}$. The population-optimal noise predictor satisfies $f^*_{\varepsilon_m,t}(Y_t) = [X_{t,m} - a_t D^*_{m,t}(Y_t)]/b_t$, where $D^*_{m,t}(Y_t) := \mathbb{E}[X_{1,m} \mid Y_t, t]$ is the population patch denoiser. This identity motivates an input-preserving realization: keep the noisy patch and denoising information separately accessible to affine readouts. Separate input recovery is not required by the identity; we impose it to characterize the exact width cost within this representation class.

Assumption 1 (Affine recovery of the noisy patch)**.** *The hidden representation $h(Y_t) \in \mathbb{R}^{d_h}$ is a measurable function of the whole noisy image with $\mathbb{E}\|h(Y_t)\|^2 < \infty$. There exist $W \in \mathbb{R}^{d_p \times d_h}$ and $b \in \mathbb{R}^{d_p}$, fixed across images, such that*

$$Wh(Y_t) + b = X_{t,m} \quad \textit{with probability one}.$$

Under Assump. 1, the clean-image estimate induced by a noise predictor with an affine output layer can also be written as $\widehat{X}_{1,m}(Y_t) = [X_{t,m} - b_t f_{\varepsilon_m,t}(Y_t)]/a_t = Ch(Y_t) + c$. Here h represents all computation before the noise predictor's final affine layer, and $C \in \mathbb{R}^{d_p \times d_h}$ and $c \in \mathbb{R}^{d_p}$ are the coefficients of the induced clean estimate, fixed across images. Our goal is to find the best denoising approximation achievable at width d_h, optimizing both h and its final affine layer while preserving the noisy patch as required by Assump. 1.

Preserving the noisy patch requires at least d_p independent feature directions. Those directions already support some denoising: any affine function of $X_{t,m}$ is also an affine function of $h(Y_t)$. Let $\ell_t(X_{t,m})$ be the best affine approximation to the population denoiser using the noisy patch alone. This baseline is available from every representation satisfying Assump. 1, so optimizing the clean readout achieves at least its accuracy without requiring any additional feature coordinates.

The additional width is needed to improve on this baseline. The part of the population denoiser that the baseline misses is $R_t = D^*_{m,t}(Y_t) - \ell_t(X_{t,m})$. This residual can reflect nonlinear dependence on the noisy patch or information from other patches that an affine function of $X_{t,m}$ alone cannot provide. Because ℓ_t is the best affine approximation, $\mathbb{E}R_t = 0$ and $\mathrm{Cov}(R_t, X_{t,m}) = 0$; Appendix A.3.1 gives the derivation. The width question thus becomes how many additional coordinates are needed to approximate R_t. The following theorem answers this through the eigenvalues of $\mathrm{Cov}(R_t)$, ordered from largest to smallest.

Theorem 5 (Optimal denoiser approximation at a fixed width)**.** *Under Assump. 1, exact affine patch recovery is impossible if $d_h < d_p$. For $d_h \geq d_p$,*

$$\inf_{h,W,b,C,c:\, Wh+b=X_{t,m}} \mathbb{E}\|\widehat{X}_{1,m}(Y_t) - D^*_{m,t}(Y_t)\|^2 = \sum_{j>d_h-d_p} \lambda_j(\mathrm{Cov}(R_t))\,. \tag{11}$$

At $d_h = d_p$, the best affine clean readout is ℓ_t, with approximation error $\mathbb{E}\|R_t\|^2$. An optimal representation uses its $d_h - d_p$ extra coordinates to capture the leading covariance directions of R_t, leaving the spectral tail as error; concentrated residual variance therefore requires less additional width. Width $2d_p$ suffices to retain the input and full residual, matching the population denoiser and attaining the irreducible Bayes risk. These optima are attainable over arbitrary measurable representations, but need not be realized by a particular architecture or training procedure.

More generally, we can use the residual spectrum to determine how much width is needed to reach a chosen approximation tolerance.

Corollary 6 (Width required at a fixed approximation tolerance)**.** *For $\delta \geq 0$, the smallest width in the preceding representation class attaining $\mathbb{E}\|\widehat{X}_{1,m}(Y_t) - D^*_{m,t}(Y_t)\|^2 \leq \delta$ is*

$$d_h = d_p + \min\left\{ k \in \{0, \ldots, d_p\} : \sum_{j>k} \lambda_j(\mathrm{Cov}(R_t)) \leq \delta \right\}.$$

Thus exact recovery of both the noisy patch and its population denoiser requires $d_p + \mathrm{rank}(\mathrm{Cov}(R_t))$ coordinates. Allowing some approximation error may reduce the number of additional coordinates needed.

4 Experiments

We examine noise recovery after compression, output preconditioning, and simultaneous input/denoiser accessibility across widths. The first suite uses JiT-B/16 (Li & He, 2025) on ImageNet256, with a $768 \to 128 \to 768$ patch embedding; the second removes this extra bottleneck and varies hidden width at two resolution/patch settings. Both use $X_t = tX_1 + (1-t)\varepsilon$, so small t means high noise. Within each suite, the objective equals clean-image squared error away from

endpoint clamps; loss weighting is not separately optimized for each target. Appendices B.1 and C give the full protocols. We measure Fréchet Inception Distance on $10{,}000$ samples (FID-10K). Bottleneck experiments use guidance scale 2.9; width experiments are unguided, so FID comparisons are made within each suite.

4.1 Prediction targets under input compression

Under the same compressed backbone and loss, image prediction markedly outperforms noise and path-velocity prediction.

We first test whether the prediction gap persists when the backbone and image-loss convention are held fixed. We train image, noise, and path-velocity predictors with the same compressed JiT-B/16 backbone and compare their generation quality. At epoch 200, image prediction reaches FID 9.13, compared with 412.11 for noise and 177.20 for path velocity (Tab. 1). To examine the underlying information loss, we next probe the learned representations.

Table 1: Guided FID-10K (↓) for prediction targets and output shortcuts with the same compressed JiT-B/16 backbone and image-loss convention. Learned variants are named by initialization; each entry is a single run.

Variant	Epoch 40	Epoch 160	Epoch 200
Image prediction	55.35	9.87	9.13
Noise prediction	412.73	411.79	412.11
Path-velocity prediction	198.03	177.92	177.20
Noise + additive shortcut	211.45	181.78	181.09
Noise + exact preconditioning	55.04	10.23	9.15
Noise + learned (noise init.)	71.64	16.71	14.99
Noise + learned (additive init.)	65.50	16.44	15.13
Noise + learned (exact init.)	52.03	10.96	9.56

4.2 Probing the noise-recovery ceiling

At high noise, noise prediction nearly saturates the compression ceiling (Thm. 1) while providing little affine access to the clean image.

To examine the information retained after compression, we fit affine probes to matched image- and noise-prediction checkpoints labeled 160, using training-consistent $[-1, 1]$ images and disjoint probe fitting and evaluation splits. We recover the clean image, sampled noise, and noisy input from the embedding and each Transformer layer at $t = 0.1$ and 0.9. We report EV, one minus the mean squared error divided by the target's total variance (Appendix B.4). For the $768 \to 128$ compression, Thm. 1 bounds population noise recovery by $128/768 = 1/6$. Tab. 2 reports the final-layer scores.

Table 2: Final-layer affine recovery at checkpoint 160 (EV, ↑), using training-consistent $[-1, 1]$ inputs and disjoint probe splits. The population noise-recovery ceiling is $1/6$.

	$\mathrm{EV}[X_1]$		$\mathrm{EV}[\varepsilon]$	
Predictor	$t = .1$	$t = .9$	$t = .1$	$t = .9$
Image	0.7689	0.9860	0.0867	0.0208
Noise	0.0047	0.0237	0.1666	0.1393

At high noise ($t = 0.1$), the noise predictor reaches $\mathrm{EV}[\varepsilon] \simeq 0.1666$ at every measured layer, but its final-layer image recovery is only 0.0047. Image prediction instead improves $\mathrm{EV}[X_1]$ from 0.5005 at the embedding to 0.7689 at the final layer, while noise recovery falls from 0.1648 to 0.0867. The discarded noise still imposes a clean-image risk floor $\sigma_t^2(d - r)$ (Cor. 2), motivating the output intervention below. Appendix B.4.1 gives the full layerwise results and fitting protocol.

4.3 Comparing output preconditioning schemes

Exact cancellation gives an image-prediction-equivalent control (Thm. 3); learned shortcuts yield substantial gains without fixed cancellation coefficients.

Keeping the compressed backbone fixed, we compare output paths that leave different amounts of the discarded-noise contribution. Exact cancellation serves as an image-prediction-equivalent control away from endpoint clamps; additive and learned shortcuts test alternatives without fixed cancellation coefficients. Appendix B.2 gives the coefficients and endpoint conventions.

At epoch 200, exact preconditioning reaches FID 9.15, compared with 412.11 for bare noise prediction and 9.13 for image prediction (Tab. 1). The additive shortcut reaches 181.09: it removes high-noise divergence of the discarded-noise term but leaves a nonzero contribution to induced clean risk (Appendix B.2). Learned shortcuts reach 14.99 and 15.13 from noise and additive initialization, and 9.56 from analytic initialization. EDM is analyzed separately in the theory. Appendices B.2 and B.3 report full FID trajectories and additional input-access interventions.

4.4 Exploring the width cost of input preservation

Increasing model width substantially improves generation quality, even at widths well below $2d_p$.

We next test how much widening improves noise prediction after removing the extra embedding bottleneck. The d_p input-preservation cost in Thm. 5 also motivates an exploratory comparison of noise-prediction width $d_p + q$ with image-prediction width q. The theorem does not predict equal FID; the offset is fixed by patch geometry rather than fitted to the results.

We replace the bottleneck with a single patch projection and hold depth and the image-loss convention fixed while varying width. ImageNet256 with 16×16 RGB patches has $d_p = 768$; ImageNet128 with 8×8 patches has $d_p = 192$. Both have 256 image tokens. Fig. 1 reports unguided FID after 40 and 80 epochs, retaining the recorded checkpoint and parameter-average choices (Appendix C).

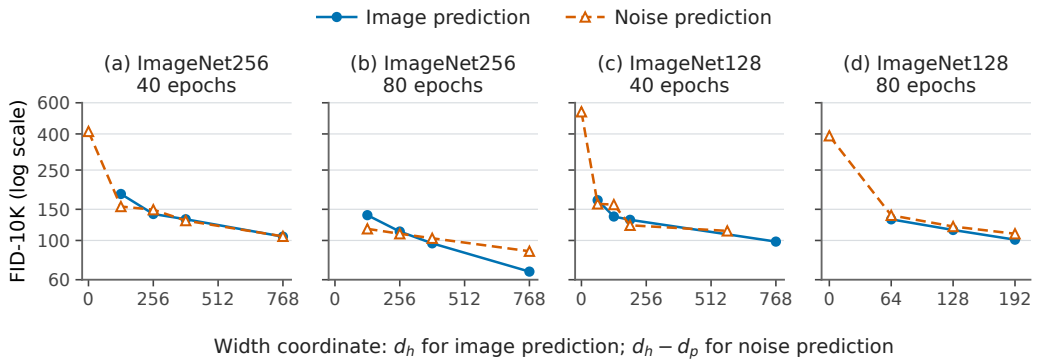


Figure 1: Width sweeps on ImageNet256/16 (left two panels) and ImageNet128/8 (right two), at 40 and 80 epochs. Unguided FID-10K (↓, logarithmic scale) is plotted against d_h for image prediction and $d_h - d_p$ for noise prediction ($d_p = 768, 192$). The shift is exploratory, not a prediction of equal FID. These settings vary resolution and patch size; widening also changes parameter count and attention configuration (Appendix C).

At epoch 40, increasing noise-prediction width from 768 to 896 reduces FID from 413.23 to 155.49 on ImageNet256; increasing it from 192 to 256 reduces FID from 533.48 to 160.93 on ImageNet128. After the prescribed shift, several curves also align closely: at epoch 80, the $q = 256$ pair on ImageNet256 reaches FID 109.11 for noise prediction and 112.33 for image prediction, while the $q = 128$ pair on ImageNet128 reaches 119.91 and 114.62. The correspondence is approximate—the widest ImageNet256 pair separates to 86.92 versus 66.67—but the gains motivate examining whether wider representations provide better simultaneous access to input and denoising information.

4.5 Testing simultaneous input and denoiser accessibility

Generation gains from widening are accompanied by stronger simultaneous affine access to input and clean-image information.

We examine simultaneous accessibility with affine readouts of 14 frozen epoch-40 models at $t = 0.5$, using the hidden representation h before the final linear head. We recover the noisy patch $X_{t,m}$ and clean patch $X_{1,m}$ from h, and compare clean readouts from the raw patch and $[X_{t,m}, h]$. The

concatenated probe refits both input and feature coefficients; the native noise-to-image conversion already uses the input but fixes its coefficient. Thus these probes diagnose accessibility, not the cause of sampling failure. At the population optimum, differences in clean-readout risk equal differences in denoiser-approximation error (Appendix C.2). Scores are $100R^2$, or $100\mathrm{EV}$; Appendix C.3 gives the protocol.

Table 3: Affine readouts of all 14 frozen epoch-40 checkpoints at $t = 0.5$, using the final-head input h. Scores are $100R^2$ (↑); the final column gives the gain from supplying the noisy input, computed before rounding.

Prediction	Width d_h	Input $h \to X_{t,m}$	Clean $h \to X_{1,m}$	Clean with input $[X_{t,m}, h] \to X_{1,m}$	Gain from $X_{t,m}$ (points)
ImageNet256/16: $d_p = 768$; raw-input baseline $X_{t,m} \to X_{1,m}$: 90.28					
Noise	768	77.85	0.67	93.71	93.05
Noise	896	98.50	88.56	94.98	6.42
Noise	1024	98.80	90.00	95.09	5.09
Noise	1152	99.54	93.42	95.32	1.90
Noise	1536	99.65	93.94	95.42	1.47
Image	128	24.08	94.08	94.12	0.04
Image	256	24.77	94.68	94.69	0.01
Image	384	26.30	94.96	94.95	-0.01
Image	768	34.18	95.35	95.31	-0.04
ImageNet128/8: $d_p = 192$; raw-input baseline $X_{t,m} \to X_{1,m}$: 86.19					
Noise	192	78.94	2.36	89.90	87.54
Noise	320	99.75	92.28	93.45	1.17
Noise	384	99.77	92.44	93.52	1.08
Image	128	26.22	93.21	93.23	0.02
Image	192	28.13	93.48	93.49	0.01

At square width $d_h = d_p = 768$ on ImageNet256, noisy-input recovery is 77.85, and clean-image recovery is only 0.67 from h. Supplying the input raises the clean score to 93.71, above the raw-input baseline of 90.28 (Tab. 3). The same pattern holds at $d_h = d_p = 192$ on ImageNet128. Thus the narrow noise predictors contain useful denoising information whose contribution to an affine readout depends strongly on access to the noisy input.

Widening to 896 on ImageNet256 raises input recovery to 98.50 and clean recovery to 88.56, reducing the benefit of supplying the input from 93.05 to 6.42 score points. On ImageNet128, widening from 192 to 320 reduces that benefit from 87.54 to 1.17 points. Image-prediction controls already provide strong clean readouts at small widths. This pattern is consistent with Sec. 3.3's input-preserving strategy, without establishing that separate affine input recovery is necessary for accurate noise prediction.

5 Conclusion

Population-equivalent diffusion targets can impose different architectural burdens once access to the noisy input and hidden representation width are constrained. We characterize these burdens through an exact input-compression risk, output-preconditioning analysis, and a conditional width–accuracy tradeoff for input-preserving representations. Our theory concerns fixed-time squared risk under Gaussian corruption: the rank results assume linear compression, and the width result requires exact affine input recovery over measurable representations. These attainable limits do not guarantee realization by a particular network or training procedure, or determine how errors accumulate during sampling. Experiments cover the tested JiT and ImageNet settings; widening changes other aspects of capacity, and finite-sample probes measure approximate affine accessibility. Appendices B and C detail the protocol qualifications. The architectural lesson is to distinguish information the learned branch must estimate from input contributions supplied analytically, and to account for the conditional width cost of keeping input and denoising features jointly accessible.

REPRODUCIBILITY STATEMENT

The code is available in an anonymous repository at https://anonymous.4open.science/r/noise-transmission/README.md. Appendix A contains the proofs of all theorems and corollaries, together with the supporting derivations. Appendices B and C document the experimental settings, output coefficients, checkpoint choices, and probe procedures, and report the additional numerical results and protocol qualifications.

AI USE STATEMENT

Generative AI tools assisted with reviewing the literature, improving the presentation of the work, developing and cleaning up code, carrying out experiments, refining proofs, providing manuscript feedback, and editing the text. The authors independently reviewed and verified all AI-assisted results, made the final scientific and methodological decisions, and take full responsibility for the contents of this paper.

REFERENCES

Shadab Ahamed, Eshed Gal, Md Shahriar Rahim Siddiqui, Simon Ghyselincks, Moshe Eliasof, and Eldad Haber. Preconditioned flow matching. *arXiv preprint arXiv:2603.02337*, 2026. URL https://arxiv.org/abs/2603.02337v2. Version 2, May 13, 2026.

Hansheng Chen, Jan Ackermann, Minseo Kim, Gordon Wetzstein, and Leonidas Guibas. Asymmetric flow models. *arXiv preprint arXiv:2605.12964*, 2026. URL https://arxiv.org/abs/2605.12964v2. Version 2, May 25, 2026.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4672–4712, 2023. URL https://proceedings.mlr.press/v202/chen23o.html.

Anne Gagneux, Ségolène Martin, Rémi Gribonval, and Mathurin Massias. Training flow matching: The role of weighting and parameterization. In *Second DeLTa Workshop at ICLR*, 2026. URL https://arxiv.org/abs/2603.06454v2. arXiv:2603.06454v2, April 17, 2026.

Qing Jin and Chaoyang Wang. Revisiting diffusion model predictions through dimensionality. *arXiv preprint arXiv:2601.21419*, 2026. URL https://arxiv.org/abs/2601.21419v2. Version 2, March 27, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2206.00364v2. Archived arXiv:2206.00364v2, October 11, 2022.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. URL https://arxiv.org/abs/2511.13720v2. Version 2, January 7, 2026.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html. ICML 2023 proceedings version.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. URL https://arxiv.org/abs/2510.11690v1. Version 1, October 13, 2025.

A PROOFS

All arguments use the model and conventions of Sec. 2. We work at the fixed time of the corresponding statement and suppress this conditioning when it is unambiguous.

A.1 INPUT COMPRESSION

A.1.1 CONDITIONAL EXPECTATION AND PROJECTED RISK

We first record the projection identities used in the proofs. For an observation Z, a square-integrable vector target T, and any measurable predictor $g(Z)$ with finite squared risk, conditional expectation gives

$$\mathbb{E}\big[T - \mathbb{E}[T \mid Z] \mid Z\big] = 0.$$

Consequently the cross term in the squared-error expansion vanishes:

$$\begin{aligned}&\mathbb{E}\,\langle T - \mathbb{E}[T \mid Z],\, \mathbb{E}[T \mid Z] - g(Z)\rangle \\ &\quad = \mathbb{E}\,\langle \mathbb{E}[T - \mathbb{E}[T \mid Z] \mid Z],\, \mathbb{E}[T \mid Z] - g(Z)\rangle = 0.\end{aligned}$$

Thus

$$\mathbb{E}\|T - g(Z)\|^2 = \mathbb{E}\|T - \mathbb{E}[T \mid Z]\|^2 + \mathbb{E}\|\mathbb{E}[T \mid Z] - g(Z)\|^2. \tag{12}$$

This proves optimality of the conditional mean and identifies the approximation error above its risk. With $Z = Y_t$ and $T = T_t$, taking the infimum over a class $\mathcal{F}$ of measurable predictors with finite squared risk gives

$$\inf_{f_{T,t}\in\mathcal{F}} \mathbb{E}\|f_{T,t}(Y_t) - T_t\|^2 = \mathbb{E}\|T_t - f^*_{T,t}(Y_t)\|^2 + \inf_{f_{T,t}\in\mathcal{F}} \mathbb{E}\|f_{T,t}(Y_t) - f^*_{T,t}(Y_t)\|^2. \tag{13}$$

The first term is the irreducible target uncertainty. The second is the minimum squared approximation error to the population predictor within $\mathcal{F}$, or Bayes excess risk.

For a deterministic orthogonal projector H, apply the same identity to HX_1 with observation Z and constant predictor $H\mathbb{E}X_1$. Since $\mathbb{E}[HX_1 \mid Z] = H\mathbb{E}[X_1 \mid Z]$,

$$\begin{aligned}&\mathbb{E}\|H(X_1 - \mathbb{E}[X_1 \mid Z])\|^2 + \mathbb{E}\|H(\mathbb{E}[X_1 \mid Z] - \mathbb{E}X_1)\|^2 \\ &\quad = \mathbb{E}\|H(X_1 - \mathbb{E}X_1)\|^2 = \operatorname{tr}\big(H\,\mathrm{Cov}(X_1)H\big) = \operatorname{tr}\big(H\,\mathrm{Cov}(X_1)\big).\end{aligned}$$

In particular,

$$0 \le \mathbb{E}\|H(X_1 - \mathbb{E}[X_1 \mid Z])\|^2 \le \operatorname{tr}\big(H\,\mathrm{Cov}(X_1)\big). \tag{14}$$

The right side is independent of the noise level when the clean distribution and projector are fixed.

A.1.2 PROOF OF THM. 1

Proof. Because A has full row rank, $AA^\top$ is invertible. Its row-space projector satisfies

$$P_A = A^\top(AA^\top)^{-1}A, \qquad P_A = P_A^\top = P_A^2, \qquad AP_A = A, \qquad \operatorname{tr}(P_A) = r.$$

Moreover,

$$P_AY_t = A^\top(AA^\top)^{-1}(AY_t), \qquad AY_t = A(P_AY_t).$$

Hence AY_t and P_AY_t generate the same observed information. This equivalence holds for an arbitrary full-row-rank encoder, without requiring orthonormal rows.

The two projected Gaussian vectors $P_A\varepsilon$ and $(I - P_A)\varepsilon$ have cross covariance

$$\mathbb{E}\big[P_A\varepsilon\{(I - P_A)\varepsilon\}^\top\big] = P_A(I - P_A) = 0.$$

Joint Gaussianity makes them independent. Independence of X_1 and ε then implies that $(I - P_A)\varepsilon$ is independent of the pair $(X_1, P_A\varepsilon)$. The observation

$$AY_t = AX_1 + \sigma_t AP_A\varepsilon$$

is a function of that pair, so

$$\mathbb{E}[(I - P_A)\varepsilon \mid AY_t, t] = 0, \qquad \mathbb{E}\|(I - P_A)\varepsilon\|^2 = \operatorname{tr}(I - P_A) = d - r.$$

This argument does not require independence between different clean-image coordinates or patches.

The observed retained coordinates satisfy $P_A Y_t = P_A X_1 + \sigma_t P_A \varepsilon$. Taking conditional expectations therefore gives

$$\begin{aligned}\mathbb{E}[\varepsilon \mid AY_t, t] &= \mathbb{E}[P_A \varepsilon \mid AY_t, t] \\ &= \frac{P_A Y_t - P_A \mathbb{E}[X_1 \mid AY_t, t]}{\sigma_t}.\end{aligned}$$

Equation (12) shows that this measurable conditional mean attains the minimum over the decoder class. It has finite second moment by the conditional-expectation contraction property.

Subtracting the conditional mean from the sampled noise yields

$$\varepsilon - \mathbb{E}[\varepsilon \mid AY_t, t] = (I - P_A)\varepsilon - \frac{P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}}{\sigma_t}.$$

The two terms lie in orthogonal subspaces for every realization. Their squared norms therefore add without a cross term:

$$\begin{aligned}\inf_{f_{\varepsilon,t} \in \mathcal{F}} \mathbb{E}\|\varepsilon - f_{\varepsilon,t}(AY_t)\|^2 &= \mathbb{E}\|(I - P_A)\varepsilon\|^2 + \sigma_t^{-2}\mathbb{E}\big\|P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\big\|^2 \\ &= d - r + \sigma_t^{-2}\mathbb{E}\big\|P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\big\|^2,\end{aligned}$$

which is Eq. (5). For a fixed clean distribution and encoder, Eq. (14) also gives

$$0 \le \inf_{f_{\varepsilon,t} \in \mathcal{F}} \mathbb{E}\|\varepsilon - f_{\varepsilon,t}(AY_t)\|^2 - (d - r) \le \frac{\operatorname{tr}(P_A \operatorname{Cov}(X_1))}{\sigma_t^2}.$$

Thus the optimal noise-recovery risk converges to $d - r$ as $\sigma_t \to \infty$. □

A.1.3 Proof of Cor. 2

Proof. For each decoder the induced clean error is exactly

$$\widehat{X}_1 - X_1 = Y_t - \sigma_t f_{\varepsilon,t}(AY_t) - X_1 = \sigma_t\{\varepsilon - f_{\varepsilon,t}(AY_t)\}.$$

The scale is deterministic at the fixed time. Multiplying Thm. 1 by σ_t^2 therefore gives

$$\inf_{f_{\varepsilon,t}} \mathbb{E}\|\widehat{X}_1 - X_1\|^2 = \sigma_t^2(d - r) + \mathbb{E}\big\|P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\big\|^2.$$

By Eq. (14), the remainder belongs to the interval

$$[0, \operatorname{tr}(P_A \operatorname{Cov}(X_1))]$$

uniformly in the noise level. This proves the asserted $O(1)$ remainder for the fixed clean distribution and encoder. □

A.2 Output preconditioning

A.2.1 Proof of Thm. 3

Proof. At the fixed time, multiplication by $c_{\text{in}} \neq 0$ is invertible on the observed space and multiplication by $c_{\text{out}} \neq 0$ is invertible on the output space. Thus the functions

$$c_{\text{out}} f(c_{\text{in}} AY_t)$$

range over all measurable functions of AY_t with finite squared risk. Indeed, for any such function $g(AY_t)$, the choice $f(u) = g(u/c_{\text{in}})/c_{\text{out}}$ realizes it. The optimization therefore reduces to conditional-mean prediction of the residual target

$$T - c_{\text{skip}} Y_t = (\hat{a}_t - c_{\text{skip}})X_1 + (\hat{b}_t - c_{\text{skip}}\sigma_t)\varepsilon.$$

Its conditional mean given AY_t is attainable by the scaled learned output, by Eq. (12).

In the retained subspace, $P_A\varepsilon = (P_A Y_t - P_A X_1)/\sigma_t$, so

$$\begin{aligned}P_A(T - c_{\text{skip}} Y_t) &= (\hat{a}_t - c_{\text{skip}})P_A X_1 + \left(\frac{\hat{b}_t}{\sigma_t} - c_{\text{skip}}\right)(P_A Y_t - P_A X_1) \\ &= \left(\hat{a}_t - \frac{\hat{b}_t}{\sigma_t}\right) P_A X_1 + \left(\frac{\hat{b}_t}{\sigma_t} - c_{\text{skip}}\right) P_A Y_t.\end{aligned}$$

The second term is observed. After subtracting the conditional mean, the retained error is therefore

$$\left(\hat{a}_t - \frac{\hat{b}_t}{\sigma_t}\right) P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}.$$

Its squared risk is the second term of Eq. (8). In particular, the skip coefficient does not affect this term, because the retained part of the skip was already available to the learned branch.

In the discarded subspace, the independence established in Appendix A.1.2 gives the conditional residual

$$\begin{aligned}&(I - P_A)\left\{T - c_{\text{skip}} Y_t - \mathbb{E}[T - c_{\text{skip}} Y_t \mid AY_t, t]\right\} \\ &\quad = (\hat{a}_t - c_{\text{skip}})(I - P_A)\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\} + (\hat{b}_t - c_{\text{skip}}\sigma_t)(I - P_A)\varepsilon.\end{aligned}$$

The clean uncertainty is a function of (X_1, AY_t); the discarded noise is independent of that pair and has mean zero. Their expected inner product is zero. The discarded risk is consequently

$$\begin{aligned}&(\hat{a}_t - c_{\text{skip}})^2 \mathbb{E}\left\|(I - P_A)\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\right\|^2 \\ &\qquad + (\hat{b}_t - c_{\text{skip}}\sigma_t)^2 (d - r).\end{aligned}$$

Retained and discarded vectors are orthogonal pointwise. Adding their risks proves the exact optimum in Eq. (8). An arbitrary decoder adds a nonnegative conditional-mean approximation error, so its risk is at least the stated discarded-noise contribution. □

A.2.2 Exact cancellation and EDM

The residual target in the general wrapper has noise coefficient $\hat{b}_t - c_{\text{skip}}\sigma_t$. Choosing $c_{\text{skip}} = \hat{b}_t/\sigma_t$ gives

$$T - c_{\text{skip}} Y_t = \left(\hat{a}_t - \frac{\hat{b}_t}{\sigma_t}\right) X_1.$$

When this coefficient is nonzero, the learned branch is an image predictor up to deterministic scaling, and target conversion removes the explicit skip from the complete clean estimate.

For EDM, apply Thm. 3 with the image target $(\hat{a}_t, \hat{b}_t) = (1, 0)$ and substitute the coefficients from Tab. 1 and Appendix B.6 of Karras et al. (2022). The discarded-noise contribution becomes

$$c_{\text{skip}}^2 \sigma_t^2 (d - r) = \frac{\sigma_{\text{data}}^4 \sigma_t^2}{(\sigma_t^2 + \sigma_{\text{data}}^2)^2}(d - r).$$

Multiplication by σ_t^2 gives a quantity tending to $\sigma_{\text{data}}^4(d - r)$, so the contribution is $O(\sigma_t^{-2})$ and tends to zero. Since $0 < c_{\text{skip}} < 1$, the sum of the two clean-uncertainty terms in the theorem is bounded by $\operatorname{tr}\operatorname{Cov}(X_1)$. The optimal total clean risk is therefore bounded. This uses an upper bound on the exact remainder in addition to the vanishing noise contribution.

The effective raw target admits the explicit expansion

$$\begin{aligned}\frac{X_1 - c_{\text{skip}} Y_t}{c_{\text{out}}} &= \frac{1 - c_{\text{skip}}}{c_{\text{out}}} X_1 - \frac{c_{\text{skip}}\sigma_t}{c_{\text{out}}}\varepsilon \\ &= \frac{\sigma_t}{\sigma_{\text{data}}\sqrt{\sigma_t^2 + \sigma_{\text{data}}^2}} X_1 - \frac{\sigma_{\text{data}}}{\sqrt{\sigma_t^2 + \sigma_{\text{data}}^2}}\varepsilon.\end{aligned}$$

The first coefficient tends to $1/\sigma_{\text{data}}$ and the second to zero. Independence and zero mean of ε give the exact mean-square difference

$$\begin{aligned}&\mathbb{E}\left\|\frac{X_1 - c_{\text{skip}} Y_t}{c_{\text{out}}} - \frac{X_1}{\sigma_{\text{data}}}\right\|^2 \\ &\quad = \left(\frac{\sigma_t}{\sigma_{\text{data}}\sqrt{\sigma_t^2 + \sigma_{\text{data}}^2}} - \frac{1}{\sigma_{\text{data}}}\right)^2 \mathbb{E}\|X_1\|^2 + \frac{d\sigma_{\text{data}}^2}{\sigma_t^2 + \sigma_{\text{data}}^2} \longrightarrow 0.\end{aligned}$$

This proves the stated convergence to scaled image prediction.

A.2.3 TARGET CONVERSION AND PROOF OF COR. 4

Proof. Substitution of $\varepsilon = (Y_t - X_1)/\sigma_t$ gives

$$T = \left(\hat{a}_t - \frac{\hat{b}_t}{\sigma_t}\right) X_1 + \frac{\hat{b}_t}{\sigma_t} Y_t.$$

When the clean coefficient is nonzero, the converted estimate in Eq. (9) satisfies the samplewise identity

$$\widehat{X}_1 - X_1 = \frac{\widehat{T} - T}{\hat{a}_t - \hat{b}_t/\sigma_t}.$$

Taking squared norms and expectations gives the target-to-clean risk scaling. Substitution of the preconditioned output also shows explicitly how its coefficients change:

$$\widehat{X}_1 = \frac{c_{\text{skip}} - \hat{b}_t/\sigma_t}{\hat{a}_t - \hat{b}_t/\sigma_t} Y_t + \frac{c_{\text{out}}}{\hat{a}_t - \hat{b}_t/\sigma_t} f(c_{\text{in}} A Y_t).$$

These are precisely $\tilde{c}_{\text{skip}}$ and $\tilde{c}_{\text{out}}$ defined in Eq. (9).

Apply Thm. 3 to the clean estimate $\widehat{X}_1 = \tilde{c}_{\text{skip}} Y_t + \tilde{c}_{\text{out}} f(c_{\text{in}} A Y_t)$: use the image target $(\hat{a}_t, \hat{b}_t) = (1, 0)$ and coefficients $\tilde{c}_{\text{skip}}, \tilde{c}_{\text{out}}$ in place of $c_{\text{skip}}, c_{\text{out}}$. The exact optimal clean risk is

$$\begin{aligned}\inf_f \mathbb{E}\|\widehat{X}_1 - X_1\|^2 = \tilde{c}_{\text{skip}}^2 \sigma_t^2 (d - r) \\ + \mathbb{E}\big\|P_A\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\big\|^2 \\ + (1 - \tilde{c}_{\text{skip}})^2 \mathbb{E}\big\|(I - P_A)\{X_1 - \mathbb{E}[X_1 \mid AY_t, t]\}\big\|^2.\end{aligned}$$

All three terms are nonnegative, which proves the lower bound for every decoder. Applying Eq. (14) to the two clean terms bounds their sum by

$$\operatorname{tr}(P_A \operatorname{Cov}(X_1)) + (1 - \tilde{c}_{\text{skip}})^2 \operatorname{tr}((I - P_A) \operatorname{Cov}(X_1)).$$

For a bounded sequence of skip coefficients, this expression is uniformly bounded. The remaining term is exactly $\tilde{c}_{\text{skip}}^2 \sigma_t^2 (d - r)$, proving Eq. (10). □

A.3 THE WIDTH CONSTRUCTION

A.3.1 THE AFFINE BASELINE AND ITS RESIDUAL

Fix the time and output patch from Sec. 3.3. The noisy-patch covariance is

$$\operatorname{Cov}(X_{t,m}) = a_t^2 \operatorname{Cov}(X_{1,m}) + b_t^2 I_{d_p} \succ 0.$$

The population denoiser is a conditional expectation, so it has finite second moment. Its optimal affine predictor from $X_{t,m}$ is consequently well-defined. Writing $\ell_t(x) = Lx + u$, minimization over the intercept gives

$$u = \mathbb{E} D^*_{m,t}(Y_t) - L \mathbb{E} X_{t,m}.$$

The normal equations for the centered variables give

$$L \operatorname{Cov}(X_{t,m}) = \operatorname{Cov}(D^*_{m,t}(Y_t), X_{t,m}),$$

so

$$L = \operatorname{Cov}(D^*_{m,t}(Y_t), X_{t,m}) \operatorname{Cov}(X_{t,m})^{-1}.$$

The residual $R_t = D^*_{m,t}(Y_t) - \ell_t(X_{t,m})$ therefore has zero mean and satisfies

$$\operatorname{Cov}(R_t, X_{t,m}) = \operatorname{Cov}(D^*_{m,t}(Y_t), X_{t,m}) - L \operatorname{Cov}(X_{t,m}) = 0.$$

This orthogonality concerns covariance, not independence.

If $Wh(Y_t) + b = X_{t,m}$, then the same representation provides

$$\ell_t(X_{t,m}) = LWh(Y_t) + (Lb + u).$$

Thus the best local affine denoising component is already available through an affine readout. No additional feature coordinates or separate representability assumption are needed for this part. The remaining target R_t is a measurable function of the whole observation, since both $D^*_{m,t}(Y_t)$ and $X_{t,m}$ are such functions.

A.3.2 PROOF OF THM. 5

Proof. The affine form of the induced clean estimate is equivalent to an affine noise output under Assump. 1. If $f_{\varepsilon_m,t}(Y_t) = Mh(Y_t) + v$, then

$$\widehat{X}_{1,m}(Y_t) = \frac{X_{t,m} - b_t f_{\varepsilon_m,t}(Y_t)}{a_t} = \frac{W - b_t M}{a_t} h(Y_t) + \frac{b - b_t v}{a_t}.$$

Thus $C = (W - b_t M)/a_t$ and $c = (b - b_t v)/a_t$. Conversely, any C, c correspond to the affine noise output with $M = (W - a_t C)/b_t$ and $v = (b - a_t c)/b_t$. Optimizing over C, c therefore optimizes the induced clean estimate over all affine noise output layers on the same representation.

The input-recovery identity $Wh(Y_t) + b = X_{t,m}$ also implies

$$\mathrm{Cov}(X_{t,m}) = W \,\mathrm{Cov}(h(Y_t))W^\top.$$

The left side has rank d_p. Hence

$$d_p \le \mathrm{rank}(W) \le \min\{d_p, d_h\},$$

which makes recovery impossible for $d_h < d_p$. For every feasible representation, W has full row rank and its kernel has dimension $d_h - d_p$.

For the lower bound, center h, $X_{t,m}$, and the denoiser and absorb the means into the affine intercepts. We use the same symbols for the centered variables in this part of the argument. The best affine baseline becomes $LX_{t,m}$ and the target is $LX_{t,m} + R_t$.

Because W has full row rank, its right inverse $W^\dagger = W^\top(WW^\top)^{-1}$ satisfies $WW^\dagger = I_{d_p}$. Thus

$$h = W^\dagger X_{t,m} + (I - W^\dagger W)h.$$

The second term belongs to $\ker W$. Choose a matrix N whose columns are an orthonormal basis of that kernel, and set $g(Y_t) = N^\top h(Y_t)$. Then

$$h = W^\dagger X_{t,m} + Ng(Y_t), \qquad Ch = KX_{t,m} + Bg(Y_t),$$

where $K = CW^\dagger$ and $B = CN$. In particular,

$$\mathrm{rank}(B) \le d_h - d_p.$$

For $d_h = d_p$, the kernel component is absent. The argument permits arbitrary dependence of g on the whole noisy image; no independence between input and additional features is assumed.

To bound the remaining error, let $S = \mathrm{col}(B)$, with orthogonal projections P_S and $P_{S^\perp} = I - P_S$. Projecting the denoiser approximation error eliminates the additional-feature term:

$$P_{S^\perp}\{Ch - D^*_{m,t}(Y_t)\} = P_{S^\perp}\{(K - L)X_{t,m} - R_t\}.$$

The expected cross term is zero because $\mathrm{Cov}(R_t, X_{t,m}) = 0$. Therefore

$$\begin{aligned}\mathbb{E}\|Ch - D^*_{m,t}(Y_t)\|^2 &\ge \mathbb{E}\|P_{S^\perp}\{(K - L)X_{t,m} - R_t\}\|^2 \\ &= \mathbb{E}\|P_{S^\perp}(K - L)X_{t,m}\|^2 + \mathbb{E}\|P_{S^\perp}R_t\|^2 \\ &\ge \mathrm{tr}\left(P_{S^\perp}\,\mathrm{Cov}(R_t)\right).\end{aligned}$$

Allowing a nonzero mean error can only add its squared norm, so this lower bound applies to the original affine predictors as well.

Let $k = d_h - d_p$ and let $\lambda_1 \ge \cdots \ge \lambda_{d_p} \ge 0$ be the eigenvalues of $\mathrm{Cov}(R_t)$, with orthonormal eigenvectors $q_1, \ldots, q_{d_p}$. If $k < d_p$, every subspace S of dimension at most k satisfies

$$\mathrm{tr}\left(P_S\,\mathrm{Cov}(R_t)\right) = \sum_{j=1}^{d_p} \lambda_j \|P_S q_j\|^2 \le \sum_{j=1}^{k} \lambda_j.$$

To see the last inequality, each weight $\|P_S q_j\|^2$ lies in $[0, 1]$, and their sum is $\mathrm{tr}(P_S) \le k$. The weighted sum of decreasing nonnegative eigenvalues is therefore at most the sum of the first k. Subtracting from the total trace gives

$$\mathrm{tr}\left(P_{S^\perp}\,\mathrm{Cov}(R_t)\right) \ge \sum_{j>k} \lambda_j.$$

For $k \ge d_p$ the right side is the empty sum, equal to zero. The same lower bound is valid in every case, proving the necessary approximation error in Eq. (11).

To attain this bound, return to the original, uncentered variables and let Q contain the leading $\min\{k, \operatorname{rank}(\operatorname{Cov}(R_t))\}$ eigenvectors with positive eigenvalues, and define

$$h(Y_t) = \big(X_{t,m},\ Q^\top R_t,\ 0\big) \in \mathbb{R}^{d_h},$$

where the final zero block pads unused coordinates. Read the noisy patch from the first d_p coordinates and define the induced clean estimate as

$$\widehat{X}_{1,m}(Y_t) = \ell_t(X_{t,m}) + QQ^\top R_t.$$

This is an affine function of h: the first block uses L, the second uses Q, and the intercept is u from the affine baseline. Its error is

$$\widehat{X}_{1,m}(Y_t) - D^*_{m,t}(Y_t) = -(I - QQ^\top)R_t.$$

Consequently,

$$\mathbb{E}\|\widehat{X}_{1,m}(Y_t) - D^*_{m,t}(Y_t)\|^2 = \sum_{j>k} \lambda_j\big(\operatorname{Cov}(R_t)\big).$$

All feature blocks have finite second moments and are measurable functions of Y_t. The construction uses fixed readout matrices across all images and all values of the other patches, so it belongs to the stated class and attains the lower bound.

The connection to clean-image risk follows from Eq. (12): with target $X_{1,m}$ and observation Y_t, it gives

$$\begin{aligned}\mathbb{E}\|\widehat{X}_{1,m}(Y_t) - X_{1,m}\|^2 = \mathbb{E}\|D^*_{m,t}(Y_t) - X_{1,m}\|^2 \\ + \mathbb{E}\|\widehat{X}_{1,m}(Y_t) - D^*_{m,t}(Y_t)\|^2.\end{aligned}$$

The first term is independent of the representation. This relates approximation of the population denoiser to the induced clean-image loss. □

A.3.3 Proof of Cor. 6

Proof. For width $d_h = d_p + k$, Thm. 5 identifies the attained minimum error as $\sum_{j>k} \lambda_j(\operatorname{Cov}(R_t))$. This sum is nonincreasing in k, and it is zero by $k = d_p$. Thus, for every $\delta \geq 0$, the set

$$\left\{ k \in \{0, \dots, d_p\} : \sum_{j>k} \lambda_j(\operatorname{Cov}(R_t)) \leq \delta \right\}$$

is nonempty. Its minimum determines the smallest feasible width at tolerance δ. A smaller width either cannot recover the noisy patch or has an optimal approximation error larger than δ. The construction in the theorem attains the stated tolerance at the proposed width.

For $\delta = 0$, all positive residual eigenvalues must be retained. This gives the exact simultaneous-recovery width

$$d_h = d_p + \operatorname{rank}\big(\operatorname{Cov}(R_t)\big).$$

In particular, if $R_t = 0$ almost surely, width d_p suffices; no extra coordinates are required in that case. □

B Input-compression and output-preconditioning experiments

We provide the coefficient details and complete results supporting the main-text tests of input compression and output preconditioning. Additional input-access interventions are included as supplementary experiments, followed by the full layerwise probe tables.

B.1 Training, sampling, and numerical conventions

Coordinates and target conversion. Using the Gaussian noise convention of Sec. 2, the experiments specialize to $a_t = t$ and $b_t = 1 - t$, with $0 < t < 1$:

$$X_t = tX_1 + (1-t)\varepsilon, \qquad Y_t = X_t/t = X_1 + \sigma_t\varepsilon, \qquad \sigma_t = (1-t)/t. \tag{15}$$

Small t therefore corresponds to high noise. A noise prediction induces the clean estimate

$$\widehat{X}_1 = \frac{X_t - (1-t)\widehat{\varepsilon}}{t} = Y_t - \sigma_t\widehat{\varepsilon}. \tag{16}$$

For bare noise prediction, $\widehat{\varepsilon} = f_{\varepsilon,t}(AY_t)$; path-velocity prediction uses $\dot{X}_t = X_1 - \varepsilon$ and $\widehat{X}_1 = X_t + (1-t)f_{\dot{X},t}(AY_t)$. For the $768 \to 128 \to 768$ embedding, $d_p = 768$ and $r_p \le 128$; with M patches, $d = Md_p$ and $r = Mr_p$. The encoder A is each trained model's linear compression, held fixed during evaluation. Embedding biases and positional offsets are deterministic and can be absorbed into the decoder. Class labels are observed, so the population conditional means in the experimental discussion include the label; corruption noise remains independent of the clean image and label.

Training and evaluation. The bottleneck baselines and output-shortcut runs use JiT-B/16 with 12 blocks, hidden width 768, and a $768 \to 128 \to 768$ embedding for each 16×16 RGB patch. Recorded configurations specify unit Gaussian noise, $\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$, training seed 0, label-drop probability 0.1, and zero attention and projection dropout. Images are center-cropped and randomly flipped, then normalized to $[-1, 1]$. The effective batch size is 1024 (512 on each of two devices). AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$ with zero weight decay; the learning rate warms up for five epochs to 2×10^{-4} and then stays constant.

The image-loss implementation uses the same clamped conversion convention specified in Eq. (23), with $\tau = 0.05$. Path-velocity prediction supplies the raw velocity output directly; noise shortcuts are applied before conversion to velocity. Consequently, the objective agrees with clean-image squared error when $\tau \le t \le 1 - \tau$, while endpoint clamps modify that identity. The exact shortcut itself divides by $\max(1-t, \tau)$ in place of $1-t$; learned exact initialization uses the same clamped baseline. The unclamped formulas in the main text identify the analytic mechanism, while the reported FID values describe these numerical implementations.

Generation uses the Heun sampler with 50 steps, an Euler final step, and 10,000 images. Guidance scale 2.9 is active over the configured interval $[0.1, 1.0]$. Evaluation uses the first exponential moving average (EMA), with per-update decay 0.9999, and the JiT ImageNet256 FID reference statistics. The baseline and shortcut table follows the reported checkpoint labels $0, 40, \ldots, 200$; the probe logs identify the stored checkpoint epoch as 160. The width suite instead counts completed training epochs explicitly. Each configuration is represented by one training run, so small FID differences do not establish statistical significance.

These settings are supported by run configurations and the inspected implementation. A historical source snapshot and a per-evaluation sampling-seed record are not available for every entry. The supplementary direct-bypass FID run also lacks a confirmed identifier linking it to the earlier probed checkpoint; its measurements remain separate below.

B.2 Output coefficients and complete FID trajectories

Normalized output and cancellation. The output-shortcut experiments retain the compressed learned branch and use

$$\widehat{\varepsilon} = c_{\mathrm{skip}} Y_t + c_{\mathrm{out}} f(c_{\mathrm{in}} AY_t), \qquad c_{\mathrm{in}} = t, \tag{17}$$

Here $c_{\mathrm{in}} AY_t = AX_t$, and time and class conditioning are implicit in f. For the noise target, $(\hat{a}_t, \hat{b}_t) = (0, 1)$, the discarded-noise term in Thm. 3 is $(1 - c_{\mathrm{skip}}\sigma_t)^2(d-r)$. Choosing $c_{\mathrm{skip}} = 1/\sigma_t$ and $c_{\mathrm{out}} = -1/\sigma_t$ makes Eq. (16) reduce to $\widehat{X}_1 = f(c_{\mathrm{in}} AY_t)$: the raw branch predicts the clean image from the compressed input. These identities hold at interior times in exact arithmetic; the implemented endpoint clamps are specified in Appendix B.1.

Coordinates and output coefficients. If a shortcut is implemented in the unnormalized coordinates as $\widehat{\varepsilon} = u(t)X_t + w(t)f(c_{\mathrm{in}} AY_t)$, then the normalized coefficients in Eq. (17) are $c_{\mathrm{skip}} = tu$ and $c_{\mathrm{out}} = w$, with $c_{\mathrm{in}} = t$. The functions u, w describe implementation coefficients only; a_t, b_t retain their corruption-path meanings. Tab. 4 lists the fixed choices. The exact noise shortcut is an instance of the cancellation construction in Thm. 3; it is distinct from the EDM coefficients discussed in Sec. 3.2.

Table 4: Shortcut coefficients in implementation and normalized coordinates.

Variant	u	w	c_{skip}	c_{out}
Bare noise prediction	0	1	0	1
Additive	1	1	t	1
Exact	$1/(1-t)$	$-t/(1-t)$	$1/\sigma_t$	$-1/\sigma_t$

What the additive shortcut removes. The additive choice $\widehat{\varepsilon} = X_t + f(c_{\text{in}} A Y_t)$ has residual target $\varepsilon - X_t = t(\varepsilon - X_1)$, which still contains noise. For noise prediction, the discarded-noise contribution in Thm. 3 is

$$(1 - c_{\text{skip}}\sigma_t)^2(d - r).$$

For a noise target, Eq. (9) gives $\tilde{c}_{\text{skip}} = 1 - \sigma_t c_{\text{skip}}$ and $\tilde{c}_{\text{out}} = -\sigma_t c_{\text{out}}$. Thus Cor. 4 gives the discarded-noise contribution $\tilde{c}_{\text{skip}}^2 \sigma_t^2 (d - r)$. Consequently, the contributions for bare, additive, and exact shortcuts are

$$\begin{aligned} \text{bare:} \quad & \sigma_t^2(d - r), \\ \text{additive:} \quad & \sigma_t^2(1 - t\sigma_t)^2(d - r) = (1 - t)^2(d - r), \\ \text{exact:} \quad & 0. \end{aligned} \tag{18}$$

The additive shortcut therefore removes the high-noise divergence of this term, but does not eliminate the term itself. The factors governing conversion of arbitrary prediction errors and those governing the unavoidable discarded-noise error should not be conflated.

Learning the coefficients. An MLP with eight sine/cosine frequency pairs, hidden width 64, and a SiLU activation produces residual coefficients $\delta_u(t), \delta_w(t)$. Its final layer is zero-initialized, giving

$$\begin{aligned} u(t) &= u_{\text{base}}(t) + \delta_u(t), & w(t) &= w_{\text{base}}(t) + \delta_w(t), \\ c_{\text{skip}}(t) &= t\{u_{\text{base}}(t) + \delta_u(t)\}, & c_{\text{out}}(t) &= w_{\text{base}}(t) + \delta_w(t). \end{aligned} \tag{19}$$

The three base pairs are $(0, 1)$, $(1, 1)$, and $(1/(1 - t), -t/(1 - t))$. The analytic identities above apply at interior times; the implementation clamps denominators as described in Appendix B.1. FID alone does not identify the learned coefficient functions: in particular, rescaling $f(c_{\text{in}} A Y_t)$ can be absorbed by $c_{\text{out}}(t)$. We therefore interpret the learned runs as evidence that an adaptive output shortcut is effective, without asserting that its coefficients uniquely recover the analytic pair.

Table 5: Full reported FID-10K trajectories for prediction parameterizations and output shortcuts. Guidance scale is 2.9.

Variant	0	40	80	120	160	200
Image	370.79	55.35	17.23	11.92	9.87	9.13
Noise	411.34	412.73	411.86	411.10	411.79	412.11
Path velocity	410.82	198.03	178.00	178.34	177.92	177.20
Additive	410.77	211.45	181.42	179.88	181.78	181.09
Exact	370.60	55.04	18.65	12.42	10.23	9.15
Learned (noise init.)	411.33	71.64	27.04	20.42	16.71	14.99
Learned (additive init.)	410.62	65.50	24.99	20.04	16.44	15.13
Learned (exact init.)	373.69	52.03	17.38	11.88	10.96	9.56

B.3 Additional input-access experiments

Injection into the hidden stream. Using the patch notation of Sec. 3.3, let $h_{i,m} \in \mathbb{R}^{d_h}$ denote the representation of image patch m after block i. We concatenate it with the uncompressed noisy patch $X_{t,m} \in \mathbb{R}^{d_p}$:

$$h_{i,m} \leftarrow W_i^{\text{inj}} \begin{bmatrix} h_{i,m} \\ X_{t,m} \end{bmatrix}, \qquad W_i^{\text{inj}} \in \mathbb{R}^{d_h \times (d_h + d_p)}. \tag{20}$$

The map is initialized as $W_i^{\text{inj}} = [I_{d_h}\ 0]$, so injection is inactive initially. The same projection is applied to every image patch at that layer. This intervention makes the raw patches available after the input embedding; the resulting network is no longer restricted to a function of AX_t. However, access to raw coordinates does not by itself guarantee that training learns to preserve them through the remaining computation.

A direct bypass after the output head. A second variant adds a time-dependent copy of the raw patches after the output head, without routing this contribution through the hidden-stream normalization layers. In image coordinates, the reported construction is

$$\widehat{\varepsilon} = f(c_{\text{in}} A Y_t) + \left(\sum_{i=1}^{12} \gamma_i(t) \right) X_t. \tag{21}$$

The bypass coefficients are initialized at zero. For scalar $\gamma_i(t)$, this is algebraically an output shortcut with $c_{\text{skip}} = t\sum_i \gamma_i(t)$ and $c_{\text{out}} = 1$. It therefore supplies a direct intact-input path; the sum over layers does not itself establish a distinct noise-recovery mechanism.

Table 6: Additional input-access variants: full reported FID-10K trajectories. Guidance scale is 2.9. All three runs are reported through epoch 200.

Variant	0	40	80	120	160	200
Bare noise prediction	411.34	412.73	411.86	411.10	411.79	412.11
Hidden-stream injection	411.33	303.88	254.46	225.63	195.76	188.89
Direct output bypass	411.31	71.14	27.76	20.02	16.28	15.81

Hidden-stream injection improves substantially over bare noise prediction, but its FID remains 188.89 at epoch 200. The updated direct-bypass run reaches 16.28 at epoch 160 and 15.81 at epoch 200. The earlier bypass checkpoint's probe scores are retained in Tab. 15 under the legacy $[0, 1]$ protocol described in Appendix B.4.2. They are separate from the corrected baseline probes, and correspondence to the updated bypass FID run has not been established. These experiments support the utility of direct output access, but do not isolate normalization as the cause of the remaining gap: injection and bypass also differ in their output parameterization and optimization.

B.4 Full layerwise probe results

Recovery metric and population ceiling. At fixed t, affine probes recover the clean image X_1, sampled noise ε, or noisy input X_t from each frozen layer. For target T, the population recovery score is

$$\mathrm{EV}[T] = 1 - \frac{\mathbb{E}\|\widehat{T} - T\|^2}{\mathbb{E}\|T - \mathbb{E}T\|^2}, \tag{22}$$

and is estimated on evaluation data. For standard Gaussian noise, the denominator is d when reconstructed patches are concatenated into a whole image. Thm. 1 gives $\mathrm{EV}[\varepsilon] \le r/d = r_p/d_p \le 1/6$ for representations depending on the noisy image only through the bottleneck; the last inequality is an equality for a full-row-rank 128-dimensional patch projection. Patch-averaged errors give the same score when target variances are pooled consistently. The population noise ceiling is unchanged by the clean-image scale, provided the stated independence and moment assumptions hold. Low affine-probe scores do not exclude information accessible to nonlinear readouts. The unrecovered noise components contribute clean-image error amplified by σ_t^2, as in Cor. 2.

B.4.1 Probes with training-consistent inputs

The image- and noise-prediction probes in Sec. 4.2 use matched JiT-B/16 checkpoints with stored epoch counter 160 (161 completed training epochs), taking the first moving-average weights with decay 0.9999. Both use the $768 \to 128 \to 768$ embedding. We evaluate $t \in \{0.1, 0.9\}$ at all layers 0–12 for the targets X_1, ε, and X_t. Tabs. 7 and 8 give the complete results, rounded from the saved arrays to four decimal places. These runs replace the earlier baseline probes using $[0, 1]$ images; corrected runs are not available here for the other times or the other variants.

Images are center-cropped to 256×256 without random flips and normalized to $[-1, 1]$, matching the backbone's training scale. A single ImageNet training-set permutation with seed 20260924 supplies 8000 fitting images followed by 2000 disjoint evaluation images. Both models use the same indices, labels, and corruption draws. Independent fitting and evaluation Gaussian streams use seeds 117 and 991, respectively. The batch size is 64. The evaluation images are held out from probe fitting, but not from backbone training.

For each layer, time, and target, we fit an affine ridge decoder pooling all patch positions with a shared intercept. The ridge coefficient is 10^{-3} times the mean diagonal of the feature Gram matrix; the intercept receives only a 10^{-8} diagonal term for numerical stability. Sufficient statistics and readout calculations use FP64; backbone inference uses BF16 autocast. The recovery-score denominator is centered separately for each within-patch coordinate over the pooled evaluation patches. Layer 0 is the input to the first block, including embedding and positional terms, and subsequent layers are block outputs with class-context tokens removed. Layer 12 precedes final-head processing. Strict checkpoint loading establishes architectural compatibility with the implementation used for probing, but does not reconstruct every detail of the original training source revision.

The paired runs correct preprocessing and enforce disjoint probe splits while retaining the original affine-readout definition. They remain finite-sample diagnostics of frozen representations. Rounded or finite-sample noise scores near $1/6$ do not establish a population-bound violation, and the probes do not determine whether the native output head uses all recoverable information. No new generation evaluation is associated with these probes.

Table 7: Image prediction: full layerwise recovery with $[-1, 1]$ inputs, matched checkpoint 160, and disjoint probe splits.

	$\mathrm{EV}[X_1]$		$\mathrm{EV}[\varepsilon]$		$\mathrm{EV}[X_t]$	
Layer	.1	.9	.1	.9	.1	.9
0	0.5005	0.9803	0.1648	0.0203	0.1697	0.9538
1	0.7179	0.9809	0.1652	0.0213	0.1693	0.9543
2	0.7384	0.9814	0.1644	0.0231	0.1684	0.9546
3	0.7463	0.9816	0.1628	0.0245	0.1668	0.9546
4	0.7501	0.9821	0.1591	0.0236	0.1632	0.9546
5	0.7507	0.9820	0.1576	0.0240	0.1616	0.9545
6	0.7516	0.9820	0.1556	0.0238	0.1597	0.9544
7	0.7521	0.9819	0.1533	0.0227	0.1574	0.9543
8	0.7531	0.9819	0.1511	0.0218	0.1551	0.9543
9	0.7555	0.9818	0.1491	0.0211	0.1532	0.9542
10	0.7593	0.9818	0.1471	0.0207	0.1512	0.9541
11	0.7654	0.9827	0.1360	0.0246	0.1401	0.9532
12	0.7689	0.9860	0.0867	0.0208	0.0908	0.9533

Table 8: Noise prediction: full layerwise recovery with $[-1, 1]$ inputs, matched checkpoint 160, and disjoint probe splits.

	$\mathrm{EV}[X_1]$		$\mathrm{EV}[\varepsilon]$		$\mathrm{EV}[X_t]$	
Layer	.1	.9	.1	.9	.1	.9
0	0.0042	0.0107	0.1666	0.1383	0.1660	0.0179
1	0.0075	0.0211	0.1666	0.1381	0.1660	0.0279
2	0.0066	0.0217	0.1666	0.1383	0.1660	0.0284
3	0.0065	0.0232	0.1666	0.1383	0.1660	0.0299
4	0.0067	0.0223	0.1666	0.1383	0.1660	0.0290
5	0.0072	0.0224	0.1666	0.1383	0.1660	0.0291
6	0.0087	0.0226	0.1666	0.1383	0.1660	0.0293
7	0.0076	0.0225	0.1666	0.1383	0.1660	0.0292
8	0.0059	0.0233	0.1666	0.1383	0.1660	0.0300
9	0.0061	0.0241	0.1666	0.1383	0.1660	0.0307
10	0.0056	0.0245	0.1666	0.1383	0.1660	0.0311
11	0.0058	0.0252	0.1666	0.1383	0.1660	0.0318
12	0.0047	0.0237	0.1666	0.1393	0.1660	0.0303

B.4.2 Legacy probes with unmatched preprocessing

The remaining seven tables retain the earlier path-velocity, shortcut, and bypass probes as preliminary supplementary records. They use $[0, 1]$ images although backbone training uses $[-1, 1]$, and are separate from the corrected results above. They provide preliminary diagnostics under the probed input distribution and are not directly comparable with the corrected scores. Their checkpoint logs identify the first moving-average weights and counter 160; the bypass checkpoint's correspondence to the updated FID run remains unverified. The five times are $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with all layers 0–12 and all three targets. Scores retain the logged precision of three decimal places.

The available launch script specifies 8000 fitting images, 2000 evaluation images, and batch size 64. Its image subsets are independently drawn from ImageNet train using seeds 0 and 1, without enforced disjointness, and corruption draws are not explicitly seeded. The implementation uses pooled affine ridge fits with coefficient 10^{-3}, an effectively unpenalized shared intercept, FP64 sufficient statistics, and BF16 activation extraction. The logs reproduce the listed scores but do not archive the precise source revision or sampled indices. These records therefore do not establish

held-out recovery under the training preprocessing. The width-readout study in Appendix C.3 uses its own disjoint probe split and training-consistent normalization.

The population noise ceiling applies to trunk representations that see the image only through the bottleneck. It does not apply to a complete shortcut predictor that also receives the intact noisy input. In particular, the direct-bypass table measures the trunk rather than the sum of its output and the bypass contribution. No nonaffine-probe comparison is included in these records.

Table 9: Legacy $[0, 1]$ probe. Path-velocity prediction: full layerwise recovery scores at epoch 160.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.255	.692	.826	.887	.909	.166	.163	.160	.155	.099	.167	.177	.220	.386	.801
1	.603	.796	.863	.897	.915	.166	.165	.162	.152	.060	.167	.177	.220	.383	.800
2	.627	.818	.878	.906	.920	.166	.165	.163	.155	.061	.167	.177	.220	.385	.804
3	.634	.823	.881	.908	.922	.166	.165	.163	.156	.071	.167	.177	.220	.386	.807
4	.637	.826	.882	.909	.922	.166	.165	.163	.157	.077	.167	.177	.220	.387	.808
5	.638	.827	.883	.909	.922	.166	.165	.163	.157	.079	.167	.177	.220	.387	.809
6	.638	.827	.883	.909	.922	.166	.165	.163	.158	.082	.167	.177	.220	.387	.809
7	.639	.827	.883	.909	.922	.166	.165	.163	.157	.080	.167	.177	.220	.386	.809
8	.639	.828	.883	.910	.923	.166	.165	.163	.156	.076	.167	.177	.220	.386	.809
9	.641	.830	.884	.910	.923	.166	.165	.163	.153	.064	.167	.177	.219	.383	.807
10	.644	.834	.888	.914	.926	.166	.164	.160	.141	.045	.167	.176	.217	.376	.807
11	.647	.838	.892	.919	.931	.166	.163	.155	.126	.032	.167	.175	.212	.367	.809
12	.648	.838	.892	.918	.932	.161	.148	.132	.116	.090	.162	.159	.191	.358	.817

Table 10: Legacy $[0, 1]$ probe. Additive output shortcut: full layerwise recovery scores at epoch 160.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.256	.692	.826	.887	.909	.166	.164	.160	.156	.114	.167	.177	.220	.386	.803
1	.599	.798	.864	.899	.916	.166	.165	.163	.156	.081	.167	.177	.220	.386	.804
2	.625	.818	.877	.906	.920	.166	.165	.163	.158	.089	.167	.177	.220	.388	.809
3	.632	.823	.881	.908	.921	.166	.165	.164	.159	.099	.167	.177	.220	.388	.811
4	.636	.826	.882	.909	.922	.166	.165	.164	.160	.108	.167	.177	.220	.389	.813
5	.637	.826	.882	.909	.922	.166	.165	.164	.160	.106	.167	.177	.220	.389	.813
6	.637	.826	.882	.909	.922	.166	.165	.164	.160	.107	.167	.177	.220	.388	.813
7	.637	.826	.882	.909	.922	.166	.165	.164	.159	.105	.167	.177	.220	.388	.812
8	.640	.829	.883	.910	.923	.166	.165	.163	.157	.093	.167	.177	.219	.386	.811
9	.642	.832	.886	.911	.924	.166	.165	.163	.155	.085	.167	.177	.219	.385	.811
10	.645	.835	.889	.914	.927	.166	.164	.159	.146	.069	.167	.175	.216	.380	.811
11	.648	.839	.894	.920	.932	.165	.162	.157	.144	.070	.166	.174	.214	.379	.815
12	.645	.837	.892	.918	.933	.130	.140	.140	.140	.131	.131	.152	.198	.375	.824

Table 11: Legacy $[0, 1]$ probe. Exact output preconditioning: full layerwise recovery scores at epoch 160.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.256	.694	.834	.911	.969	.166	.162	.155	.134	.050	.167	.178	.225	.409	.870
1	.598	.797	.872	.920	.970	.166	.163	.157	.136	.054	.167	.178	.225	.409	.872
2	.623	.818	.887	.929	.972	.166	.163	.157	.138	.057	.167	.177	.224	.408	.873
3	.633	.827	.894	.934	.972	.165	.163	.157	.139	.059	.166	.176	.224	.408	.873
4	.638	.832	.899	.939	.974	.163	.159	.152	.132	.057	.164	.173	.219	.403	.873
5	.639	.833	.899	.939	.974	.162	.158	.151	.130	.058	.163	.172	.219	.402	.873
6	.639	.834	.900	.939	.974	.161	.157	.150	.128	.057	.162	.171	.217	.401	.873
7	.640	.835	.900	.939	.974	.160	.156	.148	.125	.055	.161	.170	.216	.400	.872
8	.641	.835	.900	.939	.974	.159	.155	.146	.124	.053	.160	.169	.215	.399	.872
9	.643	.837	.901	.939	.974	.158	.154	.145	.122	.052	.159	.168	.214	.398	.872
10	.645	.840	.903	.940	.974	.157	.152	.144	.121	.051	.158	.166	.212	.397	.872
11	.648	.847	.912	.949	.977	.152	.141	.130	.111	.053	.153	.155	.198	.385	.868
12	.650	.850	.916	.954	.980	.109	.094	.084	.071	.032	.110	.107	.153	.349	.860

Table 12: Legacy $[0, 1]$ probe. Learned output shortcut, noise initialization: full layerwise recovery scores at epoch 160.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.257	.694	.834	.911	.969	.166	.162	.155	.133	.048	.167	.178	.225	.409	.869
1	.601	.795	.869	.920	.971	.166	.164	.157	.136	.056	.167	.178	.225	.409	.873
2	.625	.818	.887	.929	.971	.166	.163	.157	.139	.059	.167	.177	.224	.409	.874
3	.633	.827	.894	.933	.972	.165	.163	.157	.140	.062	.166	.177	.224	.408	.873
4	.638	.832	.897	.936	.973	.164	.162	.157	.140	.065	.165	.176	.223	.408	.873
5	.639	.833	.898	.936	.972	.163	.161	.156	.139	.065	.164	.175	.223	.407	.873
6	.639	.833	.898	.936	.973	.162	.161	.155	.138	.064	.163	.174	.222	.407	.873
7	.640	.834	.898	.935	.972	.162	.160	.155	.137	.063	.163	.174	.222	.407	.873
8	.642	.836	.899	.936	.972	.161	.159	.154	.137	.062	.162	.173	.221	.406	.873
9	.644	.839	.902	.938	.972	.160	.159	.154	.137	.059	.161	.172	.221	.406	.871
10	.647	.845	.909	.945	.973	.156	.153	.147	.131	.058	.157	.167	.214	.400	.871
11	.650	.849	.914	.950	.974	.144	.134	.124	.109	.058	.145	.148	.193	.383	.871
12	.648	.850	.917	.955	.972	.070	.061	.055	.049	.100	.071	.075	.126	.334	.865

Table 13: Legacy $[0, 1]$ probe. Learned output shortcut, additive initialization: full layerwise recovery scores at epoch 160.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.256	.694	.834	.911	.969	.166	.162	.155	.134	.050	.167	.178	.225	.409	.871
1	.597	.798	.872	.921	.968	.166	.163	.157	.136	.046	.167	.178	.225	.407	.867
2	.625	.821	.890	.931	.971	.165	.162	.156	.137	.059	.166	.176	.223	.408	.874
3	.633	.828	.895	.935	.972	.163	.161	.155	.138	.063	.164	.175	.222	.408	.874
4	.637	.832	.898	.936	.972	.162	.160	.154	.138	.064	.163	.173	.221	.407	.874
5	.638	.832	.897	.936	.972	.161	.159	.153	.137	.064	.162	.173	.221	.407	.874
6	.639	.833	.898	.936	.973	.160	.158	.152	.136	.064	.161	.172	.220	.407	.873
7	.640	.833	.897	.935	.972	.159	.157	.152	.135	.062	.160	.171	.220	.406	.873
8	.642	.836	.899	.936	.972	.158	.157	.151	.135	.062	.159	.171	.219	.406	.873
9	.644	.840	.903	.938	.972	.157	.156	.151	.135	.062	.158	.170	.218	.405	.873
10	.647	.845	.909	.945	.972	.152	.150	.145	.132	.059	.153	.164	.212	.401	.871
11	.648	.850	.914	.950	.973	.141	.135	.127	.117	.065	.142	.149	.195	.388	.872
12	.646	.850	.916	.954	.972	.078	.072	.067	.063	.097	.079	.086	.136	.342	.863

Table 14: Legacy $[0, 1]$ probe. Learned output shortcut, exact initialization: full layerwise recovery scores at epoch 160.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.255	.693	.834	.911	.969	.166	.162	.155	.134	.049	.167	.178	.225	.409	.870
1	.597	.798	.872	.921	.970	.166	.163	.157	.136	.055	.167	.178	.225	.409	.873
2	.623	.818	.887	.930	.972	.165	.163	.157	.139	.059	.166	.177	.224	.409	.873
3	.632	.827	.894	.934	.973	.164	.162	.157	.139	.061	.165	.176	.224	.408	.873
4	.637	.832	.898	.938	.974	.162	.160	.155	.136	.060	.163	.174	.222	.406	.873
5	.637	.833	.898	.938	.974	.161	.159	.154	.135	.060	.162	.173	.221	.405	.873
6	.638	.834	.899	.938	.974	.159	.158	.153	.133	.059	.161	.172	.220	.404	.873
7	.639	.834	.899	.938	.974	.158	.157	.151	.131	.057	.159	.171	.219	.403	.873
8	.639	.835	.899	.938	.974	.157	.156	.150	.130	.055	.158	.170	.218	.403	.872
9	.641	.837	.900	.938	.974	.155	.155	.149	.129	.054	.156	.169	.217	.402	.872
10	.644	.841	.904	.940	.974	.154	.154	.148	.128	.053	.155	.168	.216	.401	.872
11	.647	.847	.912	.949	.977	.145	.140	.132	.115	.052	.146	.154	.200	.387	.868
12	.647	.850	.916	.954	.980	.086	.090	.087	.075	.031	.087	.104	.155	.352	.860

Table 15: Legacy $[0, 1]$ probe. Direct-bypass trunk: full layerwise recovery scores for the previously recorded epoch-160 checkpoint. Correspondence to the updated bypass FID run in Tab. 6 is not established.

	$\mathrm{EV}[X_1]$					$\mathrm{EV}[\varepsilon]$					$\mathrm{EV}[X_t]$				
Layer	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9	.1	.3	.5	.7	.9
0	.256	.694	.834	.911	.969	.166	.162	.155	.133	.048	.167	.178	.225	.409	.869
1	.598	.788	.865	.918	.970	.166	.163	.156	.135	.056	.167	.178	.225	.409	.873
2	.619	.813	.883	.927	.971	.166	.163	.157	.138	.057	.167	.177	.224	.409	.874
3	.632	.826	.893	.933	.972	.165	.163	.157	.140	.062	.166	.176	.224	.408	.874
4	.636	.831	.897	.936	.973	.163	.161	.156	.139	.063	.164	.175	.223	.408	.874
5	.639	.832	.897	.935	.973	.161	.160	.155	.138	.063	.162	.174	.222	.407	.874
6	.639	.833	.897	.935	.973	.159	.159	.155	.137	.063	.160	.173	.222	.407	.873
7	.641	.834	.897	.935	.972	.158	.158	.154	.137	.063	.159	.172	.221	.407	.873
8	.642	.835	.898	.935	.972	.157	.158	.153	.136	.062	.158	.172	.221	.407	.873
9	.644	.839	.901	.938	.973	.155	.156	.153	.136	.062	.156	.170	.220	.406	.872
10	.647	.845	.909	.946	.973	.152	.151	.146	.133	.061	.153	.165	.214	.402	.871
11	.649	.849	.914	.950	.974	.141	.138	.133	.122	.064	.142	.152	.200	.392	.872
12	.644	.849	.917	.955	.969	.063	.062	.060	.055	.089	.064	.076	.131	.338	.858

C Hidden-width experiments and frozen-readout protocol

C.1 Architecture, optimization, and generation

Comparison design. The width sweep uses ImageNet256 with 16×16 RGB patches ($d_p = 768$) and ImageNet128 with 8×8 patches ($d_p = 192$). Depth and the image-loss convention are held fixed while width varies. The empirical width-offset comparison pairs a noise predictor of width $d_p + q$ with an image predictor of width q, using the fixed geometric offset d_p. This is a heuristic extension of Thm. 5, which characterizes denoiser approximation under affine input recovery, not FID or the learned image predictor at width q. The experiments do not estimate the residual covariance spectrum or assert that the shifted FID curves must coincide.

Architecture and widths. All width experiments use the JiT backbone family (Li & He, 2025), 12 Transformer blocks, and an ordinary learned patch projection $\mathbb{R}^{d_p} \to \mathbb{R}^{d_h}$. Attention uses the full hidden width for its query, key, value, and output projections. No attention slicing, reduced attention width, output bypass, or per-layer input injection is used in this sweep. For the probed models, head dimension is 64, except for the width-128, 256, and 384 image predictors at resolution 256, which use head dimension 32. The number of heads is d_h divided by the head dimension. Each model has 256 image tokens; class-context tokens are excluded from patch readouts. The ordinary projection still compresses its input when $d_h < d_p$; removing the extra bottleneck does not make these small

image-prediction embeddings invertible. Width also changes parameter and head counts, so these are not parameter-matched comparisons.

At resolution 256 with patch size 16, image-prediction widths are 128, 256, 384, and 768, and noise-prediction widths are 768, 896, 1024, 1152, and 1536. At resolution 128 with patch size 8, image-prediction widths are 64, 128, 192, and 768, and noise-prediction widths are 192, 256, 320, 384, and 768. All configurations are evaluated after 40 epochs. We continue the paired widths and the resolution-128 width-192 noise predictor to 80 epochs. The resolution-256 width-768 noise predictor and the resolution-128 width-768 predictors have only 40-epoch evaluations in this comparison.

Training uses the ImageNet training set, with the experiment's center-crop preprocessing and random horizontal flips, normalized to $[-1, 1]$. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$ and zero weight decay. The effective batch size is 1024 across devices and gradient-accumulation steps. The learning rate warms up over five epochs to 2×10^{-4} and then stays constant. Training uses seed 0, label-drop probability 0.1, and zero attention and projection dropout. Corruption follows Eq. (15), with unit-scale Gaussian noise and $\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$.

Implemented image-loss convention. The implementation expresses the image-space objective as a weighted velocity loss and clamps conversion denominators at $\tau = 0.05$. Specifically, with $c_-(t) = \max(1 - t, \tau)$ and $c_+(t) = \max(t, \tau)$ and complete model outputs $f_{X_1,t}(Y_t)$ or $f_{\varepsilon,t}(Y_t)$, it uses

$$\begin{aligned} v_{\text{target}} &= (X_1 - X_t)/c_-(t), \\ \widehat{v}_{\text{image}} &= (f_{X_1,t}(Y_t) - X_t)/c_-(t), \\ \widehat{v}_{\varepsilon} &= (X_t - f_{\varepsilon,t}(Y_t))/c_+(t), \\ \mathcal{L} &= \mathbb{E}\big[(1-t)^2 \|\widehat{v} - v_{\text{target}}\|^2\big]. \end{aligned} \tag{23}$$

Here $\widehat{v}$ is the converted velocity for the corresponding prediction type. Away from the clamped endpoints, $v_{\text{target}} = \dot{X}_t = X_1 - \varepsilon$ and this loss equals clean-image squared error for both prediction types. Pixel dimensions are averaged in the code. We retain this convention across the width comparisons.

Generation uses Heun with 50 sampling steps, 10,000 images, and guidance scale 1. FID uses the corresponding resolution-specific ImageNet reference statistics. The sampling seed is 0; hardware migrations can change generation batch sizes and sample ordering, so generated samples across models are not paired. The experiments use one training seed per configuration.

Checkpoint accounting. Epoch counts refer to completed epochs: the 40-epoch checkpoint has zero-based epoch index 39. Two moving-average parameter sets are maintained with per-optimizer-update decay factors 0.9999 and 0.9996. In Fig. 1, decay 0.9996 is used for the resolution-256 noise predictors of widths 1152 and 1536 at epoch 40, and for the resolution-128 width-384 noise predictor at both epochs. All remaining generation entries use decay 0.9999, including the resolution-256 widths 1152 and 1536 at epoch 80. The completed frozen-readout study uses decay 0.9996 for the resolution-256 noise predictors of widths 1152 and 1536, and decay 0.9999 for its other checkpoints. This follows the checkpoints used in the original width plot. Consequently, the resolution-128 width-384 noise predictor's probe uses a different saved parameter average from its generation-figure point. Tab. 16 records the selected FID values without changing these parameter-average choices.

C.2 Why these readouts test the width mechanism

Fix the time and output patch. The population patch denoiser is $D^*_{m,t}(Y_t) = \mathbb{E}[X_{1,m} \mid Y_t, t]$. When class labels are observed, include them in the conditioning of the denoiser and hidden features; the identities below hold with expectations over the joint data distribution. Define the population affine clean-image readout risk by $\mathcal{L}_t(u) = \inf_{B,c} \mathbb{E}\|X_{1,m} - Bu - c\|^2$. For any features u measurable with respect to that observation, $X_{1,m} - D^*_{m,t}(Y_t)$ is orthogonal in L^2 to every affine readout of u. Consequently, for fixed readout coefficients B, c,

$$\mathbb{E}\|X_{1,m} - Bu - c\|^2 = \mathbb{E}\|X_{1,m} - D^*_{m,t}(Y_t)\|^2 + \mathbb{E}\|D^*_{m,t}(Y_t) - Bu - c\|^2.$$

Taking the infimum gives

$$\mathcal{L}_t(u) = \mathbb{E}\|X_{1,m} - D^*_{m,t}(Y_t)\|^2 + \inf_{B,c} \mathbb{E}\|D^*_{m,t}(Y_t) - Bu - c\|^2. \tag{24}$$

Table 16: Complete selected generation results for the width sweep (unguided FID-10K, lower is better). A dash denotes an unavailable checkpoint. The parameter-average choices are specified above.

Resolution / patch	Prediction	d_h	Epoch 40	Epoch 80
256 / 16	Image	128	183.10	138.99
256 / 16	Image	256	141.10	112.33
256 / 16	Image	384	131.80	96.33
256 / 16	Image	768	105.10	66.67
256 / 16	Noise	768	413.23	—
256 / 16	Noise	896	155.49	116.34
256 / 16	Noise	1024	149.23	109.11
256 / 16	Noise	1152	129.47	102.88
256 / 16	Noise	1536	105.49	86.92
128 / 8	Image	64	168.87	131.62
128 / 8	Image	128	136.56	114.62
128 / 8	Image	192	130.82	101.03
128 / 8	Image	768	98.52	—
128 / 8	Noise	192	533.48	390.31
128 / 8	Noise	256	160.93	138.71
128 / 8	Noise	320	159.51	119.91
128 / 8	Noise	384	121.83	109.12
128 / 8	Noise	768	113.26	—

The irreducible error is common to all readouts, so differences in clean-image readout risk measure differences in denoiser approximation. In particular, fitting a raw-input affine predictor of $X_{1,m}$ has the same population optimum $\ell_t(X_{t,m})$ as fitting one to $D^*_{m,t}(Y_t)$. The raw-input control thus estimates the affine baseline used in Sec. 3.3.

We compare h, the raw patch $X_{t,m}$, and their concatenation $[X_{t,m}, h]$ through

$$\Delta_{\mathrm{den}} = \mathcal{L}_t(X_{t,m}) - \mathcal{L}_t([X_{t,m}, h]), \qquad \Delta_{\mathrm{input}} = \mathcal{L}_t(h) - \mathcal{L}_t([X_{t,m}, h]). \tag{25}$$

The first gain measures denoising information supplied by h beyond the local affine baseline $\ell_t(X_{t,m})$; the second measures the benefit of supplying the input to a clean-image readout of h.

What improvement beyond the raw input measures. Let $\Pi_{X_{t,m}} h$ denote the best affine prediction of h from $X_{t,m}$, and define $h_\perp = h - \Pi_{X_{t,m}} h$. Both $h_\perp$ and $R_t = D^*_{m,t}(Y_t) - \ell_t(X_{t,m})$ have zero mean and are covariance-orthogonal to $X_{t,m}$. Because $[X_{t,m}, h]$ and $[X_{t,m}, h_\perp]$ have the same affine span,

$$\begin{aligned}
\mathcal{L}_t(X_{t,m}) &= \mathbb{E}\|X_{1,m} - D^*_{m,t}(Y_t)\|^2 + \mathbb{E}\|R_t\|^2, \\
\mathcal{L}_t([X_{t,m}, h]) &= \mathbb{E}\|X_{1,m} - D^*_{m,t}(Y_t)\|^2 + \inf_C \mathbb{E}\|R_t - Ch_\perp\|^2, \\
\Delta_{\mathrm{den}} &= \mathbb{E}\|R_t\|^2 - \inf_C \mathbb{E}\|R_t - Ch_\perp\|^2.
\end{aligned} \tag{26}$$

Thus Δ_{den} measures the reduction in the theorem's residual error supplied by the frozen features after accounting for their affine dependence on the noisy patch. This residual may involve nonlinear local processing or information from other patches; the probe does not distinguish these sources. We use raw-input and concatenated readouts to estimate this gain, without fitting the oracle residual or its covariance eigenvalues.

Why supplying the input is a recovery test. If $X_{t,m} = Wh + b$, every concatenated readout satisfies

$$BX_{t,m} + Ch + c = (BW + C)h + Bb + c.$$

It is therefore already a readout of h. Conversely, a readout of h is a concatenated readout with $B = 0$. Thus

$$X_{t,m} = Wh + b \quad \Longrightarrow \quad \mathcal{L}_t(h) = \mathcal{L}_t([X_{t,m}, h]), \qquad \Delta_{\mathrm{input}} = 0. \tag{27}$$

At $d_h = d_p$, full-rank covariance of $X_{t,m}$ additionally makes W invertible. Then $h = W^{-1}(X_{t,m} - b)$, so $h_\perp = 0$ and $\Delta_{\mathrm{den}} = 0$. This is the square-width special case of the theorem's input-preserving representation class. A positive residual gain at that width must therefore be accompanied by a failure of exact affine input recovery.

For a noise predictor, the native final head is affine in h. Its converted clean estimate $(X_{t,m} - b_t f_{\varepsilon_m,t}(Y_t))/a_t$ is consequently in the affine class generated by $[X_{t,m}, h]$, even when it is not in the class generated by h alone. The concatenation control measures how access to the observed input enables the hidden features to contribute to a clean estimate.

C.3 Frozen-readout fitting and controls

The resolution-256 study includes noise widths 768, 896, 1024, 1152, and 1536, and image widths 128, 256, 384, and 768. The resolution-128 study includes noise widths 192, 320, and 384, and image widths 128 and 192. All checkpoints were frozen after 40 epochs. We extract the last-block patch activations before the final normalization and modulation, and the input to the final linear head after that processing. Sec. 4.5 reports the latter, so its representation includes all computation before the final affine head as in Sec. 3.3. Class-context tokens are removed. The three readout targets are the noisy patch $X_{t,m}$, sampled corruption ε_m, and clean patch $X_{1,m}$.

The probe fixture contains 640 ImageNet training images, with deterministic resolution-specific center crops and no random augmentation. Fit, development, and test splits contain 384, 128, and 128 disjoint images. These are held out from each other for probe fitting, not from backbone training. Within each resolution, every model sees identical images, labels, and corruption realizations. The corruption seeds at $t = 0.1, 0.5, 0.9$ are 99001, 99003, and 99005. For resolution 128, each disjoint 2×2 group of the resolution-256 Gaussian field is summed and divided by 2, preserving independent standard Gaussian pixels.

For each time, activation tap, and target, we fit a separate affine decoder. Its matrix is shared across the 256 patch positions, with a position-dependent intercept estimated from fitting images only. The population identities above apply patchwise; this shared-matrix restriction is used to estimate readouts from the pooled patch observations. Regression statistics and predictions use FP64. Ridge penalties are $\{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 1\}$ times the mean eigenvalue of the fitting feature covariance; development MSE selects the penalty independently for each target. The zero-penalty solution uses a pseudoinverse with relative eigenvalue cutoff 10^{-12}.

Backbone inference uses BF16 autocast, batch size 32, TF32 disabled, and compilation disabled for activation extraction. The saved native attention implementation is retained. We independently verify that reapplying the saved linear output head reproduces the native prediction, and evaluate the native clean-image MSE on the same examples. Controls fit $X_{t,m} \to X_{1,m}$ and $[X_{t,m}, h] \to X_{1,m}$ with the same split and regularization procedure.

Test scores use $R^2 = 1 - \mathrm{MSE}/\mathrm{Var}(\text{target})$, including prediction bias. Target variance is centered separately for each within-patch coordinate across test images and patch positions. We retain per-image errors and compute 95% percentile intervals with 1000 paired image-bootstrap resamples, recomputing target variance in each resample. These intervals measure evaluation-image uncertainty.

The population quantities in Eqs. (25) and (26) are defined by unregularized optimal affine readouts. Our finite-sample, development-selected ridge fits estimate their empirical counterparts. Population gains are nonnegative by nesting of readout classes; small negative differences can occur on the test split. We report differences in $100R^2$ as score points: for example, the empirical input-supply gain is $100(\widehat{\mathrm{MSE}}_h - \widehat{\mathrm{MSE}}_{[X_{t,m},h]})/\widehat{\mathrm{Var}}(X_{1,m})$. It is not a relative percentage reduction in MSE. Tab. 17 reports all probed checkpoints at $t = 0.5$ for both activation taps.

Noise-level and precision controls. Noisy-input accessibility depends on corruption level. For the square-width noise predictor at resolution 256, last-block input recovery is 99.56, 77.86, and 20.04 at $t = 0.1$, 0.5, and 0.9, respectively. At resolution 128, the corresponding scores are 99.18, 78.97, and 19.65. Thus, input recovery is not uniformly poor at all noise levels. An FP32 control at $t = 0.5$ recovers the input almost exactly from the square patch embeddings, while last-block scores remain approximately 77.86 and 78.97. The late-layer readout deficit therefore cannot be explained solely by BF16 rounding in the input projection.

The probe targets are realized clean images and sampled corruptions, rather than oracle population denoisers. The measurements test the simultaneous affine accessibility motivating Sec. 3.3; they do not directly estimate the theorem's residual covariance spectrum.

Table 17: Frozen affine-readout scores ($100R^2$) for all 14 probed checkpoints at epoch 40 and $t = 0.5$. Each column names the target; $X_{1,m} \mid [X_{t,m}, h]$ uses the concatenated input and hidden representation instead of h alone. The head-input columns correspond to Sec. 4.5; the last-block columns provide a control before final normalization and modulation. The raw-input $X_{t,m} \to X_{1,m}$ baselines are 90.28 on ImageNet256 and 86.19 on ImageNet128. Checkpoint averages follow the probe accounting above.

		Final linear-head input				Last Transformer block			
Prediction	d_h	$X_{t,m}$	ε_m	$X_{1,m}$	$X_{1,m} \mid [X_{t,m}, h]$	$X_{t,m}$	ε_m	$X_{1,m}$	$X_{1,m} \mid [X_{t,m}, h]$
ImageNet256/16									
Noise	768	77.85	98.04	0.67	93.71	77.86	98.05	0.67	93.72
Noise	896	98.50	98.44	88.56	94.98	98.53	98.44	88.66	94.98
Noise	1024	98.80	98.48	90.00	95.09	98.80	98.48	90.01	95.09
Noise	1152	99.54	98.56	93.42	95.32	99.56	98.56	93.48	95.32
Noise	1536	99.65	98.60	93.94	95.42	99.66	98.59	93.99	95.42
Image	128	24.08	0.43	94.08	94.12	24.26	0.77	92.70	93.22
Image	256	24.77	1.15	94.68	94.69	25.28	1.75	94.02	94.18
Image	384	26.30	2.92	94.96	94.95	27.38	4.14	94.44	94.50
Image	768	34.18	12.78	95.35	95.31	35.11	13.92	95.20	95.20
ImageNet128/8									
Noise	192	78.94	96.89	2.36	89.90	78.97	96.90	2.47	89.91
Noise	320	99.75	98.03	92.28	93.45	99.75	98.03	92.29	93.45
Noise	384	99.77	98.06	92.44	93.52	99.78	98.06	92.46	93.52
Image	128	26.22	3.14	93.21	93.23	26.80	3.83	92.56	92.73
Image	192	28.13	5.49	93.48	93.49	28.91	6.39	93.21	93.25

D Related literature

JiT invokes the manifold hypothesis to explain why image prediction can outperform noise and path-velocity prediction: clean images may lie near a low-dimensional manifold, whereas the Gaussian components in the other targets extend across the ambient space and may require greater network capacity (Li & He, 2025). Its coarse-patch Transformer experiments support this account: the target comparison holds the velocity loss fixed, and a separate embedding experiment shows that image prediction tolerates substantial input compression. JiT also reports poor noise prediction when patch dimension and hidden width are equal. This distinction matters because limited hidden width and loss of input rank are different architectural restrictions.

k-Diff develops a quantitative geometric explanation in a tractable regression model (Jin & Wang, 2026). For whitened data in a k-dimensional linear subspace of $\mathbb{R}^d$, its analysis optimizes one linear predictor shared across time. Under uniform time sampling and unweighted squared loss on a target mixing clean data and negative noise, Thm. 3.2 gives the optimal clean-data coefficient $d/(d+k)$, approaching image prediction as the ambient dimension dominates. Appendix G extends this calculation to general data second moments. The resulting explanation is tied to this predictor class and training objective. It also motivates the practical k-Diff method, which learns a target mixture from data without explicitly estimating intrinsic dimension.

Architectural locality provides a complementary account of target preference. Gagneux et al. (2026) separate the induced denoiser class from loss weighting and vary Transformer patch size at a fixed image resolution: velocity prediction performs better with small patches, whereas image prediction is more robust with large patches. Their controlled-manifold experiments further show that the effect of intrinsic dimension depends on architecture. These studies therefore offer both explanations and guidance for target choice. Our analysis asks a more specific quantitative question: with the target fixed, how do explicit input-rank and hidden-width constraints limit the best attainable denoising accuracy? We allow arbitrary downstream computation after input compression and characterize the width cost under an explicit affine input-recovery requirement.

Rank constraints connect architecture directly to attainable loss in the RAE analysis of Zheng et al. (2025). Their Thm. 1 considers a nonlinear network between linear input and output maps and lower-bounds sampled-velocity loss by a covariance tail. The proof uses the fact that every prediction lies in the column space of the final output map, regardless of the nonlinear computation preceding it. Sec. 3.1 instead permits unrestricted decoder outputs and restricts the observed input, yielding

an exact noise-recovery risk and its induced denoising error. The distinction is between directions unavailable to the output and noise components unavailable from the input.

These information constraints also motivate explicit paths from the input to the output. For data supported on a linear subspace, Chen et al. (2023) decompose the score into a learned component along that subspace and an analytic orthogonal component; their network combines an encoder–decoder with a full-input term. AsymFlow reduces the noise component of its target through a patchwise projection while retaining the full clean-data component, then reconstructs the complete velocity analytically (Chen et al., 2026). EDM combines a scaled input with a learned output using noise-dependent coefficients (Karras et al., 2022); related output wrappers appear in consistency models (Song et al., 2023) and JiT. Sec. 3.2 analyzes this scalar output wrapper under input compression. Its coefficients determine the discarded-noise contribution: exact cancellation turns the learned target into a scaled clean image, while EDM's coefficients make that contribution vanish at high noise.

A separate question is how efficiently training learns a predictor that the architecture can represent. Ahamed et al. (2026) separate conditional-mean approximation error from irreducible target uncertainty and analyze gradient descent for Gaussian flow matching. Their convergence rates depend on the intermediate covariance spectrum, motivating a transformation of the data before constructing the corruption path. This data preconditioning addresses optimization, complementing our characterization of the best attainable risk within a specified representation class. The output preconditioning in Sec. 3.2 instead changes how the learned branch and explicit input path combine to form the prediction.