# THE RANK BOTTLENECK IN DIFFUSION TRANSFORMERS: WHY AND WHEN WIDTH MUST TRACK TOKEN DIMENSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Diffusion transformers on high-dimensional latent tokens have been reported to require a width comparable to the token dimension, even when fitting a single training example. We show that this requirement can arise from a rank bottleneck in the final per-token linear layer, rather than from a limitation of the transformer backbone itself. A width-$d$ output layer can emit only a rank-$d$ subspace of an $n$-dimensional token. When the regression target contains an isotropic component, that component cannot be represented through the layer, giving an error floor of $1 - d/n$; more generally, the floor is determined by the target covariance beyond rank $d$. With nothing fitted, this floor predicts the reported failures with a mean error of 5%. The same bottleneck persists during sampling: a rank-$d$ velocity field cannot alter the component of a sample outside its output subspace, leaving the corresponding part of the initial noise unchanged. This also explains why EDM preconditioning does not remove the floor. In contrast, predicting the clean token and reconstructing the isotropic component from the network input routes that component around the rank-limited output layer. For the clean target, the remaining floor can be bounded directly from the target spectrum, giving a criterion for when the ambient token dimension is or is not the relevant width scale. Across latent and pixel representations, our experiments support this view: the observed width requirement follows the effective rank of the emitted target rather than its ambient dimension. These results identify the output parameterization, rather than transformer capacity alone, as the source of the reported width constraint.

## 1 INTRODUCTION

Diffusion and flow-matching models can be trained to predict noise, the clean sample, or a velocity field that interpolates between them. For a fixed corruption time, these targets are related by an invertible linear transformation, so at the level of regression they describe the same underlying prediction problem, up to a time-dependent weighting (Karras et al., 2022). In practice, however, the choice of target can have a much larger effect on what a model can learn.

Li & He (2026) show that predicting clean pixels allows a plain transformer to use large patches where noise prediction degrades, and attribute this behavior to a difference in the geometry of the targets: clean data occupies a structured, low-dimensional region, whereas noise fills the ambient space. A related phenomenon appears more sharply in latent diffusion. Zheng et al. (2026) replace the usual autoencoder with a frozen DINOv2 encoder and report that a diffusion transformer cannot fit even a single image unless its width is at least the latent token dimension. They explain this threshold by the fact that adding noise extends the target distribution into the full ambient latent space, and address the problem with a wider auxiliary output head.

This explanation points to a question about the parameterization itself: if the noisy target is full-dimensional, does that dimensionality have to be represented by the transformer, or only by the map that finally emits the target?

We show that, for a per-token linear output layer, it is the latter. A width-$d$ feature is mapped to an $n$-dimensional token by $W \in \mathbb{R}^{n \times d}$, whose rank is at most $d$. Consequently, every output token lies in a subspace of dimension at most $d$, regardless of the capacity of the backbone that produces



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

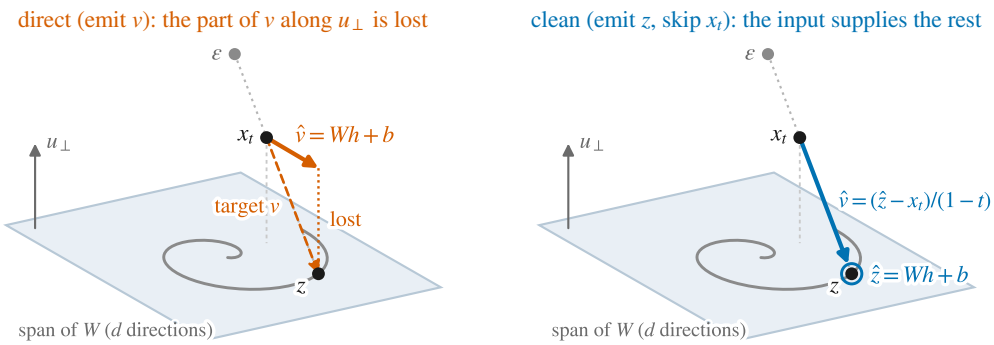


Figure 1: Where the floor comes from. A width-$d$ output layer can emit only directions in its span. Direct prediction must emit $v = z - \varepsilon$ and loses its part along $u_\perp$; clean prediction emits $\hat{z}$ inside the span and takes the rest from the input $x_t$.

the feature. If the target contains an isotropic component, that component cannot pass through the output map when $d < n$. For a unit-variance isotropic target, the resulting irreducible error is exactly $1 - d/n$; for a general target, it is the covariance mass outside the top $d$ directions (Figure 1).

This simple observation changes how the width constraint should be interpreted. It is not a threshold in transformer capacity. It is a property of the target-output interface. The same rank restriction also appears during generation: a rank-$d$ velocity field can only move a sample within its output subspace, leaving the component in the remaining directions unchanged. The initial noise in those directions therefore survives sampling, including under EDM preconditioning (Figure 2).

The parameterization also determines whether this component has to pass through the bottleneck at all. When the model predicts the clean token, the isotropic component can instead be taken directly from the network input when constructing the velocity. This removes the isotropic part of the rank floor without increasing the model width or adding parameters.

The isotropic case is only the simplest instance. For a structured target, the relevant quantity is the spectrum of its covariance: directions carrying variance outside the top $d$ dimensions contribute to the irreducible error. This gives a direct way to determine whether the ambient token dimension is actually the relevant width scale. Targets with rapidly decaying spectra can be emitted through substantially narrower representations, whereas targets whose variance remains spread across the ambient dimensions cannot.

We test these predictions on pixel patches and latent representations, including DINOv2 latents. The observed single-image thresholds follow the predicted rank floor, while clean-target parameterization removes the component attributable to isotropic noise. On full latent distributions, the remaining error is likewise bounded by the target spectrum before training. Together, these results suggest that the relevant width scale is the effective rank of the target emitted by the model, rather than the ambient token dimension itself.

**Contributions.**

- A closed-form error floor for a rank-limited output map: $1 - d/n$ for isotropic targets and the tail covariance mass beyond rank $d$ in general (Section 3).
- A sampling interpretation of the same bottleneck: a rank-$d$ velocity field preserves the component of every sample outside its output subspace, including under EDM preconditioning (Section 4.3).
- A skip parameterization that routes the isotropic component through the network input rather than the rank-limited output layer, together with a spectrum-based bound on the residual floor for full target distributions (Sections 5.2 and 5.3).
- A width criterion based on the effective rank of the emitted target, which explains the observed width dependence in latent diffusion and connects it to the behavior of large-patch clean prediction (Section 6).

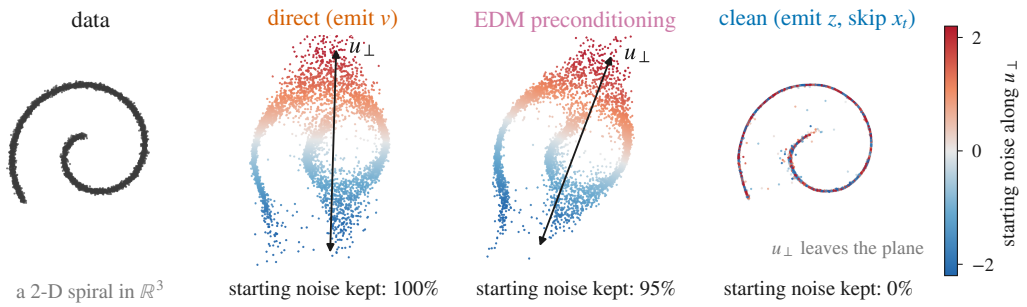


Figure 2: A width-2 output layer learning a spiral in $\mathbb{R}^3$, each sample coloured by its starting noise along $u_\perp$. Direct and EDM keep that noise (Proposition 2, Corollary 1); clean removes it (Corollary 2). Setup in Appendix B.

## 2 Setup

We use the rectified-flow interpolant. For a clean token $z \in \mathbb{R}^n$ and $\varepsilon \sim \mathcal{N}(0, I_n)$,

$$x_t = tz + (1-t)\varepsilon, \qquad t \in [0,1], \tag{1}$$

with velocity $v = z - \varepsilon$. A transformer maps the noised tokens to features of width $d$, and a final linear layer produces each output token,

$$\hat{y} = Wh + b, \qquad W \in \mathbb{R}^{n\times d},\ h \in \mathbb{R}^d. \tag{2}$$

Attention mixes information across tokens, and the argument does not depend on depth; the only structural fact we use is that (2) has rank at most $d$. We compare three parameterizations. *Direct* emits the velocity, $\hat{v} = \hat{y}$. *Clean* emits the clean token and forms the velocity from the input,

$$\hat{v} = \frac{\hat{y} - x_t}{1-t}, \tag{3}$$

and we refer to it as the skip parameterization. *EDM* uses the preconditioning of Karras et al. (2022), derived for (1) in Section 4.2. Each is trained on its natural objective: velocity error for direct and EDM, clean-token error for clean. Training the clean arm on a uniform velocity loss would weight it by $(1-t)^{-2}$ and measure loss scaling rather than parameterization. Samples are drawn with $K$ Euler steps of $\dot{x} = \hat{v}(x,t)$ from $t = 0$ to $t = 1$, identically for all arms.

## 3 A rank bottleneck

### 3.1 The floor

**Proposition 1** (Isotropic floor)**.** *Let $v = z - \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_n)$ independent of $z$, and let $\hat{v}$ be constrained to an affine subspace of dimension at most $d \le n$. Then*

$$\tfrac{1}{n}\,\mathbb{E}\,\|\hat{v} - v\|^2 \ \ge\ 1 - \tfrac{d}{n}, \tag{4}$$

*whatever the distribution of $z$. The minimum over such predictors is attained by the affine projection $\mathbb{E}v + P_d(v - \mathbb{E}v)$, with $P_d$ the projection onto the leading $d$ eigenvectors of $\mathrm{Cov}(v)$; it equals the bound when $\mathrm{rank}\,\mathrm{Cov}(z) \le d$ and exceeds it otherwise.*

*Proof.* By the Eckart–Young theorem the best $d$-dimensional affine subspace in mean square passes through $\mathbb{E}v$ and spans the leading $d$ eigenvectors of $\Sigma = \mathrm{Cov}(v)$, leaving the sum of the remaining eigenvalues. Independence gives $\Sigma = \mathrm{Cov}(z) + I_n$, so every eigenvalue is at least one, and the $n-d$ discarded eigenvalues contribute at least $(n-d)/n$ per coordinate. □

The linear algebra is classical (Eckart & Young, 1936); what is new is that it is the operative constraint in a setting where the failure was attributed to capacity. Adding isotropic noise lifts every eigenvalue of the target covariance by one, so the low-dimensional structure of clean data offers a rank-limited layer no saving once the noise is part of the target.

### 3.2 THE GENERAL FORM

Nothing in the argument is specific to $\varepsilon$. For any emitted target $y$,

$$\mathrm{floor}(y, d) = \frac{1}{n} \sum_{i>d} \lambda_i\big(\mathrm{Cov}(y)\big), \tag{5}$$

computable from cached data before training. With eigenvalues normalised to mean one, as for our standardized tokens, a target with a flat spectrum pays $1 - d/n$, while a target whose spectrum decays quickly pays almost nothing once $d$ exceeds its effective rank $r_\delta(y)$, the smallest $d$ with $\mathrm{floor}(y, d) \le \delta$. Besides this rank-induced floor, a trained network makes an estimation error inside the subspace, and the two are orthogonal parts of one mean squared error, so

$$\mathrm{error}(y, d) \approx \mathrm{floor}(y, d) + c, \tag{6}$$

where $c$ is the same parameterization's error at a width where the floor is zero. This is a model, not a consequence of Proposition 1: it assumes that a constrained network attains the floor and makes the same in-subspace error as an unconstrained one. We measure $c$; nothing is fitted.

### 3.3 COMPARISON WITH THE PUBLISHED THRESHOLD

Zheng et al. (2026) report single-image fitting losses for three DiT widths on three DINOv2 encoders. With Proposition 1 and nothing fitted, the three cells they report as failures, at $d/n = 0.50$, $0.75$ and $0.38$, are predicted at $0.500$, $0.250$ and $0.625$ against reported $0.520$, $0.270$ and $0.650$, a mean error of 5%; the six unconstrained cells are near zero as predicted (Figure 4a; Table 5).

## 4 ESCAPING THE FLOOR, AND WHAT IT COSTS TO STAY ON IT

### 4.1 THE ISOTROPIC COMPONENT IS ALREADY AVAILABLE

Proposition 1 constrains what the output layer emits, not what the network computes. Inverting (1) gives $\varepsilon = (x_t - tz)/(1-t)$, where $x_t$ is the network input. The clean parameterization (3) therefore obtains the isotropic part of the velocity from the input at no cost in output rank, and asks the rank-limited map only for $\hat{y} \approx z$, whose spectrum decays. Nothing is lost at the Bayes optimum: since $\mathbb{E}[v \mid x_t] = (\mathbb{E}[z \mid x_t] - x_t)/(1-t)$, setting $\hat{y} = \mathbb{E}[z \mid x_t]$ recovers the Bayes-optimal velocity exactly. What the construction introduces is amplification of errors in $\hat{y}$ by $1/(1-t)$ near $t = 1$.

### 4.2 EDM PRECONDITIONING

EDM writes $\hat{v} = c_{\mathrm{skip}}(t)\, x_t + c_{\mathrm{out}}(t)\, F(c_{\mathrm{in}}(t) x_t)$ and chooses coefficients that give the network a unit-variance target with the smallest $c_{\mathrm{out}}$. For (1), with $D = 2t^2 - 2t + 1$ (Appendix A),

$$c_{\mathrm{skip}} = \frac{2t-1}{D}, \qquad c_{\mathrm{out}} = c_{\mathrm{in}} = D^{-1/2}, \qquad F^\star = \frac{(1-t)z - t\varepsilon}{\sqrt{D}}. \tag{7}$$

The target handed to the network retains an isotropic component of weight $t/\sqrt{D} \ge 0.7$ for all $t \ge 1/2$, and $c_{\mathrm{skip}}(1/2) = 0$ exactly. EDM chooses its coefficients to keep the network's target at unit variance, which helps when the difficulty is conditioning. It does not reduce the rank the output layer must supply, which is the difficulty here.

More generally, let $\kappa(t)$ be the coefficient of $\varepsilon$ in the part of the velocity that must pass through the rank-$d$ map. Then

$$\mathrm{floor} = \mathbb{E}_t\big[\kappa(t)^2\big]\,\big(1 - d/n\big), \tag{8}$$

with $\kappa = 1$ for direct, $\kappa = 0$ for clean, and $\kappa = t/D$ for EDM. Under $t \sim U(0, 0.95)$ EDM keeps a fraction $\mathbb{E}[\kappa^2] = 0.771$ of the floor: its skip removes 23% of the isotropic load, not all of it.

### 4.3 THE FLOOR IN SAMPLES

The floor bounds a regression error. In generation it becomes an invariant of the sampler, and does not depend on how well the network is trained.

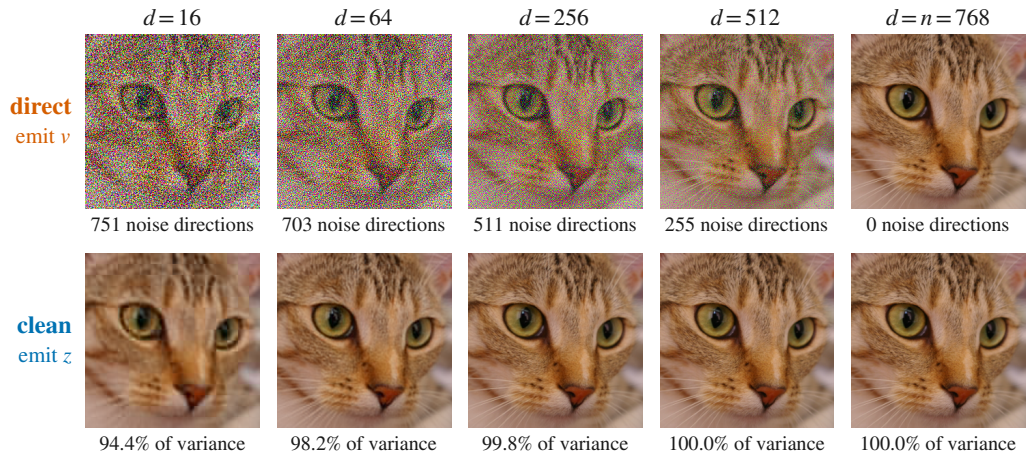


Figure 3: The best samples a model of width $d$ can produce for a photograph cut into $16 \times 16$ patches ($n = 768$). Direct keeps noise in $n - d - 1$ directions until $d = n$; clean is sharp once $d$ passes the effective rank of the patches. Construction in Appendix B.

**Proposition 2** (Residual noise)**.** *Let the velocity be $\hat{v} = Wh + b$ as in* (2)*, applied per token, and let $P_\perp$ project onto the orthogonal complement of* $\mathrm{span}(W, b)$*, of dimension at least $n - d - 1$. For any sampler whose output is $x_0$ plus a linear combination of velocity evaluations, such as Euler, Heun and explicit Runge–Kutta methods,*

$$P_\perp\, x_1 = P_\perp\, x_0 \sim \mathcal{N}(0, P_\perp), \tag{9}$$

*whatever the values of the network parameters.*

*Proof.* Each evaluation $Wh_i + b$ lies in $\mathrm{span}(W, b)$, so any linear combination of them does, and $P_\perp$ annihilates it. □

Every sample from a constrained direct model therefore carries unit-variance Gaussian noise in at least $n - d - 1$ directions per token. At $n = 768$ and $d = 256$ that is 511 of the 768 directions of each patch (Figure 3). This is not a failure of optimisation; it holds for the optimal rank-$d$ field.

**Corollary 1** (EDM keeps the noise)**.** *Under* (7)*, the component of the sample outside the network's output span evolves as $\frac{d}{dt} P_\perp x = c_{\mathrm{skip}}(t)\, P_\perp x$, so $P_\perp x_1 = \exp\big(\int_0^1 c_{\mathrm{skip}}\big)\, P_\perp x_0 = P_\perp x_0$. The integral vanishes because $c_{\mathrm{skip}} = D'/2D$ and $D(0) = D(1) = 1$.*

The EDM skip shrinks the residual noise while $t < 1/2$ and restores it while $t > 1/2$. With $K$ Euler steps the multiplier is $\prod_k (1 + c_{\mathrm{skip}}(t_k)/K)$, which keeps 95% of the residual energy at $K = 50$ and tends to all of it as $K$ grows. EDM thus lowers the training error by the factor in (8) while leaving the out-of-span component of its samples essentially unchanged.

**Corollary 2** (The skip removes it)**.** *Under* (3)*, $\frac{d}{dt} P_\perp x = -P_\perp x/(1-t)$. With Euler steps $t_k = k/K$ the multiplier telescopes, $\prod_k \left(1 - \frac{1}{K(1-t_k)}\right) = \frac{1-t_K}{1-t_0} = 0$, so $P_\perp x_1 = 0$ exactly.*

A constrained direct or EDM model keeps noise outside its $d$-dimensional span; every sample of a constrained clean model lies inside it. The clean model is therefore limited by how much of the data lies in a $d$-dimensional subspace, which is (5) applied to $z$. On trained width-256 generators, direct keeps 100.0% of the residual energy, which moves by a relative $7.6 \times 10^{-7}$ over fifty steps; EDM keeps 95.0% and clean none (Appendix G). Figure 2 shows all three behaviours in a toy model.

## 5 Experiments

All results are fp32. An earlier pass in bfloat16 produced NaN and errors over forty times the floor at widths of 384 and below; we use none of its results except to document the problem in Appendix H.

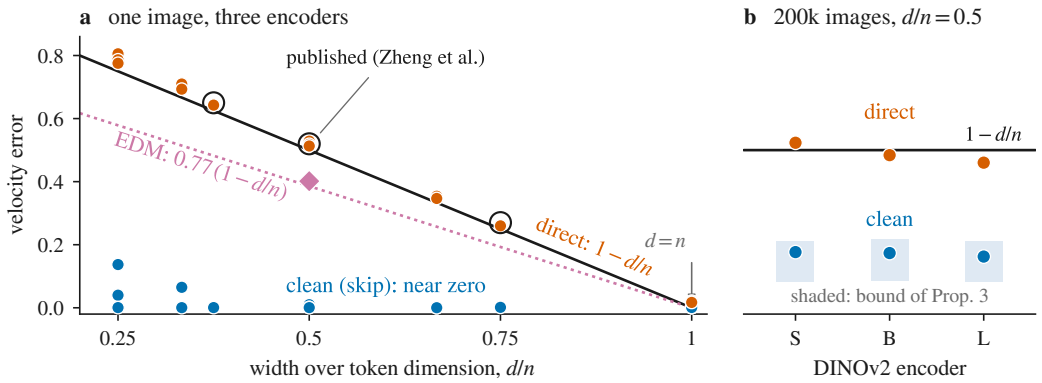


Figure 4: (a) Single-image fitting on three DINOv2 encoders; rings are the failures reported by Zheng et al. (2026). (b) Excess error on 200,000 images at $d/n = 0.5$, with the bound of Proposition 3 shaded.

### 5.1 Single-image fitting

Following Zheng et al. (2026), a transformer over frozen DINOv2 patch tokens fits one image for 20,000 steps. We score velocity error on a fixed held-out grid of $(t, \varepsilon)$, which removes sampling noise from single-point comparisons. Depth is 12 and only width varies. Figure 4a summarises the measurements; Table 2 in Appendix C lists them with seed spreads.

**The error is the floor plus a small constant.** Across the twelve constrained cells the error is 1.03 to 1.07 times the floor, and the ratio narrows as $n$ grows. The baseline $c$, measured at the unconstrained width, is 0.017 to 0.021. With it the additive model predicts all twelve cells to a mean absolute error of 1.6%, against 4.4% for $\max\{\text{floor}, c\}$, and is closer in every cell (Appendix D). Seed standard deviations there are under 1% of the mean. On the three cells Zheng et al. (2026) report as failures we measure 0.5183, 0.6427 and 0.2598 against their 0.520, 0.650 and 0.270.

**The skip parameterization avoids the floor.** At widths of 256 and above, where the 256 tokens of the image fit inside the output span, clean is roughly 180 to 1800 times below the floor. At $n = 768$, clean at width 256 reaches 0.0004 while direct at width 768, with $8.8\times$ the parameters, reaches 0.0183. Below width 256 the clean target of a single image has a tail of its own, and clean is only 5 to 51 times below the floor (widths 96 to 192 at $n = 384$, width 192 at $n = 768$): the remaining constraint is (5) applied to the clean target.

**EDM removes the fraction** (8) **predicts.** Part of the EDM velocity reaches the output through $c_{\text{skip}} x_t$, bypassing the rank-$d$ map, so EDM can sit below $1 - d/n$; Equation (8) isolates the part that must pass through the map. At $n = 1024$, $d = 512$ it predicts 0.500 for direct and 0.386 for EDM before the baseline term; in a separate three-seed comparison we measure $0.5132 \pm 0.0010$ and $0.4015 \pm 0.0007$, that is 1.03 and 0.80 times $1 - d/n$.

**The architectural changes we tested do not remove the floor.** Long skip connections as in U-ViT (Bao et al., 2023) route shallow features to deep layers but keep the final linear layer, so Proposition 1 still applies. A non-linear readout is not covered by it, so we test it directly (Table 4, Appendix E). These fp32 ablations use an earlier protocol, so they are compared only with each other.

Both variants recover under 3% of the distance to zero despite added parameters, against 97 to 99% for the skip parameterization with none, and the long skips land within 1.4% of the additive prediction. Here only a change of target removed the floor.

### 5.2 Training on a full distribution

Fitting one image drives the Bayes error close to zero, which isolates the rank term but says nothing about held-out behaviour. We therefore train on 200,000 DINOv2-encoded ImageNet images for

40,000 steps at $d/n = 0.5$. The Bayes error is now far from zero, so the rank penalty is read as the excess over the same arm at an unconstrained width, where the baseline cancels. Velocity error carries a $1/(1-t)$ factor for the clean arm, so clean is scored in clean space.

On a full distribution the clean arm no longer regresses $z$ but its conditional mean $m_t = \mathbb{E}[z \mid x_t]$, which is shrunk towards the data mean at high noise. This makes (5) an upper bound rather than a prediction, and gives a matching lower bound from the same spectrum.

**Proposition 3** (Full-distribution bound)**.** *Let $\lambda_i$ be the eigenvalues of the per-token covariance of $z$, normalised so that their mean is one. For Bayes-optimal predictors, the excess clean-space error of a width-$d$ model over an unconstrained one satisfies*

$$\frac{1}{n}\sum_{i>d}\mathbb{E}_t\left[\frac{t^2\lambda_i^2}{t^2\lambda_i+(1-t)^2}\right] \;\le\; \text{excess} \;\le\; \frac{1}{n}\sum_{i>d}\lambda_i. \tag{10}$$

*Proof.* The excess is the tail eigenvalue mass of $\bar{C} = \mathbb{E}_t[\mathrm{Cov}(m_t)]$. Conditioning on less information shrinks the covariance of a conditional mean, so the per-token linear estimator $\ell_t$, which sees only its own token, satisfies $\mathrm{Cov}(\ell_t) \preceq \mathrm{Cov}(m_t) \preceq \mathrm{Cov}(z)$. Loewner order bounds every eigenvalue, hence every tail sum. $\mathrm{Cov}(\ell_t)$ shares the eigenbasis of $\mathrm{Cov}(z)$ with eigenvalues $t^2\lambda_i^2/(t^2\lambda_i+(1-t)^2)$. □

Figure 4b and Table 6 show the result. For direct, the excess is within 3 to 8% of $1 - d/n$ at all three token dimensions. For clean, all three measurements fall inside (10), at 84, 80 and 77% of its upper end as the token dimension grows. For pixel patches the whole interval lies below $0.003$, which is why the skip parameterization removes almost all of the floor there. The residual of about $0.17$ on DINOv2 latents is over ten times the single-image values, and Corollary 2 explains why it must appear: a constrained clean model generates only within its $d$-dimensional span, and DINOv2 latents need 81 to 82% of their dimensions to hold 95% of their variance, against 3 to 8% for pixel patches (Appendix F). A single image hides this: its 256 tokens span at most 255 dimensions.

The escape is therefore narrower than a single image suggests: the skip removes the isotropic part of the floor, and what remains is bounded on both sides by quantities measurable before training.

### 5.3 Generation

We train generative models on raw pixel patches, where the output is the image and no decoder mediates the comparison. Patch 16 gives token dimension 768 at both $64 \times 64$ and $256 \times 256$, so the boundary sits at the same width as for DINOv2-B tokens while the sequence grows from 16 tokens to 256. The $64 \times 64$ images are area-downsampled from the same $256 \times 256$ centre crops. We report FID and KID, whose standard estimator is unbiased where that of FID is not, with Inception Score, precision and recall, from 10,000 samples and $K = 50$ Euler steps.

Three of eighteen $64 \times 64$ runs diverged to NaN weights. Diverged models emit constant images (FID 414.7 for any seed), so we exclude them from all means and report the pattern below.

**What the residual noise costs.** Proposition 2 guarantees that no amount of training removes the residual noise from a constrained direct model; the ideal rows of Table 1 shows its cost, and Figure 3 shows it on a photograph. Samples exact inside the span of the trained width-256 head, with the retained noise outside it, score FID 222.7 under direct and 220.1 under EDM, with recall zero. The same construction under clean scores 9.2, two points from the real images at 7.3. At this resolution a width-256 clean model therefore carries almost no representational penalty, while the residual noise alone costs over 200 FID. The trained models sit above their ideal counterparts. For clean that gap owes nothing to residual noise, which the parameterization removes; it comes mainly from finite training and capacity. For direct the ideal sample already accounts for 215 of the 348 FID points between the trained model and real images, about three fifths. The ideal rows are not a formal lower bound on FID, since a model need not match the data inside its span to minimise it, but they isolate the cost that Proposition 2 makes unavoidable. Their high precision with zero recall is expected: each ideal image sits near a real one, and together they cover none of its variety.

Table 1: ImageNet $64 \times 64$, patch 16, $n = 768$. Trained rows use finite seeds only; for the two constrained arms with diverged seeds, only FID was recomputed. Ideal rows take 10,000 held-out real images, keep them exactly inside the span of each trained width-256 head, and add outside it the noise that Proposition 2 and Corollary 1 say each parameterization retains.

| $d$ | Arm | Seeds | FID | KID | IS | Prec. | Rec. |
|---|---|---|---|---|---|---|---|
| 768 | clean | 3 | $93.75 \pm 2.47$ | $.068 \pm .002$ | $4.45 \pm .21$ | .557 | .086 |
| 768 | EDM | 3 | $161.49 \pm 3.86$ | $.137 \pm .004$ | $2.17 \pm .07$ | .619 | .001 |
| 768 | direct | 3 | $173.76 \pm 11.27$ | $.160 \pm .012$ | $2.43 \pm .14$ | .395 | .001 |
| 256 | clean | 3 | $109.80 \pm 2.84$ | $.082 \pm .002$ | $3.33 \pm .20$ | .613 | .031 |
| 256 | EDM | 2 | $338.48 \pm 37.17$ | | | | |
| 256 | direct | 1 | 355.11 | | | | |
| real images, held out | | | 7.25 | .000 | 61.60 | .738 | .735 |
| 256 | ideal clean | | 9.23 | .001 | 51.93 | .713 | .720 |
| 256 | ideal EDM | | 220.08 | .165 | 1.46 | .897 | .000 |
| 256 | ideal direct | | 222.67 | .168 | 1.44 | .901 | .000 |

**How much of the gap is rank.** Using clean as the control for the cost of a smaller model, shrinking from width 768 to 256 costs clean 16.0 FID, EDM 177.0 and direct 181.4. The rank-attributable excess is $+161$ for EDM (two seeds) and $+165$ for direct (one seed). EDM is no better than direct in samples although its training error is lower by the factor (8), as Corollary 1 predicts.

**Training stability.** All three diverged runs are at the constrained width, two under direct and one under EDM, the two parameterizations whose output must carry the full token. None of the nine unconstrained runs and none of the three constrained clean runs diverged. With so few runs we report this as an observation rather than a claim.

**The gap without a rank constraint.** At width 768 there is no floor and no residual noise, yet clean is 80 FID ahead of direct. This matches the target-geometry effect reported by Fu et al. (2026), whose setting lies entirely above the boundary. EDM, which regresses velocity with a variance-normalised weighting, recovers 12.3 of those 80 points and is not separable from direct at three seeds. Clean remains 67.7 points ahead of it, with seed standard deviations of 2.5 and 3.9, so EDM-style reweighting of the velocity loss does not close the gap. Clean raises recall from 0.001 to 0.086: at this width direct covers a narrow part of the distribution, which FID alone does not show.

**Higher resolution.** At $256 \times 256$, with sixteen times more tokens, clean is ahead of direct at both widths, by 150 FID at $d = 768$ and 102 at $d = 256$ (Table 8), so the advantage does not depend on the short 16-token sequence. We report no rank decomposition here: the unconstrained direct model scores worse than the constrained one, which no account of width predicts in expectation. All four runs completed 100,000 steps, but the width-768 model costs twice as much per step and every arm is far from converged, so at one seed per cell the width difference is not interpretable.

## 6 A DESIGN RULE

Equation (5) yields one rule for width. A model emitting the velocity pays $\text{floor} = (1 - d/n) + \text{tail}_z(d)$, so keeping it below $\delta$ requires $d \geq n(1 - \delta)$, the requirement of Zheng et al. (2026). A model emitting the clean token pays $\text{tail}_z(d)$ alone, so it needs only $d \geq r_\delta(z)$, the effective rank of the data. The token dimension enters only because the noise makes the velocity target full rank.

Table 7 shows the two regimes. For pixel patches the clean target is so nearly low-rank that a modest width suffices, which may explain why clean prediction lets a plain transformer use very large patches (Li & He, 2026). For DINOv2 latents the saving is only about 15% of the width: their spectrum shows they are themselves high-rank, so clean prediction cannot give them the reduction it gives pixels. In both cases the requirement is predictable from the spectrum of the emitted target.

## 7 Related work

**Prediction targets.** Noise prediction (Ho et al., 2020), velocity regression (Salimans & Ho, 2022; Lipman et al., 2023; Liu et al., 2023) and the EDM design space (Karras et al., 2022) frame the choice of target. Li & He (2026) argue for clean prediction from the geometry of pixel data, and Fu et al. (2026) extend the argument to latent transformers with a local Gaussian analysis in which velocity regression inherits an isotropic covariance floor. That account concerns the geometry of the target; ours derives a width-dependent floor from the rank of the output map, together with the sampling invariant of Proposition 2. Our unconstrained results are consistent with theirs.

**Architectures and samplers.** Peebles & Xie (2023) introduced the diffusion transformer and Zheng et al. (2026) the representation-encoder latents whose width constraint we study. Their wide head, inspired by DDT (Wang et al., 2026), lets the final projection reach full rank $n$, which by Proposition 1 is what removes the floor. U-ViT (Bao et al., 2023) adds long skip connections, which we show leave the floor intact. SiT (Ma et al., 2024) studies interpolants and samplers within one framework. Proposition 2 covers deterministic samplers built from velocity evaluations; stochastic samplers inject fresh noise at each step, and whether they mitigate a constrained model is a question we leave open. Guo et al. (2026) report that skip connections help pixel diffusion under clean prediction and fail under velocity prediction, which they attribute to high-frequency detail; the asymmetry is consistent with Table 4.

**Low-dimensional structure.** Diffusion models are known to exploit low-dimensional structure in data (Wang et al., 2024; Stanczuk et al., 2024). We use that structure in the opposite direction: it is its absence from the noise target, combined with a rank-limited output map, that produces the floor. The nearest analogue outside diffusion is the softmax bottleneck of language models (Yang et al., 2018), where a rank-limited output layer bounds the distributions a model can express; here the same limit bounds a regression error and survives into every sample.

## 8 Limitations

The single-image experiments isolate the rank term because the Bayes error is nearly zero; on full distributions of DINOv2 latents a residual remains, which we bound rather than predict, and the lower bound uses an estimator that ignores information from other tokens. Three of eighteen generation runs diverged, all at the constrained width under direct or EDM, so the constrained direct figures rest on one surviving seed and we report only FID for the constrained arms with diverged seeds. The ideal samples are not a formal lower bound on FID. Our generative models train on raw patches without a decoder and are far from converged, so absolute FID is high; at $256 \times 256$ each cell is a single seed. FID and KID use torchvision's InceptionV3 and are not comparable to published numbers. Proposition 2 applies to deterministic samplers only, and Proposition 1 to a final linear layer; for non-linear readouts we have only the empirical result of Table 4, for one MLP design. The constant $c$ in (6) is measured per parameterization, not universal, and the ablations of Table 4 use an earlier scoring protocol. Reduced precision is unreliable at narrow width (Appendix H).

## 9 Conclusion

The width requirement reported for diffusion over representation latents is accounted for by a rank-limited linear output layer that must emit a target with an isotropic component. In closed form the floor matches the published failure values and our measurements at three token dimensions to within a few percent. It also constrains sampling: a constrained velocity model leaves the out-of-span component of its initial noise in every sample, whatever it has learned, and EDM preconditioning leaves it too; at width 256 on ImageNet $64 \times 64$ that noise alone costs over 200 FID. Predicting the clean token and forming the velocity from the input removes the isotropic part of the floor and the residual noise at no cost in parameters; what remains is bounded by the clean target's spectrum, measurable before training. Under a per-token linear output layer, the width a model needs is therefore the effective rank of what it emits: far below the token dimension for pixel patches, close to it for representation latents. The two architectural changes we tested did not help.

### AI USE STATEMENT

Generative AI tools (Claude, Anthropic, and ChatGPT, OpenAI) were used for drafting, language editing and code formulation. The authors are responsible for all research, experiments, analysis, results, and final content.

### REPRODUCIBILITY STATEMENT

All experiments use public encoders and datasets. Model definitions, training and evaluation code, and per-run hyperparameter records are in the supplementary material. Every result in the paper comes from an fp32 run; the bfloat16 values of Table 11 are shown only to document the precision problem. Runs that diverged to non-finite weights are excluded from all means and reported in Section 5.3.

## REFERENCES


Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A ViT backbone for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Funing Fu, Tenghui Wang, Guanyu Zhou, Junyong Cen, and Qichao Zhu. JLT: Clean-latent prediction in latent diffusion transformers. *arXiv preprint arXiv:2605.27102*, 2026.

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. PixelU: A U-shaped transformer for efficient end-to-end pixel diffusion. In *European Conference on Computer Vision*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

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*, 2022.

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

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

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

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023.

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

Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *International Conference on Machine Learning*, 2024.

Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Breaking the curse of dimensionality: Diffusion models efficiently learn low-dimensional distributions. *arXiv preprint arXiv:2409.02426*, 2024.

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

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026.

## A Derivation of the EDM coefficients

Write $\hat{v} = c_{\text{skip}} x_t + c_{\text{out}} F$ under (1). The network's target is $F^\star = (v - c_{\text{skip}} x_t)/c_{\text{out}}$, and

$$v - c_{\text{skip}} x_t = (1 - c_{\text{skip}} t)\, z - \big(1 + c_{\text{skip}}(1-t)\big)\, \varepsilon.$$

Assuming $z$ and $\varepsilon$ uncorrelated and every coordinate of $z$ of unit variance, which holds for our per-dimension standardized tokens, unit variance requires $c_{\text{out}}^2 = (1 - c_{\text{skip}} t)^2 + (1 + c_{\text{skip}}(1-t))^2$. Minimising over $c_{\text{skip}}$ gives $c_{\text{skip}} = (2t-1)/D$ with $D = t^2 + (1-t)^2$. Then $1 - c_{\text{skip}} t = (1-t)/D$ and $1 + c_{\text{skip}}(1-t) = t/D$, so $c_{\text{out}} = D^{-1/2}$ and $F^\star = ((1-t)z - t\varepsilon)/\sqrt{D}$. The velocity-space coefficient of $\varepsilon$ passing through the network is $c_{\text{out}} \cdot t/\sqrt{D} = t/D$, which is $\kappa$ in (8). Since $D' = 2(2t-1)$, $c_{\text{skip}} = D'/2D$, the identity behind Corollary 1.

## B Experimental details

All results are fp32. The ablations of Table 4 ran on cloud GPUs and all other runs on NVIDIA H100 GPUs. Latent experiments use DINOv2 (Oquab et al., 2024) patch tokens at $224 \times 224$, standardized per dimension, depth 12, AdamW with warmup, gradient clipping at 1.0, and $t \sim U(0, 0.95)$. Pixel experiments use $256 \times 256$ centre crops, with $64 \times 64$ images area-downsampled from the same crops, depth 12 and $t \sim U(0, 1)$; the $256 \times 256$ runs train for 100,000 steps. The MLP readout uses GELU with the tanh approximation. A variant of the clean parameterization that caps the $1/(1-t)$ factor at $\lambda = 10$ behaves like clean in our fits and is reported in the supplementary material. For the ideal samples, the FID reference is the first 10,000 entries of a fixed permutation of 200,000 cached images and the ideal samples are built from the next 10,000, so the two sets are disjoint by construction. Per-run settings are recorded in the hyperparameter table of the supplementary material.

The toy model of Figure 2 places a two-dimensional spiral on a random plane in $\mathbb{R}^3$. A four-layer MLP of width 256 with sinusoidal time features ends in a width-2 feature and a linear head to $\mathbb{R}^3$, the three parameterizations are trained for 20,000 steps with Adam, and samples use 50 Euler steps from shared noise. Along the direction orthogonal to each head's column space, the sample variance goes from 0.997 to 0.997 for direct, from 1.012 to 0.961 for EDM, and from 1.021 to 0.000 for clean.

Figure 3 uses four sample photographs from scikit-image (CC0), resized to $256 \times 256$ and cut into $16 \times 16$ patches. The output span of width $d$ is the patch mean together with the leading $d$ principal directions of their patches, of which 19 hold 95% of the variance. For the cat photograph the clean row shows each patch projected onto that span, and the direct row adds one fixed draw of standard Gaussian noise projected onto its orthogonal complement.

## C FITTING MEASUREMENTS

Table 2: Velocity error against width, single-image fitting, fp32. Entries with $\pm$ are means over three seeds. Ratio is measured error over $1 - d/n$.

| $n$ | $d$ | Floor | Direct | Ratio | Clean |
|---|---|---|---|---|---|
| 384 | 96 | 0.750 | 0.8054 | 1.07 | 0.1371 |
| 384 | 128 | 0.667 | $0.7096 \pm 0.0033$ | 1.06 | $0.0649 \pm 0.0060$ |
| 384 | 192 | 0.500 | $0.5284 \pm 0.0051$ | 1.06 | $0.0098 \pm 0.0006$ |
| 384 | 256 | 0.333 | 0.3537 | 1.06 | 0.0002 |
| 384 | 384 | 0 | $0.0210 \pm 0.0031$ | | $0.0008 \pm 0.0007$ |
| 768 | 192 | 0.750 | 0.7864 | 1.05 | 0.0399 |
| 768 | 256 | 0.667 | $0.6933 \pm 0.0016$ | 1.04 | $0.0004 \pm 0.0001$ |
| 768 | 384 | 0.500 | $0.5183 \pm 0.0020$ | 1.04 | $0.0013 \pm 0.0014$ |
| 768 | 512 | 0.333 | 0.3467 | 1.04 | 0.0002 |
| 768 | 768 | 0 | $0.0183 \pm 0.0017$ | | $0.0004 \pm 0.0001$ |
| 1024 | 256 | 0.750 | 0.7754 | 1.03 | 0.0006 |
| 1024 | 384 | 0.625 | $0.6427 \pm 0.0008$ | 1.03 | $0.0006 \pm 0.0008$ |
| 1024 | 512 | 0.500 | $0.5130 \pm 0.0004$ | 1.03 | $0.0005 \pm 0.0005$ |
| 1024 | 768 | 0.250 | 0.2598 | 1.04 | 0.0014 |
| 1024 | 1024 | 0 | $0.0170 \pm 0.0011$ | | $0.0009 \pm 0.0005$ |

## D COMBINATION RULES

Table 3: The two combination rules on the constrained cells of Table 2. $c = 0.021, 0.018, 0.017$ for $n = 384, 768, 1024$.

| $n$ | $d$ | Measured | $\max\{f, c\}$ | Error | $f + c$ | Error |
|---|---|---|---|---|---|---|
| 384 | 96 | 0.8054 | 0.750 | 6.9% | 0.771 | 4.3% |
| 384 | 128 | 0.7096 | 0.667 | 6.1% | 0.688 | 3.1% |
| 384 | 192 | 0.5284 | 0.500 | 5.4% | 0.521 | 1.4% |
| 384 | 256 | 0.3537 | 0.333 | 5.8% | 0.354 | 0.2% |
| 768 | 192 | 0.7864 | 0.750 | 4.6% | 0.768 | 2.3% |
| 768 | 256 | 0.6933 | 0.667 | 3.8% | 0.685 | 1.2% |
| 768 | 384 | 0.5183 | 0.500 | 3.5% | 0.518 | 0.0% |
| 768 | 512 | 0.3467 | 0.333 | 3.8% | 0.352 | 1.4% |
| 1024 | 256 | 0.7754 | 0.750 | 3.3% | 0.767 | 1.1% |
| 1024 | 384 | 0.6427 | 0.625 | 2.8% | 0.642 | 0.1% |
| 1024 | 512 | 0.5130 | 0.500 | 2.5% | 0.517 | 0.8% |
| 1024 | 768 | 0.2598 | 0.250 | 3.8% | 0.267 | 2.8% |
| | | | mean | 4.4% | | 1.6% |

## E ADDITIONAL TABLES

Table 4: Architectural attempts to escape the floor at $n = 768$, fp32. Recovered: fraction of the distance from the linear readout to zero error. Parameters (widths 256, 384): linear readout and skip parameterization 14.9M and 33.2M, MLP readout 17.9M and 36.4M, long skips 15.7M and 34.9M.

| | Width 256 | | Width 384 | |
|---|---|---|---|---|
| Variant | Error | Recovered | Error | Recovered |
| floor $+ c$ | 0.698 | | 0.531 | |
| linear readout | 0.7169 | | 0.5383 | |
| MLP readout, hidden $4n$, GELU | 0.6992 | 2.5% | 0.5355 | 0.5% |
| U-ViT long skips | 0.6971 | 2.8% | 0.5235 | 2.7% |
| skip parameterization | 0.0098 | 98.6% | 0.0143 | 97.3% |

Table 5: The closed form against the published table of Zheng et al. (2026). Constrained cells are those with $d < n$; error is relative to the prediction.

| Encoder ($n$) | Model ($d$) | $d/n$ | Predicted | Reported | Error |
|---|---|---|---|---|---|
| DINOv2-B (768) | DiT-S (384) | 0.50 | 0.500 | 0.520 | 4% |
| DINOv2-L (1024) | DiT-B (768) | 0.75 | 0.250 | 0.270 | 8% |
| DINOv2-L (1024) | DiT-S (384) | 0.38 | 0.625 | 0.650 | 4% |
| six cells with $d \geq n$ | | $\geq 1$ | 0 | 0.001–0.036 | |

Table 6: Rank penalty on 200,000 images at $d/n = 0.5$, as a difference across width. For direct the prediction is the isotropic floor $1 - d/n$ of Proposition 1; for clean the interval is (10), computed from the exact token spectrum with $t \sim U(0, 0.95)$.

| Encoder | $n$ | Arm | Excess | Predicted |
|---|---|---|---|---|
| DINOv2-S/14 | 384 | direct | $+0.5233$ | 0.500 |
| DINOv2-B/14 | 768 | direct | $+0.4840$ | 0.500 |
| DINOv2-L/14 | 1024 | direct | $+0.4602$ | 0.500 |
| DINOv2-S/14 | 384 | clean | $+0.1766$ | $[0.0814,\ 0.2114]$ |
| DINOv2-B/14 | 768 | clean | $+0.1735$ | $[0.0835,\ 0.2164]$ |
| DINOv2-L/14 | 1024 | clean | $+0.1624$ | $[0.0811,\ 0.2111]$ |

Table 7: Width needed for a floor of at most $0.05$, from the spectra of Appendix F. Clean is $r_{0.05}(z)$, the 95% column of Table 9. Direct is $n(1-0.05)$ rounded up, where the isotropic part alone reaches $0.05$; the clean tail raises it slightly for DINOv2 latents, so the savings shown are lower bounds.

| Target | $n$ | Direct | Clean | Saving |
|---|---|---|---|---|
| DINOv2-S/14 latents | 384 | 365 | 311 | $1.2\times$ |
| DINOv2-B/14 latents | 768 | 730 | 631 | $1.2\times$ |
| DINOv2-L/14 latents | 1024 | 973 | 835 | $1.2\times$ |
| ImageNet $64^2$, patch 16 | 768 | 730 | 62 | $11.8\times$ |
| ImageNet $256^2$, patch 16 | 768 | 730 | 26 | $28.1\times$ |

Table 8: ImageNet $256 \times 256$, patch 16, 256 tokens, $n = 768$, one seed per cell, 100,000 steps.

| $d$ | Arm | FID | KID | IS | Prec. | Rec. |
|---|---|---|---|---|---|---|
| 768 | clean | 173.19 | .160 | 5.40 | .144 | .056 |
| 768 | direct | 322.79 | .351 | 1.97 | .039 | .000 |
| 256 | clean | 185.15 | .155 | 3.40 | .203 | .017 |
| 256 | direct | 287.54 | .236 | 1.52 | .033 | .000 |

## F TARGET SPECTRA

Table 9: Spectra of the clean target $z$. Rank $p\%$ is the number of directions holding $p$ percent of the variance, so the 95% column is $r_{0.05}(z)$. The participation ratio is $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$.

| Target | $n$ | Part. ratio | 90% | 95% | 99% |
|---|---|---|---|---|---|
| DINOv2-S/14 latents | 384 | 278.2 | 266 | 311 | 362 |
| DINOv2-B/14 latents | 768 | 559.4 | 539 | 631 | 728 |
| DINOv2-L/14 latents | 1024 | 731.3 | 712 | 835 | 966 |
| ImageNet $64^2$, patch 16 | 768 | 10.5 | 23 | 62 | 226 |
| ImageNet $256^2$, patch 16 | 768 | 4.8 | 8 | 26 | 150 |

Higher-resolution patches are more nearly low-rank, with participation ratio 4.8 against 10.5, because neighbouring pixels are more correlated. At $d/n = 0.5$ the interval (10) is $[0.0001, 0.0022]$ for $64 \times 64$ patches and $[0.0000, 0.0010]$ for $256 \times 256$ patches.

## G VERIFICATION OF THE SAMPLING PROPOSITIONS

We sample 256 token sequences from Gaussian noise with 50 Euler steps from each trained width-256 generator on ImageNet $64 \times 64$, and measure the energy in the orthogonal complement of $\mathrm{span}(W, b)$, of dimension 511, before and after.

Table 10: Residual energy per token outside the output span, trained width-256 generators. The direct checkpoint is its one finite seed.

| Arm | Before | After | Kept | Relative change |
|---|---|---|---|---|
| direct | 511.5 | 511.5 | 100.0% | $7.6 \times 10^{-7}$ |
| EDM | 510.7 | 485.1 | 95.0% | $2.5 \times 10^{-2}$ |
| clean | 510.7 | 0.0 | 0.0% | 1.0 |

The EDM value is the Euler product of Corollary 1, which keeps 90.2, 95.0, 97.5 and 99.7% of the energy at $K = 25$, 50, 100 and 1000 steps. The same three values are obtained with arbitrary untrained weights. In the ideal samples of Table 1, the per-token data energy inside the span is 189.9 for direct against 510.9 of residual noise, so the noise dominates each patch by more than two to one.

## H NUMERICAL PRECISION AT NARROW WIDTH

An earlier pass used bfloat16 autocast. At widths of 384 and below it produced errors non-monotone in width, over forty times the floor, and NaN in several configurations; identical configurations and seeds in fp32 fall on the predicted floor.

Table 11: Identical configurations and seeds, differing only in precision.

| Configuration | Precision | Error | Floor | Above floor |
|---|---|---|---|---|
| $n$=384, $d$=192, direct, seed 2 | bfloat16 | NaN | 0.500 | |
| $n$=384, $d$=192, direct, seed 2 | fp32 | 0.5225 | 0.500 | +4.5% |
| $n$=384, $d$=128, direct, seed 0 | bfloat16 | 30.29 | 0.667 | +4440% |
| $n$=384, $d$=128, direct, seed 0 | fp32 | 0.7124 | 0.667 | +6.8% |

At width 512 and above the two precisions agree within one percent; we did not isolate the cause of the narrow-width failures. The fp32 generation runs that diverged (Section 5.3) are a separate phenomenon: they occur at full precision and only at the constrained width.