REPRESENTATION GEOMETRY IN DIFFUSION TRANSFORMERS: WHAT MATTERS FOR GENERATION QUALITY?

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow-matching models are the dominant paradigm for image generation, but are notoriously expensive to train. A recent line of work accelerates training and improves generative performance by shaping the internal representations of these models through auxiliary objectives. However, it is unclear which properties actually matter and how to reliably identify them. To address this, we measure eight properties of diffusion transformer (SiT-XL/2) representations and test whether they track generative performance in three complementary settings: a single training trajectory, 16 REPA and iREPA models that differ only in the vision encoder they are aligned to, and models trained with different representation guidance objectives. Several properties correlate strongly with generative performance, measured by Fréchet Inception Distance (FID), along the training trajectory, yet several of these properties do not separate better from worse models once training has converged. This shows that correlation along a trajectory alone is insufficient evidence that a property matters for generation. We confirm this with interventional experiments, in which turning such properties into auxiliary training objectives yields no or only marginal gains. We find that properties that remain predictive throughout our experiments correspond largely to those that are explicitly or implicitly optimized by recent successful methods. These results offer principled guidance for the design of new representation-shaping objectives.

1 INTRODUCTION

Diffusion and flow models have emerged as the dominant paradigm for generative modeling in the image domain, achieving photorealistic results (Ma et al., 2024; Peebles & Xie, 2023; Rombach et al., 2022). However, high-quality generation comes at the cost of long training times, even on specialized modern hardware.

A recent line of work has established a fruitful link between such generative models and representation learning. While learning to generate, these models build hidden representations of images that are also useful for other tasks such as classification (Xiang et al., 2023; Yang & Wang, 2023; Chen et al., 2024), and during training, generation quality and linear-probe accuracy have been observed to improve together (Xiang et al., 2023; Yu et al., 2024). Building on this link, several methods add an auxiliary objective, a second training loss that shapes the hidden representations. Some align them with the representations of a pretrained vision encoder (Yu et al., 2024; Singh et al., 2025; Wu et al., 2025; Leng et al., 2025), and others use self-contained objectives that need no external model (Jiang et al., 2025; Wang & He, 2025; Yun et al., 2025; Chefer et al., 2026; Yang et al., 2026). Both kinds consistently improve the Fréchet Inception Distance (FID) (Heusel et al., 2017), a standard measure of generated image quality, and reduce training cost. Knowing which properties of the model's own representations make them good for generation would make it possible to design objectives that target these properties directly and reduce training cost further. This raises the question that guides this paper.

Which properties of diffusion representations matter for generation quality?

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.

Testing a candidate property directly is expensive, because each candidate requires its own auxiliary loss and a model trained from scratch. A common heuristic avoids this cost by observing which properties emerge during training while generation quality improves, taking this as evidence that they matter, and designing objectives that encourage them (Yu et al., 2024; Leng et al., 2025; Wu et al., 2025). Along a single training run, however, generation quality improves with training time, so any property that also changes steadily during training correlates with it, whether or not it matters for generation. Is emergence during training then a reliable indicator of what matters? A property that matters should also distinguish better from worse models that were trained independently of each other.

Systematic comparisons across independently trained models have already proven useful for the components that feed into a diffusion model. iREPA (Singh et al., 2025) compares diffusion models aligned with many different pretrained encoders, relates properties of each encoder to the resulting generation quality, and uses the result to design its method. Similarly, a systematic study across many autoencoders relates properties of their latent spaces, in which the diffusion model generates, to generation quality (Zhong et al., 2026). Both analyses concern what the diffusion model is aligned with or trained on, not its own hidden representations. Comparisons of the hidden representations across models do exist, but they examine a single property, appear as side analyses, or mix checkpoints of one training run with independently trained models (Xiang et al., 2023; Chen et al., 2024; Yu et al., 2024; Li et al., 2026a). To the best of our knowledge, a systematic comparison of the properties of hidden representations across independently trained models is still missing.

We provide such a systematic cross-model evaluation. It only requires evaluating existing trained models, so testing a candidate property costs an evaluation instead of a training run. We measure eight properties of the hidden representations of diffusion transformers (Peebles & Xie, 2023; Ma et al., 2024), namely linear-probe accuracy, effective rank, ICR, CKNNA, uniformity, LDS, augmentation invariance and the class-scatter ratio, which capture different aspects of the representations (Fig. 1). We compare them in three settings. The first follows checkpoints of a single standard training run and applies the heuristic at face value. The second compares models trained with the same alignment method but different pretrained encoders, and the third compares models trained with different auxiliary objectives, including one trained without any. We report every association as an estimate with a confidence interval and read the results as degrees of support.

Nearly every property we examine correlates strongly with FID during training, yet several associations weaken across independently trained models.

Some of these associations even change sign, while others remain strong, so emergence during training alone is not a reliable indicator of what matters. Linear-probe accuracy, effective rank, ICR and CKNNA remain predictive, keeping the sign of their training-run correlation in both comparisons of independently trained models. Of these, only linear-probe accuracy has a confidence interval that excludes zero in both. For effective rank, we further show that under simplifying assumptions, fitting the flow-matching objective well requires a minimum effective rank, a necessary condition that does not imply that raising the effective rank improves generation.

We relate this evidence to how well the properties work as targets of auxiliary losses. Published objectives with large FID gains already encourage two of the properties that remain predictive. LSEP optimizes linear-probe accuracy directly with a classification loss (Yun et al., 2025), and REPA aligns the features with a pretrained encoder, which implicitly increases the similarity measured by CKNNA (Yu et al., 2024). Conversely, when we train models with auxiliary losses on LDS and augmentation invariance, two properties whose associations weaken across models, generation quality does not improve consistently over a control run in the tested settings.

We therefore propose using cross-model evidence to prioritize which representation properties to target with auxiliary objectives. Properties whose correlation with FID is strong and keeps its sign across independently trained models are the better candidates for training experiments.

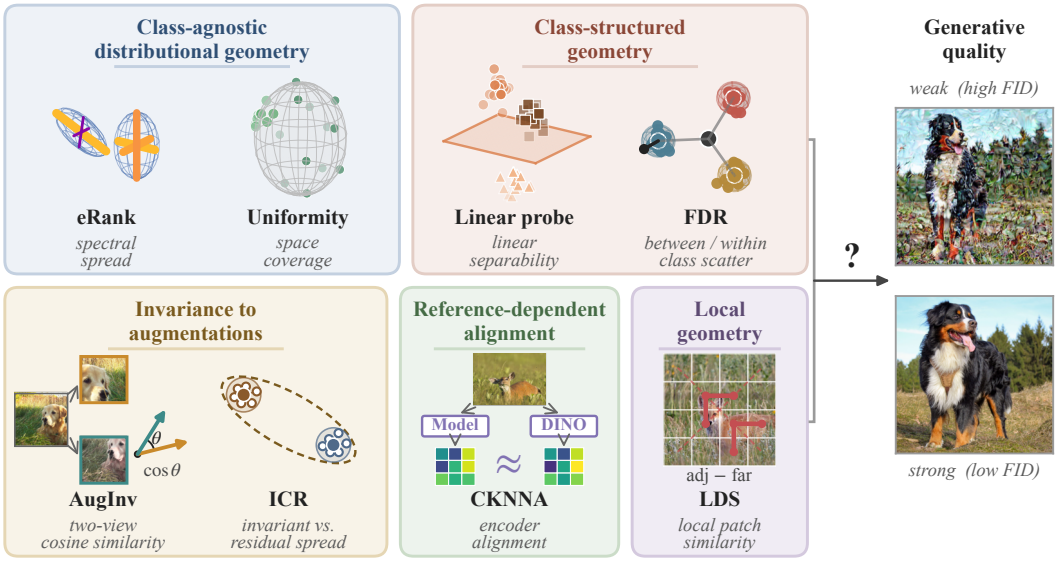


Figure 1: The eight properties, grouped by the aspect of the representation they describe. We ask which of them matter for generation quality, measured by FID.

Contributions.

- **A systematic cross-model evaluation.** We provide a systematic evaluation of how well eight properties of the hidden representations predict generation quality across independently trained models.
- **Relation to auxiliary losses.** We show that the properties that remain predictive across models include those that published objectives with large FID gains optimize, explicitly or implicitly. For two properties that do not, LDS and augmentation invariance, we show with interventional training runs that auxiliary losses on them improve FID at most marginally.
- **Theoretical analysis.** We link effective rank to the flow-matching loss. Under simplifying assumptions, fitting this loss well requires a minimum effective rank, which ties effective rank to the generative objective itself.

2 Many Plausible Properties Emerge and Track FID During Training

Since models that generate better also tend to have better representations (Section 1), the properties that representation learning values are natural places to look for what makes a representation good for generation. We study eight, grouped by the aspect of the representation they describe (Fig. 1). Section C gives precise definitions, Section D.2 implementation details and Section B prior work on each.

- **Spread.** Effective rank (eRank) (Roy & Vetterli, 2007) and uniformity (Wang & Isola, 2020) measure how many directions the features use and how evenly they fill the space. Contrastive self-supervised learning encourages uniform features (Wang & Isola, 2020) and can suffer from features collapsing onto few directions (Jing et al., 2021); a generator plausibly needs both properties to keep different images apart, and a loss that spreads diffusion features out improves generation (Wang & He, 2025).
- **Class structure.** Linear-probe accuracy (Alain & Bengio, 2016) and the class-scatter ratio (Fisher discriminant ratio, FDR), computed from the between- and within-class scatter matrices (Rao, 1948), measure how well a linear classifier trained on the frozen features recognizes the class and how far apart the class means lie relative to the spread within classes. A class-

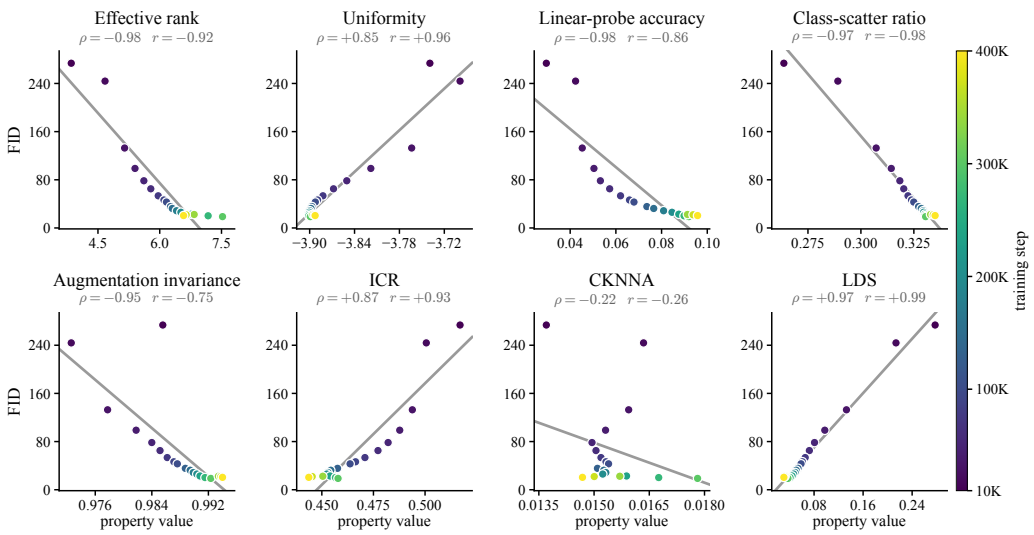


Figure 2: FID against each property across the 19 checkpoints of the training run, at layer 12 and $t = 0.25$. Each point is a checkpoint, colored by training step, and lines are least-squares fits. Panel headers give the Spearman (ρ) and Pearson (r) correlations with FID.

conditional generator plausibly has to represent which class it is drawing, and a loss that makes diffusion features linearly separable by class improves generation (Yun et al., 2025).

- **Invariance.** Augmentation invariance (AugInv) is the cosine similarity between the features of two differently augmented and noised views of an image, and ICR (Invariant Contamination Ratio) compares the variance the two views share with the variance in which they differ. Self-supervised learning pulls differently augmented views of an image together (Chen et al., 2020; He et al., 2020; Grill et al., 2020; Caron et al., 2021), a moderate degree of invariance could let a generator separate what an image shows from its crop or color, and ICR decreases together with FID during training (Li et al., 2026b).
- **Similarity to a pretrained encoder.** CKNNA (Huh et al., 2024) measures how similar the features of the diffusion model are to those of a pretrained encoder, here DINOv2 (Oquab et al., 2023). Aligning diffusion features with such encoders improves generation (Yu et al., 2024; Singh et al., 2025; Wu et al., 2025; Leng et al., 2025).
- **Local spatial structure.** LDS compares how similar the features are at nearby and at distant locations of an image. Images are locally coherent, and iREPA finds that this spatial structure, measured on the encoder used for alignment, predicts generation quality better than the encoder's linear-probe accuracy (Singh et al., 2025).

If a property matters for generation, a model trained only to generate should develop it on its own. Accordingly, properties are often supported by showing that they develop during training while FID improves (Li et al., 2026b; Yu et al., 2024; Xiang et al., 2023; Wu et al., 2025). We therefore measure the eight properties along the training trajectory of a SiT-XL/2 diffusion transformer (Ma et al., 2024) trained on ImageNet-256 (Deng et al., 2009) without any auxiliary objective, using 19 checkpoints up to 400K steps (Section D.1). For this, we extract the features of real images at layer 12 of 28 and noise level $t = 0.25$, where $t = 0$ is the clean image and $t = 1$ pure noise. Diffusion features are most informative at middle layers and small noise levels (Xiang et al., 2023). We investigate the robustness of our findings to these choices in Section E.1. We use the Spearman correlation with FID, computed from 50K samples without guidance (Section D.3), as the central correlation coefficient in figures and tables, and also report the Pearson correlation. We center on Spearman because it captures any monotone relation and is less sensitive to outliers than the Pearson correlation.

Seven of the eight properties track FID closely, with training-run correlations between 0.85 and 0.98 in absolute value (Fig. 2). As FID improves, effective rank, linear-probe accuracy, the class-scatter

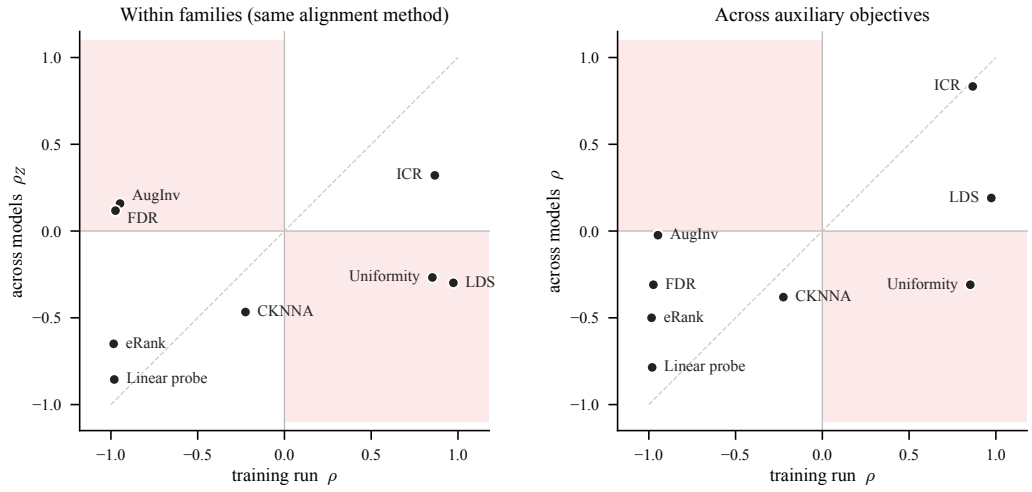


Figure 3: Spearman correlations with FID across independently trained models against those along the training run, one point per property, at layer 12 and $t = 0.25$. Left: partial correlations within families, among models with the same alignment method. Right: correlations across auxiliary objectives. Shaded quadrants mark correlations whose sign differs from the training run, and the dashed line marks equal values. Table 1 gives the intervals.

ratio and augmentation invariance increase, while uniformity, ICR and LDS decrease, where lower uniformity and ICR mean more uniform and more invariant features. CKNNA is the exception, with a training-run correlation of -0.22. Taken as evidence in the usual way, the run supports seven of the eight properties at once. To check whether it is good evidence that these properties matter for generation, the next section puts them to a second test, across independently trained models.

3 Do These Properties Still Tell Good Models from Bad Ones?

Across independently trained models, a property can no longer track FID merely because both change with training time. Such models can, however, differ in many ways at once, so we first compare models that differ in only one. Singh et al. (2025) released SiT-XL/2 models trained for 400K steps with REPA (Yu et al., 2024) or iREPA, with eight different pretrained encoders as the alignment target and all other settings fixed (Section D.1). In our evaluation, the encoder alone changes FID from 7.5 to 10.1 for REPA and from 6.9 to 8.2 for iREPA. The method is a second, known factor that shifts FID and several properties. Since every encoder appears with both methods, the two factors are not entangled, and a partial Spearman correlation removes the effect of the method cleanly (Section D.4). We call this the within-family comparison.

A property that makes representations good for generation should matter however the representation is shaped, not only through alignment. We therefore also compare eight SiT-XL/2 models that solve the same task, class-conditional generation on ImageNet-256, with different objectives: a standard SiT-XL/2 without an auxiliary objective (Ma et al., 2024), four models aligned with a pretrained encoder, REPA, iREPA, REG and REPA-E (Yu et al., 2024; Singh et al., 2025; Wu et al., 2025; Leng et al., 2025), and three with self-contained objectives, SRA, Dispersive Loss and Self-Flow (Jiang et al., 2025; Wang & He, 2025; Chefer et al., 2026). We call this the auxiliary-objective comparison. Each objective changes many properties at once and in its own way, so a property that matters should still go with FID across these models, while one that changes only as a side effect of a particular objective should not. We evaluate the released checkpoints, trained for 400K to 7M steps (Table 4). All of them are trained into a strong generative regime, in which the budget no longer orders them by FID (Spearman correlation -0.07).

Figure 3 plots the correlations within families, among models with the same alignment method, and across auxiliary objectives against those along the training run, and Table 1 gives their intervals.

Table 1: Spearman (ρ) and Pearson (r, gray) correlations with FID at layer 12 and $t = 0.25$. A negative correlation means that a higher value of the property goes with lower FID. Training-run values are those of Fig. 2. Within families, both are partial correlations that remove the difference between REPA and iREPA. Intervals are 95% bootstrap intervals from 5000 resamples, drawn separately for REPA and iREPA models within families. Where Spearman and Pearson differ strongly across auxiliary objectives (LDS, augmentation invariance), the Pearson value rests on one model.

	Training run 19 checkpoints		Within families 16 models			Across auxiliary objectives 8 models		
Property	ρ	r	ρ_Z	95% CI	r_Z	ρ	95% CI	r
Effective rank	-0.98	-0.92	-0.65	$[-0.89, +0.00]$	-0.79	-0.50	$[-0.95, +0.32]$	-0.33
Uniformity	$+0.85$	$+0.96$	-0.27	$[-0.69, +0.30]$	-0.40	-0.31	$[-0.92, +0.54]$	-0.13
Linear-probe accuracy	-0.98	-0.86	-0.86	$[-0.99, -0.17]$	-0.86	-0.79	$[-1.00, -0.14]$	-0.68
Class-scatter ratio	-0.97	-0.98	$+0.12$	$[-0.46, +0.76]$	$+0.11$	-0.31	$[-0.92, +0.54]$	-0.34
Augmentation invariance	-0.95	-0.75	$+0.16$	$[-0.46, +0.68]$	$+0.08$	-0.02	$[-0.80, +0.75]$	-0.40
ICR	$+0.87$	$+0.93$	$+0.32$	$[-0.12, +0.73]$	$+0.70$	$+0.83$	$[+0.32, +1.00]$	$+0.62$
CKNNA	-0.22	-0.26	-0.47	$[-0.80, +0.18]$	-0.74	-0.38	$[-0.93, +0.44]$	-0.43
LDS	$+0.97$	$+0.99$	-0.30	$[-0.67, +0.32]$	-0.42	$+0.19$	$[-0.85, +1.00]$	$+0.81$

Along the training run, seven of the eight properties correlate with FID almost equally strongly. Across independently trained models, their correlations spread out, and those of uniformity, LDS and augmentation invariance reverse within families, changing sign relative to the training run. They do so at most layers and noise levels we test (Section E.1). The class-scatter ratio also changes sign within families at this layer and noise level, but only in 6 of the 15 we test. A single training run therefore does not tell these properties apart, because training time moves every steadily changing property together with FID.

The cross-model correlations also separate properties that describe the same aspect of the representation. Linear-probe accuracy and the class-scatter ratio both measure class structure and tracked FID equally closely during training, but within families only linear-probe accuracy keeps its correlation. ICR and augmentation invariance are computed from the same augmented views, yet only ICR keeps the sign of its training-run correlation in both comparisons. The evidence therefore attaches to a specific measurement rather than to a broad aspect such as class structure or invariance.

Linear-probe accuracy has the clearest support across models. Its correlation within families is the strongest and the only one whose interval excludes zero, and it keeps its sign at every layer and noise level in both comparisons. Effective rank and CKNNA have moderate correlations within families; across auxiliary objectives, they keep their sign at every layer and noise level, but their intervals are wide. ICR is weak within families but strong across auxiliary objectives, where its interval excludes zero. We read CKNNA cautiously, since it measures similarity to a pretrained encoder and most of these models were trained to align with one. Uniformity, LDS, augmentation invariance and the class-scatter ratio have weaker or less consistent correlations. The cross-model evidence thus gives weaker support to several properties than their training-run correlations suggest. Testing directly whether the properties whose predictiveness weakens do not matter for generation quality requires training models with objectives that change these properties, which Section 4 does for LDS and augmentation invariance. Section 5 then puts the properties with more consistent support into context and discusses what this support does and does not imply.

4 Building Losses on Selected Properties

Within families, uniformity, LDS and augmentation invariance reverse the sign of their training-run correlation at most layers and noise levels (Section 3). These reversals suggest that the training-run correlations overstate how much the three properties matter for generation, and if they do, pushing them further along their training-run trend should not improve FID. We test this prediction with auxiliary losses on LDS and augmentation invariance, whereas uniformity needs no new run because a published loss already pushes it in this direction.

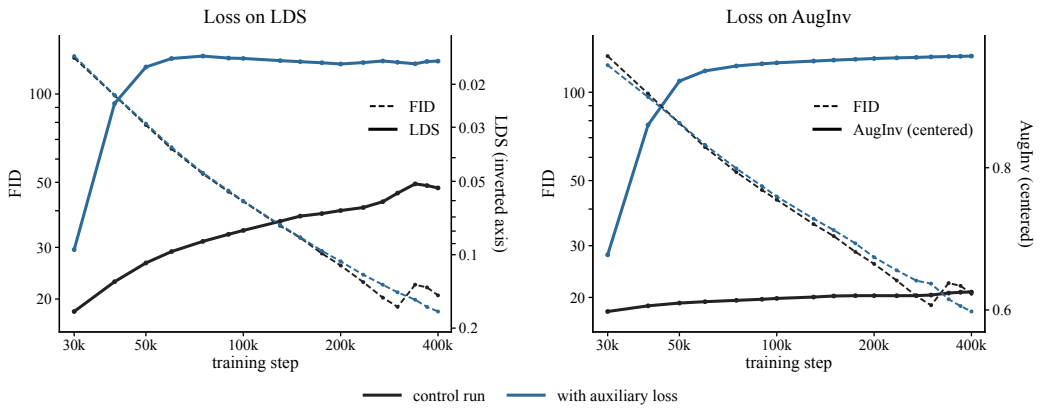


Figure 4: Auxiliary losses on LDS (left) and augmentation invariance (right), each branching off the control run at 25K steps. Solid lines show the targeted property at layer 8 and $t = 0.25$ (right axis), dashed lines FID (left axis). Augmentation invariance is measured without the component shared by all images (Section D.5). The LDS axis is inverted, so that in both panels up is the direction in which the loss pushes the property. FID, LDS and training steps are on log scales.

Each run with an auxiliary loss starts from the training run of Section 2 at 25K steps and continues to 400K steps with the recipe otherwise unchanged, so that the training run serves as the control run. We start at 25K steps because before then the model is still forming its representation, and until about 20K steps both properties move against the direction they follow for the rest of training. Following the heuristic of Section 1, each loss pushes its property in the direction in which it moves while FID improves, lowering LDS and raising augmentation invariance (Fig. 2). Optimizing a metric directly, however, can move it without the intended change in the representation. LDS falls whenever one feature direction dominates every token, because nearby and distant tokens then look equally similar, and augmentation invariance rises whenever a component shared by all images grows, because any two images then look alike. The losses are therefore built to rule out such shortcuts (Section D.5). Both losses act on layer 8, where REPA, iREPA and LSEP also act in SiT-XL/2 (Yu et al., 2024; Singh et al., 2025; Yun et al., 2025), and we measure each property at this layer to check that its loss moved it as intended.

By 40K steps, 15K steps after branching, each loss has moved its property beyond any value the control run reaches in 400K steps, and the property remains there for the rest of training (Fig. 4). FID, in contrast, stays within 7% of that of the control run up to 200K steps. The runs with the auxiliary loss as an intervention end below the control run after its FID rises between 300K and 340K steps, a rise that their own FID does not show, and the best FID along each run lies 3.6% (LDS) and 4.8% (augmentation invariance) below the best FID of the control run, from one run each. In the tested setting, pushing LDS or augmentation invariance in the direction of its training-run correlation therefore does not consistently improve generation.

The published loss on uniformity is Dispersive Loss (Wang & He, 2025), which pushes the hidden features of different images apart with an objective of the same form as uniformity, computed on mini-batches of features that are neither centered nor normalized. We report its gains together with those of other objectives that need no pretrained encoder, each with its training length and guidance setting. Relative to a standard SiT-XL/2 trained for 7M steps, Dispersive Loss lowers FID by 10.0% without guidance and by 4.4% with guidance after about 6M steps, whereas Self-Flow (Chefer et al., 2026) lowers it by 31% without guidance after 4M steps. After 400K steps without guidance, LSEP (Yun et al., 2025) lowers FID by 40% relative to a standard SiT-XL/2 trained equally long, compared with 14% for Dispersive Loss, each relative to its own baseline and with a different sampler. LSEP makes the features linearly separable by class and thus targets linear-probe accuracy, the property with the clearest cross-model support. Section 5 discusses this property together with the others whose cross-model support is more consistent.

5 The Properties with More Consistent Support, in Context

Linear-probe accuracy has served as the main evidence that diffusion models which generate better also learn better representations. This evidence comes from single training runs or from variants of one method. Along training, and across REPA models of different sizes or with different alignment encoders, linear-probe accuracy improves together with FID (Xiang et al., 2023; Yu et al., 2024). Across the independently trained models we compare, including models trained with different objectives, linear-probe accuracy has the most consistent association with FID of all eight properties. This agrees with the gains of LSEP, which targets linear separability directly and, after 400K steps without guidance, lowers FID by 40% for SiT-XL/2 and by 35% for SiT-L/2 relative to standard models trained equally long (Yun et al., 2025). The two kinds of evidence, from comparing trained models and from changing the objective, point the same way.

ICR was proposed more recently (Li et al., 2026b) and has been supported mainly as a measure of representation quality that needs neither labels nor an external network. Its authors show that ICR is lowest at the noise levels where linear-probe accuracy peaks, and that in training on limited data, ICR reaches its minimum before the model starts to memorize its training images, which makes it a signal for when to stop training. Its link to generation quality, however, comes from training runs, along which ICR and FID decrease together. We extend this link to independently trained models, using our reimplementation of ICR (Section D.2). Across auxiliary objectives, ICR is associated with FID as strongly as linear-probe accuracy, whereas within families its association is weaker.

CKNNA, the similarity of the features to those of DINOv2, has likewise been tracked alongside FID during training and framed as a desirable property (Wu et al., 2025; Leng et al., 2025). Its cross-model associations are moderate and imprecise, and they may partly reflect how the models were trained rather than what generation requires. All models in the within-family comparison and four of the eight in the auxiliary-objective comparison were trained with REPA or a method built on it, which aligns their features with those of a pretrained encoder and so implicitly encourages what CKNNA measures.

Effective rank, by contrast, is not targeted explicitly by any of the objectives we compare, and for diffusion models it has mainly served to measure how informative the features are at different noise levels (Yang & Wang, 2023). Yet it stays positively associated with generation quality in all three settings, and relating it to the excess risk of the flow-matching objective gives intuition for why a model that fits this objective well should have a reasonably high effective rank. Let Σ be the covariance, across images, of the pooled features, i.e. the features averaged over all tokens, at a given layer and noise level. Effective rank is computed from Σ, whose total variance is $\operatorname{Tr}\Sigma$ and whose largest variance along a single direction is $\lambda_{\max}(\Sigma)$. The objective trains the network to predict, for each token, the velocity of the path between data and noise, and we write $\lambda^\star$ for the largest variance of the token-averaged velocity that minimizes the objective.

Proposition 5.1 (Informal version of Theorem F.1)**.** *Fix a layer and a noise level, and assume that*

(i) from this layer on, the network computes each token's predicted velocity from the token's features through a fixed affine map shared by all tokens,
(ii) the direction in which the features vary most is mostly one that this map reads out, and
(iii) the features have total variance $\operatorname{Tr}\Sigma \geq \tau > 0$.

If the flow-matching loss at this noise level is close to its minimum, then

$$\operatorname{eRank}(\Sigma) \;\geq\; \frac{\operatorname{Tr}\Sigma}{\lambda_{\max}(\Sigma)} \;\geq\; \frac{c\,\tau}{\lambda^\star + \delta}, \tag{1}$$

where δ vanishes as the loss approaches its minimum and $c > 0$ measures how strongly the map passes the variance along the top feature direction on to the predicted velocity.

The proof reads Eq. (1) from right to left (Section F). A model whose loss is close to its minimum predicts a velocity whose covariance is close to that of the optimal velocity, so the variance of its prediction along any direction is at most $\lambda^\star+\delta$ (Proposition F.2). By assumption (i), this prediction is the readout of the features, and by assumption (ii), the readout passes on at least c times the variance along the top feature direction. No single feature direction can therefore hold more variance than $(\lambda^\star + \delta)/c$ (Lemma F.3). Since effective rank is always at least the total variance divided by the largest variance (Lemma F.4), assumption (iii) then bounds it from below.

The three assumptions differ in strength. Assumption (i) is the strong simplification. It holds exactly only for the features that enter the output layer, and at layer 12 it replaces the remaining blocks, with their nonlinearity and the attention between tokens, by a single affine map applied to each token separately, which ignores the class label. Assumption (ii) is intuitive, because the network has to use the variance it puts into its features for its prediction, and a dominant direction that the prediction ignores would carry the largest variance without affecting the output. Assumption (iii) is mild. It only rules out features that collapse to near-zero variance, and because rescaling the features rescales the map inversely, the product $c\tau$ does not depend on their scale. The objective does not supply τ, though, and without (iii) it cannot guarantee an effective rank above 16, the output dimension per token (Section F.6). The result is therefore a necessary condition under simplifying assumptions, and it does not imply that raising the effective rank improves generation.

For all four properties, cross-model support strengthens the evidence but cannot show that raising the property improves generation, because the properties are observed in existing models rather than set by an intervention. For linear-probe accuracy, LSEP adds evidence of this second kind, whereas none of the objectives we compare targets effective rank, ICR or CKNNA directly. The more consistent cross-model associations can nonetheless help practitioners design new auxiliary losses, as a firmer starting point than the training-run correlations that nearly every property shows.

6 Conclusion

Along a single training run, nearly every property we examined tracks FID closely, but so does any property that changes steadily while the model trains. Across independently trained models, several of these associations weaken or reverse, while linear-probe accuracy has the clearest support. Auxiliary losses on LDS and augmentation invariance, two of the properties whose associations weaken, do not consistently improve generation in the tested setting. Emergence during training is not sufficient evidence that a property matters for generation. Comparing independently trained models strengthens or weakens the case for each property without settling it.

Before building an auxiliary loss around a representation property, we therefore recommend checking whether the property still distinguishes better from worse models that were trained independently. If its association with generation quality weakens or reverses across such models, the motivation drawn from the training run weakens with it, and further evidence is worth gathering before an expensive training run is committed to the property. A property that emerges during training is not thereby a bad target, only one that the evidence usually given for it does not yet support.

Several limitations of our analysis remain, which we leave for future work.

- All cross-model comparisons use SiT-XL/2 on ImageNet-256, and only the training run is repeated at two smaller sizes, where its correlations transfer (Section E.3). The comparisons can be extended to other model sizes, to other datasets and to DiT (Peebles & Xie, 2023).
- Generation quality is measured mainly by FID without guidance. Four further measures show the same weakening across independently trained models (Section E.2), and generation with guidance remains to be evaluated.
- The two cross-model settings contain 16 and 8 models, so the confidence intervals are broad. Adding models as more are released would narrow them.
- The effective-rank result rests on strong simplifying assumptions and does not predict the cross-model findings for effective rank. Weakening these assumptions is a natural next step.

While these limitations point to ways of extending the analysis, our findings and the cross-model perspective already let practitioners narrow the design space of auxiliary objectives, at the cost of evaluating existing checkpoints rather than training new models. We hope that this leads to a more principled design of auxiliary objectives.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022.

Rajendra Bhatia. *Matrix analysis*. Springer Science & Business Media, 2013.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9640–9649, 2021.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Bradley Efron and Trevor Hastie. *Computer age statistical inference, student edition: algorithms, evidence, and data science*, volume 6. Cambridge University Press, 2021.

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 370–382, 2025.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

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

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

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

Peter Holderrieth and Ezra Erives. Introduction to flow matching and diffusion models, 2026. URL `https://diffusion.csail.mit.edu/`.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2013.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

Olav Kallenberg. *Foundations of modern probability*. Springer, 1997.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2013.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025.

Binglei Li, Mengping Yang, Zhiyu Tan, Xiaomeng Yang, Zhizhong Huang, Junping Zhang, and Hao Li. Diversedit++: Quantifying, analyzing, and promoting representation diversity in diffusion transformers. *arXiv preprint arXiv:2608.03082*, 2026a.

Xiao Li, Yixuan Jia, Zekai Zhang, Xiang Li, Lianghe Shi, Jinxin Zhou, Zhihui Zhu, Liyue Shen, and Qing Qu. Evaluating the representation space of diffusion models via self-supervised principles. *arXiv preprint arXiv:2606.09718*, 2026b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2017.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

C Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

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

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? *arXiv preprint arXiv:2512.10794*, 2025.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

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

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15802–15812, 2023.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Ddae++: enhancing diffusion models towards unified generative and discriminative learning. *arXiv preprint arXiv:2505.10999*, 2025.

Mengping Yang, Zhiyu Tan, Binglei Li, Xiaomeng Yang, Hesen Chen, and Hao Li. Diversedit: Towards diverse representation learning in diffusion transformers. *arXiv preprint arXiv:2603.04239*, 2026.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18938–18949, 2023.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Junno Yun, Yaşar Utku Alçalar, and Mehmet Akçakaya. No alignment needed for generation: Learning linearly separable representations in diffusion models. *arXiv preprint arXiv:2509.21565*, 2025.

Zekai Zhang, Xiao Li, Xiang Li, Lianghe Shi, Meng Wu, Molei Tao, and Qing Qu. Generalization of diffusion models arises with a balanced representation space. *arXiv preprint arXiv:2512.20963*, 2025.

Tianxiong Zhong, Xingye Tian, Xuebo Wang, Xin Tao, and Pengfei Wan. Diffusing in the right space: A systematic study of latent diffusability. *arXiv preprint arXiv:2606.03578*, 2026.

Daniel Zwillinger and Stephen Kokoska. *CRC Standard Probability and Statistics Tables and Formulae*. Chapman & Hall/CRC, Boca Raton, FL, 1st edition, 2000. ISBN 978-1584880592.

APPENDIX CONTENTS

A Background and Notation 15

A.1 Flow Matching 15

A.2 Architecture 16

A.3 Hidden Representations 16

B Related Work 17

C Definitions of the Properties 18

C.1 Class-Agnostic Distributional Geometry 18

C.2 Class-Structured Geometry 19

C.3 Reference-Dependent Alignment 19

C.4 Local Geometry 20

C.5 Invariance to Augmentation 20

C.6 Orientation 21

D Implementation Details 21

D.1 Checkpoint Sets 21

D.2 Property Implementations 22

D.3 FID Protocol 24

D.4 Statistical Analysis 24

D.5 Interventional Training Runs 24

D.6 Compute 26

E Additional Results 26

E.1 Robustness Across Blocks and Times 27

E.2 Quality Measures Besides FID 30

E.3 Transfer Across Model Sizes 30

F Proof of the Effective-Rank Result 33

F.1 Setup 34

F.2 Covariance of the Pooled Bayes-Optimal Velocity 34

F.3 Operator-Norm Closeness of the Velocity Covariances 36

F.4 Spectral Control of the Feature Covariance 36

F.5 Main Result: An Effective-Rank Lower Bound 39

F.6 Scope and Connection to the Empirical Results 39

G Use of Large Language Models 40

Throughout the appendix, Set 1 denotes the training run of Section 2, Set 2 the within-family comparison and Set 3 the auxiliary-objective comparison of Section 3.

A Background and Notation

This appendix fixes the notation used throughout the appendix. It summarizes flow matching, its connection to score-based diffusion, the SiT architecture and the hidden representations whose properties we measure. The exposition is partly inspired by Holderrieth & Erives (2026), to which we refer for a more detailed introduction.

A.1 Flow Matching

All models considered in this work are trained to generate samples from the data distribution p_{data}, which throughout is the distribution underlying the ImageNet dataset (Deng et al., 2009). Stochastic interpolants (Albergo et al., 2025) connect the data distribution $p_0 := p_{\mathrm{data}}$ to a noise distribution p_1 through a stochastic process $(X_t)_{t\in[0,1]}$.

Definition A.1 (Stochastic interpolant)**.** *Given random variables $X_0 \sim p_0$ and $X_1 \sim p_1$, a stochastic interpolant is*

$$X_t := \alpha_t X_0 + \beta_t X_1, \qquad t \in [0,1], \tag{2}$$

where α_t and β_t are strictly monotonic, continuously differentiable real-valued functions of t with $\alpha_0 = 1 = \beta_1$ and $\alpha_1 = 0 = \beta_0$.

This is the spatially linear interpolant without a latent variable, a special case of the general construction of Albergo et al. (2025). Throughout, X_0 and $X_1 \sim \mathcal{N}(0, I)$ are drawn independently, and all models considered in this work use the linear interpolant $\alpha_t = 1 - t$, $\beta_t = t$. Hence $t = 0$ is data and $t = 1$ is pure noise.

A generative model evolves a process $(Y_t)_{t\in[0,1]}$ backward in time, from $Y_1 \sim p_1$ at $t = 1$ to $t = 0$, through a velocity field $u_t^\theta(x)$ parameterized by a neural network, that is, through the neural ODE $\frac{\mathrm{d}}{\mathrm{d}t} Y_t = u_t^\theta(Y_t)$ (Chen et al., 2018). The goal is that the marginals of Y_t match those of X_t for every t, in particular at $t = 0$. Because the interpolant already has the right marginals and its time derivative $\dot{X}_t = \dot{\alpha}_t X_0 + \dot{\beta}_t X_1$ is available in closed form given (X_0, X_1), the network can be trained by regression onto $\dot{X}_t$ (Lipman et al., 2022; Liu et al., 2022; Albergo & Vanden-Eijnden, 2022).

Definition A.2 (Flow-matching objective)**.** *The flow-matching objective is*

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, X_0,\, X_1}\Big[\|u_t^\theta(X_t) - \dot{X}_t\|^2\Big]. \tag{3}$$

For the linear interpolant, $\dot{X}_t = X_1 - X_0$.

The proof of Theorem F.1 in Section F uses the population minimizer of this objective, which we record here with its proof.

Proposition A.1 (Flow-matching minimizer)**.** *Let $u_t^\star(x) := \mathbb{E}\Big[\, \dot{X}_t \;\Big|\; X_t = x \,\Big]$. The flow-matching objective satisfies*

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\, X_0,\, X_1}\big[\|u_t^\theta(X_t) - u_t^\star(X_t)\|^2\big] + C, \tag{4}$$

where C does not depend on θ. In particular, $u_t^\star$ is the unique population minimizer of $\mathcal{L}$ up to almost-everywhere equality.

Proof. By the tower property, $\mathcal{L}(\theta) = \mathbb{E}_{t,\, X_0,\, X_1}\Big[\mathbb{E}\Big[\, \|u_t^\theta(X_t) - \dot{X}_t\|^2 \;\Big|\; X_t \,\Big]\Big]$. Adding and subtracting $u_t^\star(X_t)$ and expanding the squared norm,

$$\begin{aligned}\mathbb{E}\Big[\, \|u_t^\theta(X_t) - \dot{X}_t\|^2 \;\Big|\; X_t \,\Big] &= \|u_t^\theta(X_t) - u_t^\star(X_t)\|^2 \\ &\quad + 2\big\langle u_t^\theta(X_t) - u_t^\star(X_t),\; \mathbb{E}\Big[\, \dot{X}_t \;\Big|\; X_t \,\Big] - u_t^\star(X_t)\big\rangle \\ &\quad + \mathbb{E}\Big[\, \|\dot{X}_t - u_t^\star(X_t)\|^2 \;\Big|\; X_t \,\Big]. \end{aligned} \tag{5}$$

The cross term vanishes since $\mathbb{E}\Big[\, \dot{X}_t \;\Big|\; X_t \,\Big] = u_t^\star(X_t)$ by definition. Setting $C := \mathbb{E}_{t,\, X_0,\, X_1}\big[\mathbb{E}\Big[\, \|\dot{X}_t - u_t^\star(X_t)\|^2 \;\Big|\; X_t \,\Big]\big]$ and taking the outer expectation yields equation 4. Uniqueness follows since $\mathbb{E}_{t,X_0,X_1}[\|u_t^\theta(X_t) - u_t^\star(X_t)\|^2] = 0$ if and only if $u^\theta = u^\star$ almost everywhere. □

Connection to score-based diffusion. Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020) and Noise-Conditional Score Networks (NCSN) (Song & Ermon, 2019) generate samples with a stochastic rather than a deterministic process. Albergo et al. (2025) show that subtracting a multiple of the score $s_t(x) := \nabla_x \log p_t(x)$ from the drift of the ODE driven by $u_t^\star$ and injecting noise with any diffusion coefficient $\sigma(t) \geq 0$ gives a family of SDEs,

$$\mathrm{d}Y_t = \left[u_t^\star(Y_t) - \tfrac{\sigma(t)^2}{2}\, s_t(Y_t)\right] \mathrm{d}t + \sigma(t)\, \mathrm{d}\bar{B}_t, \qquad Y_1 \sim p_1, \tag{6}$$

integrated backward from $t = 1$ to $t = 0$ with a reverse-time Brownian motion $\bar{B}_t$, whose marginals coincide with those of the interpolant. Setting $\sigma \equiv 0$ recovers the ODE. Song et al. (2020) showed that DDPM and NCSN can be viewed as discretizations of SDEs, and the family equation 6 on $[0, 1]$ recovers those SDEs for nonzero σ and specific schedules (α_t, β_t), up to the terminal-distribution approximation of those works. For Gaussian $X_1 \sim \mathcal{N}(0, I)$, the score and the optimal velocity are related in closed form (Albergo et al., 2025),

$$u_t^\star(x) = a_t\, s_t(x) + r_t\, x, \qquad a_t := \beta_t^2 \frac{\dot{\alpha}_t}{\alpha_t} - \dot{\beta}_t \beta_t, \qquad r_t := \frac{\dot{\alpha}_t}{\alpha_t}. \tag{7}$$

A network trained on the flow-matching objective alone therefore determines the score, so one learns only u^θ and chooses at inference whether to integrate the ODE or an SDE of the family equation 6. All FID values we compute use an SDE sampler (Section D.3).

A.2 Architecture

All models considered in this work have the SiT-XL/2 architecture (Ma et al., 2024), which parameterizes the velocity field u^θ with the Diffusion Transformer backbone (Peebles & Xie, 2023).

Latent space. Following Rombach et al. (2022), SiT operates in the latent space of a pretrained variational autoencoder (VAE, Kingma & Welling (2013)). The SD-VAE encoder maps an image $x \in \mathbb{R}^{256\times256\times3}$ to a latent in $\mathbb{R}^{32\times32\times4}$, a 32×32 grid with 4 channels, and u^θ is learned in this latent space. At inference, one samples $Y_1 \sim \mathcal{N}(0, I)$ in the latent space, integrates the ODE or an SDE of the family equation 6 from $t = 1$ to $t = 0$ with the learned velocity field u^θ, and decodes the result Y_0 to pixels with the VAE decoder.

Tokens and blocks. Following the Vision Transformer (Dosovitskiy et al., 2020), the latent is divided into non-overlapping $p \times p$ patches with $p = 2$. Each patch is flattened and linearly projected to a d-dimensional token, and fixed sinusoidal 2D positional embeddings (Vaswani et al., 2017) are added. This gives $N = 256$ tokens on a 16×16 grid with $d = 1152$. The tokens pass through $L = 28$ transformer blocks, each consisting of multi-head self-attention with 16 heads and a position-wise MLP, both added as residuals. Each block is conditioned on a vector in $\mathbb{R}^d$, the sum of a small MLP applied to a sinusoidal embedding of t and a learned embedding of the class label $y \in \{1, \ldots, 1000\}$. The conditioning enters through adaptive layer normalization (adaLN): a per-block MLP maps this vector to scale, shift and gating parameters that modulate the normalized activations before the attention and MLP sublayers. After the last block, a linear readout maps every token to $\mathbb{R}^{d_v}$ with $d_v = 4p^2 = 16$, and the tokens are reassembled into the velocity $u_t^\theta \in \mathbb{R}^{32\times32\times4}$. The model has 675M parameters.

Null class. SiT models are trained with class dropout for classifier-free guidance (Ho & Salimans, 2022), so the class embedding table contains an additional null class $\varnothing$ that replaces the label with a fixed probability during training. We write $c \in \{y, \varnothing\}$ for the class input of a forward pass. Under *true-class conditioning* the network receives the image's label y. Under *null conditioning* it receives $\varnothing$ and therefore has no access to the label.

A.3 Hidden Representations

We denote the hidden representation of an input at block $\ell \in \{1, \ldots, L\}$ and time t by $\mathbf{h}_\ell^{(t)} \in \mathbb{R}^{N\times d}$, the matrix whose N rows are the token vectors produced after block ℓ. It is a function of the noised latent X_t, the time t and the class input c. For image i of a sample of n images, we write $[\mathbf{h}_\ell^{(t)}]_i \in \mathbb{R}^{N\times d}$ for its representation and $h_{i,m} := [\mathbf{h}_\ell^{(t)}]_{i,m} \in \mathbb{R}^d$ for its m-th token, suppressing ℓ

and t where they are clear from context. Each image carries a class label $y_i \in \{1, \dots, 1000\}$. Most properties collapse the spatial dimension by global average pooling (GAP),

$$\bar{h}_i := \frac{1}{N} \sum_{m=1}^{N} h_{i,m} \in \mathbb{R}^d, \tag{8}$$

and we write $\bar{h}$, without an index, for the pooled representation of a random image. For uniformity we instead concatenate all tokens of an image into a single vector $\mathrm{vec}([\mathbf{h}_\ell^{(t)}]_i) \in \mathbb{R}^{Nd}$. Each property is computed from the representation at a single block and time, and the anchor used in the main text is block $\ell = 12$ at $t = 0.25$.

B Related Work

This appendix discusses related work in more detail than the introduction. It places diffusion models within representation learning, reviews work on representation geometry in self-supervised learning and in diffusion models, and discusses auxiliary objectives that shape the hidden representations of diffusion models. It closes with prior comparisons of independently trained models.

Representation learning and self-supervised learning. Representation learning aims to learn an embedding of the data that can be used effectively for downstream tasks such as classification, segmentation and retrieval. Self-supervised learning obtains such embeddings from pretext tasks that need no labels. Contrastive methods such as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) maximize agreement between augmented views of the same image while pushing the representations of different images apart. Self-distillation methods such as BYOL (Grill et al., 2020) and DINO (Caron et al., 2021; Oquab et al., 2023; Siméoni et al., 2025) train a student network to match a momentum-updated teacher without explicit negative pairs. Masked modeling methods such as BEiT (Bao et al., 2022) and MAE (He et al., 2022) reconstruct masked parts of the input. Denoising autoencoders (Vincent et al., 2008) learn representations by recovering a clean input from a corrupted one, and Vincent (2011) showed that the optimal denoiser implicitly estimates the score $\nabla_x \log p_t(x)$, which connects denoising-based representation learning to score-based generative modeling. Flow-matching and diffusion models can be viewed as denoising autoencoders at every time t. From the noised input X_t, they learn to predict the velocity $\dot{X}_t = X_1 - X_0$, the difference between the noise and the data. Their hidden representations $\mathbf{h}_\ell^{(t)}$ therefore encode structure useful for denoising at each time, which makes them a natural object of representation analysis.

Representation geometry in self-supervised learning. Wang & Isola (2020) decompose the contrastive objective into an alignment term, which measures how close the features of positive pairs are, and a uniformity term, and show that both are needed for good representations. Jing et al. (2021) describe dimensional collapse, the concentration of representations on a low-dimensional subspace, and connect it to reduced effective rank and worse downstream performance. Huh et al. (2024) hypothesize that sufficiently capable models trained on different modalities and objectives converge toward similar representations. Under this hypothesis, a diffusion model whose representation is closer to that of a strong vision encoder may have learned a better representation, so the alignment between the two models is a candidate property of representation quality. The same work introduces CKNNA, which we use to measure this cross-model alignment.

Representation geometry in diffusion models. Xiang et al. (2023) show that intermediate features of diffusion models become linearly separable by class, and increasingly uniform, through generative pretraining alone, and they use linear-probe accuracy as a diagnostic of diffusion representations. Yang & Wang (2023) study the effective rank of intermediate features across time levels and find that it tracks representation quality for discriminative tasks. Zhang et al. (2025) link generalization to balanced intermediate activations. Representation alignment trains a diffusion model so that its features match those of a pretrained vision encoder, which we call the alignment encoder. For this setting, Singh et al. (2025) find that the spatial structure (LDS) of the alignment encoder's features predicts generation quality better than their semantic structure as measured by linear probing. Their LDS is computed on the alignment encoder, not on the hidden features of the diffusion model. The ICR paper (Li et al., 2026b) evaluates diffusion representations with a self-supervised

invariance criterion, the Invariant Contamination Ratio, and supports its relevance to generation by showing that it decreases together with FID during training. These studies mostly examine a property within one model or along one training run. Whether the property still predicts generation quality across independently trained models is the question our cross-model evaluation addresses.

Representation-shaping objectives. A parallel line of work adds an auxiliary representation loss to the flow-matching objective,

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}(\theta) + \lambda\, \mathcal{L}_{\text{repr}}(\theta), \qquad \lambda \geq 0, \tag{9}$$

where $\mathcal{L}$ is the objective of Definition A.2 and $\mathcal{L}_{\text{repr}}$ targets some property of the intermediate features. One group aligns the features to those of an alignment encoder. REPA (Yu et al., 2024) adds a patch-wise cosine-similarity loss between the features of an early SiT block ($\ell = 8$) and the patch tokens of the frozen alignment encoder, which speeds up convergence and improves FID substantially over vanilla SiT. REPA thus optimizes the patch-wise cosine similarity to the alignment encoder directly, and the CKNNA between the diffusion model and that encoder only implicitly. iREPA (Singh et al., 2025) replaces REPA's MLP projection head with a convolutional layer to preserve the spatial structure of the feature map during alignment, motivated by the spatial structure (LDS) of the alignment encoder's features. REG (Wu et al., 2025) builds on REPA and additionally learns to generate the class token of the alignment encoder, so that the encoder's class information is available during inference. REPA-E (Leng et al., 2025) propagates the alignment objective into the VAE encoder, which makes the latent compression and the generator trainable end to end. A second group needs no alignment encoder. Dispersive Loss (Wang & He, 2025) optimizes a uniformity objective on the hidden features directly, penalizing configurations in which features cluster rather than spread across the ambient space. SRA (Jiang et al., 2025) aligns the features of early blocks to those of later blocks of an exponential-moving-average teacher. Self-Flow (Chefer et al., 2026) assigns different times to different tokens within one forward pass and also aligns earlier representations with later ones. LSEP (Yun et al., 2025) trains a classification head on pooled intermediate features jointly with the generator, and so optimizes linear separability explicitly. Each of these methods improves FID in its reported setting. An auxiliary loss can improve FID for reasons other than the property it was designed around, so the gain alone does not show that the named property is what matters. Our cross-model evaluation asks the complementary question of whether the level of the property in a trained model predicts that model's generation quality relative to other trained models.

Comparing independently trained models. Comparing converged, independently trained models is not new in representation analysis (Huh et al., 2024; Li et al., 2026a). Our focus is doing so systematically, for a broad set of representation properties, and for generative models, where the relevant outcome is generation quality rather than a downstream discriminative score. Because the comparison needs only forward passes on existing checkpoints, it can strengthen or weaken the evidence for a candidate property without training a model for it.

C Definitions of the Properties

This appendix defines the eight properties. Implementation choices, such as sample sizes and numerical safeguards, are given in Section D.2. We use the notation of Section A. A random image x with label y is encoded to the latent X_0 and noised to $X_t = \alpha_t X_0 + \beta_t X_1$ with $X_1 \sim \mathcal{N}(0, I)$ independent of the image. The representation $\mathbf{h}_\ell^{(t)}$ is computed from X_t and a class input $c \in \{y, \varnothing\}$ at a fixed block ℓ and time t, and expectations and covariances are taken over the image and the noise. Where a population form becomes awkward (the linear probe, the nearest-neighbor mask of CKNNA and the finite sets of token pairs in LDS), the definition is stated for a sample of n images.

C.1 Class-Agnostic Distributional Geometry

Definition C.1 (Effective rank (Roy & Vetterli, 2007))**.** *Effective rank measures how many directions the pooled features effectively spread over. Let* $\Sigma := \mathrm{Cov}(\bar{h}) \in \mathbb{R}^{d\times d}$ *with eigenvalues* $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$*, and let* $\bar{\lambda}_k := \lambda_k / \sum_j \lambda_j$*. Then*

$$\mathrm{eRank}(\Sigma) := \exp\Big(-\sum_{k=1}^{d} \bar{\lambda}_k \ln \bar{\lambda}_k\Big), \tag{10}$$

with the convention $0 \ln 0 = 0$. It equals 1 if and only if Σ has rank one, and d if and only if all eigenvalues are equal. The features are not normalized before the covariance is formed.

Definition C.2 (Uniformity (Wang & Isola, 2020))**.** *Uniformity measures how evenly the features of different images spread over the unit sphere. Let $z := \mathrm{vec}(\mathbf{h}_\ell^{(t)}) \in \mathbb{R}^{Nd}$ be the flattened representation of an image and $\mu := \mathbb{E}[z]$ its mean. The centered and L2-normalized representation is $x := (z-\mu)/\|z-\mu\|_2$. For x, x' the representations of two independently drawn images, each with its own noise,*

$$\mathrm{Unif} := \log \mathbb{E}_{x,\,x'} \left[\exp\left(-2 \left\| x - x' \right\|^2\right) \right]. \tag{11}$$

Lower values mean more uniform coverage. This is the Gaussian-potential uniformity of Wang & Isola (2020) on L2-normalized features, applied after removing the mean that all images share. The training loss of Dispersive Loss (Wang & He, 2025) is the logarithm of a related quantity, evaluated within each mini-batch on the raw flattened features, without centering or normalization and, in the official implementation, with the squared distance scaled by $1/(Nd)$. The two are therefore related, but not the same quantity.

C.2 Class-Structured Geometry

Definition C.3 (Linear-probe accuracy (Alain & Bengio, 2016))**.** *Linear-probe accuracy measures how linearly separable the classes are in the pooled features. Given a training sample $\{(\bar{h}_i, y_i)\}$, a probe $g(\bar{h}) = W\,\mathrm{BN}(\bar{h}) + w_0$ with $W \in \mathbb{R}^{1000\times d}$ and $w_0 \in \mathbb{R}^{1000}$ is fitted by minimizing the cross-entropy loss, where* BN *standardizes each feature coordinate with batch-normalization statistics and has no learnable parameters. The property is the top-1 accuracy of g on held-out images. It is defined by this training procedure rather than as the accuracy of the best linear classifier.*

Definition C.4 (Class-scatter ratio (FDR) (Rao, 1948))**.** *The class-scatter ratio compares how far apart the class means lie with how widely each class spreads. Let $\pi_k := \mathrm{P}(y = k)$, $\mu_k := \mathbb{E}[\bar{h} \mid y = k]$ and $\mu := \mathbb{E}[\bar{h}]$. The between-class and within-class scatter matrices are*

$$S_B := \sum_{k=1}^{1000} \pi_k \, (\mu_k - \mu)(\mu_k - \mu)^\top, \qquad S_W := \sum_{k=1}^{1000} \pi_k \, \mathbb{E}\left[(\bar{h} - \mu_k)(\bar{h} - \mu_k)^\top \,\middle|\, y = k \right], \tag{12}$$

and the class-scatter ratio is

$$\mathrm{FDR} := \frac{\mathrm{Tr}(S_B)}{\mathrm{Tr}(S_W)}. \tag{13}$$

It is computed on the raw pooled features. Since $S_B + S_W = \mathrm{Cov}(\bar{h})$, a larger ratio means that a larger share of the total variance lies between classes.

C.3 Reference-Dependent Alignment

Definition C.5 (CKNNA (Huh et al., 2024))**.** *CKNNA measures how similarly the diffusion model and a reference encoder arrange the same images, restricted to pairs of images that are nearest neighbors in both. Take n images, and let $\phi_i := \bar{h}_i/\|\bar{h}_i\|_2 \in \mathbb{R}^d$ and $\psi_i := \bar{g}_i/\|\bar{g}_i\|_2 \in \mathbb{R}^{d_{\mathrm{ref}}}$ be the L2-normalized pooled features of image i from the diffusion model and from the reference encoder, DINOv2-ViT-g/14 (Oquab et al., 2023). Form the kernel matrices $K_{ij} := \langle \phi_i, \phi_j \rangle$ and $L_{ij} := \langle \psi_i, \psi_j \rangle$. For a kernel matrix M, let $\mathrm{knn}_M(i;k)$ be the k indices $j \neq i$ with the largest M_{ij}. For a pair of kernel matrices (M, M'), the mutual k-nearest-neighbor mask is*

$$\alpha_{M,M'}(i,j) := \mathbb{1}\left[\, i \neq j,\ j \in \mathrm{knn}_M(i;k),\ j \in \mathrm{knn}_{M'}(i;k) \,\right]. \tag{14}$$

The neighborhood-restricted alignment and its normalization are

$$\mathrm{Align}(M, M') := \frac{1}{(n-1)^2} \sum_{i,j} \alpha_{M,M'}(i,j) \Big(M_{ij} - \tfrac{1}{n}\textstyle\sum_l M_{il} \Big) \Big(M'_{ij} - \tfrac{1}{n}\textstyle\sum_l M'_{il} \Big), \tag{15}$$

$$\mathrm{CKNNA} := \frac{\mathrm{Align}(K, L)}{\sqrt{\mathrm{Align}(K, K)\,\mathrm{Align}(L, L)}}. \tag{16}$$

The mask is indexed by the kernel pair on which the alignment is evaluated. In the normalization, $\mathrm{Align}(K, K)$ therefore uses the neighborhoods of K alone and $\mathrm{Align}(L, L)$ those of L alone.

C.4 Local Geometry

Definition C.6 (Spatial self-similarity (LDS) (Singh et al., 2025))**.** *LDS measures whether spatially nearby tokens of an image have more similar features than distant tokens. Arrange the $N = 256$ tokens of image i on the 16×16 grid, let $\mathrm{dist}(m, m')$ be the Manhattan distance between the grid positions of tokens m and m', and let*

$$\mathrm{sim}_i(m, m') := \frac{\langle h_{i,m},\, h_{i,m'} \rangle}{\|h_{i,m}\|_2 \, \|h_{i,m'}\|_2}. \tag{17}$$

The LDS score of image i is

$$\mathrm{LDS}_i := \mathbb{E}_{m,m'}\big[\mathrm{sim}_i(m, m') \mid \mathrm{dist}(m, m') < 4\big] - \mathbb{E}_{m,m'}\big[\mathrm{sim}_i(m, m') \mid \mathrm{dist}(m, m') \geq 4\big], \tag{18}$$

where the expectations are over uniformly drawn token pairs (m, m') with $m \neq m'$. The property is the average $\frac{1}{n}\sum_i \mathrm{LDS}_i$ over images. LDS uses the individual tokens, not the pooled feature.

C.5 Invariance to Augmentation

Both invariance properties compare two views of the same image. For $j \in \{1, 2\}$, the image x is augmented with a random image augmentation T_j and encoded to a latent $X_0^{(j)}$, and the *view* $X_t^{(j)} := \alpha_t X_0^{(j)} + \beta_t X_1^{(j)}$ is this latent noised at time t with its own noise $X_1^{(j)} \sim \mathcal{N}(0, I)$. The augmentations T_1, T_2 and the noises $X_1^{(1)}, X_1^{(2)}$ are drawn independently, so the two views differ in both the augmentation and the noise. Let $\bar{h}^{(1)}$ and $\bar{h}^{(2)}$ be the pooled representations of the two views under null conditioning.

Definition C.7 (Augmentation invariance (Wang & Isola, 2020))**.** *Augmentation invariance measures how close the model places the pooled representations of two views of the same image,*

$$\mathrm{AugInv} := \mathbb{E}\left[\frac{\langle \bar{h}^{(1)}, \bar{h}^{(2)} \rangle}{\|\bar{h}^{(1)}\|_2 \, \|\bar{h}^{(2)}\|_2}\right], \tag{19}$$

where the expectation is over images, augmentations and noises. Higher values mean more invariance. It is the cosine form of the alignment term of Wang & Isola (2020): on L2-normalized pooled features, the expected squared distance between the two views equals $2(1 - \mathrm{AugInv})$. The representation depends on the class input, so the null conditioning is part of the definition.

Definition C.8 (Invariant Contamination Ratio (ICR) (Li et al., 2026b))**.** *ICR compares, direction by direction, the variance that the two views share with the variance in which they differ. Let $s := (\bar{h}^{(1)} + \bar{h}^{(2)})/2$ and $\delta := \bar{h}^{(1)} - \bar{h}^{(2)}$, and define the residual and invariant covariances*

$$\Sigma_\xi := \tfrac{1}{2}\mathrm{Cov}(\delta), \qquad \Sigma_s := \mathrm{Cov}(s) - \tfrac{1}{4}\mathrm{Cov}(\delta). \tag{20}$$

Let $\lambda_1, \ldots, \lambda_d$ be the generalized eigenvalues of $\Sigma_s v = \lambda\, \Sigma_\xi v$. Then

$$\mathrm{ICR} := \frac{1}{1 + \frac{1}{d}\sum_{i=1}^{d} \lambda_i}, \tag{21}$$

with all d eigenvalues entering the mean. Lower values mean more invariance.

The covariances in Eq. (20) follow from an additive model in which each view is $\bar{h}^{(j)} = \sigma + \xi^{(j)}$, with a component σ shared by both views and residuals $\xi^{(1)}, \xi^{(2)}$ that are independent of σ and of each other with covariance Σ_ξ. Then $\mathrm{Cov}(\delta) = 2\Sigma_\xi$ and $\mathrm{Cov}(s) = \mathrm{Cov}(\sigma) + \frac{1}{2}\Sigma_\xi$, so $\Sigma_s = \mathrm{Cov}(\sigma)$. Augmentation invariance and ICR are thus two operationalizations of the same intuition, built on the same views. Augmentation invariance averages a per-image angle, while ICR compares shared and view-specific variance along every direction of the representation.

Class conditioning. Linear-probe accuracy, the class-scatter ratio, augmentation invariance and ICR are defined on the null-conditioned representation ($c = \varnothing$). Effective rank, uniformity, CKNNA and LDS are defined on the representation conditioned on the true class ($c = y$).

Table 2: Orientation of the eight properties. The last column gives the sign of the unoriented Spearman correlation ρ between the measured value and FID that means "more of the property goes together with better FID".

Property	More of the property	ρ with FID when $\tilde{\rho} > 0$
Effective rank	higher eRank	negative
Uniformity	lower Unif	positive
Linear-probe accuracy	higher accuracy	negative
Class-scatter ratio	higher FDR	negative
CKNNA	higher CKNNA	negative
LDS	higher LDS	negative
Augmentation invariance	higher AugInv	negative
ICR	lower ICR	positive

C.6 Orientation

For each property, Table 2 states which direction of the measured value counts as "more of the property". Let ρ be the Spearman correlation between the measured value and FID across a set of models. The oriented correlation is $\tilde{\rho} := -\rho$ for properties where a higher value means more of the property, and $\tilde{\rho} := \rho$ for uniformity and ICR, where a lower value means more of it. A positive $\tilde{\rho}$ therefore means that more of the property goes together with better (lower) FID. A property *reverses* when its oriented correlation along the training run (Set 1) and its oriented correlation across independently trained models (Set 2 or Set 3) have opposite signs. The figures in Section E show the unoriented ρ, and the last column of Table 2 gives the sign of ρ that corresponds to $\tilde{\rho} > 0$.

D Implementation Details

This appendix describes how the results are obtained. It describes the models in the three checkpoint sets and how each property of Section C is estimated from their features. It then gives the FID protocol, the statistical analysis, the interventional training runs and the compute used.

D.1 Checkpoint Sets

Set 1: training run. Set 1 consists of 19 checkpoints of a vanilla SiT-XL/2 trained for 400K iterations in the iREPA codebase with representation alignment disabled. Training follows the hyperparameters of Ma et al. (2024): AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017) with learning rate 10^{-4}, batch size 256, EMA decay 0.9999, the linear path, velocity prediction, and class dropout for classifier-free guidance (Ho & Salimans, 2022) with probability 0.1. Table 3 lists the checkpoint steps.

Table 3: Training steps of the 19 Set 1 checkpoints.

10,000	20,000	30,000	40,000	50,000
60,000	75,000	90,000	100,000	130,000
150,000	175,000	200,000	235,000	270,000
300,000	340,000	370,000	400,000	

Set 2: same method family. Set 2 consists of 16 SiT-XL/2 models, 8 trained with REPA (Yu et al., 2024) and 8 with iREPA (Singh et al., 2025), each for 400K iterations. Within each family, the models differ only in the alignment encoder, the pretrained vision encoder whose features the objective aligns to: CLIP-ViT-L (Radford et al., 2021), DINOv2-ViT-B (Oquab et al., 2023), DINOv3-ViT-B/16 and DINOv3-ViT-H/16+ (Siméoni et al., 2025), LangPE-ViT-G (Bolya et al., 2026), MoCoV3-ViT-B (Chen et al., 2021), PE-ViT-G (Bolya et al., 2026), and WebSSL-ViT-DINO-1B (Fan et al., 2025). Architecture, training budget, loss and all other hyperparameters are held fixed. The checkpoints are those released by Singh et al. (2025). For FID, each family is decoded with the decoder its own codebase uses by default, `sdvae-ft-mse-f8d4` for iREPA and `sd-vae-ft-ema` for

REPA. The decoder is therefore constant within each family, and controlling for the family indicator (Section D.4) also controls for it.

Set 3: different objectives. Set 3 consists of eight SiT-XL/2 models trained with different objectives, listed in Table 4. For each method we use the released checkpoint. Set 3 mixes training budgets, from 400K to 7M iterations, because these are the checkpoints each method reports and releases, and all of them are strong generators.

Table 4: Set 3 models. Epochs refer to ImageNet epochs at the method's training budget. VAE: the encoder used to compute latents for feature extraction, and the decoder used to decode samples for FID. `sd-vae-ft-mse` and `sd-vae-ft-ema` share the same encoder and differ only in the decoder.

Method	Iter.	Epochs	Encoder (features)	Decoder (FID)	Source
iREPA (Singh et al., 2025)	400K	80	SD-VAE	`sdvae-ft-mse-f8d4`	HuggingFace
Dispersive Loss (Wang & He, 2025)	400K	80	SD-VAE	`sd-vae-ft-ema`	Google Drive
REPA-E (Leng et al., 2025)	400K	80	jointly trained	jointly trained	HuggingFace
REPA (Yu et al., 2024)	4M	800	SD-VAE	`sd-vae-ft-ema`	Dropbox
SRA (Jiang et al., 2025)	4M	800	SD-VAE	`sd-vae-ft-ema`	HuggingFace
REG (Wu et al., 2025)	4M	800	SD-VAE	`sd-vae-ft-ema`	HuggingFace
Self-Flow (Chefer et al., 2026)	4M	800	SD-VAE	`sd-vae-ft-ema`	HuggingFace
SiT (Ma et al., 2024)	7M	1400	SD-VAE	`sd-vae-ft-ema`	Dropbox

For feature extraction, every model except REPA-E receives latents from the SD-VAE encoder (Rombach et al., 2022). REPA-E fine-tunes the VAE encoder with its alignment loss, so it receives latents from its own jointly trained VAE. For FID, each model is decoded with the decoder its own method uses by default (Table 4).

REG additionally takes as input a class token derived from its alignment encoder. During feature extraction, this token is the DINOv2-ViT-B class token of the input image, noised to the same time t as the latent with seeded noise and prepended to the image tokens. Null conditioning replaces only the class label, so the class token is supplied in both conditionings. The token is removed from the hidden representation before pooling, so every property is computed from the N image tokens alone. For augmentation invariance and ICR, the class token is recomputed from each augmented view, so that it follows the same augmentation as the latent.

D.2 Property Implementations

This subsection describes how each definition of Section C is estimated. Unless stated otherwise, features come from the 50,000 images of the ImageNet validation set. Table 5 summarizes the settings.

Effective rank. We pool the features of each image, form the sample covariance $\hat{\Sigma}$ with denominator $n - 1$ in float64, clamp its eigenvalues at zero and evaluate Eq. (10). Features are extracted from all 50,000 validation images under true-class conditioning.

Uniformity. We flatten the features of each image to a vector of dimension $Nd = 294{,}912$. A first pass over the 50,000 validation images accumulates their mean μ in float64. A second pass repeats the same forward pass, centers each feature by μ and L2-normalizes it. We estimate the expectation in Eq. (11) from all ordered pairs of distinct images within each 128-image batch, pooled across the batches. The validation images are read in class order, 50 per class, so each batch spans two to four classes, and 33.5% of these pairs show two images of the same class, against about 0.1% of pairs drawn at random. The estimate therefore gives same-class pairs more weight than the expectation over independently drawn images in Eq. (11). Section E.1 repeats the analysis with random pairs, which pair every validation image with each image of a fixed set of 1,024 validation images drawn uniformly at random. For each pair we compute $-2\|x - x'\|^2$ in float64, and we combine these values across batches with a streaming log-sum-exp. The estimate of Unif is the log-sum-exp minus the logarithm of the total number of pairs. Features use true-class conditioning. A training loss such as Dispersive Loss instead evaluates the logarithm within each mini-batch. We pool the pairs before taking the logarithm, since averaging per-batch logarithms would add a further bias, the logarithm being concave.

Linear-probe accuracy. The diffusion model is frozen. We extract null-conditioned pooled features and train a probe consisting of a batch-normalization layer without learnable parameters followed by a linear map $\mathbb{R}^d \to \mathbb{R}^{1000}$. The probe is trained on a class-balanced subset of 100,000 ImageNet training images (100 per class) and evaluated on the 50,000 validation images, and we report top-1 accuracy. The probe is trained with Adam (Kingma & Ba, 2014) at learning rate 7.5×10^{-4} with cosine decay, for 30 epochs with batch size 256 and without data augmentation. The probe head, optimizer, schedule, number of epochs and learning rate are adapted from the released DDAE++ configuration for its largest model (Xiang et al., 2025). Unlike that protocol, which trains on the full ImageNet training set, we train on pre-extracted features of the 100,000-image subset and use a smaller batch.

Class-scatter ratio. We use null-conditioned, unnormalized pooled features of the 50,000 validation images. Per-class sums, counts and squared norms are accumulated in float64, and the traces of S_B and S_W are computed directly from them, without forming the $d \times d$ matrices, with π_k replaced by the empirical class frequency.

CKNNA. The reference encoder is DINOv2-ViT-g/14 (Oquab et al., 2023). Its features are the average of its patch tokens, without the class token. We use $k = 10$ and 10,000 validation images, with true-class conditioning for the diffusion model. We implement CKNNA following the REPA implementation (Yu et al., 2024).

LDS. We L2-normalize every token, form the per-image 256×256 Gram matrix of cosine similarities, and average it over the pairs with Manhattan distance 1 to 3 and over the pairs with distance at least 4. Every image contributes the same number of pairs to each group, so the streaming average over all pairs equals the average of LDS_i over images. We use the 50,000 validation images with true-class conditioning. The definition follows iREPA (Singh et al., 2025), with our own distance thresholds.

Augmentation invariance and ICR. Both properties use one fixed set of views. The images are a seeded, class-balanced subset of 20,000 ImageNet training images (20 per class), shared by all models, blocks and times. The augmentation is RandomResizedCrop to 256×256 with scale $(0.8, 1.0)$, a random horizontal flip, and ColorJitter with brightness, contrast and saturation 0.4 and hue 0.1. It is applied in pixel space before VAE encoding. Each view then receives its own Gaussian noise. The augmentation and the noise are seeded by image and view, so every model sees identical inputs. All forward passes use null conditioning. Augmentation invariance is the mean of the per-image cosine similarity in Eq. (19). For ICR, both covariances in Eq. (20) are mean-centered with denominator $n - 1$ and accumulated in float64. To keep the generalized eigenproblem well posed, we add a ridge $\tau_{\text{eff}} I$ to Σ_ξ with $\tau_{\text{eff}} = \tau_{\text{ridge}} \operatorname{Tr}(\Sigma_\xi)/d$ and $\tau_{\text{ridge}} = 10^{-10}$. Scaling the ridge with $\operatorname{Tr}(\Sigma_\xi)$ keeps ICR invariant to a global rescaling of the features, as the unregularized definition is. Negative sample eigenvalues can occur because Σ_s is a difference of two sample covariances, and they are kept rather than clipped. Our ICR implementation is our own, based on the definition of Li et al. (2026b). It deviates from that paper's setup in two respects: we read features from block 12 rather than block 14, and at $t = 0.25$ rather than $t = 0.2$.

Table 5: Feature-extraction settings per property. $d = 1152$, $N = 256$. For the linear probe, the number of images and the split are given as training / evaluation. "Two views" means the pooled features of two views of each image, which differ in both the augmentation and the noise (Definition C.7).

Property	Input features	L2-normalization	Images	Split	Conditioning
Effective rank	pooled ($\mathbb{R}^d$)	no	50K	val	true class
Uniformity	flattened, centered ($\mathbb{R}^{Nd}$)	yes	50K	val	true class
Linear-probe accuracy	pooled ($\mathbb{R}^d$)	no	100K / 50K	train / val	null
Class-scatter ratio	pooled ($\mathbb{R}^d$)	no	50K	val	null
CKNNA	pooled ($\mathbb{R}^d$)	yes	10K	val	true class
LDS	per token ($\mathbb{R}^d$)	per token	50K	val	true class
Augmentation invariance	pooled, two views	through cosine	20K	train	null
ICR	pooled, two views	no	20K	train	null

D.3 FID Protocol

Generation quality is measured by the Fréchet Inception Distance (FID) (Heusel et al., 2017), computed with the ADM evaluation suite (Dhariwal & Nichol, 2021) against its precomputed ImageNet 256×256 reference statistics. Every model generates 50,000 class-balanced samples (50 per class) without classifier-free guidance (Ho & Salimans, 2022), using the EMA weights and an Euler–Maruyama SDE sampler with 250 steps, following the evaluation protocol of Ma et al. (2024). All FID values are recomputed in-house with this protocol, so the values reported here can differ from those in the original papers. sFID (Nash et al., 2021), Inception Score (Salimans et al., 2016), precision and recall (Kynkäänniemi et al., 2019) are computed by the same suite from the same samples.

D.4 Statistical Analysis

Correlation measures. The primary association measure is the Spearman rank correlation, the Pearson correlation of the rank vectors (Zwillinger & Kokoska, 2000, Section 14.7), which captures any monotone association and is not dominated by single extreme values. We report the Pearson correlation alongside it as a point estimate. Point estimates, confidence intervals, leave-one-out ranges and the robustness sweeps of Section E all use Spearman.

Partial correlation in Set 2. In Set 2, the method family is a binary confound. Let $Z_i = 0$ for REPA models and $Z_i = 1$ for iREPA models. The partial Spearman correlation ρ_Z between a property X and FID Y controlling for Z replaces each variable by its rank vector and then applies the partial-correlation formula. With ρ_{XY}, ρ_{XZ} and ρ_{YZ} the pairwise Spearman correlations,

$$\rho_Z = \frac{\rho_{XY} - \rho_{XZ}\,\rho_{YZ}}{\sqrt{(1-\rho_{XZ}^2)(1-\rho_{YZ}^2)}}. \tag{22}$$

Bootstrap confidence intervals. Asymptotic confidence intervals are unreliable with 8 or 16 models, so we use percentile bootstrap intervals (Efron & Hastie, 2021) with 5000 resamples and seed 42. The 95% interval spans the 2.5th to the 97.5th percentile of the bootstrap distribution. In Set 2, models are resampled with replacement separately within each family, which preserves the balance of 8 REPA and 8 iREPA models, and the statistic is ρ_Z. In Set 3, the eight models are resampled jointly as (property, FID) pairs, and the statistic is the Spearman correlation.

Leave-one-out range. In Set 2 we also report the range of ρ_Z over the 16 subsets obtained by removing one model at a time. A range that stays on one side of zero shows that no single model determines the sign. This is an influence diagnostic, not a significance test.

Oriented correlation and reversal. Findings are stated in terms of the oriented correlation $\tilde{\rho}$ of Section C.6, which is positive when more of the property goes together with better FID. A property reverses when $\tilde{\rho}$ has opposite signs in Set 1 and in Set 2 or Set 3.

D.5 Interventional Training Runs

For each of LDS and augmentation invariance, we train a SiT-XL/2 with an auxiliary loss of the form Eq. (9) on the features of block 8. Both runs branch off the Set 1 training run (Section D.1) at iteration 25K, with the weights and EMA weights restored and a freshly initialized AdamW optimizer, and continue to 400K iterations. Apart from the auxiliary loss, the recipe is that of Set 1: batch size 256, a constant learning rate of 10^{-4}, EMA decay 0.9999, bfloat16 precision and seed 42, with the denoising loss unchanged. The control is the Set 1 run itself over the same iterations. We evaluate both runs at the 17 checkpoint steps of Set 1 between 30K and 400K iterations, and FID follows Section D.3. Each run is trained once. For each run, we compare its best FID over these 17 checkpoints with the best FID of the control over the same checkpoints.

LDS. The auxiliary loss is a hinge on the LDS of Definition C.6, computed on the hidden states after block 8 without a projection,

$$\mathcal{L}_{\text{LDS}} = \max\big(\overline{\text{LDS}} - 0.02,\ 0\big), \tag{23}$$

where $\overline{\mathrm{LDS}}$ is the mean of LDS_i over the 64 images of each GPU's batch. It is computed in float32 on the denoising forward pass, so each image enters at its own sampled time and with its class label, which is dropped with probability 0.1. The loss weight is $\lambda = 1$. The loss decreases LDS, the direction in which LDS moves as FID improves along the training run (at block 8 and $t = 0.25$, $\rho = +0.967$ over the Set 1 checkpoints). The floor of 0.02 keeps the loss from being met by a degenerate solution. Without a floor, the loss was met by growing a single component of the block-8 features that dominates the direction of every token and is canceled by later blocks. The block-8 activations grew by about two orders of magnitude, and training in bfloat16 diverged. Because LDS approaches zero only asymptotically, a floor at zero acts like no floor. With the floor at 0.02, the hinge switches off while the activations stay bounded, and 0.02 is still below the block-8 LDS of every model in Sets 2 and 3 (minimum 0.043 at $t = 0.25$). Because the loss constrains the mean over the sampled training times, LDS at a single time can lie slightly below the floor. The weight $\lambda = 1$ was chosen in pilot runs with SiT-B/2, where it gave about 90% of the attainable change in LDS at a 0.1% increase of the denoising loss, while $\lambda = 3$ cost five times as much for little further change. The run was resumed once, from its 30K checkpoint with the optimizer state, after a file-system failure. The evaluated LDS at block 8 decreases faster than in the control run. At $t = 0.25$ it is 0.024 at 40K iterations and stays between 0.015 and 0.017 from 50K iterations on, whereas the LDS of the control decreases from 0.108 at 50K to 0.053 at 400K iterations. At the anchor block 12, the difference to the control is small and shrinks over training, from -0.021 at 30K to -0.002 at 400K iterations. To describe the scale and composition of the block-8 features, we use the share $\|\mu\|^2/\mathbb{E}\|z\|^2$ of their energy that lies in the component shared by all images, with the flattened representation z and its mean μ over images as in Definition C.2. At 400K iterations and $t = 0.25$, this share is 0.91 in the LDS run against 0.77 in the control, and the pooled features of the augmentation-invariance views have about three times the mean squared norm of the control. The best FID of the LDS run is 18.13 (at 400K iterations), 3.6% below the best FID of the control, 18.81 (at 300K iterations). This difference does not reflect a consistent improvement. The FID of the LDS run is higher than that of the control at every checkpoint up to 300K iterations, by 0.13 to 2.27, and lower only from 340K iterations on, where the FID of the control itself rises from 18.81 to 22.38. In this setting, decreasing LDS far below the level of every model we measured does not consistently improve FID.

Augmentation invariance. The auxiliary loss compares two views of each training image. The first view is the training latent itself, the SD-VAE encoding of the center crop. The second view applies the augmentation of Section D.2 (RandomResizedCrop with scale $(0.8, 1.0)$, a horizontal flip with probability 0.5, and ColorJitter with brightness, contrast and saturation 0.4 and hue 0.1) before encoding with the posterior mean of the SD-VAE. Both views are noised at the same time t, each with its own noise. The loss uses a separate forward pass of both views through blocks 1 to 8, with one time $t \sim \mathcal{U}[0, 1]$ shared by the whole batch and the null class. With the per-sample times and labels of the denoising pass, the two views of an image would share a conditioning vector that all other images lack, and the loss could be met through this vector instead of through the content. A first run built this way diverged. The denoising pass is unchanged. The features are the block-8 token grids of the two views. They are centered by the mean over the global batch of both views at every token position and channel, which removes the input-independent positional component, then flattened to $Nd = 294{,}912$ dimensions and L2-normalized. The loss is the symmetric NT-Xent loss (Chen et al., 2020) over the global batch: for each of the 2×256 rows, the positive is the other view of the same image and the negatives are the remaining 510 rows, with temperature 0.1, averaged over all rows. Gradients flow through both views. The total loss is $\mathcal{L} + 0.5\,\mathcal{L}_{\text{NT-Xent}}$, so $\lambda = 0.5$, without warmup. The loss and the property of Definition C.7 share the augmentation and the null class, but they differ in three respects. The loss compares a center crop with an augmented view, whereas the property compares two independently augmented views. The loss uses centered, flattened token features, whereas the property uses uncentered, pooled features. The loss is contrastive, so it also pushes the views of different images apart. At block 8, where the loss acts, we therefore evaluate, here and in Section 4, a centered variant of the property,

$$\mathrm{AugInv}_{\mathrm{c}} := \mathbb{E}\left[\frac{\langle \bar{h}^{(1)} - \mu,\, \bar{h}^{(2)} - \mu\rangle}{\|\bar{h}^{(1)} - \mu\|_2\, \|\bar{h}^{(2)} - \mu\|_2}\right], \tag{24}$$

where μ is the mean pooled feature over all images and both views, with the views and conditioning of Section D.2. Most of the energy of the pooled block-8 features lies in this mean, a component shared by all images, so the uncentered cosine is close to one for any two images and mostly reflects

the size of that component. Averaging the centered token features of the loss over the token positions gives the pooled feature minus the mean pooled feature of the batch, so the centered variant removes the same shared component as the loss, with the mean taken over the evaluation views instead of the training batch. The centered augmentation invariance of the run exceeds that of the control at every checkpoint and at every time $t \in \{0, 0.25, 0.5\}$. At $t = 0.25$ it rises from 0.677 at 30K to 0.957 at 400K iterations, whereas the control stays between 0.598 and 0.625. Along the control run, $\mathrm{AugInv}_{\mathrm{c}}$ has $\rho = -0.956$ with FID over the 17 checkpoints from 30K iterations on. The centered variant is evaluated only at block 8. At the anchor block 12, the uncentered augmentation invariance of the run is lower than that of the control at every checkpoint (0.989 against 0.994 at 400K iterations). At block 8, the features of the run become dominated by the component shared by all images. At 400K iterations and $t = 0.25$, this component carries a share of 0.99 of their energy, against 0.77 in the control and 0.91 in the LDS run, and the pooled features of the augmentation-invariance views have about 125 times the mean squared norm of the control. The image-specific part is smaller than in the control, with a mean variance per coordinate $\mathbb{E}\|z - \mu\|^2/(Nd)$ of 1.10 against 2.57, but it spreads over more directions, with an effective rank of the pooled features of 16.3 against 5.9. The images therefore do not collapse onto one another, but the block-8 features are dominated by a large component that does not depend on the input. This resembles the degenerate solution that the floor of the LDS loss prevents. At block 12, the shared component carries a share of 0.82 of the energy, against 0.89 in the control. Training is stable, with median gradient norms between 0.18 and 0.32 and no spikes. The denoising loss falls from 0.788 over iterations 25K to 50K to 0.756 over iterations 375K to 400K, against 0.754 to 0.760 for the control over iterations 235K to 400K. The best FID of the augmentation-invariance run is 17.91 (at 400K iterations), 4.8% below the best FID of the control. This improvement is not consistent either. The FID of the run is lower than that of the control at 30K and 40K iterations, higher at every checkpoint from 50K to 300K iterations, by 0.25 to 3.47, and lower again only from 340K iterations on, where the FID of the control rises. In this setting, increasing augmentation invariance at block 8 does not consistently improve FID.

D.6 Compute

FID sampling was performed on a single NVIDIA H100 SXM5 94 GB GPU, and FID evaluation on a single NVIDIA A100 SXM4 80 GB GPU. Feature extraction and all analysis computations were performed on NVIDIA A100 SXM4 80 GB GPUs partitioned into 40 GB MIG slices. The Set 1 model was trained on 4 NVIDIA H100 SXM5 94 GB GPUs. The LDS run was trained on 4 NVIDIA H100 GPUs in about 19 hours, and the augmentation-invariance run on 8 NVIDIA GH200 GPUs in about 27 hours.

E Additional Results

The main text correlates each property with FID at a single block and time, the anchor. This appendix first checks how far the findings of the main text depend on these two choices, by reporting the correlations at other blocks and times and with other measures of generation quality. It then reports whether the findings transfer to other model sizes.

Feature extraction. Features are extracted from blocks $\ell \in \{4, 8, 12, 16, 20\}$ at times $t \in \{0, 0.25, 0.5\}$, which gives fifteen (ℓ, t) cells per model. In Set 2, effective rank, linear-probe accuracy and CKNNA were additionally extracted at blocks 24 and 27. The main text reports the anchor $(\ell, t) = (12, 0.25)$. The three times match the levels used by Yu et al. (2024) and Wu et al. (2025). Block 12 lies past the alignment depth $\ell = 8$ used by the alignment-based objectives in this study (Yu et al., 2024; Singh et al., 2025; Wu et al., 2025; Leng et al., 2025), and Xiang et al. (2023) find linearly separable features in the intermediate layers of diffusion networks. Features are extracted in separate passes for groups of properties, since the properties differ in conditioning, images and views. The VAE latent is the posterior mean rather than a sample, and the Gaussian noise is drawn from a seeded generator. The forward pass runs in float16 with the EMA weights, and features are cast to float32 before the properties are computed (with float64 accumulation where stated in Section D.2). The released SiT, Dispersive Loss and Self-Flow checkpoints were trained with $t = 0$ as noise and receive $1 - t$. At $t = 0$ the input is the clean latent, so the two views of augmentation invariance and ICR differ only in the augmentation.

E.1 Robustness Across Blocks and Times

Figure 5 shows, for every property and all fifteen (ℓ, t) cells, the partial Spearman correlation ρ_Z in Set 2 and the Spearman correlation ρ in Set 3. Figures 6 and 7 show the same values as functions of the block at $t = 0.25$ and of the time at $\ell = 12$. All three figures show the unoriented correlation with FID. For uniformity and ICR, a positive value means that more of the property goes together with better FID, and for the other six properties a negative value does (Table 2). The Set 1 values quoted below come from the same sweep and are not plotted.

Properties with more consistent support. Linear-probe accuracy has a negative correlation with FID in every cell of all three sets. In Set 2, ρ_Z ranges from -0.862 to -0.278, and in Set 3, ρ ranges from -0.976 to -0.714. Effective rank has a negative correlation in every cell of Sets 1 and 3. In Set 2 it is negative in 12 of 15 cells, and the three exceptions are the cells at block 20, where ρ_Z lies between $+0.067$ and $+0.276$. Across the models of Set 2, the association of effective rank with FID is therefore weak at block 16 and reverses at block 20. ICR has a positive correlation with FID in every cell of Sets 1 and 3, with ρ between 0.823 and 0.974 in Set 1 and between 0.619 and 1.000 in Set 3. In Set 2 its ρ_Z falls with depth, from between $+0.758$ and $+0.849$ at block 4 to between -0.486 and -0.163 at block 20, where it takes the opposite sign. CKNNA has a negative correlation in every cell of Set 3 and in 13 of 15 cells of Set 2, the exceptions being block 20 at $t = 0.25$ and $t = 0.5$ ($\rho_Z = +0.008$ and $+0.228$). Along the training run, however, CKNNA has no consistent sign: its correlation is positive in 7 of 15 cells, reaching $+0.723$ at block 8 and $t = 0.5$, and it is -0.225 at the anchor.

Properties with weaker or less consistent associations. Uniformity has $\rho = +0.853$ with FID at the anchor of Set 1 and a positive correlation in 11 of 15 cells, so along training more uniform features go together with better FID. In Set 2 its ρ_Z is negative in 12 of 15 cells, between -0.437 and $+0.203$, which is the opposite direction. In Set 3 it has no consistent sign, with ρ between -0.571 and $+0.381$ and -0.310 at the anchor. LDS has ρ between $+0.965$ and $+0.977$ in every cell of Set 1, so along training LDS decreases as FID improves. In Set 2 its ρ_Z is negative in 12 of 15 cells, which is the opposite direction, and the three exceptions, all at block 20, are close to zero ($+0.010$ to $+0.143$). In Set 3, LDS keeps the Set 1 sign in 12 of 15 cells. The exceptions are the three cells at block 8, where ρ lies between -0.429 and -0.310. Block 8 is the block at which the alignment-based objectives in Set 3 align their features to an encoder. Singh et al. (2025) report that the LDS of the alignment encoder's features correlates negatively with FID. Our measurement concerns the diffusion model's own features, so the sign at block 8 neither confirms nor contradicts their finding. Augmentation invariance has a negative correlation in 14 of 15 cells of Set 1. In Set 2 its ρ_Z is positive in 9 of 15 cells, and the remaining values are small (-0.180 to 0.000). In Set 3 it is negative in 10 of 15 cells, the same sign as in Set 1, but at the anchor its correlation is -0.024. The class-scatter ratio has a negative correlation in 13 of 15 cells of Set 1 (the exceptions are two cells at block 4) and in 13 of 15 cells of Set 3. In Set 2 it has no consistent sign, with ρ_Z between -0.300 and $+0.352$. For the class-scatter ratio, the loss of predictive power is therefore specific to the controlled comparison of Set 2.

Uniformity with random pairs. Our estimate of uniformity uses the pairs within each 128-image batch, about a third of which show two images of the same class (Section D.2). We therefore repeat the analysis with random pairs. Along the training run, the two estimates agree: at the anchor, $\rho = +0.816$ with random pairs against $+0.853$ within batches, and both are positive in 11 of 15 cells. Across independently trained models, they do not. At the anchor, random pairs give $\rho_Z = +0.496$ in Set 2, with bootstrap interval $[-0.299, +0.906]$ and leave-one-out range $[+0.204, +0.714]$, against $\rho_Z = -0.268$, $[-0.694, +0.297]$ and $[-0.388, -0.040]$ within batches. In Set 3, random pairs give $\rho = +0.357$, $[-0.524, +0.846]$, against $\rho = -0.310$, $[-0.923, +0.544]$. Over the fifteen cells, random pairs give a positive correlation in 11 of 15 cells of Set 2 and in all 15 cells of Set 3, where ρ lies between $+0.190$ and $+0.667$. With random pairs, uniformity therefore keeps its Set 1 direction across independently trained models, whereas with pairs within batches it reverses. The direction of the cross-model association of uniformity thus depends on the pair set, and at the anchor neither estimate excludes a zero association.

Sensitivity of the ICR estimator. ICR averages generalized eigenvalues whose denominators are the eigenvalues of Σ_ξ, so it is dominated by the smallest directions of Σ_ξ and depends on the ridge

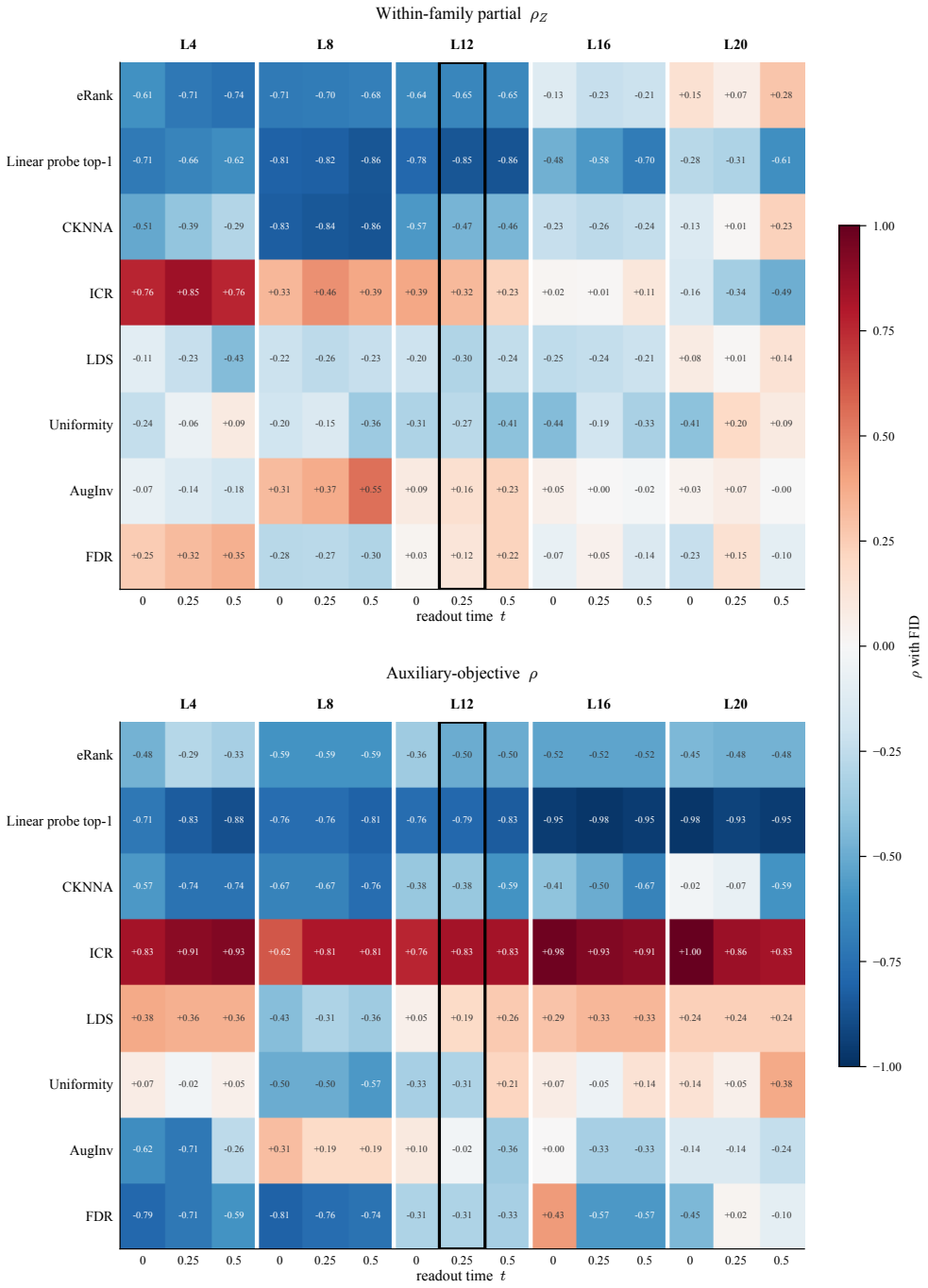


Figure 5: Correlation of each property with FID across the fifteen (ℓ, t) cells: partial Spearman ρ_Z in Set 2 (top, 16 models) and Spearman ρ in Set 3 (bottom, 8 models). Blue cells are negative and red cells positive. The correlations are not oriented (Table 2). The anchor cell $(\ell, t) = (12, 0.25)$ is boxed.

τ_{ridge}. In a sweep over τ_{ridge} on an early, weakly trained checkpoint, ICR stays flat for $\tau_{\text{ridge}} \leq 10^{-10}$ and rises steeply for larger ridges, while on a converged model the same sweep changes it only slightly. Our choice $\tau_{\text{ridge}} = 10^{-10}$ lies on the flat part.

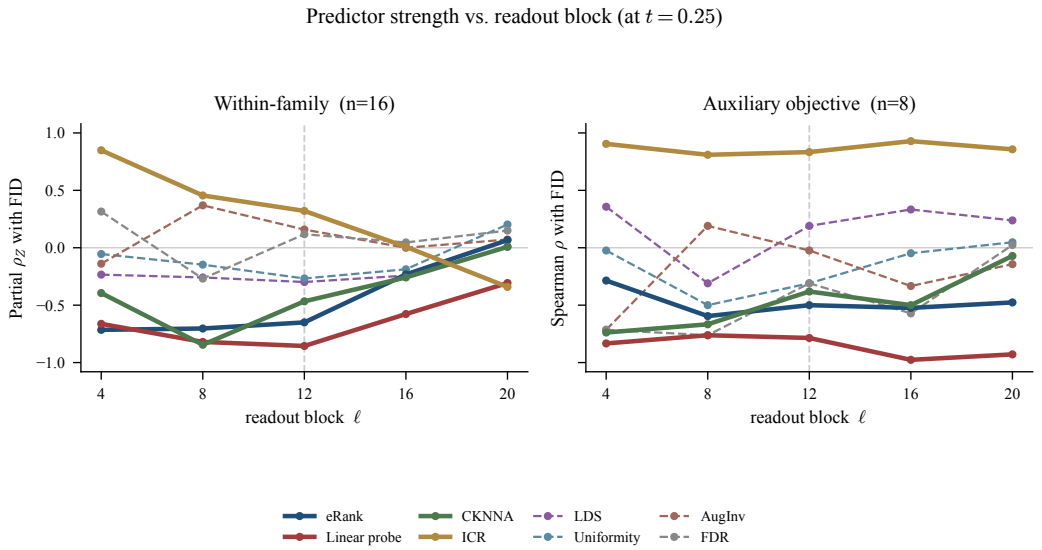


Figure 6: Correlation of each property with FID as a function of the block ℓ at $t = 0.25$: partial Spearman ρ_Z in Set 2 (left, 16 models) and Spearman ρ in Set 3 (right, 8 models). Solid lines are the properties with more consistent support in the main text, dashed lines those with weaker or less consistent associations. The anchor block $\ell = 12$ is marked.

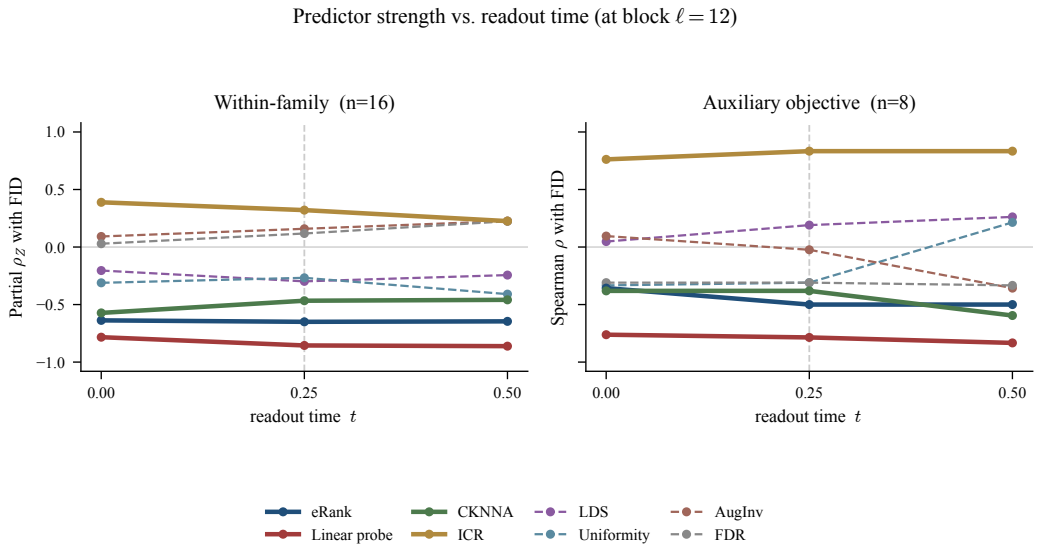


Figure 7: As Fig. 6, as a function of the time t at block $\ell = 12$. The anchor $t = 0.25$ is marked.

Table 6: Spearman correlation of each property with sFID, Inception Score (IS), precision (Prec.) and recall (Rec.) at the anchor $(\ell, t) = (12, 0.25)$. Set 2 reports the partial correlation ρ_Z controlling for the method family.

	Set 1 (training run)				Set 2 (same family)				Set 3 (different objectives)			
Property	sFID	IS	Prec.	Rec.	sFID	IS	Prec.	Rec.	sFID	IS	Prec.	Rec.
Effective rank	-0.89	$+0.98$	$+0.99$	$+0.93$	-0.11	$+0.62$	$+0.28$	$+0.36$	-0.74	$+0.43$	$+0.62$	-0.45
Linear-probe accuracy	-0.91	$+0.99$	$+0.97$	$+0.99$	-0.04	$+0.54$	$+0.69$	$+0.47$	-0.52	$+0.81$	$+0.69$	$+0.10$
CKNNA	-0.06	$+0.20$	$+0.26$	$+0.05$	$+0.52$	$+0.66$	$+0.25$	$+0.64$	-0.40	$+0.38$	$+0.38$	-0.17
ICR	$+0.96$	-0.88	-0.86	-0.96	$+0.10$	-0.14	-0.39	-0.24	$+0.62$	-0.83	-0.76	$+0.07$
LDS	$+0.92$	-0.98	-0.96	-0.99	$+0.10$	$+0.38$	$+0.28$	$+0.18$	-0.48	-0.38	0.00	-0.62
Uniformity	$+0.85$	-0.84	-0.87	-0.79	$+0.56$	$+0.73$	-0.03	$+0.71$	$+0.14$	$+0.48$	$+0.10$	$+0.50$
Augmentation invariance	-0.90	$+0.95$	$+0.94$	$+0.96$	$+0.17$	-0.24	-0.25	-0.06	$+0.48$	$+0.24$	-0.19	$+0.60$
Class-scatter ratio	-0.92	$+0.98$	$+0.96$	$+0.99$	$+0.65$	$+0.21$	-0.26	$+0.34$	-0.07	$+0.40$	$+0.14$	$+0.33$

E.2 Quality Measures Besides FID

FID summarizes generation quality in one number and has well-known shortcomings as a single summary statistic. We therefore repeat the comparison with four further measures computed from the same samples (Section D.3): sFID (Nash et al., 2021), Inception Score (Salimans et al., 2016), precision and recall (Kynkäänniemi et al., 2019). Like FID, sFID and Inception Score summarize generation quality in one number. Precision and recall do not measure performance as a single number. They separate the fidelity of the samples from their coverage of the data distribution, and we report them as additional context. Table 6 gives the Spearman correlation of each property with each measure at the anchor. For Set 2, the correlation is the partial correlation ρ_Z. For sFID, lower is better, as for FID. For Inception Score, precision and recall, higher is better, so their orientation is the reverse of that of FID. Figures 8 to 10 show the correlations oriented as in Section C.6, extended to every measure, so that a positive value means that more of the property goes together with better generation under that measure.

The central observation of the main text is not specific to FID. Along the training run, every property except CKNNA has an absolute Spearman correlation of at least 0.83 with FID, sFID and Inception Score, and the median absolute correlation over the eight properties is 0.961 for FID, 0.904 for sFID and 0.966 for Inception Score. Across the converged models of Set 3, the median absolute correlation falls to 0.345, 0.476 and 0.417, respectively, and for each of the three measures one or two properties change sign between Set 1 and Set 3. Within the method families of Set 2, the median absolute partial correlation is 0.310, 0.141 and 0.459, and three or four properties change sign between Set 1 and Set 2 for each of the three measures. Precision and recall show the same pattern. Under every measure, strong correlations along a training run therefore do not carry over consistently to independently trained models.

E.3 Transfer Across Model Sizes

We repeat the analysis of the training run for two smaller models, SiT-B/2 with 12 blocks of width 768 and SiT-L/2 with 24 blocks of width 1024 (Ma et al., 2024). Both are trained with the same recipe as the Set 1 model and evaluated at the same 19 checkpoints, with FID following Section D.3. We read their features at the extracted block closest in relative depth to the anchor, block 6 of 12 for SiT-B/2 and block 10 of 24 for SiT-L/2, against block 12 of 28 for SiT-XL/2, at $t = 0.25$. Table 7 gives the Spearman correlation of each property with FID, and Fig. 11 shows how each property evolves over training at the three sizes.

The correlations along the training run transfer across model sizes. For seven of the eight properties, the correlation with FID has the same sign at all three sizes and an absolute value of at least 0.78. Their trajectories also have nearly the same shape: over the 19 checkpoints, the Spearman correlation between the SiT-B/2 or SiT-L/2 trajectory of a property and its SiT-XL/2 trajectory lies between 0.91 and 1.00. The exception is CKNNA, which is already weak along the SiT-XL/2 run. Its correlation with FID is -0.974 for SiT-B/2, $+0.600$ for SiT-L/2 and -0.225 for SiT-XL/2, and its trajectory differs in shape across the sizes. Each size is a single training run, so the transfer concerns the evidence from the training run. Sets 2 and 3 are not repeated at other sizes.

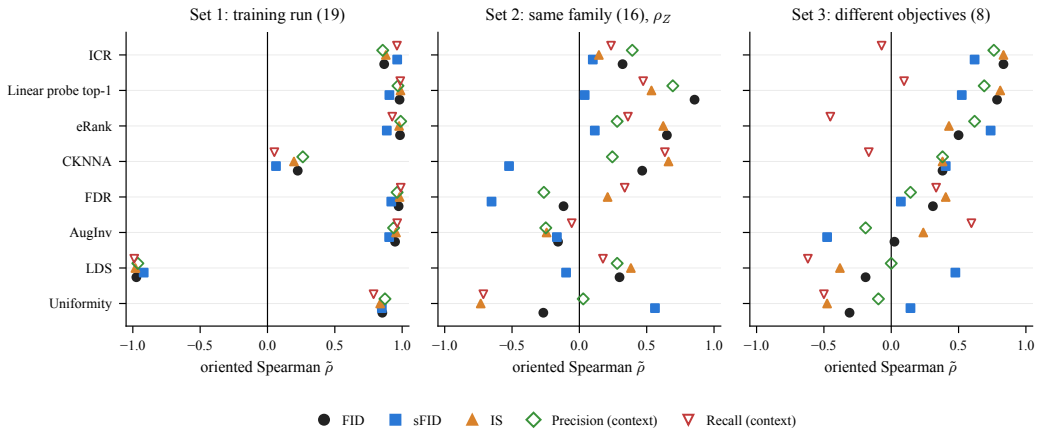


Figure 8: Oriented Spearman correlation $\tilde{\rho}$ of each property with five measures of generation quality at the anchor $(\ell, t) = (12, 0.25)$, along the training run (Set 1, left), within the method families of Set 2 (partial correlation ρ_Z, middle) and across the converged models of Set 3 (right). A positive value means that more of the property goes together with better generation under that measure. Filled markers are FID, sFID and Inception Score (IS), and hollow markers are precision and recall, reported as context. Rows are ordered by the Set 3 value under FID.

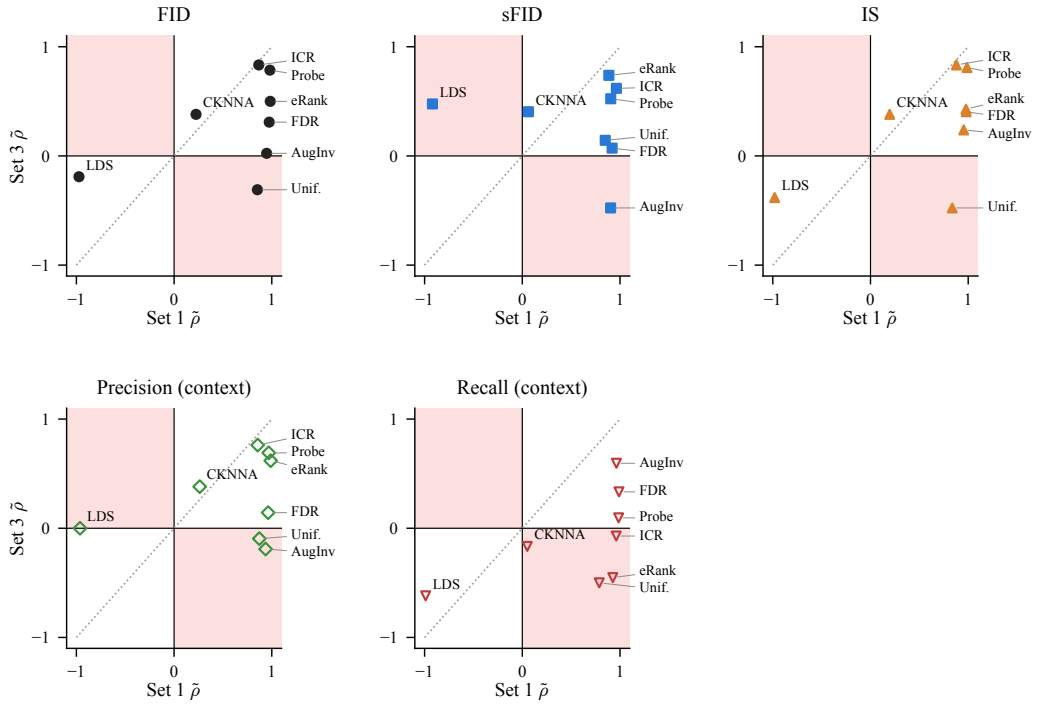


Figure 9: Oriented correlation along the training run (Set 1, horizontal) against the oriented correlation across the converged models of Set 3 (vertical), one panel per measure of generation quality, at the anchor $(\ell, t) = (12, 0.25)$. Shaded quadrants mark a sign change between the two sets, and the dotted line marks equal values. Precision and recall are reported as context.

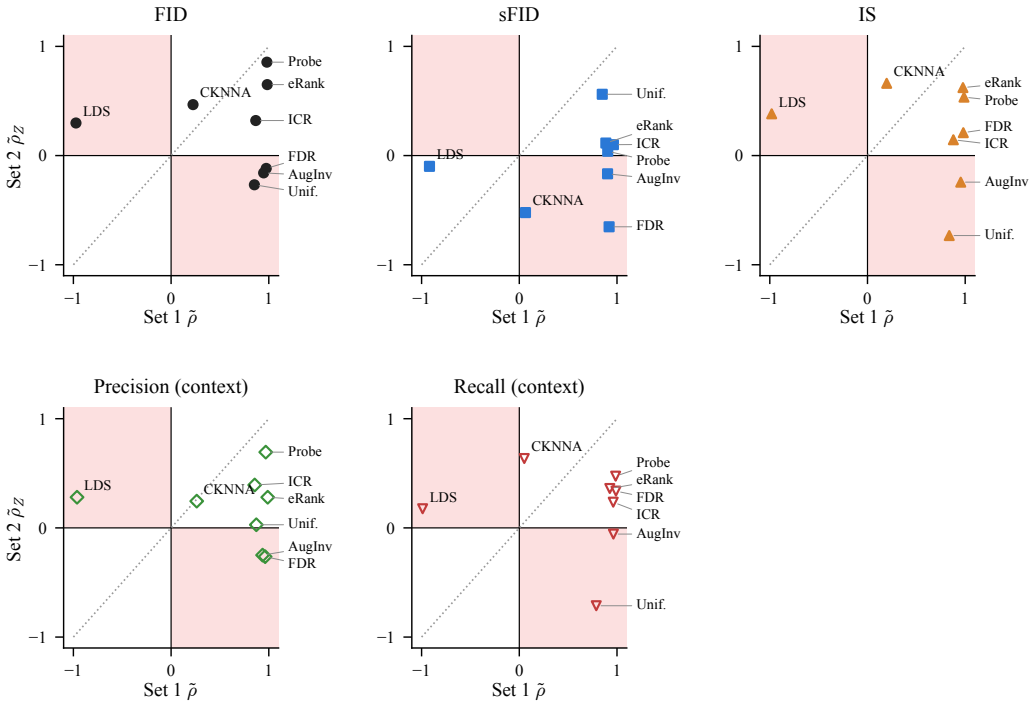


Figure 10: As Fig. 9, with the oriented partial correlation $\tilde{\rho}_Z$ across the 16 models of Set 2 on the vertical axis.

Table 7: Spearman correlation of each property with FID along the training runs of three model sizes, over the 19 checkpoints of Set 1 at $t = 0.25$. Each size is read at the block closest in relative depth to the anchor.

Property	SiT-B/2 block 6 of 12	SiT-L/2 block 10 of 24	SiT-XL/2 block 12 of 28
Effective rank	-1.000	-1.000	-0.984
Linear-probe accuracy	-1.000	-1.000	-0.981
CKNNA	-0.974	$+0.600$	-0.225
ICR	$+1.000$	$+1.000$	$+0.867$
LDS	$+1.000$	$+1.000$	$+0.974$
Uniformity	$+0.788$	$+0.933$	$+0.853$
Augmentation invariance	-0.839	-0.904	-0.947
Class-scatter ratio	-1.000	-1.000	-0.974

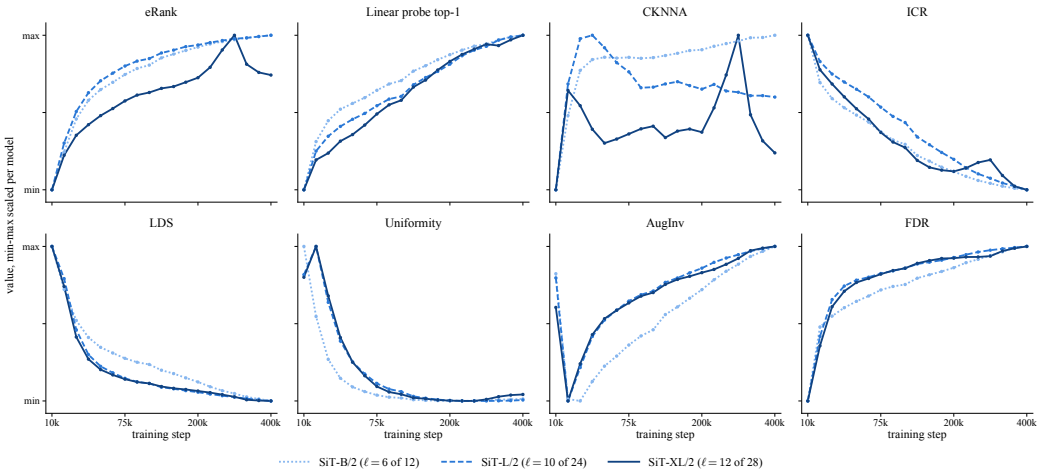


Figure 11: Evolution of the eight properties over the 19 checkpoints of Set 1 for SiT-B/2, SiT-L/2 and SiT-XL/2 at $t = 0.25$, each read at the block closest in relative depth to the anchor. Each curve is min-max scaled over its own checkpoints, so that only the shapes are compared.

F Proof of the Effective-Rank Result

We first state the result formally. The notation and the assumptions are introduced in the remainder of this appendix, which reproduces the full argument.

Theorem (Effective-rank lower bound; formal statement of Theorem F.1)**.** *Fix $t \in (0,1)$ and a block ℓ, let $X_1 \sim \mathcal{N}(0, I)$ be independent of X_0, and let $\Sigma_t := \mathrm{Cov}(\bar{h}_\ell^{(t)}(X_t))$ be the population covariance of the pooled hidden feature. Let $\Gamma_t^\star$ be the covariance of the pooled Bayes-optimal velocity with largest eigenvalue $\lambda_1^\star$, let $\varepsilon_t^{\mathrm{pool}}$ be the pooled per-t excess risk of the network, and let $\delta_t := \varepsilon_t^{\mathrm{pool}} + 2\sqrt{\varepsilon_t^{\mathrm{pool}}\,\|\Gamma_t^\star\|_{\mathrm{op}}}$. Assume (Assumption F.1) that there are deterministic $A_t \in \mathbb{R}^{d_v \times d}$ and $b_t \in \mathbb{R}^{d_v}$, depending on t but not on (X_0, X_1), with $u_m^\theta(x_t, t) = A_t\, h_{\ell,m}^{(t)}(x_t) + b_t$ for every token m, where $d_v \le d$ and A_t has full row rank with smallest singular value $b > 0$. Assume (Assumption F.2) that $\lambda_{d_v}(\Sigma_t) > \lambda_{d_v+1}(\Sigma_t)$ and that $\|P_A - P_{\mathrm{top}}\|_{\mathrm{op}} \le \eta$ for some $\eta \in [0,1)$, where P_A is the orthogonal projector onto the row space of A_t and P_{top} the orthogonal projector onto the top-d_v eigenspace of Σ_t. Assume (Assumption F.3) that $\mathrm{Tr}(\Sigma_t) \ge \tau_t$ for some $\tau_t > 0$. Then*

$$\mathrm{eRank}(\Sigma_t) \;\ge\; \frac{\tau_t\, b^2(1-\eta^2)}{\lambda_1^\star + \delta_t}.$$

Effective rank has not been a central object of study for the hidden representations of diffusion models: none of the representation-shaping objectives in Section B targets it. This appendix aims to give theoretical intuition for why it is coupled to the flow-matching objective at all. Under an idealization of the network after layer ℓ (Assumption F.1), fitting the objective well forces the covariance of the pooled features to spread over several directions. This is intuition for why a spread covariance is necessary for a good fit, not an explanation of the correlation with FID measured in the main text.

The argument proceeds in four steps. We first analyze the structure of the covariance of the pooled Bayes-optimal velocity (Section F.2): its decomposition has spectral properties promising a spread spectrum. This motivates the second step, which controls the covariance of the learned pooled velocity through that of the optimal one and shows that they are operator-norm close (Section F.3). Idealizing the network after layer ℓ as a shared per-token affine map and assuming approximate eigenspace alignment, we then transfer this operator-norm control from the learned velocity covariance to the feature covariance Σ_t as per-eigenvalue bounds (Section F.4). Combining the resulting eigenvalue bounds with an elementary entropy inequality lower-bounds the effective rank. Under a lower bound assumption on the trace of Σ_t, we obtain a bound on $\mathrm{eRank}(\Sigma_t)$ that no longer depends

on the per-eigenvalue lower bounds. The only property of training used anywhere below is that the per-t excess risk is small. The chain results in Theorem F.1.

F.1 Setup

Three objects from Sections A and C recur throughout. The stochastic interpolant $X_t = \alpha_t X_0 + \beta_t X_1$ defines the noising schedule, where we assume $X_1 \sim \mathcal{N}(0, I)$ is independent of X_0. The Bayes-optimal velocity $u^\star = \mathbb{E}[\dot{X}_t \mid X_t]$ is the unique population minimizer of the flow-matching objective $\mathcal{L}$ (Proposition A.1). The GAP-pooled hidden feature $\bar{h}_\ell^{(t)}$ at layer ℓ of the SiT architecture (Section A.3 and Eq. (8)) is the object whose covariance is analyzed below. The population analog of the empirical covariance $\hat{\Sigma}$ is

$$\Sigma_t := \mathrm{Cov}\big(\bar{h}_\ell^{(t)}(X_t)\big) \in \mathbb{R}^{d\times d}. \tag{25}$$

For SiT-XL/2, $d = 1152$, $N = 256$, and the per-token output dimension is $d_v = 16$. Throughout, $t \in (0, 1)$ is fixed.

Remark F.1 (Linear interpolant)**.** *The linear interpolant used in the experiments is $\alpha_t = 1 - t$, $\beta_t = t$, giving $\dot{\alpha}_t = -1$, $\dot{\beta}_t = 1$, and $\dot{X}_t = X_1 - X_0$.*

Convention for randomness. Recall from Section A.3 that $\mathbf{h}_\ell^{(t)} \in \mathbb{R}^{N\times d}$ is the matrix of N token representations at block ℓ and noise level t; we write $h_{\ell,m}^{(t)}$ for its m-th row. The only source of randomness is (X_0, X_1) at fixed t, and every other random object is a deterministic function of it. For brevity we write $u^\theta := u^\theta(X_t, t)$ and $h_{\ell,m}^{(t)} := h_{\ell,m}^{(t)}(X_t)$. Expectations and covariances are taken over the joint law of (X_0, X_1), unless an explicit conditional bar $\mid X_t$ appears.

Pooled velocities. Let $u^\theta_{\text{full}}, u^\star_{\text{full}} \in \mathbb{R}^{Nd_v} \cong \mathbb{R}^{h\times w\times C}$ be the stacked network and Bayes-optimal velocities. We pool by token-averaging via the map

$$S_v := \tfrac{1}{N}(\mathbf{1}_N^\top \otimes I_{d_v}) \in \mathbb{R}^{d_v\times Nd_v}, \tag{26}$$

where $\mathbf{1}_N$ is the all-ones vector in $\mathbb{R}^N$ and $\otimes$ is the Kronecker product. In words, S_v averages the N per-token blocks of size d_v. We set $\bar{u^\theta} := S_v u^\theta_{\text{full}}$, $\bar{\dot{X}}_t := S_v \dot{X}_t$, and the pooled Bayes-optimal velocity $\bar{u^\star} := \mathbb{E}[\bar{\dot{X}}_t \mid X_t] = S_v u^\star_{\text{full}}$, the last equality holding because S_v is deterministic and linear. The full and pooled per-t excess risks are

$$\varepsilon_t^{\text{full}} := \mathbb{E}\big[\, \|u^\theta_{\text{full}} - u^\star_{\text{full}}\|^2 \,\big], \qquad \varepsilon_t^{\text{pool}} := \mathbb{E}\big[\, \|\bar{u^\theta} - \bar{u^\star}\|^2 \,\big]. \tag{27}$$

Remark F.2 (Per-t excess risk and the global objective)**.** *The flow-matching objective (Definition A.2) minimizes $\mathbb{E}_t[\varepsilon_t^{\text{full}}]$, the average per-$t$ excess risk, not $\varepsilon_t^{\text{full}}$ at each t individually. Since $t \sim \mathrm{Uniform}[0, 1]$ in Definition A.2, Markov's inequality gives, for any $\delta > 0$,*

$$\lambda\big(\{t \in [0, 1] : \varepsilon_t^{\text{full}} > \delta\}\big) \;\le\; \frac{\mathbb{E}_t[\varepsilon_t^{\text{full}}]}{\delta}. \tag{28}$$

The results below therefore apply for all t outside a set of Lebesgue measure at most $\mathbb{E}_t[\varepsilon_t^{\text{full}}]/\delta$. As the loss approaches zero, this measure goes to zero, so the results hold for Lebesgue-almost-every t.

Spectral notation. For a positive-semidefinite (PSD) matrix M, $\lambda_k(M)$ is its k-th largest eigenvalue. For a matrix A, $\sigma_k(A)$ is its k-th largest singular value, and $\|\cdot\|_{\text{op}}$ is the operator (spectral) norm.

F.2 Covariance of the Pooled Bayes-Optimal Velocity

Recall from Section A.1 that the Bayes-optimal velocity is the conditional expectation $u^\star = \mathbb{E}[\dot{X}_t \mid X_t]$, the best X_t-measurable predictor of $\dot{X}_t$ in mean square. Applying S_v gives the pooled Bayes-optimal velocity $\bar{u^\star} = \mathbb{E}[\bar{\dot{X}}_t \mid X_t]$. A small excess risk means the trained network is close to it. The next proposition records the covariance of $\bar{u^\star}$.

Proposition F.1 (Covariance of the pooled Bayes-optimal velocity)**.** *For every* $t \in (0,1)$*,*

$$\mathrm{Cov}(\bar{u^\star}) = \frac{\dot\beta_t^2}{N}\, I_{d_v} + \dot\alpha_t^2\, S_v \mathrm{Cov}(X_0) S_v^\top - \frac{\Delta_t^2}{\beta_t^2}\, S_v \mathbb{E}\big[\,\mathrm{Cov}(X_0 \mid X_t)\,\big] S_v^\top, \tag{29}$$

where $\Delta_t := \alpha_t\dot\beta_t - \dot\alpha_t\beta_t$*.*

Proof. The argument uses the *law of total covariance* (Kallenberg, 1997, Lemma 8.2): for a square-integrable random vector W,

$$\mathrm{Cov}(W) = \mathbb{E}\big[\,\mathrm{Cov}(W \mid X_t)\,\big] + \mathrm{Cov}\big(\mathbb{E}[W \mid X_t]\big). \tag{30}$$

We apply Eq. (30) to $W = \bar{\dot X}_t = S_v \dot X_t$ and evaluate its three terms. We repeatedly use the following properties of covariance. For a deterministic matrix M and a random vector Z, $\mathrm{Cov}(MZ) = M\,\mathrm{Cov}(Z)\,M^\top$. Adding a deterministic vector leaves the covariance unchanged. For independent Z, Z', $\mathrm{Cov}(Z+Z') = \mathrm{Cov}(Z) + \mathrm{Cov}(Z')$.

Unconditional covariance. From $\dot X_t = \dot\alpha_t X_0 + \dot\beta_t X_1$ we have

$$\bar{\dot X}_t = S_v \dot X_t = \dot\alpha_t\, S_v X_0 + \dot\beta_t\, S_v X_1. \tag{31}$$

Using independence of X_0 and X_1 and $\mathrm{Cov}(MZ) = M\,\mathrm{Cov}(Z)\,M^\top$ with $M = S_v$, we get

$$\begin{aligned}\mathrm{Cov}(\bar{\dot X}_t) &= \dot\alpha_t^2\,\mathrm{Cov}(S_v X_0) + \dot\beta_t^2\,\mathrm{Cov}(S_v X_1) \\ &= \dot\alpha_t^2\, S_v\,\mathrm{Cov}(X_0)\, S_v^\top + \dot\beta_t^2\, S_v\,\mathrm{Cov}(X_1)\, S_v^\top \\ &= \dot\alpha_t^2\, S_v\,\mathrm{Cov}(X_0)\, S_v^\top + \frac{\dot\beta_t^2}{N}\, I_{d_v}.\end{aligned} \tag{32}$$

Here we used $\mathrm{Cov}(X_1) = I_{Ndv}$ and $S_v S_v^\top = \frac{1}{N}\, I_{d_v}$. The latter follows as

$$\begin{aligned}S_v S_v^\top &= \tfrac{1}{N}(\mathbf{1}_N^\top \otimes I_{d_v})(\tfrac{1}{N}(\mathbf{1}_N^\top \otimes I_{d_v}))^\top \\ &= \tfrac{1}{N}(\mathbf{1}_N^\top \otimes I_{d_v})(\tfrac{1}{N}(\mathbf{1}_N \otimes I_{d_v})) \\ &= \tfrac{1}{N^2}\,(\mathbf{1}_N^\top \mathbf{1}_N) \otimes (I_{d_v} I_{d_v}) \\ &= \tfrac{1}{N^2}\, N \cdot I_{d_v} = \tfrac{1}{N}\, I_{d_v},\end{aligned} \tag{33}$$

where in the second equality we used that the transpose operator distributes over the Kronecker product and its bilinearity, and in the third equality we used the mixed-product property of the Kronecker product.

Conditional covariance. For $t \in (0,1)$ we have $\beta_t > 0$, so we may solve $X_t = \alpha_t X_0 + \beta_t X_1$ for $X_1 = \beta_t^{-1}(X_t - \alpha_t X_0)$. Substituting into $\dot X_t = \dot\alpha_t X_0 + \dot\beta_t X_1$ and using $\dot\alpha_t - \dot\beta_t\alpha_t/\beta_t = -\Delta_t/\beta_t$,

$$\dot X_t = \frac{\dot\beta_t}{\beta_t}\, X_t - \frac{\Delta_t}{\beta_t}\, X_0, \qquad \text{so} \qquad \bar{\dot X}_t = \frac{\dot\beta_t}{\beta_t}\, S_v X_t - \frac{\Delta_t}{\beta_t}\, S_v X_0. \tag{34}$$

Conditional on X_t, the first term on the right-hand side of Eq. (34) is deterministic and contributes nothing to the covariance, so $\mathrm{Cov}(\bar{\dot X}_t \mid X_t) = (\Delta_t^2/\beta_t^2)\,\mathrm{Cov}(S_v X_0 \mid X_t)$.

Posterior-mean covariance. By definition $\mathbb{E}[\bar{\dot X}_t \mid X_t] = \bar{u^\star}$, so the second term on the RHS of Eq. (30) is $\mathrm{Cov}(\bar{u^\star})$.

Substituting the three terms into Eq. (30) and solving for $\mathrm{Cov}(\bar{u^\star})$ yields Eq. (29). □

The right-hand side of Eq. (29) gives intuition for why $\mathrm{Cov}(\bar{u^\star})$ should have a spread spectrum. The identity term $\frac{\dot\beta_t^2}{N} I_{d_v}$ contributes uniformly across all d_v pooled velocity dimensions. The data-covariance term $\dot\alpha_t^2\, S_v \mathrm{Cov}(X_0) S_v^\top$ is positive semidefinite and adds further spread. The subtracted posterior term reduces this contribution, but its magnitude depends on the data geometry and cannot be bounded in closed form without further assumptions.

Equation (29) identifies $\mathrm{Cov}(\bar{u^\star})$ as a well-defined PSD matrix whose spectrum depends on the interpolant through $(\alpha_t, \beta_t, \Delta_t)$. Only its eigenvalues are used below, not the explicit form of the posterior term. We therefore abbreviate

$$\Gamma_t^\star := \mathrm{Cov}(\bar{u^\star}) \in \mathbb{R}^{d_v \times d_v}, \qquad \lambda_k^\star := \lambda_k(\Gamma_t^\star), \quad k = 1, \dots, d_v. \tag{35}$$

F.3 Operator-Norm Closeness of the Velocity Covariances

Lemma F.1 (Pooling does not increase excess risk)**.** $\varepsilon_t^{\mathrm{pool}} \le N^{-1}\varepsilon_t^{\mathrm{full}}$.

Proof. With $E := u_{\mathrm{full}}^{\theta} - u_{\mathrm{full}}^{\star}$ we have $\bar{u^{\theta}} - \bar{u^{\star}} = S_v E$. By definition of the operator norm, $\|S_v E\| \le \|S_v\|_{\mathrm{op}}\|E\|$, for each realization of X_t, so

$$\varepsilon_t^{\mathrm{pool}} = \mathbb{E}\big[\,\|S_v E\|^2\,\big] \le \|S_v\|_{\mathrm{op}}^2\,\mathbb{E}\big[\,\|E\|^2\,\big] = \|S_v\|_{\mathrm{op}}^2\,\varepsilon_t^{\mathrm{full}}. \tag{36}$$

Since $S_v S_v^{\top} = N^{-1} I_{d_v}$, all eigenvalues of $S_v S_v^{\top}$ equal N^{-1}, hence $\|S_v\|_{\mathrm{op}}^2 = N^{-1}$ and the claim follows. □

Proposition F.2 (Operator-norm closeness)**.** *Given the per-t excess risk $\varepsilon_t^{\mathrm{pool}}$, we can bound the operator-norm difference between the pooled velocity covariances as*

$$\big\|\mathrm{Cov}(\bar{u^{\theta}}) - \Gamma_t^{\star}\big\|_{\mathrm{op}} \le \varepsilon_t^{\mathrm{pool}} + 2\sqrt{\varepsilon_t^{\mathrm{pool}}\,\|\Gamma_t^{\star}\|_{\mathrm{op}}} =: \delta_t. \tag{37}$$

Proof. Write $\bar{E} := \bar{u^{\theta}} - \bar{u^{\star}}$. Expanding the covariance of the sum $\bar{u^{\theta}} = \bar{u^{\star}} + \bar{E}$, we get

$$\mathrm{Cov}(\bar{u^{\theta}}) - \Gamma_t^{\star} = \mathrm{Cov}(\bar{u^{\star}}, \bar{E}) + \mathrm{Cov}(\bar{E}, \bar{u^{\star}}) + \mathrm{Cov}(\bar{E}). \tag{38}$$

For the autocovariance term, a PSD matrix has operator norm equal to its largest eigenvalue and therefore at most its trace, so

$$\begin{aligned}\|\mathrm{Cov}(\bar{E})\|_{\mathrm{op}} &\le \mathrm{Tr}\,\big(\mathrm{Cov}(\bar{E})\big) = \mathrm{Tr}\,\big(\mathbb{E}((\bar{E} - \mathbb{E}[\bar{E}])(\bar{E} - \mathbb{E}[\bar{E}])^{\top})\big) \\ &= \mathbb{E}\big[\mathrm{Tr}\,\big((\bar{E} - \mathbb{E}[\bar{E}])^{\top}(\bar{E} - \mathbb{E}[\bar{E}])\big)\big] = \mathbb{E}\big[\|\bar{E}\|^2\,\big] - \|\mathbb{E}[\bar{E}]\|^2 \\ &\le \mathbb{E}\big[\,\|\bar{E}\|^2\,\big] = \varepsilon_t^{\mathrm{pool}}.\end{aligned} \tag{39}$$

In the second line, we used that expectation and trace commute, and that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ for matrices A, B of correct sizes. For the cross terms in Eq. (38), take unit vectors $u, v \in \mathbb{R}^{d_v}$. For any unit vector w, $\mathrm{Var}(w^{\top}W) = w^{\top}\mathrm{Cov}(W)\,w \le \|\mathrm{Cov}(W)\|_{\mathrm{op}}$. Applying the Cauchy–Schwarz inequality $\mathrm{Cov}(P, R) \le \sqrt{\mathrm{Var}(P)\,\mathrm{Var}(R)}$ to the scalars $P = u^{\top}\bar{u^{\star}}$ and $R = v^{\top}\bar{E}$, we obtain

$$u^{\top}\mathrm{Cov}(\bar{u^{\star}}, \bar{E})\,v = \mathrm{Cov}(u^{\top}\bar{u^{\star}},\, v^{\top}\bar{E}) \le \sqrt{\|\Gamma_t^{\star}\|_{\mathrm{op}}\,\varepsilon_t^{\mathrm{pool}}}, \tag{40}$$

where the variance of $v^{\top}\bar{E}$ is bounded by $\|\mathrm{Cov}(\bar{E})\|_{\mathrm{op}} \le \varepsilon_t^{\mathrm{pool}}$ from the autocovariance bound in Eq. (39). By swapping the roles of $\bar{E}$ and $\bar{u^{\star}}$, we obtain the same bound for $\mathrm{Cov}(\bar{E}, \bar{u^{\star}})$. Taking the supremum over unit u, v bounds $\|\mathrm{Cov}(\bar{u^{\star}}, \bar{E})\|_{\mathrm{op}}$, and applying the triangle inequality on Eq. (38) assembles the three terms into Eq. (37). □

By Lemma F.1 and Remark F.2, $\delta_t \to 0$ Lebesgue-a.e. as the model trains, so a well-trained network has $\mathrm{Cov}(\bar{u^{\theta}})$ operator-norm close to $\Gamma_t^{\star}$ Lebesgue-a.e. Turning this into a spectral control of the feature covariance Σ_t requires a structural assumption that connects Σ_t to $\mathrm{Cov}(\bar{u^{\theta}})$. The next section introduces this.

F.4 Spectral Control of the Feature Covariance

Assumption F.1 (A1: Affine linear feature-to-velocity map)**.** *At layer ℓ, the map from features to velocity is a shared per-token affine linear map: there exist deterministic $A_t \in \mathbb{R}^{d_v \times d}$ and $b_t \in \mathbb{R}^{d_v}$, depending on t but not on (X_0, X_1), such that*

$$u_m^{\theta}(x_t, t) = A_t\, h_{\ell,m}^{(t)}(x_t) + b_t \qquad \textit{for every token } m = 1, \dots, N, \tag{41}$$

and A_t has full row rank, $\mathrm{rank}(A_t) = d_v$ *(with $d_v \le d$).*

We call A_t the *feature-to-velocity map*: it sends the layer-ℓ feature of a token to that token's velocity output. Assumption F.1 is a modeling assumption: we idealize the sub-network from layer ℓ to the output as a shared per-token affine map. The idealization ignores the mixing of tokens through attention in the remaining blocks, the nonlinearity of these blocks, and the dependence of the map

on the class label. It holds exactly only at the input of the output head, taking h after the final LayerNorm and with a fixed label. The matrix A_t is the same for every token; it does not depend on the summation index m and therefore factors out of the token average:

$$\bar{u^\theta} = \frac{1}{N}\sum_{m=1}^{N}\big(A_t\, h_{\ell,m}^{(t)} + b_t\big) = A_t\Big(\frac{1}{N}\sum_{m=1}^{N} h_{\ell,m}^{(t)}\Big) + b_t = A_t\, \bar{h}_\ell^{(t)} + b_t. \tag{42}$$

Thus, the same matrix A_t maps the pooled feature to the pooled velocity. We write

$$B := \sigma_{\max}(A_t), \qquad b := \sigma_{\min}(A_t) > 0, \qquad \kappa := B/b, \tag{43}$$

where $b > 0$ by the full-row-rank assumption.

The bias b_t in Eq. (42) is deterministic and does not affect the covariance, so under Assumption F.1 the pooled learned velocity covariance and the pooled feature covariance are related by $\mathrm{Cov}(\bar{u^\theta}) = A_t\, \Sigma_t\, A_t^\top$.

By the spectral theorem, the symmetric PSD matrix Σ_t admits an eigendecomposition $\Sigma_t = Q\Lambda Q^\top$ with $Q \in \mathbb{R}^{d\times d}$ orthogonal and $\Lambda = \mathrm{diag}(\lambda_1(\Sigma_t), \dots, \lambda_d(\Sigma_t))$, the eigenvalues in non-increasing order. Let $Q_1 \in \mathbb{R}^{d\times d_v}$ be the submatrix of Q containing the top-d_v eigenvectors (i.e. the first d_v columns of Q) and $P_{\mathrm{top}} := Q_1 Q_1^\top \in \mathbb{R}^{d\times d}$ the orthogonal projector onto the top-d_v eigenspaces. Let $A_t = U\,\mathrm{diag}(\sigma_1, \dots, \sigma_{d_v})\, V_A^\top$ be the compact singular value decomposition with $U \in \mathbb{R}^{d_v\times d_v}, V_A \in \mathbb{R}^{d\times d_v}$ both orthogonal, and $P_A := V_A V_A^\top \in \mathbb{R}^{d\times d}$ the orthogonal projector onto the row space of A_t.

Assumption F.2 (A2: approximate spectral alignment)**.** *The feature covariance has a spectral gap, $\lambda_{d_v}(\Sigma_t) > \lambda_{d_v+1}(\Sigma_t)$, and the row space of the feature-to-velocity map is approximately aligned with the top-d_v eigenspaces of the feature covariance: $\|P_A - P_{\mathrm{top}}\|_{\mathrm{op}} \le \eta$ for some $\eta \in [0, 1)$.*

The spectral gap ensures P_{top} is uniquely defined.

The operator-norm bound from Proposition F.2, combined with Weyl's perturbation inequality, lower-bounds each of the leading d_v feature eigenvalues individually.

Lemma F.2 (Lower control of the leading feature eigenvalues)**.** *Under Assumption F.1, for every $k = 1, \dots, d_v$,*

$$\lambda_k(\Sigma_t) \;\ge\; \frac{(\lambda_k^\star - \delta_t)_+}{B^2}, \qquad (x)_+ := \max\{x, 0\}. \tag{44}$$

Proof. Write $M_t := A_t\Sigma_t A_t^\top = \mathrm{Cov}(\bar{u^\theta})$. By Proposition F.2 and *Weyl's perturbation inequality* (Bhatia, 2013, Cor. III.2.6), which states that $|\lambda_k(P) - \lambda_k(Q)| \le \|P - Q\|_{\mathrm{op}}$ for symmetric matrices P, Q, we get

$$\lambda_k^\star \overset{\triangle}{\le} \lambda_k(M_t) + |\lambda_k^\star - \lambda_k(M_t)| \overset{\text{Weyl}}{\le} \lambda_k(M_t) + \|M_t - \Gamma^\star\|_{\mathrm{op}} \le \lambda_k(M_t) + \delta_t \tag{45}$$
$$\iff \lambda_k(M_t) \ge \lambda_k^\star - \delta_t$$

Define $\Sigma_t^{1/2} := Q\Lambda^{1/2}Q^\top := Q\,\mathrm{diag}(\lambda_1(\Sigma_t)^{1/2}, \dots, \lambda_d(\Sigma_t)^{1/2})Q^\top$. Then $M_t = (A_t\Sigma_t^{1/2})(A_t\Sigma_t^{1/2})^\top$, and as the eigenvalues of a matrix $XX^\top$ are the squared singular values of X, we have

$$\lambda_k(M_t) = \sigma_k(A_t\Sigma_t^{1/2})^2. \tag{46}$$

The *singular-value product bound* $\sigma_k(MN) \le \sigma_{\max}(M)\,\sigma_k(N)$ (Horn & Johnson, 2013, §7.3), applied with $M = A_t$ and $N = \Sigma_t^{1/2}$, gives $\sigma_k(A_t\Sigma_t^{1/2}) \le \sigma_{\max}(A_t)\,\sigma_k(\Sigma_t^{1/2}) = B\sqrt{\lambda_k(\Sigma_t)}$. Hence, we obtain

$$\lambda_k(M_t) \le B^2\lambda_k(\Sigma_t). \tag{47}$$

Combining the two bounds Eqs. (45) and (47) on $\lambda_k(M_t)$ gives $\lambda_k^\star - \delta_t \le B^2\lambda_k(\Sigma_t)$, and $\lambda_k(\Sigma_t) \ge 0$ permits the positive part. □

Next, we want to upper bound the largest eigenvalue of Σ_t.

Lemma F.3 (Upper control of the largest feature eigenvalue)**.** *Under Assumptions F.1 and F.2,*

$$\lambda_1(\Sigma_t) \;\le\; \frac{\lambda_1^\star + \delta_t}{b^2(1-\eta^2)}. \tag{48}$$

Proof. The argument sandwiches $\lambda_1(M_t)$ between two bounds: an upper bound from the objective side via Weyl and Proposition F.2, and a lower bound that brings $\lambda_1(\Sigma_t)$ into the picture. Combining the two then yields an upper bound on $\lambda_1(\Sigma_t)$. Throughout, $M_t = A_t\,\Sigma_t\,A_t^\top = \mathrm{Cov}(\bar{u^\theta})$ as in Lemma F.2.

For the upper bound on $\lambda_1(M_t)$, Proposition F.2 and Weyl's perturbation inequality, applied in the upper direction (the mirror of Eq. (45)), give

$$\lambda_k(M_t) \overset{\triangle}{\le} \lambda_k^\star + |\lambda_k(M_t) - \lambda_k^\star| \overset{\text{Weyl}}{\le} \lambda_k^\star + \|M_t - \Gamma_t^\star\|_{\mathrm{op}} \le \lambda_k^\star + \delta_t. \tag{49}$$

This bound does not yet involve $\lambda_1(\Sigma_t)$. To translate it into a bound on $\lambda_1(\Sigma_t)$, we now lower-bound $\lambda_1(M_t)$ in terms of $\lambda_1(\Sigma_t)$.

Let q_1 be a unit top eigenvector of Σ_t. Since q_1 lies in the top-d_v eigenspace, $P_{\mathrm{top}}q_1 = q_1$. Hence $(I - P_A)q_1 = (P_{\mathrm{top}} - P_A)q_1$, and so

$$\begin{aligned}\|(I-P_A)q_1\| = \|(P_{\mathrm{top}} - P_A)q_1\| &\le \|P_{\mathrm{top}} - P_A\|_{\mathrm{op}}\,\|q_1\| \\ &= \|P_{\mathrm{top}} - P_A\|_{\mathrm{op}} \le \eta,\end{aligned} \tag{50}$$

where the last inequality is Assumption F.2. Because P_A is an orthogonal projector, $P_A q_1$ and $(I - P_A)q_1$ are orthogonal, so $1 = \|q_1\|^2 = \|P_A q_1\|^2 + \|(I-P_A)q_1\|^2$ and therefore

$$\|P_A q_1\|^2 \;\ge\; 1 - \eta^2. \tag{51}$$

Recall from Eq. (46) that $\lambda_1(M_t) = \sigma_1(A_t\,\Sigma_t^{1/2})^2$. The largest singular value of any matrix A can be characterized as $\sigma_1(A) = \sup_{\|w\|=1}\|Aw\|$, so for any unit vector w,

$$\sigma_1(A_t\,\Sigma_t^{1/2})^2 \;\ge\; \|A_t\,\Sigma_t^{1/2}\,w\|^2. \tag{52}$$

Choosing $w = q_1$ and using $\Sigma_t^{1/2}q_1 = \sqrt{\lambda_1(\Sigma_t)}\,q_1$,

$$\lambda_1(M_t) \;\ge\; \left\|A_t\,\Sigma_t^{1/2}\,q_1\right\|^2 \;=\; \left\|\sqrt{\lambda_1(\Sigma_t)}\,A_t\,q_1\right\|^2 \;=\; \lambda_1(\Sigma_t)\,\|A_t\,q_1\|^2. \tag{53}$$

It remains to lower-bound $\|A_t\,q_1\|^2$. Decompose $q_1 = P_A q_1 + (I - P_A)q_1$. Since A_t annihilates the orthogonal complement of its row space, $A_t\,(I - P_A)\,q_1 = 0$ and therefore $A_t\,q_1 = A_t\,P_A\,q_1$. By definition of the smallest singular value, $\|A_t\,v\| \ge b\,\|v\|$ for every v in the row space of A_t. Since $P_A q_1$ lies in this row space,

$$\|A_t\,q_1\| \;=\; \|A_t\,P_A\,q_1\| \;\ge\; b\,\|P_A\,q_1\|, \tag{54}$$

and combining with Eq. (51),

$$\|A_t\,q_1\|^2 \;\ge\; b^2\,(1-\eta^2). \tag{55}$$

Substituting Eq. (55) into Eq. (53) yields the lower bound on $\lambda_1(M_t)$ in terms of $\lambda_1(\Sigma_t)$:

$$\lambda_1(M_t) \;\ge\; b^2(1-\eta^2)\,\lambda_1(\Sigma_t). \tag{56}$$

Sandwiching $\lambda_1(M_t)$ between Eq. (56) and Eq. (49) at $k = 1$ gives

$$b^2(1-\eta^2)\,\lambda_1(\Sigma_t) \;\le\; \lambda_1(M_t) \;\le\; \lambda_1^\star + \delta_t, \tag{57}$$

and dividing by $b^2(1-\eta^2) > 0$ finally gives Eq. (48). □

The two lemmas are the spectral content of the chain: low excess risk lower-bounds the leading d_v feature eigenvalues and upper-bounds the largest one. Lemma F.2 uses only the linear feature-to-velocity map (Assumption F.1); the alignment assumption (Assumption F.2) enters only the upper bound, where misalignment costs the factor $1 - \eta^2$.

F.5 Main Result: An Effective-Rank Lower Bound

Write $T_\Sigma := \mathrm{Tr}(\Sigma_t)$ and $\bar{\lambda}_j := \lambda_j(\Sigma_t)/T_\Sigma$ for the normalized eigenvalues. Recall the effective rank (Roy & Vetterli, 2007) $\mathrm{eRank}(\Sigma_t) := \exp\big(-\sum_{j=1}^{d} \bar{\lambda}_j \log \bar{\lambda}_j\big)$.

Lemma F.4 (Effective rank versus trace-to-top ratio)**.** *For any PSD matrix $\Sigma_t \neq 0$, $\mathrm{eRank}(\Sigma_t) \geq T_\Sigma/\lambda_1(\Sigma_t)$.*

Proof. Let $\bar{\lambda}_{\max} := \max_j \bar{\lambda}_j = \lambda_1(\Sigma_t)/T_\Sigma$. Since $\bar{\lambda}_j \leq \bar{\lambda}_{\max}$, we have $-\log \bar{\lambda}_j \geq -\log \bar{\lambda}_{\max}$ for every j, so the entropy satisfies $-\sum_j \bar{\lambda}_j \log \bar{\lambda}_j \geq (-\log \bar{\lambda}_{\max}) \sum_j \bar{\lambda}_j = -\log \bar{\lambda}_{\max}$. Exponentiating, $\mathrm{eRank}(\Sigma_t) \geq 1/\bar{\lambda}_{\max} = T_\Sigma/\lambda_1(\Sigma_t)$. □

This bound needs only an upper bound on the largest eigenvalue and a lower bound on the trace, which is exactly what the previous section provides.

Corollary F.1 (Objective-only effective-rank bound)**.** *Under Assumptions F.1 and F.2,*

$$\mathrm{eRank}(\Sigma_t) \;\geq\; \kappa^{-2}(1-\eta^2)\,\frac{\sum_{k=1}^{d_v}(\lambda_k^\star - \delta_t)_+}{\lambda_1^\star + \delta_t}. \tag{58}$$

Proof. By Lemma F.4, $\mathrm{eRank}(\Sigma_t) \geq T_\Sigma/\lambda_1(\Sigma_t)$. Discarding the nonnegative tail eigenvalues and applying Lemma F.2, $T_\Sigma \geq \sum_{k=1}^{d_v} \lambda_k(\Sigma_t) \geq B^{-2} \sum_{k=1}^{d_v} (\lambda_k^\star - \delta_t)_+$. Applying Lemma F.3 to $\lambda_1(\Sigma_t)$ and dividing gives $T_\Sigma/\lambda_1(\Sigma_t) \geq (b^2(1-\eta^2)/B^2) \sum_k (\lambda_k^\star - \delta_t)_+/(\lambda_1^\star + \delta_t)$, which is Eq. (58) since $\kappa = B/b$. □

In the small-risk limit $\delta_t \to 0$, Eq. (58) becomes $\mathrm{eRank}(\Sigma_t) \gtrsim \kappa^{-2}(1-\eta^2)\ \mathrm{Tr}(\Gamma_t^\star)/\lambda_1(\Gamma_t^\star)$. The ratio $\mathrm{Tr}(\Gamma_t^\star)/\lambda_1(\Gamma_t^\star)$ is the spectral spread of the pooled Bayes-optimal velocity covariance. Since $\Gamma_t^\star \in \mathbb{R}^{d_v \times d_v}$, it is at most d_v. The objective alone, therefore, certifies $\mathrm{eRank}(\Sigma_t)$ only up to order d_v. To obtain a bound that can exceed d_v, one extra ingredient is needed.

Assumption F.3 (A3: nontrivial pooled feature energy)**.** *There is a $\tau_t > 0$ with $\mathrm{Tr}(\Sigma_t) \geq \tau_t$.*

Assumption F.3 only rules out a degenerate, near-zero pooled covariance. It says nothing about how the variance is distributed across directions. The trace $\mathrm{Tr}(\Sigma_t)$ is directly estimable from the same covariance used to compute the effective rank.

Theorem F.1 (Effective-rank lower bound)**.** *Under Assumptions F.1 to F.3,*

$$\mathrm{eRank}(\Sigma_t) \;\geq\; \frac{\tau_t\, b^2(1-\eta^2)}{\lambda_1^\star + \delta_t}. \tag{59}$$

Proof. By Lemma F.4 and Assumption F.3, $\mathrm{eRank}(\Sigma_t) \geq T_\Sigma/\lambda_1(\Sigma_t) \geq \tau_t/\lambda_1(\Sigma_t)$. Substituting the upper bound Eq. (48) on $\lambda_1(\Sigma_t)$ gives Eq. (59). □

F.6 Scope and Connection to the Empirical Results

Corollary F.1 uses only low flow-matching excess risk and the geometry of the feature-to-velocity map, but the resulting lower bound on $\mathrm{eRank}(\Sigma_t)$ cannot exceed d_v. The objective alone therefore cannot certify $\mathrm{eRank}(\Sigma_t) > d_v$. Theorem F.1 reaches beyond d_v by taking the trace τ_t as an empirical input. The alignment parameter η is similarly an input, not a consequence of the objective. The trace-to-top inequality of Lemma F.4 is tight only for a flat spectrum, and the condition-number factor κ^{-2} loosens Corollary F.1 further. The result is a necessary condition. Low excess risk forces a lower bound on $\mathrm{eRank}(\Sigma_t)$ under Assumptions F.1 to F.3, but the appendix does not establish sufficiency, nor does it predict the magnitude of the empirical effective-rank correlation reported in the main text. The analysis also treats u^θ, $u^\star$ and Σ_t as functions of X_t alone, without a class input. The models in the main text are class-conditional, and effective rank is measured with the true class label as input (Section C), so the result applies to them only through this idealization. What the result offers is therefore intuition: if the network after layer ℓ acts roughly linearly on the pooled feature, fitting the objective well forces the feature covariance to spread over several directions. It is intuition for this necessity, not an explanation of the measured correlation.

G USE OF LARGE LANGUAGE MODELS

Aid or polish writing. All text was revised by the authors, manually and with the help of the language models Claude Opus 5 and Claude Opus 5.5 (Anthropic), which were used to polish the language.

Retrieval and discovery. The authors used deep-research tools based on language models to search for related work and other sources. The retrieved sources were evaluated manually, and all references were selected and added by the authors.

Drafting sections. The authors wrote bullet-point outlines of every section by hand. Claude Opus 5 and Claude Opus 5.5 drafted the sections from these outlines, and the authors then revised the drafts as described above.

Research ideation or execution. The research question, the high-level approach and the design of the experiments are the authors' own. Language models were used to implement the experiments that the authors had planned.

Proving mathematical claims. The authors developed the effective-rank result of Section F and its proof. Claude Opus 4.6 was then used to improve the result, in particular to weaken its assumptions, and to stress-test the argument.

The authors take full responsibility for all content of this paper and have verified every claim, number and proof.