PURIFY BEFORE YOU ALIGN: IMPROVING REPRESENTATION ALIGNMENT FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation alignment improves and accelerates diffusion training by encouraging intermediate features to match a pretrained visual encoder. Yet choosing the right encoder remains largely empirical: DINOv2 provides substantial gains, whereas MAE, despite retaining more reconstructive information, offers little benefit. We introduce *local appearance purification* (LAP), which improves generation quality and makes encoders such as MAE effective alignment targets by removing locally predictable content and aligning to the residual. LAP modifies only the alignment target and requires no changes to diffusion training or inference. Through controlled interventions within a fixed encoder, we show that effective alignment depends on what information the target emphasizes, not simply how much image information it contains. Our theory interprets standard alignment as maximizing a variational lower bound on the information that the denoiser's hidden state carries about the target, with purified alignment yielding a corresponding bound conditioned on clean latent patches. We further derive an exact constant-readout baseline that measures how well the target can be matched without using the hidden state. LAP improves generation across five pretrained representations, with the largest gains for MAE: on ImageNet-256 with SiT-B/2, it reduces FID from 53.1 to 42.3. Its gains persist when combined with stronger alignment recipes and tokenizers. Across 13 encoders, the constant-readout score predicts the benefit of purification before diffusion training (Spearman $\rho = 0.94$).

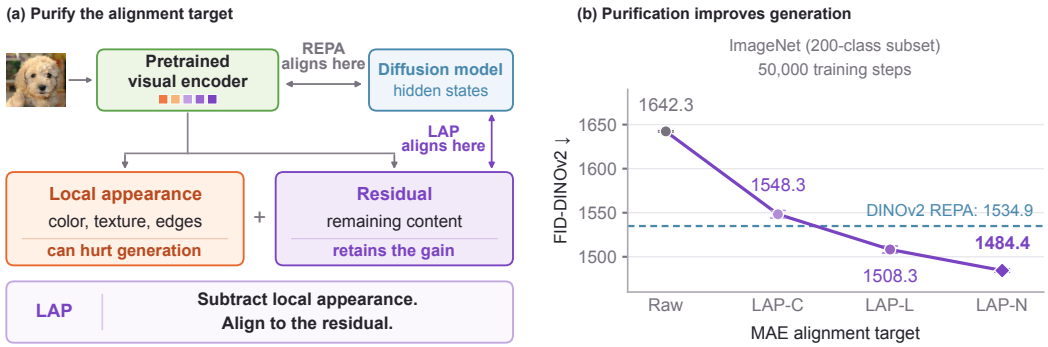
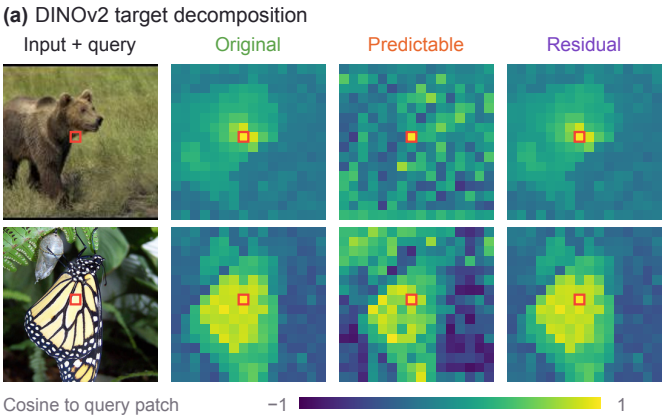


Figure 1: **Purification improves generation by changing the alignment target.** (a) The DINOv2 split (Figure 2) suggests that local reconstruction content can duplicate denoising supervision. LAP subtracts a local prediction and aligns to the residual, which may still contain local information. (b) All three LAP variants improve MAE-based generation for SiT-B/2. LAP-L and LAP-N also surpass DINOv2 REPA in the development setting (200 classes, 50K steps, FID-DINOv2).

1 INTRODUCTION

Representation alignment has become part of the standard recipe for training diffusion models. REPA (Yu et al., 2025) encourages intermediate denoiser features to match those of a frozen pretrained encoder, substantially accelerating training and improving sample quality. A growing line of work builds on this idea (Leng et al., 2025; Wang et al., 2025b; Wu et al., 2025; Yang et al., 2026). However, one important choice remains largely empirical: which encoder should provide

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.



(b) Alignment outcome

Alignment target	FID ↓
Original	41.75
Predictable component	56.71
Residual (LAP)	41.47
No alignment	53.92

Predictable part hurts

Residual keeps the benefit

Figure 2: **One encoder, opposite alignment effects.** (a) DINOv2 features split into a locally predictable component and a residual. Each map shows the similarity of every patch to the red-boxed patch (details in Appendix D). (b) Aligning to the locally predictable component hurts generation, whereas the residual (LAP) keeps the benefit.

the alignment target? Different pretrained encoders lead to very different generation quality, and we still lack a clear principle for what makes a representation useful for alignment.

The puzzle is sharpest for MAE (He et al., 2022). Since diffusion models learn to recover clean images from noise, features that preserve reconstructive information might seem especially suitable. MAE features predict clean-image content far better than DINOv2 (Oquab et al., 2024), yet aligning to them barely improves generation, whereas DINOv2 provides a large gain (Figure 3). Recent work shows that target processing matters, including spatial operations in iREPA (Singh et al., 2026) and other target transformations (Shin et al., 2026; Ning et al., 2026). Appendix K reviews related work. We ask a more basic question:

> *What information should an alignment target ask the denoiser to match?*

To answer this, we hold the pretrained encoder fixed and intervene on the information selected for alignment. For DINOv2, aligning only to the locally predictable component performs worse than no alignment, while its residual retains the full benefit (Figure 2). Likewise, making the aligned representation more predictive of the denoising target improves prediction but worsens generation (Section 3.1). Together, these results show that effective alignment depends on what information the target emphasizes, not simply how much image information it contains.

These results suggest a simple view: the denoising objective already provides a strong reconstruction signal, so alignment is most useful when it emphasizes information beyond what denoising already encourages. This view also suggests a simple fix without replacing the encoder. *Local appearance purification* (LAP) removes target content predictable from local appearance and aligns the denoiser to the residual, using constant, linear, or nonlinear predictors (Figure 4). Across multiple visual teachers, LAP consistently improves generation, with the largest gains for MAE. On ImageNet-256 with SiT-B/2, LAP reduces MAE-based FID from 53.1 to 42.3, while also improving AIMv2 and CLIP. Its gains persist when combined with stronger alignment recipes and tokenizers, showing that target purification complements improvements elsewhere in the alignment pipeline. Beyond natural images, LAP could make self-supervised reconstructive encoders useful alignment teachers in domains such as medical and scientific imaging, where suitable discriminative teachers may be unavailable or poorly matched to the data.

We also develop an information-theoretic view of purified alignment. Standard alignment maximizes a variational lower bound on the information that the hidden state carries about the teacher target, while purified alignment yields a corresponding bound conditioned on local appearance. However, a high alignment score need not imply an informative hidden state: if target tokens share a common direction, even a constant readout that ignores the hidden state can achieve a high cosine score. We derive the exact optimum of this constant readout in closed form, giving a diagnostic that can be computed before diffusion training. In a matched sweep of 13 encoders, this score strongly pre-

dicts the benefit of purification (Spearman $\rho = 0.94$), providing a practical criterion for identifying teachers likely to benefit from LAP (Section 5.5).

Contributions. (1) Through fixed-encoder interventions, we show that alignment quality depends on what information the target emphasizes, not simply how much image information it contains. In particular, locally predictable content can hurt generation even when it is easy to match. (2) We introduce LAP, a family of target transformations that removes locally predictable content before alignment. LAP improves generation across multiple visual teachers, composes with stronger alignment recipes and tokenizers, and requires no encoder change or inference overhead. (3) We develop an information-theoretic view of raw and purified alignment and derive a closed-form constant-readout diagnostic for alignment achievable without the hidden state. Across 13 encoders, this score predicts the benefit of purification before diffusion training.

2 Background

2.1 Representation Alignment in Diffusion Training

We follow the latent flow-matching setup of REPA (Yu et al., 2025; Ma et al., 2024; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023). A frozen SD-VAE (Rombach et al., 2022) encodes an image I into a clean latent X. The denoiser receives $x_t = (1-t)X + t\varepsilon$, $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$, and predicts the velocity $v^* = \varepsilon - X$ with the flow-matching loss $\mathcal{L}_{\text{flow}} = \mathbb{E}\,\|v_\theta(x_t, t) - v^*\|^2$. REPA adds supervision from a frozen visual encoder f: a trainable projector g_ψ maps the hidden state at block l to an aligned representation $a_t = g_\psi(h_t^l)$ and matches it token-wise to features of the clean image,

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_a \mathcal{L}_{\text{align}}, \qquad \mathcal{L}_{\text{align}} = -\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \cos\left(a_t^{(i)}, y^{(i)}\right)\right], \tag{1}$$

where N is the number of patch tokens. Standard REPA uses the raw encoder output $y = z = f(I)$. The raw or purified teacher target occupies the same slot y. See Figure 4 for illustration.

2.2 Measuring Representation Content and Use

To understand alignment, we distinguish the information available in a representation from the information the denoiser actually uses. We measure the former with linear probes and the latter with hidden-state subspace ablations. For features U and responses V, let $R^2(U \to V)$ denote the held-out coefficient of determination of a ridge regression,

$$R^2(U \to V) = 1 - \frac{\sum_{\text{eval}} \|V - \hat{W}^\top U - \hat{b}\|^2}{\sum_{\text{eval}} \|V - \bar{V}\|^2}, \tag{2}$$

where $(\hat{W}, \hat{b})$ is fit on a separate set of images from those used for evaluation, and $\bar{V}$ is the evaluation-set mean. Let X_p and v_p^* denote the patchified clean latent and velocity target. We use three probes. *Prediction utility*, $R^2(y \to X_p)$, measures how much clean-image information is available in the frozen alignment target and can therefore be evaluated before diffusion training. *Presence*, $R^2(h_t^l \to z)$, measures how much of the teacher representation can be decoded from the denoiser hidden state. *Velocity sufficiency*, $R^2(a_t \to v_p^*)$, measures how predictive the aligned representation is of the denoising target.

Linear probes establish that information is decodable, but not that the denoiser uses it. To measure reliance, we ablate the hidden-state subspace associated with the aligned representation. Let Π project onto the span of the top r left singular vectors of the probe from h_t^l to a_t, and replace $h_t^l \leftarrow (\mathbf{I} - \Pi)h_t^l + \Pi\bar{h}_t$, where $\bar{h}_t$ is the mean hidden state in the same noise-level bin. Let $\mathcal{E}_\Pi$, $\mathcal{E}_{\text{full}}$, and $\mathcal{E}_0$ denote the velocity MSE after subspace ablation, full-layer mean filling, and no ablation, respectively. We normalize the effect of the subspace ablation by the effect of ablating the full layer: $U_\Pi = (\mathcal{E}_\Pi - \mathcal{E}_0)/(\mathcal{E}_{\text{full}} - \mathcal{E}_0)$. We compare U_Π against rank-matched random subspaces. The detailed experimental setup for these diagnostics is summarized in Section 5.1.

3 Why Purify the Alignment Target?

Across five pretrained encoders, better clean-image prediction accompanies worse generation (Figure 3). MAE predicts clean-latent patches best but offers little alignment benefit, whereas targets

with lower prediction utility improve generation more. This raises a question about what the denoiser needs from its teacher: reconstruction information, or a different constraint on its intermediate representation? A cross-encoder comparison cannot identify the cause because architecture, data, and pretraining objective change together. We therefore hold DINOv2 fixed and intervene in two ways. First, we encourage the aligned representation to predict the denoising target more accurately. Second, we separate locally predictable target content from its residual and align to each alone. These interventions test the role of reconstruction content without replacing the encoder.

3.1 Better Denoising Prediction Can Hurt Generation

We first encourage the aligned representation a_t to predict the denoising target. A small auxiliary head r_ω adds the loss $\lambda_s \mathbb{E}\|r_\omega(a_t, t) - v^*\|^2$ to REPA. At full scale, it raises held-out velocity-prediction R^2 from 0.11 to 0.53, yet worsens FID-50K from 41.75 to 46.79 (Tables 21 and 27). Better prediction through the aligned representation therefore does not translate into better generation. The loss sweep (Figure 12) and controls using other projections (Appendix D) show the same qualitative behavior. The unmodified REPA model shows why prediction alone is insufficient. DINOv2 features are decodable from the hidden state (presence $R^2 = 0.207$), but the denoiser relies only weakly on the aligned subspace: ablating it increases velocity error by only 12.2% of the full-layer effect, compared with 22.7% for rank-matched random subspaces and 14.0% for the untrained projector subspace in the unaligned model (Figure 12). Decodability does not imply reliance. This ablation tests reliance on the specific aligned subspace, but does not imply that all reconstruction-related information in the hidden state is unimportant.

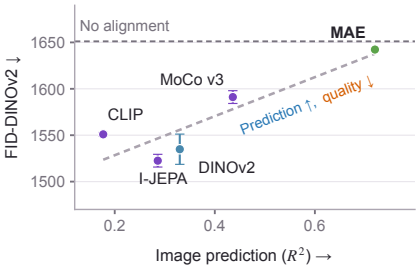


Figure 3: Image prediction does not track generation quality (Table 28).

3.2 Locally Predictable Content Does Not Drive the Alignment Benefit

We next ask whether locally predictable content is responsible for the benefit of a strong alignment target. We use an affine predictor from clean-latent patches to DINOv2 tokens to decompose each target feature into a locally predictable component and a residual, then align to each component separately while keeping the encoder and training recipe fixed. The residual corresponds to the linear LAP variant defined in Section 4.1. The two components have sharply different effects (Figure 2). Aligning only to the locally predictable component worsens FID from 53.92 without alignment to 56.71, whereas aligning to the residual reaches 41.47, preserving the gain of raw DINOv2 at 41.75. This difference is not explained by alignment difficulty: the predictable component is easier to match despite producing worse generation. Together with the previous result, this supports a view in which alignment is useful not because it provides more reconstruction information, but because it can shape intermediate representations beyond what denoising already encourages. This motivates purification. We next formalize this idea as a target construction, develop its information-theoretic interpretation, and introduce a diagnostic for evaluating alignment targets.

4 Local Appearance Purification for Representation Alignment

The results above suggest a simple design principle: alignment should emphasize information that complements the reconstruction signal already provided by denoising. We turn this idea into *local appearance purification* (LAP), which removes locally predictable content from a target representation and aligns the denoiser to the residual. To analyze this objective, we develop a variational interpretation of what alignment encourages the hidden state to encode and a constant-readout diagnostic that measures how much alignment can be achieved even without using the hidden state.

4.1 Local Appearance Purification

We define locally predictable content as the part of an encoder representation predictable from local clean-image appearance. Let $z = f(I)$ be the frozen encoder features and X_p the patchified clean latent. We fit a predictor d_ϕ from X_p to z and align to the residual:

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\,\|z - d_\phi(X_p)\|^2\,, \qquad y^{\text{pure}} = \text{sg}[z - d_\phi(X_p)]\,. \tag{3}$$

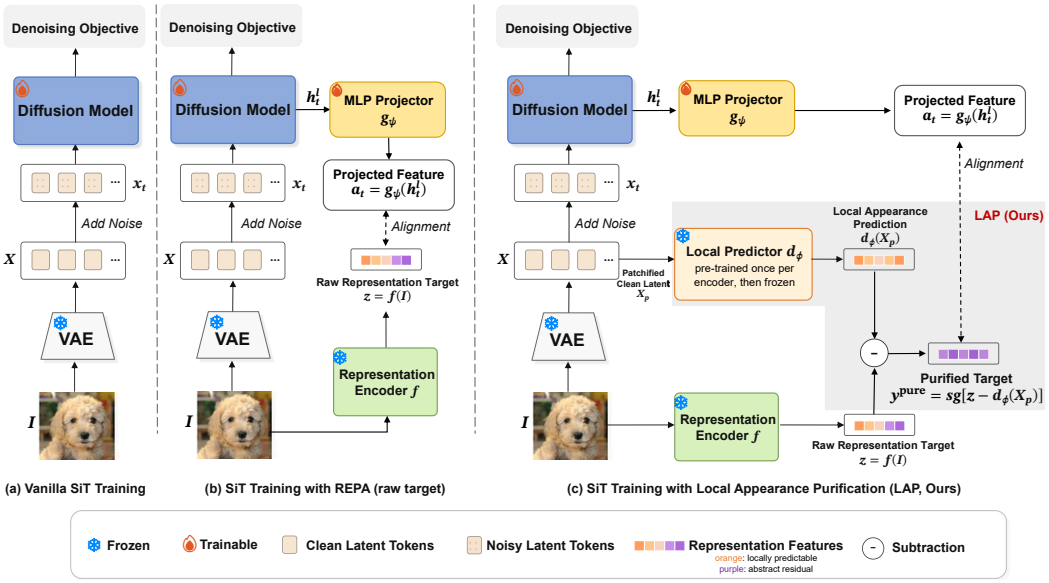


Figure 4: **Our LAP-N framework.** Compared with SiT training with REPA, LAP-N modifies only the alignment target by subtracting the reconstruction content predicted by a frozen local predictor d_ϕ. Training and inference are otherwise unchanged.

Here sg denotes stop-gradient. Replacing y with y^{pure} in Equation (1) gives the LAP objective. The predictor selects what to remove and receives no gradients from diffusion training. The denoising objective, model architecture, and inference procedure remain unchanged (Figure 4). We consider three variants of d_ϕ, from a constant offset to a nonlinear local predictor.

LAP-C: constant purification. Restricting d_ϕ to a constant yields batch-mean centering: we subtract the mean target feature over all images and token positions in each minibatch. LAP-C therefore is a degenerate member of the family that removes only the component shared across the batch.

LAP-L: linear purification. We fit an affine map from each clean-latent patch to its corresponding target token and subtract its prediction. This gives the residual used in Section 3.2. The affine predictor is refitted independently on each minibatch, with no gradient through the fit (Appendix D).

LAP-N: nonlinear purification. We train a small convolutional predictor to recover the target from a local neighborhood of clean-latent patches. The predictor is fitted before diffusion training, frozen, and reused throughout training. Details are given in Appendix E.

What LAP removes depends on the predictor: local patches may still contain semantic information (Brendel & Bethge, 2019), and a limited predictor may not capture all locally predictable content. We therefore do not interpret the residual as a purely semantic representation, but instead evaluate its content and effect on generation empirically in Section 5. Purification is used only to construct training targets and adds no inference cost.

4.2 A Variational View of Purified Alignment

The denoising objective trains the model to predict velocity, while alignment encourages the hidden state to preserve information about the teacher. We interpret this alignment objective probabilistically, which lets us connect cosine alignment to a variational information bound. For fixed l, t, write $h = h_t^l$ and $a = g_\psi(h)$. For each nonzero token, let $\hat{y}^{(i)} = y^{(i)}/\|y^{(i)}\|$ and define $\hat{a}^{(i)}$ analogously. We interpret cosine alignment as predicting the direction of each teacher token from the hidden state. Specifically, we use a von Mises–Fisher decoder (Banerjee et al., 2005) whose likelihood increases with the cosine similarity between the normalized readout $\hat{a}^{(i)}$ and target $\hat{y}^{(i)}$:

$$p(\hat{y}^{(i)} \mid h) \propto \exp\left(\kappa\, \hat{a}^{(i)\top} \hat{y}^{(i)}\right), \qquad \kappa > 0.$$

Under this decoder, maximizing the expected log-likelihood is equivalent, up to constants and scaling, to minimizing the cosine alignment loss. Thus, lower alignment loss corresponds to a better

probabilistic prediction of the teacher representation from the hidden state. This probabilistic decoder lets us connect alignment loss to the information that the hidden state carries about the teacher target. Applying the variational bound of Barber & Agakov (2003) gives

$$\mathcal{I}(h; z) \geq c_\sigma - \kappa N \, \mathcal{L}^{(t)}_{\text{align}}(z), \qquad \mathcal{I}(h; z \mid X_p) \geq c'_\sigma - \kappa N \, \mathcal{L}^{(t)}_{\text{align}}(y^{\text{pure}}). \tag{4}$$

Here $\mathcal{I}$ denotes mutual information, with losses evaluated at fixed t. The first bound interprets standard alignment as encouraging h to retain teacher information. For a fixed purifier such as LAP-N, the second instead measures teacher information retained beyond what is predictable from clean latent patches X_p. Since $\kappa N > 0$, minimizing either loss maximizes its lower bound. The constants c_σ and c'_σ are independent of the trainable denoiser. Derivations and assumptions are given in Appendix B. These bounds may be loose and do not by themselves establish information separation or explain generation gains. We next introduce a directly measurable diagnostic of alignment targets.

4.3 The Constant-Readout Test

The variational view links alignment score to how well the teacher target can be predicted from the hidden state. However, a high score can also arise from structure shared across the targets themselves. If many target tokens point in a similar direction, a readout that ignores the hidden state and always outputs the same vector can still achieve a high cosine score. We therefore ask how well such a constant readout can align to the target. For projector g_ψ, define the alignment score

$$C_\psi(y) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \cos\left(g_\psi(h)^{(i)}, y^{(i)}\right)\right]. \tag{5}$$

Now consider a readout that outputs the same unit vector μ for every image and token. Let $m_y = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \hat{y}^{(i)}\right]$. The expected cosine score of this constant readout is $\mu^\top m_y$. Maximizing over all unit vectors gives the exact baseline

$$C_0(y) = \max_{\|\mu\|=1} \mu^\top m_y = \|m_y\|. \tag{6}$$

Thus, C_0 measures how much cosine alignment can be obtained from shared target structure alone, without using the hidden state. Since it depends only on the target distribution, C_0 can be computed before diffusion training by normalizing the target tokens, averaging them across images and positions, and taking the norm. After training, comparing C_ψ with C_0 measures how much the learned readout gains beyond this constant baseline. Under the same directional decoder, the optimal gain over C_0 also admits an interpretation as predictive $\mathcal{V}$-information (Xu et al., 2020) (derivations in Appendix B.5). We test this prediction across encoders in Section 5.5.

5 Experiments

We now evaluate LAP across teacher encoders, alignment settings, training scales, and datasets, and ask when purification provides the largest generation gains. We also test whether the constant-readout score can identify promising targets before diffusion training.

5.1 Experimental Setup

Unless stated otherwise, we train class-conditional SiT-B/2 on ImageNet-1k at 256×256 resolution for 400K steps with batch size 64, the SD-VAE, and representation alignment at block 8. We evaluate generation without guidance under the ADM protocol (Dhariwal & Nichol, 2021), reporting FID-50K, Inception Score (IS), precision, and recall. Results are averaged over two training seeds. Evaluation details and individual runs are provided in Appendices H and L. For development experiments, we use a 200-class subset (*dev-200*) and report FID-DINOv2 on 10K samples (Stein et al., 2023). These values are not directly comparable to FID-50K. Hidden-state diagnostics use disjoint probe-fitting and evaluation images, noise levels $t \in [0.3, 0.7)$, and ablation rank 256 (Appendix C).

5.2 Generation Quality across Teachers

Raw MAE barely improves over no alignment, whereas LAP-N reduces FID from 53.05 to 42.30, approaching DINOv2-based REPA at 41.75 (Table 1). AIMv2 and CLIP also improve in both seeds,

Table 1: **Purification across five teachers.** SiT-B/2 on ImageNet-256 over two training seeds. Teachers are ordered by raw-target C_0 on dev-200. Shaded rows use LAP variants. Bold marks the best mean per teacher and metric. Full results are in Tables 29 and 30.

Teacher	C_0	Method	FID ↓	IS ↑	Prec. ↑	Rec. ↑
No alignment			53.92 ±0.03	25.34 ±0.16	0.435 ±0.002	0.602 ±0.006
MAE ViT-L	0.93	REPA	53.05 ±0.90	26.30 ±0.45	0.438 ±0.004	0.599 ±0.001
		+LAP-C	45.17 ±0.14	31.55 ±0.28	0.477 ±0.000	0.617 ±0.000
		+LAP-L	45.15 ±0.48	31.44 ±0.25	0.480 ±0.005	0.617 ±0.003
		+LAP-N	**42.30** ±1.08	**33.85** ±0.44	**0.496** ±0.006	**0.622** ±0.010
AIMv2 ViT-L	0.66	REPA	47.97 ±0.15	29.88 ±0.01	0.464 ±0.000	0.615 ±0.007
		+LAP-C	44.96 ±0.88	31.98 ±0.78	0.482 ±0.005	0.618 ±0.005
		+LAP-L	**44.31** ±0.13	**32.67** ±0.32	**0.484** ±0.000	**0.622** ±0.006
		+LAP-N	44.56 ±0.82	32.40 ±0.57	0.483 ±0.004	0.621 ±0.005
CLIP ViT-L	0.61	REPA	44.74 ±0.01	32.21 ±0.13	0.481 ±0.003	0.615 ±0.007
		+LAP-C	42.60 ±0.29	34.05 ±0.25	0.495 ±0.003	**0.626** ±0.003
		+LAP-L	42.69 ±0.87	33.91 ±0.65	0.493 ±0.004	0.621 ±0.004
		+LAP-N	**42.21** ±0.16	**34.38** ±0.23	**0.495** ±0.001	0.624 ±0.009
DINOv3 ViT-B	0.52	REPA	42.73 ±0.86	34.25 ±1.01	0.486 ±0.004	0.623 ±0.003
		+LAP-C	42.91 ±1.25	34.20 ±0.75	0.487 ±0.011	0.629 ±0.003
		+LAP-L	**42.07** ±1.63	**34.91** ±1.14	**0.490** ±0.011	**0.633** ±0.000
		+LAP-N	43.27 ±0.24	34.02 ±0.23	0.485 ±0.002	0.628 ±0.003
DINOv2 ViT-B	0.22	REPA	41.75 ±0.56	35.50 ±0.62	0.492 ±0.001	0.630 ±0.005
		+LAP-C	42.64 ±1.19	34.62 ±1.13	0.490 ±0.006	0.624 ±0.002
		+LAP-L	**41.47** ±1.21	**35.53** ±1.07	**0.495** ±0.010	0.621 ±0.002
		+LAP-N	42.11 ±0.30	35.26 ±0.34	0.493 ±0.002	**0.632** ±0.002

with lower FID accompanied by higher IS, precision, and recall. Using LAP-L as a common comparison across all five teachers, mean FID improves by 7.9, 3.7, and 2.0 for MAE, AIMv2, and CLIP, but by only 0.7 and 0.3 for DINOv3 and DINOv2, respectively. The latter differences are within seed variation. Purifier capacity also matters differently across teachers: for MAE, LAP-C and LAP-L already reach about 45.2 FID, while LAP-N further improves to 42.3. For CLIP, simpler purification captures most of the gain, whereas DINOv2 changes little across variants. Overall, purification can substantially improve otherwise weak alignment targets, but the magnitude of the benefit and the useful purifier capacity depend on the teacher.

5.3 LAP Complements Different REPA-style Recipes and Tokenizers

Since LAP changes only the alignment target, we ask whether its gains persist when other parts of the alignment pipeline are improved. We combine LAP with spatial operations in iREPA (Singh et al., 2026), the alignment schedule in VA-REPA (Yang et al., 2026), the entangled class token in REG (Wu et al., 2025), and the structural loss in sREPA (Xu et al., 2026). We also evaluate LAP on different latent tokenizers such as SD-VAE (Rombach et al., 2022), REPA-E VAE (Leng et al., 2025), and EQ-VAE (Kouzelis et al., 2025).

LAP adds to stronger alignment recipes. For MAE, LAP-N improves FID from 43.06 to 41.32 with spatial operations from iREPA and from 52.82 to 41.60 with the VA-REPA schedule (Table 2). Spatial operations substantially improve MAE on their own, yet they do not subsume purification. REG and sREPA leave raw MAE relatively weak (52.92 and 50.43 FID), whereas adding LAP-N lowers them to 41.65 and 38.46, respectively. For DINOv2, the selected LAP variants reduce mean FID across all four recipes by 0.3–1.2, with lower FID in both seeds and higher mean IS and precision. Thus, purification remains useful across changes to the alignment loss, schedule, and architecture, although the gains are smaller for an already strong DINOv2 target.

LAP makes MAE effective across tokenizers. Stronger tokenizers substantially improve the generative backbone, but do not solve the target-selection problem: raw MAE remains ineffective with SD-VAE, EQ-VAE, and REPA-E VAE (Table 3). LAP changes this consistently. It reduces FID from 53.05 to 42.30 with SD-VAE, from 52.41 to 40.01 with EQ-VAE, and from 33.74 to 27.56

Table 2: **LAP on top of REPA-family recipes.** Same protocol as Table 1. Each +LAP row uses the variant with the best FID. Bold marks the better one. Full results are in Table 21.

Teacher	Method	FID ↓	IS ↑	Prec. ↑	Rec. ↑
MAE ViT-L	iREPA	43.06 ±0.62	33.23 ±0.66	0.491 ±0.004	0.617 ±0.006
	+LAP	**41.32** ±0.28	**34.70** ±0.37	**0.502** ±0.002	**0.626** ±0.001
	VA-REPA	52.82 ±0.90	26.32 ±0.58	0.437 ±0.001	0.606 ±0.006
	+LAP	**41.60** ±0.91	**34.06** ±0.53	**0.503** ±0.004	**0.622** ±0.004
	REG	52.92 ±0.53	26.33 ±0.41	0.434 ±0.002	0.608 ±0.002
	+LAP	**41.65** ±0.06	**34.31** ±0.16	**0.500** ±0.003	**0.626** ±0.002
	sREPA	50.43 ±0.53	27.41 ±0.22	0.454 ±0.002	0.606 ±0.009
	+LAP	**38.46** ±0.62	**36.89** ±0.30	**0.519** ±0.003	**0.622** ±0.002
DINOv2 ViT-B	iREPA	41.32 ±0.19	35.65 ±0.01	0.496 ±0.004	**0.635** ±0.003
	+LAP	**40.55** ±0.39	**36.29** ±0.16	**0.503** ±0.003	0.632 ±0.004
	VA-REPA	40.71 ±0.14	36.27 ±0.24	0.500 ±0.002	0.635 ±0.001
	+LAP	**40.39** ±0.02	**36.54** ±0.04	**0.502** ±0.001	**0.637** ±0.006
	REG	29.24 ±0.82	57.08 ±2.10	0.559 ±0.003	0.610 ±0.001
	+LAP	**28.67** ±0.26	**57.49** ±0.07	**0.567** ±0.005	**0.615** ±0.004
	sREPA	38.13 ±0.01	39.55 ±0.01	0.515 ±0.005	**0.635** ±0.004
	+LAP	**36.92** ±0.42	**41.11** ±0.46	**0.522** ±0.002	0.631 ±0.005

Table 3: **LAP with other tokenizers.** MAE ViT-L teacher, same protocol as Table 1. Each +LAP row uses LAP-N. Bold marks the best. Full results are in Table 21.

Tokenizer	Method	FID ↓	IS ↑	Prec. ↑	Rec. ↑
SD-VAE	No alignment	53.92 ±0.03	25.34 ±0.16	0.435 ±0.002	0.602 ±0.006
	REPA	53.05 ±0.90	26.30 ±0.45	0.438 ±0.004	0.599 ±0.001
	+LAP	**42.30** ±1.08	**33.85** ±0.44	**0.496** ±0.006	**0.622** ±0.010
EQ-VAE	No alignment	52.50 ±0.04	25.99 ±0.05	0.449 ±0.003	0.599 ±0.004
	REPA	52.41 ±1.37	26.31 ±0.91	0.452 ±0.004	0.597 ±0.004
	+LAP	**40.01** ±0.25	**35.24** ±0.36	**0.523** ±0.003	**0.608** ±0.005
REPA-E VAE	No alignment	32.30 ±0.04	40.61 ±0.09	0.584 ±0.002	0.604 ±0.005
	REPA	33.74 ±0.45	39.32 ±0.88	0.574 ±0.005	**0.611** ±0.003
	+LAP	**27.56** ±0.40	**47.43** ±0.44	**0.606** ±0.003	0.607 ±0.000

with REPA-E VAE. Notably, with the stronger REPA-E tokenizer, raw MAE is worse than no alignment (33.74 vs. 32.30), whereas LAP turns it into a substantial gain (27.56), while also improving IS from 39.32 to 47.43 and precision from 0.574 to 0.606. These results show that LAP remains valuable even after large improvements to the latent representation: the tokenizer improves the generative substrate, while purification makes MAE useful as an alignment target. The benefit also extends beyond MAE, with LAP improving DINOv2 from 26.10 to 24.94 FID on REPA-E VAE (Table 21). All tokenizer comparisons use the released VAEs frozen, without joint VAE training.

5.4 LAP Gains Persist Across Training, Scale, and Task

LAP improves generation early and throughout training. By 100K steps, LAP lowers FID-10K by 14.4 for MAE, 4.3 for AIMv2, and 4.5 for CLIP (Figure 5). These gains persist through 400K steps: MAE REPA remains close to the unaligned model, whereas MAE LAP reaches the same final FID as DINOv2 REPA (45.1). Purification therefore improves the full training trajectory.

LAP remains effective across training settings. With MAE, LAP continues to improve generation at larger batch size, with SiT-L/2, and at a different alignment depth, reaching FID 28.98, 12.06, and 38.8, respectively (Appendices F and F.2). The gains also persist under classifier-free guidance, where MAE, AIMv2, and CLIP improve in both seeds (Table 19). Thus, the benefit of purification is not specific to the base training configuration.

LAP also transfers beyond ImageNet. LAP with MAE achieves the best FID on two aerial-scene datasets: RESISC45 and AID (Table 4). It improves over no alignment by 6.6 FID on RESISC45 and 2.3 on AID, and outperforms DINOv2-based REPA by 4.9 and 3.6 FID, respectively. Notably, on AID, neither raw MAE nor DINOv2 improves over no alignment, whereas purified MAE does.

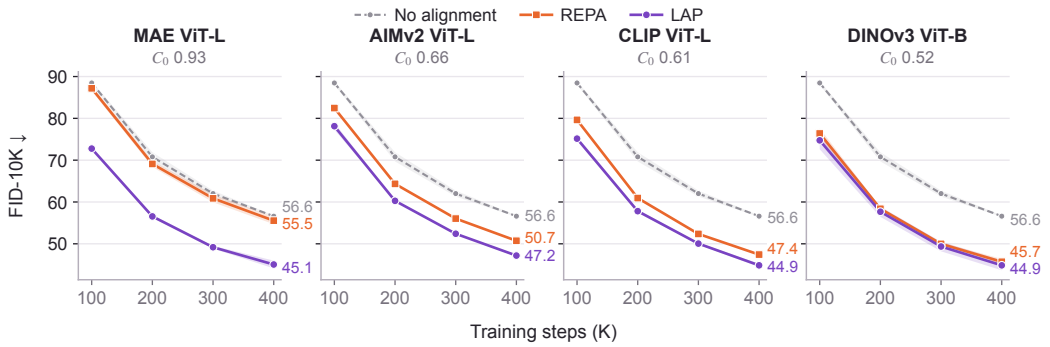


Figure 5: **Purification improves throughout training.** Two-seed FID-10K means and ranges for SiT-B/2, using LAP-N for MAE/CLIP and LAP-L for AIMv2/DINOv3. Full trajectories and evaluation details: Appendix H.4.

Table 4: **LAP on aerial-scene datasets.** Bold marks the best. Details: Appendix J.

Dataset	Method	FID ↓	IS ↑	Prec. ↑	Rec. ↑
RESISC45	No alignment	68.06 ±5.10	4.75 ±0.17	0.421 ±0.014	0.200 ±0.020
	REPA (DINOv2)	66.34 ±3.70	4.80 ±0.15	0.426 ±0.008	0.184 ±0.016
	REPA (MAE)	64.65 ±4.38	4.94 ±0.23	0.414 ±0.013	0.205 ±0.024
	LAP (MAE)	**61.47** ±2.33	**4.96** ±0.04	**0.433** ±0.010	**0.228** ±0.009
AID	No alignment	62.41 ±2.06	6.47 ±0.02	0.365 ±0.003	**0.222** ±0.003
	REPA (DINOv2)	63.68 ±1.35	6.36 ±0.02	0.376 ±0.002	0.200 ±0.003
	REPA (MAE)	66.13 ±1.47	6.40 ±0.06	0.368 ±0.007	0.191 ±0.003
	LAP (MAE)	**60.13** ±3.29	**6.52** ±0.09	**0.382** ±0.001	0.214 ±0.026

These results suggest that purification can recover useful alignment guidance even when the pretrained teacher is transferred outside its original image domain.

5.5 The Constant-Readout Score Predicts Purification Benefit

C_0 predicts which teachers benefit from purification. For raw MAE, $C_0 = 0.93$, close to the trained projector's cosine score of 0.957, indicating that much of its alignment score is attainable without using the hidden state. In a matched 13-encoder sweep, purification gain is strongly correlated with raw-target C_0 (Spearman $\rho = 0.94$), with a partial rank correlation of 0.83 after controlling for raw-target FID (Figure 6 and Appendix H.2). Before diffusion training, C_0 provides a cheap diagnostic for identifying teachers likely to benefit from purification.

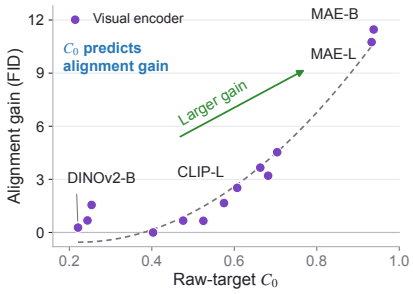


Figure 6: **C_0 predicts alignment gain.** Full results: Appendix H.2.

C_0 does not explain the full benefit of LAP. For MAE, all three LAP variants reduce C_0 to about 0.01–0.02, yet LAP-N still outperforms LAP-C and LAP-L. Its residual retains class information (68.7% vs. 66.1%) while removing 35% of held-out target variance (Appendix E). Thus, LAP-N removes additional image-dependent local content beyond the shared direction captured by C_0.

6 Conclusion

Prediction utility and alignment benefit need not coincide. Our fixed-encoder interventions motivate LAP, which removes locally predictable content before alignment and makes otherwise weak targets such as MAE effective. LAP complements existing REPA recipes and latent tokenizers without changing inference. Variational information bounds clarify what the alignment objective rewards, while the constant-readout score C_0 measures alignment attainable without consulting the hidden state and helps identify teachers likely to benefit from purification.

Limitations. Evaluation centers on SiT and ImageNet; larger-model checks use one seed and cross-domain runs are short. Gains depend on the teacher and recipe. LAP does not guarantee a semantic residual, and C_0 does not provide calibrated gain predictions.

AI USE STATEMENT

We used AI tools to polish the writing and assist with coding and experiments.

REPRODUCIBILITY STATEMENT

All experiments use the public REPA codebase as the base recipe with the modifications described in Sections 2 to 5. Hyperparameters, probe and ablation protocols, seed conventions, purifier training, control specifications, and the pre-specified sweep design are given in Appendices A to H, H.2, I, J and L, with compute disclosed in Appendix I. Code, checkpoints, and the toolkit will be released.

REFERENCES

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6(46): 1345–1382, 2005.

David Barber and Felix Agakov. The IM algorithm: A variational approach to information maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2003.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2036.

Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In *International Conference on Learning Representations (ICLR)*, 2019.

Hesen Chen, Junyan Wang, Zhiyu Tan, and Hao Li. SARA: Structural and adversarial representation alignment for training-efficient diffusion models. *arXiv preprint arXiv:2503.08253*, 2025.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024. doi: 10.1016/j.imavis.2024.105171.

Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrisi da Costa, Louis Béthune, Zhe Gan, Alexander T. Toshev, Marcin Eichner, Moin Nabi, Yinfei Yang, Joshua M. Susskind, and Alaaeldin El-Nouby. Multimodal autoregressive pre-training of large vision encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9641–9654, 2025.

Robert M. Gray. *Entropy and Information Theory*. Springer, 2 edition, 2011.

Yasaman Haghighi, Bastien van Delft, Mariam Hassan, and Alexandre Alahi. LayerSync: Self-aligning intermediate layers. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.12581.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher D. Manning. Conditional probing: Measuring usable information beyond a baseline. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, et al. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2505.02831.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. EQ-VAE: Equivariance regularized latent space for improved generative image modeling. In *International Conference on Machine Learning (ICML)*, 2025.

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

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2210.01351.

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

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 (ECCV)*, 2024.

Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. John Wiley & Sons, 2000.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Ruibin Min, Yexin Liu, Aimin Pan, Changsheng Lu, Jiafei Wu, Kelu Yao, et al. AHPA: Adaptive hierarchical prior alignment for diffusion transformers. *arXiv preprint arXiv:2605.03317*, 2026.

Mang Ning, Mingxiao Li, Le Zhang, Lanmiao Liu, Matthew B. Blaschko, Albert Ali Salah, and Itir Onal Ertugrul. Spectrum matching: a unified perspective for superior diffusability in latent diffusion. *arXiv preprint arXiv:2603.14645*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. arXiv:2304.07193.

Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. MIT course 6.441, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026. Accepted to ECCV 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, et al. What matters for representation alignment: Global information or spatial structure? In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2512.10794.

George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-REPA: Aligning diffusion U-Nets to ViTs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2503.18414.

Tri Ton, Ji Woo Hong, and Chang D. Yoo. TARO: Timestep-adaptive representation alignment with onset-aware conditioning for synchronized video-to-audio synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2504.05684.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Chenyu Wang, Cai Zhou, Sharut Gupta, Zongyu Lin, Stefanie Jegelka, Stephen Bates, et al. Learning diffusion models with flexible representation guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2507.08980.

Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, et al. SRA 2: Variational autoencoder self-representation alignment for efficient diffusion training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32978–32987, 2026.

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

Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. InfoDiffusion: Representation learning using information maximizing diffusion models. In *International Conference on Machine Learning (ICML)*, 2023.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, et al. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2507.01467.

Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.

Shaodong Xu, Zhendong Wang, Litong Gong, Zexian Li, Wengang Zhou, Tiezheng Ge, et al. Beyond point-wise matching: Structural representation alignment for accelerating diffusion transformers. *arXiv preprint arXiv:2605.16949*, 2026.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations (ICLR)*, 2020.

Donglin Yang, Yongxing Zhang, Xin Yu, Liang Hou, Xin Tao, Pengfei Wan, et al. Stable velocity: A variance perspective on flow matching. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.05435.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.06940.

Pengfei Zhang, Tianxin Xie, Minghao Yang, and Li Liu. AG-REPA: Causal layer selection for representation alignment in audio flow matching. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2603.01006.

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

APPENDIX CONTENTS

A Linear-Gaussian Analysis of Decodability and Use **16**
A.1 Model and Prediction Error . . . 16
A.2 Decodability Can Be High When Use Is Zero . . . 16
A.3 Prediction Through a Restricted Subspace . . . 16
A.4 Scope and Numerical Verification . . . 17

B Alignment Targets as Variational Information Bounds **17**
B.1 Setup and Notation . . . 17
B.2 From the vMF Decoder to the Alignment Loss . . . 18
B.3 General Lemmas . . . 19
B.4 Raw and Purified Targets . . . 21
B.5 An Exact Identity for Usable Information . . . 24
B.6 Scope . . . 26

C Experimental and Diagnostic Protocols **27**
C.1 Training and Evaluation Settings . . . 27
C.2 Linear Probes and Subspace Interventions . . . 27
C.3 Image-Disjoint Results and Split Sensitivity . . . 28
C.4 Representation Changes after Purification . . . 29
C.5 Diagnostics Across Noise Levels . . . 29
C.6 Reference Subspaces, Resampling, and Rank . . . 30
C.7 Component Readout and Easy-to-Match Content . . . 31
C.8 Constant-Readout Evaluation . . . 32
C.9 Position-Dependent Purification (LAP-P) . . . 33
C.10 Clean-Latent Prediction Across Noise Levels . . . 33
C.11 Seed Coverage and Uncertainty . . . 34

D Controls for Predictive Supervision and Target Content **34**
D.1 Auxiliary Prediction Controls . . . 34
D.2 The Affine Target Decomposition . . . 35
D.3 Separating Rank, Instability, and Predictable Content . . . 36
D.4 Where the Predictable-Only Target Becomes Harmful . . . 36

E Purifier Training, Validation, and Ablations **36**
E.1 Architecture and Validation Criteria 36
E.2 Visualizing the MAE Target Transformation 38
E.3 Receptive-Field Selection and the CLIP Check 40
E.4 Interaction with iREPA-Style Operations 40
E.5 A Check on Relational Structure 41

F Robustness to Batch Size, Capacity, and Guidance **41**
F.1 Larger Batch and Model 42
F.2 Alignment Depth 42
F.3 Classifier-Free Guidance 43

G An Encoder-Free Alignment Control **44**
G.1 Construction and Evaluation 44
G.2 What the Comparison Establishes 44

H Generation Evaluation and Additional Results **45**
H.1 Standard Evaluation and Mechanism Controls 45
H.2 Pre-Specified Encoder Sweep and an External Check of C_0 48
H.3 Checking the Development Metric 49
H.4 Complete Training Trajectories 50
H.5 Qualitative Samples 51

I Computational Resources **52**

J Two Aerial-Scene Datasets **53**

K Related Work **53**
K.1 Where and When Representations Guide Generation 53
K.2 Transforming the Target or Matching Objective 54
K.3 Mechanistic Analysis and Distillation 54

L Individual Runs for the Main Comparisons **55**
L.1 Auxiliary Prediction Sweep 55
L.2 Encoder Targets and Purification 56
L.3 Standard FID by Seed 57

For the theoretical distinction between decodability and use, start with Appendix A. Appendix B reads the alignment losses as variational information bounds. For experimental definitions and reliability checks, see Appendices C and D. Purifier validation and robustness results are in Appendices E and F. Appendix L collects the individual runs behind the main comparisons.

A Linear-Gaussian Analysis of Decodability and Use

This analysis asks a limited question: can a feature be easy to decode without helping predict a second target? The answer is yes, even with optimal predictors. The model below explains that distinction. It does not model diffusion training or prove that an auxiliary loss harms optimization.

A.1 Model and Prediction Error

Let $\xi \in \mathbb{R}^k$ be a latent factor vector. Define

$$\xi \sim \mathcal{N}(0, I_k), \qquad h = \xi + \eta, \qquad y = C\xi, \qquad v = B\xi, \tag{7}$$

where $\eta \sim \mathcal{N}(0, \sigma^2 I_k)$ is independent of ξ, $\sigma^2 \geq 0$, and B and C are fixed matrices. Here y represents an alignment target and v a prediction target. The hidden state h is fixed by this construction, rather than learned.

For an orthogonal projector $P = P^\top = P^2$, let $\mathcal{R}(P)$ be the minimum mean-squared error for predicting v from Ph. Joint Gaussianity makes the conditional mean linear. Writing $s = 1 + \sigma^2$, we obtain

$$\mathbb{E}[v \mid Ph] = \frac{1}{s} BPh, \tag{8}$$

$$\mathcal{R}(P) = \operatorname{tr}(BB^\top) - \frac{1}{s} \operatorname{tr}(BPB^\top). \tag{9}$$

To see this, $\operatorname{Cov}(v, Ph) = BP$ and $\operatorname{Cov}(Ph) = sP$. The pseudoinverse of sP is P/s. Substituting these quantities into the Gaussian conditional covariance gives $BB^\top - BPB^\top/s$. Its trace is the scalar prediction error.

A.2 Decodability Can Be High When Use Is Zero

Assume $C \neq 0$, and let P_y project onto the row space of C. We define presence using a variance-weighted population coefficient of determination:

$$R^2(h \to y) = 1 - \frac{\mathbb{E}\|y - \mathbb{E}[y \mid h]\|^2}{\operatorname{tr}(CC^\top)} = \frac{1}{1 + \sigma^2}. \tag{10}$$

This score depends on observation noise but not on B.

Proposition 1. Define information-level use as the extra optimal prediction error after removing a subspace:

$$U_{\text{opt}}(P) = \mathcal{R}(I - P) - \mathcal{R}(I) = \frac{\|BP\|_F^2}{1 + \sigma^2}. \tag{11}$$

For $B \neq 0$, the normalized use of the target subspace is therefore

$$\frac{U_{\text{opt}}(P_y)}{U_{\text{opt}}(I)} = \frac{\|BP_y\|_F^2}{\|B\|_F^2}. \tag{12}$$

If $\operatorname{rank}(C) < k$, a nonzero B can be chosen with rows in the orthogonal complement of $\operatorname{row}(C)$. Then $BP_y = 0$, so target-subspace use is zero while presence approaches one as $\sigma^2 \to 0$. If C has full column rank, $P_y = I$ and its normalized use is instead one.

Proof. Apply Equation (9) to $I - P$ and I, then subtract. Since P is symmetric and idempotent, $\operatorname{tr}(BPB^\top) = \|BP\|_F^2$. Equation (10) follows by replacing B with C and retaining the full hidden state. The zero-use construction requires a nontrivial orthogonal complement, which is exactly the stated rank condition. □

A.3 Prediction Through a Restricted Subspace

Proposition 2. For a unit vector u, predicting v only from $u^\top h$ reduces error relative to predicting the mean by

$$\operatorname{tr}(BB^\top) - \mathcal{R}(uu^\top) = \frac{\|Bu\|^2}{1 + \sigma^2}. \tag{13}$$

Thus, a direction helps prediction exactly when $Bu \neq 0$. A subspace P retains all of the full-state prediction benefit exactly when $B(I - P) = 0$.

Proof. The first statement is Equation (9) with $P = uu^\top$. For the second, the error gap from restricting prediction to P is $\|B(I - P)\|_F^2/(1 + \sigma^2)$. This vanishes precisely under the stated condition. □

The relevant notion is non-orthogonal overlap, not a literal intersection of row spaces. For example, two distinct one-dimensional subspaces can have zero intersection but still have a nonzero projection onto one another. The formula above captures that case.

A.4 Scope and Numerical Verification

The experimental use score holds the trained network fixed and measures its response to an intervention. In contrast, U_{opt} compares optimal predictors under an isotropic Gaussian model. In this special model, evaluating the full-state optimal predictor on $(I - P)h$ also attains the retained-subspace optimum. That equivalence need not hold for a nonlinear denoiser. Experimental use scores need not be nonnegative or bounded by the full-layer effect in every possible setting.

The model has no learning dynamics. Adding an auxiliary task cannot change h, and an unrestricted optimal velocity predictor already uses all available information. Consequently, these propositions do not establish gradient competition, semantic content, or the safety of residual alignment during training. Those interpretations must be assessed using the experiments.

The accompanying `theory/verify_linear_gaussian.py` checks the covariance identities, the rank condition, and orthogonal, shared, and oblique subspace examples. It also compares the analytical errors with held-out Monte Carlo predictions using 4×10^5 samples. These checks verify the implementation of the algebra, rather than adding empirical evidence about diffusion models.

B Alignment Targets as Variational Information Bounds

This appendix derives the information bounds and constant-readout identity of Sections 4.2 and 4.3. We first construct a directional decoder whose negative log-likelihood is the cosine alignment loss, up to a positive scale and a constant (Appendix B.2). We then derive the raw and purified bounds, including the smoothing correction needed for the actual training targets (Appendices B.3 and B.4). In both cases, minimizing the loss maximizes the lower bound. Finally, we solve for the best constant readout and derive the exact predictive-score identity (Appendix B.5). The distinction between optimizing a bound and increasing information, and the connection to InfoDiffusion, are discussed in Appendix B.6.

B.1 Setup and Notation

Mutual information. The main text writes I for the image, so we write $\mathcal{I}(\cdot\,;\cdot)$ for mutual information. All random variables take values in Euclidean spaces or in products of spheres, so regular conditional distributions exist. For random variables U, W, and S, with joint law P_{UW} and marginals P_U, P_W,

$$\mathcal{I}(U;W) = \mathrm{KL}(P_{UW} \,\|\, P_U \otimes P_W) \in [0,\infty], \tag{14}$$

$$\mathcal{I}(U;W \mid S) = \int \mathrm{KL}\big(P_{UW|S=s} \,\big\|\, P_{U|S=s} \otimes P_{W|S=s}\big)\, P_S(ds). \tag{15}$$

We use two standard facts, which hold in this generality (Cover & Thomas, 2006; Gray, 2011). The first is the chain rule,

$$\mathcal{I}\big(U;(W,S)\big) = \mathcal{I}(U;S) + \mathcal{I}(U;W \mid S). \tag{16}$$

The second is the data-processing inequality: for probability measures P, R and a measurable map T,

$$\mathrm{KL}\big(P \circ T^{-1} \,\big\|\, R \circ T^{-1}\big) \le \mathrm{KL}(P \,\|\, R), \tag{17}$$

with equality when T is a bijection onto its image with measurable inverse.

Entropy relative to a reference measure. Let ν be a σ-finite measure on the space of W. If P_W has density p with respect to ν, we write $H_\nu(W) = -\mathbb{E}\log p(W)$. If, for P_S-almost every s, the conditional law $P_{W|S=s}$ has density $p(\cdot \mid s)$ with respect to ν, we write $H_\nu(W \mid S) = -\mathbb{E}\log p(W \mid S)$. Both are used only when the expectation is finite.

Alignment variables. Fix a noise time $t \in (0, 1]$. The hidden state $h = h_t^l$ is a deterministic function of x_t and the conditioning vector e, with parameters θ. Here e is the sum of the time embedding and the class embedding after label dropout. The aligned representation is $a = g_\psi(h) \in \mathbb{R}^{N\times D}$ with tokens $a^{(i)}$, which we assume nonzero. The encoder features are $z = f(I) \in \mathbb{R}^{N\times D}$. Since f is frozen, the law of z does not depend on (θ, ψ). For any $y \in \mathbb{R}^{N\times D}$ with nonzero tokens, let $\hat{y}^{(i)} = y^{(i)}/\|y^{(i)}\|$ and $\hat{y} = (\hat{y}^{(1)}, \dots, \hat{y}^{(N)}) \in (\mathbb{S}^{D-1})^N$. Let ω be the surface measure on the unit sphere $\mathbb{S}^{D-1}$ and $\nu = \omega^{\otimes N}$ the product measure on $(\mathbb{S}^{D-1})^N$. We assume $D \geq 2$.

Smoothed targets. Let $\xi \in \mathbb{R}^{N\times D}$ have independent standard normal entries and be independent of all other variables. For $\sigma > 0$ and any array y, we write $y_\sigma = y + \sigma\xi$. The tokens of y_σ are nonzero almost surely, so $\hat{y}_\sigma$ is well defined. Smoothing is a device of the analysis. Training uses the unsmoothed targets.

B.2 From the vMF Decoder to the Alignment Loss

A likelihood for directions. Cosine alignment compares directions and ignores token norms. A natural probabilistic decoder therefore assigns a density on the unit sphere. For a mean direction $\mu \in \mathbb{S}^{D-1}$ and concentration $\kappa > 0$, the von Mises–Fisher (vMF) density with respect to ω is (Mardia & Jupp, 2000)

$$\mathrm{vMF}(w; \mu, \kappa) = c_D(\kappa)\exp\big(\kappa\,\mu^\top w\big), \qquad c_D(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2}\,\mathcal{B}_{D/2-1}(\kappa)}, \tag{18}$$

where $\mathcal{B}_\alpha$ is the modified Bessel function of the first kind. The normalizer satisfies $c_D(\kappa)^{-1} = \int \exp(\kappa\mu^\top w)\,\omega(dw)$ and is independent of μ by rotation invariance. We keep κ fixed throughout the analysis. The denoiser and projector choose only the mean direction of each token. This gives the product decoder

$$q_\psi(w \mid h) = \prod_{i=1}^{N} \mathrm{vMF}\left(w^{(i)};\, \frac{a^{(i)}}{\|a^{(i)}\|},\, \kappa\right), \qquad a = g_\psi(h), \tag{19}$$

which is a probability density with respect to ν for every h. Factorizing the decoder is a modeling choice, not an assumption that the true target tokens are independent.

Recovering the cosine objective. Taking the logarithm turns the product over tokens into a sum. Substituting the normalized readout and target gives

$$\begin{aligned}\log q_\psi(\hat{y} \mid h) &= \sum_{i=1}^{N}\left[\log c_D(\kappa) + \kappa\frac{(a^{(i)})^\top y^{(i)}}{\|a^{(i)}\|\,\|y^{(i)}\|}\right] \\ &= N\log c_D(\kappa) + \kappa\sum_{i=1}^{N}\cos\Big(a^{(i)}, y^{(i)}\Big).\end{aligned} \tag{20}$$

At a fixed noise time t, write $\mathcal{L}_{\mathrm{align}}^{(t)}(y) = -\mathbb{E}[\frac{1}{N}\sum_i \cos(a^{(i)}, y^{(i)})]$ for the alignment loss of Equation (1) with target y, evaluated at noise time t. Taking the negative expectation in Equation (20) yields

$$\mathbb{E}\big[-\log q_\psi(\hat{y} \mid h)\big] = -N\log c_D(\kappa) + \kappa N\,\mathcal{L}_{\mathrm{align}}^{(t)}(y). \tag{21}$$

Thus minimizing the cosine loss is exactly equivalent to maximizing the decoder's expected log-likelihood for fixed κ. This identity requires no density for the target distribution. Also, $|\log q_\psi(\hat{y} \mid h)| \leq N|\log c_D(\kappa)| + \kappa N$, so its expectations are finite.

From likelihood to an information bound. Relating this score to Shannon information requires an additional step. For a smoothed target $W = \hat{y}_\sigma$ with the finite marginal and conditional entropies established below (Lemma 4 with S constant and with $S = h$ supplies the densities, and

Lemma 1 gives the gap), adding and subtracting $\mathbb{E}\log q_\psi(W \mid h)$ gives the Barber–Agakov decomposition (Barber & Agakov, 2003),

$$\begin{aligned}\mathcal{I}(h;W) &= H_\nu(W) + \mathbb{E}\log q_\psi(W \mid h) + \mathbb{E}_h\,\mathrm{KL}\big(P_{W|h} \,\big\|\, q_\psi(\cdot \mid h)\nu\big) \\ &\geq H_\nu(W) + N\log c_D(\kappa) - \kappa N\,\mathcal{L}^{(t)}_{\mathrm{align}}(y_\sigma). \end{aligned} \tag{22}$$

The inequality drops a nonnegative decoder-approximation gap. The target entropy is fixed because the encoder and target construction do not change during diffusion training. Consequently, decreasing the alignment loss raises the lower bound. Training uses y, not y_σ, so Appendix B.4 bounds the change in cosine caused by smoothing and absorbs it into a model-independent constant. Conditioning this argument on X_p gives the purified-target bound. The lemmas below justify these steps even when the original targets have no density.

B.3 General Lemmas

The first lemma is the bound of Barber & Agakov (2003). We prove it without assuming that W has a conditional density given U, because h is a deterministic function of the noisy input and such a density need not exist.

Lemma 1. Let P_W have density p with respect to ν, with $\mathbb{E}|\log p(W)| < \infty$. Let $q(w \mid u) \geq 0$ be jointly measurable with $\int q(w \mid u)\,\nu(dw) = 1$ for every u, and $\mathbb{E}|\log q(W \mid U)| < \infty$. Then

$$\mathcal{I}(U;W) \;\geq\; H_\nu(W) + \mathbb{E}\log q(W \mid U). \tag{23}$$

If, in addition, $P_{W|U=u}$ has a density $p(\cdot \mid u)$ with respect to ν for P_U-almost every u and $\mathcal{I}(U;W) < \infty$, the gap in Equation (23) equals $\mathbb{E}_U\,\mathrm{KL}\big(p(\cdot \mid U) \,\|\, q(\cdot \mid U)\big)$.

Proof. Write $P = P_{UW}$ and $R = P_U \otimes P_W$. If P is not absolutely continuous with respect to R, then $\mathcal{I}(U;W) = \infty$ by Equation (14) and the claim holds, since the right-hand side of Equation (23) is finite. Otherwise let $\rho = dP/dR$, so that $\mathcal{I}(U;W) = \mathbb{E}_P \log\rho$. If this is infinite, the claim again holds. Assume it is finite.

Define $g(u,w) = q(w \mid u)/p(w)$ on $\{p(w) > 0\}$. Since $p(W) > 0$ almost surely, g is defined P-almost everywhere and R-almost everywhere. The integrability assumptions give $\mathbb{E}_P|\log g| < \infty$ and

$$\mathbb{E}_P \log g = \mathbb{E}\log q(W \mid U) - \mathbb{E}\log p(W) = \mathbb{E}\log q(W \mid U) + H_\nu(W). \tag{24}$$

Since $\rho > 0$ holds P-almost surely and both $\mathbb{E}_P\log\rho$ and $\mathbb{E}_P\log g$ are finite,

$$\mathcal{I}(U;W) - \mathbb{E}_P\log g = -\mathbb{E}_P\log\frac{g}{\rho} \;\geq\; -\log\mathbb{E}_P\left[\frac{g}{\rho}\right], \tag{25}$$

by Jensen's inequality for the concave logarithm. Changing measure from P to R,

$$\begin{aligned}\mathbb{E}_P\left[\frac{g}{\rho}\right] &= \int_{\{\rho>0\}} \frac{g}{\rho}\,\rho\,dR = \int_{\{\rho>0\}} g\,dR \;\leq\; \int g\,dR \\ &= \int P_U(du)\int_{\{p>0\}} \frac{q(w \mid u)}{p(w)}\,p(w)\,\nu(dw) \;\leq\; \int P_U(du)\int q(w \mid u)\,\nu(dw) = 1. \end{aligned} \tag{26}$$

The first inequality uses $g \geq 0$, the next equality is Tonelli's theorem, and the last inequality enlarges the domain of integration. Therefore $\mathcal{I}(U;W) - \mathbb{E}_P\log g \geq -\log 1 = 0$, and Equation (23) follows from Equation (24).

For the second statement, the additional assumption gives $\rho(u,w) = p(w \mid u)/p(w)$, so $\mathcal{I}(U;W) = \mathbb{E}\log p(W \mid U) + H_\nu(W)$. Subtracting the right-hand side of Equation (23) leaves $\mathbb{E}\big[\log p(W \mid U) - \log q(W \mid U)\big] = \mathbb{E}_U\,\mathrm{KL}\big(p(\cdot \mid U) \,\|\, q(\cdot \mid U)\big)$. □

Lemma 2. Suppose that, for P_S-almost every s, the conditional law $P_{W|S=s}$ has density $p(\cdot \mid s)$ with respect to ν, with $\mathbb{E}|\log p(W \mid S)| < \infty$. Let $q(w \mid u,s) \geq 0$ be jointly measurable with $\int q(w \mid u,s)\,\nu(dw) = 1$ for every (u,s), and $\mathbb{E}|\log q(W \mid U,S)| < \infty$. Then

$$\mathcal{I}(U;W \mid S) \;\geq\; H_\nu(W \mid S) + \mathbb{E}\log q(W \mid U,S). \tag{27}$$

Proof. The integrability assumptions imply that, for P_S-almost every s, both $\mathbb{E}\big[|\log p(W \mid s)| \,\big|\, S = s\big]$ and $\mathbb{E}\big[|\log q(W \mid U, s)| \,\big|\, S = s\big]$ are finite. For each such s, apply Lemma 1 to the pair (U, W) under the conditional law $P_{UW|S=s}$, with density $p(\cdot \mid s)$ and decoder $q(\cdot \mid \cdot, s)$. This gives

$$\mathrm{KL}\big(P_{UW|S=s} \,\big\|\, P_{U|S=s} \otimes P_{W|S=s}\big) \geq -\mathbb{E}\big[\log p(W \mid s) \mid S = s\big] + \mathbb{E}\big[\log q(W \mid U, s) \mid S = s\big]. \tag{28}$$

Integrating both sides against P_S and using Equation (15) gives Equation (27). □

Lemma 3. Let U, W, S be random variables.

(a) Let $(s, w) \mapsto \varphi_s(w)$ be jointly measurable. Then $\mathcal{I}\big(U; \varphi_S(W) \mid S\big) \leq \mathcal{I}(U; W \mid S)$.

(b) Let m be measurable. Then $\mathcal{I}\big((U, m(S)); W \mid S\big) = \mathcal{I}(U; W \mid S)$.

(c) Let ξ be independent of (U, W, S). Then $\mathcal{I}\big(U; (W, \xi) \mid S\big) = \mathcal{I}(U; W \mid S)$.

Taking S constant gives the unconditional versions.

Proof. Fix s and consider the integrand of Equation (15). For (a), the map $(u, w) \mapsto (u, \varphi_s(w))$ sends $P_{UW|S=s}$ to the conditional law of $(U, \varphi_S(W))$ given $S = s$, and sends the product $P_{U|S=s} \otimes P_{W|S=s}$ to the product of the corresponding marginals. Equation (17) gives the inequality for the integrand. For (b), conditionally on $S = s$ the variable $m(S)$ equals the constant $m(s)$. The map $(u, w) \mapsto \big((u, m(s)), w\big)$ is a bijection onto its image with measurable inverse, so the integrand is unchanged by Equation (17). For (c), independence gives $P_{(U,W,\xi)|S=s} = P_{UW|S=s} \otimes P_\xi$ and $P_{U|S=s} \otimes P_{(W,\xi)|S=s} = P_{U|S=s} \otimes P_{W|S=s} \otimes P_\xi$. Multiplying both arguments of a divergence by the same factor P_ξ changes neither absolute continuity nor the density ratio, so the integrand is unchanged. Integrating over P_S completes each case. □

Lemma 4 (smoothed targets have densities). Let $y \in \mathbb{R}^{N \times D}$ be a random array with $\mathbb{E}\log(1 + \|y^{(i)}\|) < \infty$ for every token i, and let S be any random variable. Let ξ be as in Appendix B.1, independent of (y, S), and let $\sigma > 0$. Then, for P_S-almost every s, the conditional law of $\hat{y}_\sigma$ given $S = s$ has a density with respect to ν, and

$$-N \log K_D - (D-1) \sum_{i=1}^{N} \mathbb{E} \log\left(1 + \frac{\|y^{(i)}\|}{\sigma}\right) \;\leq\; H_\nu(\hat{y}_\sigma \mid S) \;\leq\; N \log \omega(\mathbb{S}^{D-1}), \tag{29}$$

where $K_D = (2\pi)^{-(D-1)/2}\, 2^{D-2} \max\{1, \mathbb{E}|\gamma|^{D-1}\}$ with $\gamma \sim \mathcal{N}(0, 1)$. In particular, $H_\nu(\hat{y}_\sigma \mid S)$ is finite.

Proof. Step 1: one token with a fixed mean. Let $m \in \mathbb{R}^D$ and $Y = m + \sigma\zeta$ with $\zeta \sim \mathcal{N}(0, \mathbf{I}_D)$. In polar coordinates $Y = \varrho u$ with $\varrho > 0$ and $u \in \mathbb{S}^{D-1}$, Lebesgue measure factorizes as $\varrho^{D-1} d\varrho\, \omega(du)$. Here ω is unnormalized surface measure, so no additional sphere-area factor is needed. Integrating out the radius gives the direction density

$$f_m(u) = \int_0^\infty (2\pi\sigma^2)^{-D/2} \exp\left(-\frac{\|\varrho u - m\|^2}{2\sigma^2}\right) \varrho^{D-1}\, d\varrho \tag{30}$$

with respect to ω. Let $b = u^\top m$. The orthogonal decomposition $m = bu + (m - bu)$ gives

$$\|\varrho u - m\|^2 = (\varrho - b)^2 + \|m - bu\|^2 \geq (\varrho - b)^2. \tag{31}$$

Dropping the nonnegative second term increases the exponential. Extending the resulting nonnegative integral to the real line, with $|\varrho|^{D-1}$ on the added negative half-line, gives

$$\begin{aligned} f_m(u) &\leq (2\pi\sigma^2)^{-D/2} \int_{-\infty}^{\infty} |\varrho|^{D-1} e^{-(\varrho-b)^2/(2\sigma^2)}\, d\varrho \\ &= (2\pi)^{-(D-1)/2} \sigma^{-(D-1)}\, \mathbb{E}|b + \sigma\gamma|^{D-1}. \end{aligned} \tag{32}$$

For $p = D - 1 \geq 1$ and $x, x' \geq 0$, convexity gives $(x + x')^p \leq 2^{p-1}(x^p + x'^p)$. Hence $\mathbb{E}|b + \sigma\gamma|^{D-1} \leq 2^{D-2}\big(\|m\|^{D-1} + \sigma^{D-1}\mathbb{E}|\gamma|^{D-1}\big)$, and

$$f_m(u) \leq (2\pi)^{-(D-1)/2} 2^{D-2} \left(\left(\frac{\|m\|}{\sigma}\right)^{D-1} + \mathbb{E}|\gamma|^{D-1}\right) \leq K_D \left(1 + \frac{\|m\|}{\sigma}\right)^{D-1}, \tag{33}$$

where the last step uses $x^p + 1 \le (1+x)^p$ for $x \ge 0$.

Step 2: the array. Since ξ is independent of (y, S), conditionally on $(y, S) = (m, s)$ the tokens of y_σ are independent with laws $\mathcal{N}(m^{(i)}, \sigma^2 \mathbf{I}_D)$. By Step 1, $\hat{y}_\sigma$ then has the conditional density $p(w \mid m, s) = \prod_i f_{m^{(i)}}(w^{(i)})$ with respect to ν. By Tonelli's theorem, the mixture $p(w \mid s) = \mathbb{E}\big[p(w \mid y, s) \mid S = s\big]$ is a density of the conditional law of $\hat{y}_\sigma$ given $S = s$.

Step 3: upper bound and integrability of the negative part. Write $M = \nu((\mathbb{S}^{D-1})^N) = \omega(\mathbb{S}^{D-1})^N$. For any density q with respect to ν,

$$\int q(\log q)^- \, d\nu \le M/e, \qquad (\log q)^- = \max\{-\log q, 0\}, \tag{34}$$

because $-x \log x \le 1/e$ on $[0, 1]$, with $0 \log 0 = 0$. Thus $\int q \log q \, d\nu$ is well defined, though it may be $+\infty$. Nonnegativity of $\mathrm{KL}(q\nu \,\|\, \nu/M)$ gives $-\int q \log q \, d\nu \le \log M$, which yields the upper entropy bound after averaging over S.

Step 4: lower bound. Put $A(y) = N \log K_D + (D-1) \sum_i \log(1 + \|y^{(i)}\|/\sigma)$. By Equation (33), $\log p(w \mid y, S) \le A(y)$, and $\mathbb{E}|A(y)| < \infty$ by assumption. Together with Step 3 this gives absolute integrability of $\log p(\hat{y}_\sigma \mid y, S)$. Convexity of $x \log x$ and the mixture formula in Step 2 give

$$\int p(w \mid s) \log p(w \mid s) \, \nu(dw) \le \mathbb{E}\left[\int p(w \mid y, s) \log p(w \mid y, s) \, \nu(dw) \,\middle|\, S = s \right]. \tag{35}$$

Jensen's inequality and the interchange of integrals are valid by applying them to the nonnegative convex function $x \log x + 1/e$. Averaging over S bounds the left-hand side by $\mathbb{E}A(y) < \infty$. Step 3 bounds its negative part, so $\mathbb{E}|\log p(\hat{y}_\sigma \mid S)| < \infty$. Negating the inequality gives the lower bound in Equation (29), without assuming the entropy's finiteness in advance. □

Lemma 5 (cosine perturbation). For nonzero $x, y, a \in \mathbb{R}^D$, $|\cos(a, x) - \cos(a, y)| \le 2\|x - y\|/\|y\|$.

Proof. Write $\hat{x} - \hat{y} = (x - y)/\|y\| + x\,(1/\|x\| - 1/\|y\|)$. The norm of the second term is $\big|\|y\| - \|x\|\big|/\|y\| \le \|x - y\|/\|y\|$. Hence $\|\hat{x} - \hat{y}\| \le 2\|x - y\|/\|y\|$, and $|\cos(a, x) - \cos(a, y)| = |\hat{a}^\top(\hat{x} - \hat{y})| \le \|\hat{x} - \hat{y}\|$ by the Cauchy–Schwarz inequality. □

B.4 Raw and Purified Targets

Why the targets are smoothed. Lemmas 1 and 2 require the target to have a density with respect to the reference measure. In our setting the normalized target $\hat{z}$ has none, for two independent reasons. First, training images are 8-bit and finitely many, so the law of $\hat{z}$ is atomic. Second, even if images were continuous, the encoder input has $3 \cdot 224^2 = 150{,}528$ coordinates for DINOv2 and $3 \cdot 256^2 = 196{,}608$ for MAE, whereas $(\mathbb{S}^{D-1})^N$ has dimension $N(D-1) = 196{,}352$ for $D = 768$ and $261{,}888$ for $D = 1024$. The encoder and the normalization are locally Lipschitz, and a locally Lipschitz image of a lower-dimensional set has ν-measure zero. The law of $\hat{z}$ is therefore singular with respect to ν, and the same holds conditionally on X_p. We therefore state the bounds for the smoothed targets of Appendix B.1, for which Lemma 4 proves the required densities and finite entropies, and we use Lemma 5 to return to the training loss.

For an array y with nonzero tokens, let $\bar{e}(y) = \frac{1}{N} \sum_i \mathbb{E}\big[1/\|y^{(i)}\|\big]$. This is finite for a target with finite support and no zero tokens. For continuously varying targets, including residuals formed from stochastic latents, nonzero tokens alone do not imply a finite inverse-norm moment. Across the 2,048-image samples of Appendix C.8, the smallest residual token norms under LAP-N are approximately 24.3 for MAE, 20.6 for DINOv2, and 8.9 for CLIP. The sample moments are therefore finite, but this does not establish finiteness of the population moments. A sufficient condition is that the law of $r^{(i)}$ has a bounded density near the origin on a subspace of dimension at least two, since $\int_0^1 \rho^{k-2} \, d\rho < \infty$ for $k \ge 2$; this is the generic case for residuals formed from a sampled latent.

Proposition 3 (raw target). Assume $\mathbb{E}\log(1 + \|z^{(i)}\|) < \infty$ for every i. Then, for every $\sigma > 0$, every $\kappa > 0$, and every (θ, ψ),

$$\mathcal{I}(h; z) \;\ge\; \mathcal{I}(h; \hat{z}_\sigma) \;\ge\; H_\nu(\hat{z}_\sigma) + N \log c_D(\kappa) + \kappa N \, \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \cos\Big(a^{(i)}, z_\sigma^{(i)}\Big) \right], \tag{36}$$

where $H_\nu(\hat{z}_\sigma)$ is finite and does not depend on (θ, ψ). If moreover $\bar{e}(z) < \infty$, then

$$\mathcal{I}(h; z) \;\geq\; H_\nu(\hat{z}_\sigma) + N \log c_D(\kappa) - \kappa N \Big(\mathcal{L}^{(t)}_{\text{align}}(z) + 2\sigma\sqrt{D}\,\bar{e}(z) \Big), \tag{37}$$

with $\mathcal{L}^{(t)}_{\text{align}}$ the alignment loss at noise time t defined below Equation (20).

Proof. Step 1. Since ξ is independent of (h, z), Lemma 3(c) gives $\mathcal{I}(h; z) = \mathcal{I}\big(h; (z, \xi)\big)$. Since $\hat{z}_\sigma$ is a measurable function of (z, ξ), Lemma 3(a) gives $\mathcal{I}\big(h; (z, \xi)\big) \geq \mathcal{I}(h; \hat{z}_\sigma)$.

Step 2. By Lemma 4 with $y = z$ and S constant, $\hat{z}_\sigma$ has a density with respect to ν and $H_\nu(\hat{z}_\sigma)$ is finite. It depends only on the data, the frozen encoder, and σ. Apply Lemma 1 with $U = h$, $W = \hat{z}_\sigma$, and the decoder q_ψ of Equation (19). Its integrability assumption holds by the bound below Equation (20). Substituting Equation (20) with $y = z_\sigma$ into Equation (23) gives Equation (36).

Step 3. Lemma 5 with $x = z^{(i)}_\sigma$ and $y = z^{(i)}$ gives $\cos(a^{(i)}, z^{(i)}_\sigma) \geq \cos(a^{(i)}, z^{(i)}) - 2\sigma\|\xi^{(i)}\|/\|z^{(i)}\|$. Since ξ is independent of z and $\mathbb{E}\|\xi^{(i)}\| \leq (\mathbb{E}\|\xi^{(i)}\|^2)^{1/2} = \sqrt{D}$, taking expectations and averaging over tokens gives $\mathbb{E}\big[\frac{1}{N}\sum_i \cos(a^{(i)}, z^{(i)}_\sigma)\big] \;\geq\; -\mathcal{L}^{(t)}_{\text{align}}(z) - 2\sigma\sqrt{D}\,\bar{e}(z)$. Substituting into Equation (36) gives Equation (37). □

Averaging over noise times. The time t is drawn independently of the image and of ξ, so the constants in Equation (37) are the same for every t. Taking the expectation over t and using Equation (15) with $S = t$ gives $\mathcal{I}(h_t; z \mid t) \geq H_\nu(\hat{z}_\sigma) + N \log c_D(\kappa) - \kappa N \big(\mathcal{L}_{\text{align}} + 2\sigma\sqrt{D}\,\bar{e}(z) \big)$. The same remark applies to every bound below.

For the purified target, let $r = z - d_\phi(X_p)$ be the residual of Equation (3), so that $r_\sigma = r + \sigma\xi = z_\sigma - d_\phi(X_p)$. The predictor d_ϕ is fixed and measurable.

Proposition 4 (purified target). Assume $\mathbb{E}\log(1 + \|r^{(i)}\|) < \infty$ for every i. Then, for every $\sigma > 0$, every $\kappa > 0$, and every (θ, ψ),

$$\mathcal{I}(h; z \mid X_p) \;\geq\; \mathcal{I}(h; \hat{r}_\sigma \mid X_p) \;\geq\; H_\nu(\hat{r}_\sigma \mid X_p) + N \log c_D(\kappa) + \kappa N\, \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \cos\Big(a^{(i)}, r^{(i)}_\sigma \Big) \right], \tag{38}$$

where $H_\nu(\hat{r}_\sigma \mid X_p)$ is finite and does not depend on (θ, ψ). If moreover the tokens of r are nonzero almost surely and $\bar{e}(r) < \infty$, then

$$\mathcal{I}(h; z \mid X_p) \;\geq\; H_\nu(\hat{r}_\sigma \mid X_p) + N \log c_D(\kappa) - \kappa N \Big(\mathcal{L}^{(t)}_{\text{align}}\big(y^{\text{pure}}\big) + 2\sigma\sqrt{D}\,\bar{e}(r) \Big). \tag{39}$$

Proof. Step 1. Since ξ is independent of (h, z, X_p), Lemma 3(c) gives $\mathcal{I}(h; z \mid X_p) = \mathcal{I}\big(h; (z, \xi) \mid X_p\big)$.

Step 2. Define $\varphi_{x_p}(w, e)$ as the token-wise normalization of $w + \sigma e - d_\phi(x_p)$, which is jointly measurable in (x_p, w, e) and defined almost surely. Then $\hat{r}_\sigma = \varphi_{X_p}(z, \xi)$, and Lemma 3(a) gives $\mathcal{I}\big(h; (z, \xi) \mid X_p\big) \geq \mathcal{I}(h; \hat{r}_\sigma \mid X_p)$.

Step 3. The residual r is a function of (z, X_p), so ξ is independent of (r, X_p). By Lemma 4 with $y = r$ and $S = X_p$, the conditional law of $\hat{r}_\sigma$ given X_p has a density with respect to ν, and $H_\nu(\hat{r}_\sigma \mid X_p)$ is finite. It depends only on the data, the frozen encoder, the fixed predictor, and σ. Apply Lemma 2 with $U = h$, $W = \hat{r}_\sigma$, $S = X_p$, and the decoder $q(w \mid h, x_p) = q_\psi(w \mid h)$, which is allowed to ignore x_p. Substituting Equation (20) with $y = r_\sigma$ into Equation (27) gives Equation (38).

Step 4. The argument of Step 3 in the proof of Proposition 3, with r in place of z, gives Equation (39), because $\mathcal{L}^{(t)}_{\text{align}}(r) = \mathcal{L}^{(t)}_{\text{align}}(y^{\text{pure}})$. □

The constants of Equation (4). Equations (37) and (39) are the two inequalities of Equation (4), with

$$\begin{aligned} c_\sigma &= H_\nu(\hat{z}_\sigma) + N \log c_D(\kappa) - 2\kappa N \sigma \sqrt{D}\,\bar{e}(z), \\ c'_\sigma &= H_\nu(\hat{r}_\sigma \mid X_p) + N \log c_D(\kappa) - 2\kappa N \sigma \sqrt{D}\,\bar{e}(r). \end{aligned} \tag{40}$$

Both depend on the data, the frozen encoder, the fixed predictor, σ, and κ, and neither depends on (θ, ψ). The moment conditions required for these bounds are $\mathbb{E}\log(1+\|y^{(i)}\|) < \infty$ and $\bar{e}(y) < \infty$ for the respective target $y \in \{z, r\}$, together with almost surely nonzero residual tokens for LAP. The main text states the bounds at a fixed noise time; the remark on averaging over noise times above extends them to the time-averaged loss.

Optimization direction. Write either right-hand side of Equation (4) as $B_\sigma(y) = c_\sigma(y) - \kappa N \mathcal{L}^{(t)}_{\text{align}}(y)$, where $c_\sigma(y)$ is the appropriate constant above, $c_\sigma(z) = c_\sigma$ and $c_\sigma(r) = c'_\sigma$. Since $\kappa N > 0$, minimizing the alignment loss is equivalent to maximizing $B_\sigma(y)$. The sign is also explicit in the combined objective:

$$\mathcal{L}^{(t)}_{\text{flow}} + \lambda_a \mathcal{L}^{(t)}_{\text{align}}(y) = \mathcal{L}^{(t)}_{\text{flow}} - \frac{\lambda_a}{\kappa N} B_\sigma(y) + \frac{\lambda_a}{\kappa N} c_\sigma(y). \tag{41}$$

For $\lambda_a > 0$, minimizing this objective rewards a larger information lower bound alongside a smaller denoising error. No entropy estimate or additional information loss is needed in training. This is an equivalence between objectives, not a guarantee that either the bound or mutual information increases at every optimization step. The constants differ across targets, so alignment losses alone cannot rank their mutual information.

The role of σ. Both propositions hold for every $\sigma > 0$ under their stated moment assumptions. If $\bar{e}(y) < \infty$, the correction $2\sigma\sqrt{D}\,\bar{e}(y)$ vanishes as $\sigma \downarrow 0$. The entropy term, however, tends to $-\infty$ when the unsmoothed conditional target law is singular. The following argument covers both propositions.

Let S be constant and $y = z$ for the raw target, or let $S = X_p$ and $y = r$ for LAP. Assume the tokens of y are nonzero almost surely and $P_{\hat{y}|S=s} \perp \nu$ for P_S-almost every s. Set $Q = \nu/M$, where $M = \omega(\mathbb{S}^{D-1})^N$, and $P_\sigma = P_{(S,\hat{y}_\sigma)}$. Using the same (y, S, ξ) for every σ gives $\hat{y}_\sigma \to \hat{y}$ almost surely, by continuity of normalization at nonzero tokens. Thus $P_\sigma \Rightarrow P_0 = P_{(S,\hat{y})}$. The KL chain rule gives

$$\mathrm{KL}(P_\sigma \,\|\, P_S \otimes Q) = \log M - H_\nu(\hat{y}_\sigma \mid S). \tag{42}$$

The singular conditional limit has $\mathrm{KL}(P_0 \,\|\, P_S \otimes Q) = \infty$. Weak lower semicontinuity of KL (Polyanskiy & Wu, 2016, Theorem 3.6) therefore implies, for every sequence $\sigma_n \downarrow 0$,

$$\infty = \mathrm{KL}(P_0 \,\|\, P_S \otimes Q) \le \liminf_{n\to\infty} \mathrm{KL}(P_{\sigma_n} \,\|\, P_S \otimes Q). \tag{43}$$

Consequently $H_\nu(\hat{y}_\sigma \mid S) \to -\infty$. No monotonicity in σ is needed. Raw-target singularity was established above. For LAP with discrete encoder inputs, fixing $X_p = x_p$ translates the atomic law of z by the fixed vector $d_\phi(x_p)$, so its normalized residual law is also atomic. This concerns the conditional residual law, not its marginal. With fixed κ, the bounded cosine term cannot offset the diverging entropy. These lower bounds therefore become vacuous as $\sigma \downarrow 0$, even though the loss correction vanishes. Appendix B.5 avoids this smoothing limitation.

The squared-error form. The same construction is more transparent with a Gaussian decoder whose mean adds a correction from h to the local prediction. Assume $\mathbb{E}\|r\|^2 < \infty$ and $\mathbb{E}\|g_\psi(h)\|^2 < \infty$. Conditionally on X_p, the smoothed target z_σ has a Lebesgue density, since it is a Gaussian convolution. Its conditional entropy $H_{\text{Leb}}(z_\sigma \mid X_p)$ is at least that of $\sigma\xi$, which is $\frac{ND}{2}\log(2\pi e\sigma^2)$, and it is at most $\frac{ND}{2}\log\big(2\pi e\,\mathbb{E}\|r_\sigma\|^2/(ND)\big)$, by translation invariance and the Gaussian maximum-entropy bound. Lemma 3(c) and Lemma 3(a) give $\mathcal{I}(h; z \mid X_p) \ge \mathcal{I}(h; z_\sigma \mid X_p)$. For $\tau > 0$, take $q(z_\sigma \mid h, x_p) = \mathcal{N}\big(z_\sigma;\, d_\phi(x_p) + g_\psi(h),\, \tau^2\mathbf{I}\big)$, so that $\log q(z_\sigma \mid h, X_p) = -\frac{ND}{2}\log(2\pi\tau^2) - \frac{1}{2\tau^2}\|r_\sigma - g_\psi(h)\|^2$. Since ξ has mean zero and is independent of (r, h), $\mathbb{E}\|r_\sigma - g_\psi(h)\|^2 = \mathbb{E}\|r - g_\psi(h)\|^2 + \sigma^2 ND$. Lemma 2 then gives

$$\mathcal{I}(h; z \mid X_p) \;\ge\; H_{\text{Leb}}(z_\sigma \mid X_p) - \frac{ND}{2}\log(2\pi\tau^2) - \frac{1}{2\tau^2}\Big(\mathbb{E}\big\|r - g_\psi(h)\big\|^2 + \sigma^2 ND\Big). \tag{44}$$

Minimizing residual regression error therefore maximizes a variational lower bound on conditional information, with the local prediction supplied to the decoder as side information.

Batchwise coefficients. Proposition 4 covers LAP-N and the frozen dataset-level linear predictor of Appendix D. For LAP-L, the affine coefficients $\widehat{W}$ are refitted on each batch and are therefore random. Under the same moment assumptions, the proof applies with side information $S = (X_p, \widehat{W})$

and the map that subtracts the corresponding affine prediction. It then bounds $\mathcal{I}(h; z \mid X_p, \widehat{W})$. For LAP-C, the predictor ignores X_p and $z \mapsto z - \bar{c}$ is a bijection given the batch mean $\bar{c}$, so the natural side information is $\bar{c}$ alone and the bound concerns $\mathcal{I}(h; z \mid \bar{c})$. Conditioning on this random batch statistic need not be equivalent to unconditional information. These fits use only clean latents and frozen encoder features, so their laws do not depend on (θ, ψ). Stop-gradient prevents backpropagation through the target construction.

What the two bounded quantities reward. Let m be any measurable function. By Lemma 3(b),

$$\mathcal{I}\big((h, m(X_p)); z \mid X_p\big) = \mathcal{I}(h; z \mid X_p). \tag{45}$$

Appending any function of the local appearance to the hidden state leaves the quantity bounded in Proposition 4 unchanged. The quantity bounded in Proposition 3 behaves differently. By Equation (16),

$$\mathcal{I}\big((h, m(X_p)); z\big) = \mathcal{I}(h; z) + \mathcal{I}\big(m(X_p); z \mid h\big) \;\geq\; \mathcal{I}(h; z). \tag{46}$$

If $\mathcal{I}(h; z) < \infty$, the inequality is strict whenever $\mathcal{I}(m(X_p); z \mid h) > 0$. For example, if h is independent of (z, X_p), the left-hand side equals $\mathcal{I}(m(X_p); z)$ while $\mathcal{I}(h; z) = 0$. This distinction applies to the bounded information quantities. It does not imply that the residual loss is invariant to adding functions of X_p to h. The raw-target decoder can also be inserted into a conditional bound, so a conditional bound alone does not distinguish LAP from raw alignment. Equation (16) gives

$$\mathcal{I}\big(h; (z, X_p)\big) = \mathcal{I}(h; X_p) + \mathcal{I}(h; z \mid X_p), \tag{47}$$

which separates information about the clean latent from information about the target beyond it. Proposition 4 bounds only the second term.

Relation to the denoising loss. In our SiT implementation, every block after block l and the final layer act on the residual stream together with the conditioning vector e. There are no other skip connections. Hence $v_\theta(x_t, t)$ is a function of (h, e). The velocity $v^* = \varepsilon - X$ has a smooth Lebesgue density, since ε is Gaussian and independent of X, and its entropy is finite when X has finite second moments. No smoothing is needed here. Write d for the dimension of the latent and assume $\mathbb{E}\|v_\theta(x_t, t) - v^*\|^2 < \infty$. Lemma 1 with $U = (h, e)$, $W = v^*$, Lebesgue reference measure, and the decoder $\mathcal{N}(v^*; v_\theta, \tau^2 \mathbf{I})$ gives

$$\mathcal{I}\big((h, e); v^*\big) \;\geq\; H_{\mathrm{Leb}}(v^*) - \frac{d}{2}\log(2\pi\tau^2) - \frac{1}{2\tau^2}\,\mathbb{E}\big\|v_\theta(x_t, t) - v^*\big\|^2. \tag{48}$$

The last expectation is $\mathcal{L}_{\mathrm{flow}}$ evaluated at time t. Minimizing it therefore maximizes a lower bound on the information that (h, e) carries about v^*. The data-dependent part of the velocity is $-X$, but this does not identify $\mathcal{L}_{\mathrm{flow}}$ with the first term of Equation (47). The connection motivates supervision that complements reconstruction, without proving that the two objectives capture disjoint information.

Bounds versus bounded quantities. Raising a lower bound does not by itself raise the bounded quantity. For the smoothed target, the Barber–Agakov step has an expected conditional-KL gap. The full bound on the original target also includes a data-processing gap and, when used, the cosine-perturbation slack. These gaps need not remain fixed during training.

B.5 An Exact Identity for Usable Information

Shannon information is a blunt description of a deterministic network (Appendix B.6). A more direct comparison asks how well a projector can predict the target relative to a constant readout. Following the predictive-score construction of Xu et al. (2020), this comparison has an exact expression with no density assumption and no smoothing.

Let $w \in (\mathbb{S}^{D-1})^N$ be any random target, such as $\hat{z}$ or $\hat{r}$. Consider the predictive families

$$\mathcal{V}_h = \{q_\psi(\cdot \mid h) : \psi \in \Psi\}, \qquad \mathcal{V}_\varnothing = \Big\{ \textstyle\prod_i \mathrm{vMF}\big(\cdot\,; \mu, \kappa\big) : \mu \in \mathbb{S}^{D-1} \Big\}, \tag{49}$$

where $\mathcal{V}_\varnothing$ uses one constant direction for all tokens and all inputs. Define the predictive entropies $H_{\mathcal{V}}(w \mid h) = \inf_\psi \mathbb{E}\big[-\log q_\psi(w \mid h)\big]$ and $H_{\mathcal{V}}(w \mid \varnothing) = \inf_{q \in \mathcal{V}_\varnothing} \mathbb{E}\big[-\log q(w)\big]$, and the usable information $\mathcal{I}_{\mathcal{V}}(h \to w) = H_{\mathcal{V}}(w \mid \varnothing) - H_{\mathcal{V}}(w \mid h)$.

Assumption 1. For every $\mu \in \mathbb{S}^{D-1}$ there is $\psi \in \Psi$ with $g_\psi(h)^{(i)} = \mu$ for all h and i. This holds for the token-wise MLP projector, by setting the weights of its last layer to zero and its bias to μ.

For our shared token-wise projector, write $g_\psi(h)^{(i)} = G_\psi(h^{(i)})$ and let J be an independent uniform token index. The score defined above is N times the usual $\mathcal{V}$-information from $h^{(J)}$ to $w^{(J)}$, with J itself not supplied to the decoder. Assumption 1 supplies every constant vMF direction required by optional ignorance in this token-level formulation. The shared-direction baseline is not the family of all input-independent distributions over whole arrays. Position-dependent constant readouts are excluded from the baseline by design; since h carries positional embeddings, realizing them requires reading h.

Proposition 5. Let $\bar{m} = \mathbb{E}\big[\frac{1}{N}\sum_{i=1}^{N} w^{(i)}\big] \in \mathbb{R}^D$ be the mean of the normalized target tokens. Then

$$\mathcal{I}_{\mathcal{V}}(h \to w) = \kappa N \left(\sup_{\psi \in \Psi} \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \cos\Big(g_\psi(h)^{(i)}, w^{(i)} \Big) \right] - \|\bar{m}\| \right), \tag{50}$$

and under Assumption 1 this quantity is nonnegative.

Proof. Step 1: the score with access to h. Let $C_\psi = \mathbb{E}[\frac{1}{N}\sum_i \cos(g_\psi(h)^{(i)}, w^{(i)})]$. By Equation (20), the predictive entropy is

$$\begin{aligned} H_{\mathcal{V}}(w \mid h) &= \inf_{\psi} \big\{ -N \log c_D(\kappa) - \kappa N C_\psi \big\} \\ &= -N \log c_D(\kappa) - \kappa N \sup_{\psi} C_\psi. \end{aligned} \tag{51}$$

The infimum becomes a supremum because the coefficient of C_ψ is negative.

Step 2: the best constant readout. A shared unit direction μ earns expected cosine

$$\mathbb{E}\left[\frac{1}{N} \sum_{i} \mu^\top w^{(i)} \right] = \mu^\top \bar{m} \le \|\bar{m}\|, \tag{52}$$

by linearity of expectation and the Cauchy–Schwarz inequality. If $\bar{m} \neq 0$, equality holds at $\mu^* = \bar{m}/\|\bar{m}\|$. If $\bar{m} = 0$, every unit direction earns zero. Thus the optimum is $C_0 = \|\bar{m}\|$, and

$$H_{\mathcal{V}}(w \mid \varnothing) = -N \log c_D(\kappa) - \kappa N C_0. \tag{53}$$

Step 3: subtracting the baseline. The normalizing constants cancel when the two predictive entropies are subtracted, giving

$$H_{\mathcal{V}}(w \mid \varnothing) - H_{\mathcal{V}}(w \mid h) = \kappa N \big(\sup_{\psi} C_\psi - C_0 \big). \tag{54}$$

Under Assumption 1, the projector can realize the optimal constant direction, so $\sup_\psi C_\psi \ge C_0$. This proves both the identity and its nonnegativity. □

In the notation of Section 4.3, $\|\bar{m}\| = C_0$ and the supremum is $\sup_\psi C_\psi$, so Equation (50) reads $\mathcal{I}_{\mathcal{V}}(h \to w) = \kappa N(\sup_\psi C_\psi - C_0)$: the usable information is the gain of the best projector over the constant-readout baseline of Equation (6).

Computing the baseline. Given n target arrays with N tokens each, normalize each token, average the resulting directions, and take the norm:

$$\widehat{C}_0(y) = \left\| \frac{1}{nN} \sum_{j=1}^{n} \sum_{i=1}^{N} \frac{y_j^{(i)}}{\|y_j^{(i)}\|} \right\|. \tag{55}$$

This is the exact constant-readout optimum for the empirical target distribution and an estimate of the population baseline. It requires only target features, with no denoiser or fitted probe. For batchwise purification, we first form the targets in the same batches as the training procedure, then pool their normalized directions (Appendix C.8). The baseline uses one direction shared across positions and images, as specified above.

The constant-readout reference. The best shared constant direction attains cosine $C_0 = \|\bar{m}\|$ without reading h. Appendix C.8 measures C_0 for every target. For raw MAE it is 0.93, compared

with 0.957 for the trained projector. Their difference, about 0.03, is illustrative because the two estimates use different samples. Note that C_0 is computed from unit-normalized tokens, so a target with zero mean can still have $C_0 > 0$, and subtracting a mean does not by itself remove the baseline. The measured difference is not a certified population bound or an upper bound on Equation (50), which optimizes over the entire projector family. LAP-C, LAP-L, and LAP-N subtract a constant (the batch mean token), a batchwise affine prediction, and a frozen nonlinear local prediction, respectively, before this comparison is made.

Prediction gain over a fixed baseline. With the Gaussian families $\{\mathcal{N}(z;\, d_\phi(X_p) + b,\, \tau^2\mathbf{I}) : b \in \mathbb{R}^D\}$ and $\{\mathcal{N}(z;\, d_\phi(X_p) + g_\psi(h),\, \tau^2\mathbf{I}) : \psi \in \Psi\}$, where b is shared across tokens, the analogous log-score improvement is

$$\Delta_{\mathrm{pred}}(h; d_\phi) = \frac{1}{2\tau^2}\left(\inf_b \mathbb{E}\|r - b\|^2 - \inf_\psi \mathbb{E}\|r - g_\psi(h)\|^2\right). \tag{56}$$

Here $r - b$ subtracts b from every token. With a finite residual second moment and a correction family containing every shared constant b, this score is finite and nonnegative. It measures improvement over the specified frozen baseline. We do not identify it with conditional $\mathcal{V}$-information in the full sense of Hewitt et al. (2021), which imposes an optional-ignorance condition on the joint predictive family. The fixed baseline and restricted correction family need not satisfy that condition.

B.6 Scope

Deterministic hidden states. Since h is a function of (x_t, e), Equation (17) gives $\mathcal{I}(h; z) \le \mathcal{I}\big((x_t, e); z\big)$, with equality whenever that function is injective. Shannon information therefore does not distinguish representations that differ by a bijection, and it cannot increase beyond what the noisy input contains. Training directly changes the score of the current projector. Proposition 5 characterizes the best score in the family, which a trained projector need not attain. Neither is an identity for Shannon information.

Conditioning on the clean latent. X_p contains every latent patch and nearly determines the image. $\mathcal{I}(h; z \mid X_p)$ can then be small and $H_\nu(\hat{r}_\sigma \mid X_p)$ very negative, so Equation (38) can be loose. The operational content of LAP lies in the predictor class. d_ϕ has a limited receptive field and capacity, and an unrestricted predictor of z from the full latent would leave almost no residual. Thus, LAP defines an operational separation without guaranteeing a semantic residual (Section 6).

Residualization is not conditional independence. A residual can still be predictable from X_p. For example, let S be uniform on $\{-1, 1\}$, take one token with $X_p = S$ and $z = (S, 1)$, and use the best constant predictor $d_\phi = (0, 1)$. Then $r = (S, 0)$ and $\mathcal{I}(h; z \mid X_p) = 0$ for every h, since z is determined by X_p. Yet reading $h = S$ achieves residual cosine 1, whereas every constant direction has expected cosine 0. The same happens with an affine predictor, as in LAP-L. Let S be uniform on $\{-1, 0, 1\}$, $X_p = S$, and $z = (S^2, 1)$. Since S and S^2 are uncorrelated, the least-squares affine prediction is the constant $(2/3, 1)$, so $r = (S^2 - 2/3, 0)$. Reading $h = S$ again achieves cosine 1 through $g(h) = (h^2 - 2/3, 0)$, while the best constant direction attains only $1/3$. A conditional information bound therefore does not establish that the residual loss ignores local content. LAP's selectivity comes from its chosen predictor class and must be assessed empirically.

No account of harm. Neither the information bounds nor the predictive-score identity establishes that redundant content harms generation. Explaining why the predictable-only target can be worse than no alignment requires an argument about optimization or model capacity, which this analysis does not provide.

Relation to information-maximizing diffusion. InfoDiffusion (Wang et al., 2023) learns an auxiliary latent u through an encoder $q_\phi(u \mid x_0)$, written u here to avoid a clash with the teacher features z, and conditions the reverse process on it. Its regularized objective has the form

$$\mathbb{E}_q\big[\mathcal{L}_D(x_0)\big] + \zeta\, \mathcal{I}_q(x_0; u) - \beta\, \mathrm{D}\big(q_\phi(u) \,\|\, p(u)\big), \tag{57}$$

which is maximized for positive weights ζ and β. Here $\mathcal{L}_D$ is the diffusion evidence lower bound, $\mathcal{I}_q(x_0; u)$ is the data–latent information under the learned joint distribution, and $q_\phi(u)$ is the aggregate posterior. The positive information term encourages informative latents and addresses their being ignored by the diffusion decoder. InfoDiffusion obtains a tractable objective through a decomposition involving the encoder and aggregate-posterior divergences.

Alignment instead starts with a frozen teacher $z = f(I)$. Its purpose is to shape intermediate denoiser features so that a projector can read the chosen teacher content from them. The teacher is a training target and is not supplied to the denoiser at inference. The vMF argument gives a lower bound on $\mathcal{I}(h; z)$, and Equation (41) shows that the alignment term rewards a larger bound. This differs from learning the auxiliary latent and regularizing its posterior in InfoDiffusion.

LAP further selects the target content through a fixed local predictor. Its residual objective admits the conditional bound of Equation (39), which connects the method to the goal of guidance beyond local reconstruction. The conditional bound alone does not establish such a separation: a raw-target decoder also admits a conditional bound, and residuals can remain locally predictable. The substantive change is the target supplied to the restricted projector. Its benefit is established by the interventions and generation results, while the information analysis clarifies the objective and its optimization direction.

C Experimental and Diagnostic Protocols

This appendix defines the measurements used in the main text and shows how they vary with noise level, subspace rank, and intervention type. Unless stated otherwise, diagnostics use seed 0 of a 400K-step ImageNet-1k checkpoint. Generation results use the seed coverage stated in each table.

C.1 Training and Evaluation Settings

Our base model is SiT-B/2 with 12 blocks and hidden width 768. The SD-VAE produces a $4\times32\times32$ latent. Patches of size 2×2 give 256 tokens, each containing 16 latent values. Alignment is applied after block 8 through a three-layer MLP with hidden width 2048 and SiLU activations. The alignment coefficient is $\lambda_a = 0.5$. The main teacher comparison uses MAE ViT-L/16, AIMv2 ViT-L/14, CLIP ViT-L/14 (Radford et al., 2021), DINOv3 ViT-B/16, and DINOv2 ViT-B/14. Development comparisons also include I-JEPA ViT-H/14 (Assran et al., 2023) and MoCo v3 ViT-B/16 (Chen et al., 2021). The encoder preprocessing gives each target a 16×16 patch grid.

Base runs use batch 64, AdamW with learning rate 10^{-4}, fp16, and uniformly sampled noise times. Target comparisons on dev-200 train for 50K steps. The DINOv2 auxiliary-prediction sweep and its projection controls train for 100K steps. The full ImageNet-1k comparisons train for 400K steps. The larger-batch and larger-model settings are listed separately in Appendix F.

The development metric, FID-DINOv2, compares DINOv2 ViT-B features from 10K EMA-generated images with those from the training images. Sampling uses 50 Euler ODE steps without guidance. Standard FID-50K uses a different feature space and sampling protocol, described in Appendix H. The two metrics should be compared only within their respective settings.

C.2 Linear Probes and Subspace Interventions

The main hidden-state diagnostics use 1,024 images for fitting and 256 disjoint images for evaluation. A seed-20260916 permutation selects these images from the ImageNet-1k training data. They are held out from probe fitting, not from diffusion training. Every checkpoint receives the same images, VAE posterior samples, noise, and times, with $t \sim \mathcal{U}[0.3, 0.7)$. Sampling 80 patch tokens per image gives 81,920 fit tokens and 20,480 evaluation tokens. The component probes use the separate sampling rule below.

For a probe from features H to responses Y, we center both matrices on the fit split and solve

$$W = (H_c^\top H_c + \lambda I)^{-1} H_c^\top Y_c, \qquad \lambda = 10^{-3}\frac{\operatorname{tr}(H_c^\top H_c)}{d}. \tag{58}$$

Here d is the feature dimension of H. The intercept restores the fit-split means. We report evaluation-split R^2, computed as one minus total squared prediction error divided by response variation around the evaluation-split mean. Negative values indicate worse prediction than this mean baseline. The relative penalty scales with feature magnitude. The image-disjoint evaluation uses FP32 inference and FP64 ridge solves, without tuning on evaluation scores.

What each probe measures. Presence probes the hidden state for target features. Prediction utility probes the target for clean-latent patches. Velocity sufficiency probes the aligned representation for

Table 5: **Image-disjoint diagnostics.** ImageNet-1k/400K, mid-noise, seed 0 unless marked. Presence always probes raw DINOv2. "Historical" and "paired" denote token splits. Velocity R^2 and use are evaluated on disjoint images. Use is a percentage of full-layer mean-fill ablation.

	DINOv2 presence R^2			Velocity	Use
Model	Historical	Paired	Image	R^2	(%)
Unaligned	0.196	0.166	0.130	0.451	14.03
LAP-L (MAE)	0.268	0.226	0.172	0.114	15.38
LAP-N (MAE)	0.290	0.240	0.178	0.004	10.61
REPA	0.343	0.286	0.207	0.115	12.21
LAP-L (DINOv2)	0.342	0.289	0.213	−0.016	6.67
LAP-N (DINOv2)	0.361	0.297	0.213	−0.066	4.18
REPA + prediction loss	0.297	0.255	0.197	0.534	18.86
REPA (seed 1)	0.348	0.291	0.208	0.112	10.64
REPA + prediction loss (seed 1)	0.300	0.256	0.197	0.533	11.51

velocity patches. These quantities have different inputs and outputs and should not be treated as interchangeable.

Which directions are removed. We form the aligned subspace from the top-r left singular vectors of the probe matrix for $h \to a_t$. A target-probe control instead uses the matrix for $h \to y$. The main-text subspaces use only the fitting images and rank $r = 256$. Supplementary rank sweeps are reported separately below.

For a projector Π and bin-conditional hidden-state mean $\bar{h}_b$, mean filling replaces the hidden state by

$$h^{\text{ablate}} = (I - \Pi)h + \Pi\bar{h}_b. \tag{59}$$

The remaining network then runs normally. For the main-text results, $\bar{h}_b$ is estimated from all tokens of the fitting images. Baseline, full-layer, and subspace effects use identical evaluation images and noise draws. We divide the increase in velocity MSE by the full-layer increase. The random reference averages five fixed rank-256 subspaces. A supplementary resampling control replaces Πh using a random permutation of images in the batch, preserving patch positions.

These are interventions on one layer, not measurements of every pathway by which alignment affects training. Downstream computation can compensate for a local ablation (McGrath et al., 2023). Compensation may differ between subspace and full-layer interventions, so the normalized scores are most useful for matched comparisons.

C.3 Image-Disjoint Results and Split Sensitivity

Table 5 reports the main-text diagnostics and compares presence under three evaluation protocols. The paired token split uses exactly the same feature pool and fit/evaluation token counts as the image split, but allows tokens from every image on both sides. Historical scores use the earlier dense-token collection. Changes from historical scores therefore reflect both sampling and the split, and cannot all be attributed to image overlap.

The main comparisons persist across images. LAP-N (MAE) closes 62.8% of the unaligned-to-REPA presence gap (95% interval: 60.5–65.0%). For the auxiliary loss, velocity R^2 rises from $0.115/0.112$ to $0.534/0.533$ in seeds 0/1. Aligned-subspace use rises from $12.2/10.6\%$ to $18.9/11.5\%$, with a smaller change in seed 1. Under the same protocol, the five rank-256 random subspaces of the seed-0 REPA model give 22.2–23.0%, with mean 22.7% (95% interval: 21.7–23.7%). This is the random reference in Figure 12(a). For DINOv2 purification, presence is retained while velocity prediction and normalized use decrease. We show absolute probe R^2 in Figure 7(b), including values below zero, and display use on a separate axis.

The image-bootstrap intervals in Figures 7, 8 and 12 use 2,000 resamples of evaluation images, with identical resamples for paired comparisons. Each fitted probe remains fixed, and each bootstrap R^2 uses the response mean of that resample. Component intervals also condition on the fitted batch

decompositions. These intervals quantify evaluation-image variability. They do not cover probe-refitting uncertainty, additional training seeds, or new random-subspace draws.

C.4 Representation Changes after Purification

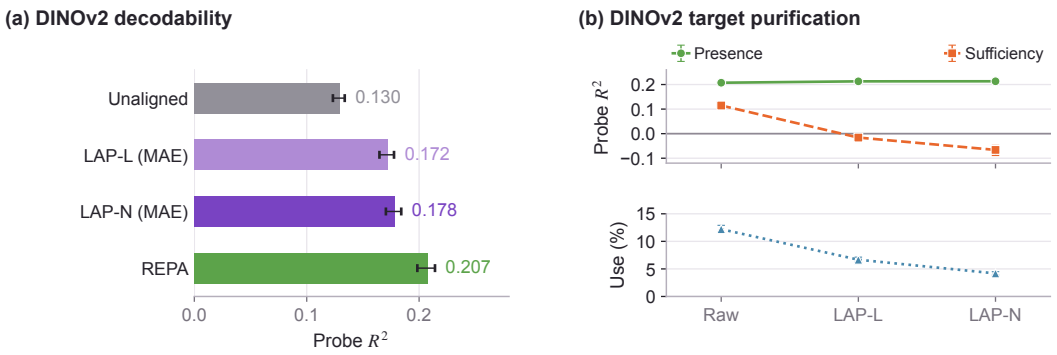


Figure 7: **Purification preserves readable features while reducing prediction coupling.** (a) DINOv2 decodability after MAE-based alignment. (b) Absolute probe R^2 (top) and normalized ablation effect (bottom) for DINOv2 targets. Negative R^2 is worse than the evaluation-mean baseline. Presence always probes raw DINOv2. Whiskers give 95% image-bootstrap intervals. Seed 0, mid-noise, 400K steps, disjoint fit and evaluation images (Appendix C.3).

Purified MAE yields features that are readable as DINOv2. Although DINOv2 is never used in its training, a model aligned to purified MAE becomes more DINOv2-decodable. LAP-N raises the held-out probe score from 0.130 in the unaligned model to 0.178, closing about 63% of the gap to explicit DINOv2 alignment at 0.207 (Figure 7(a)). The probes are evaluated on images held out from probe fitting, not from diffusion training.

Purification reduces coupling to velocity prediction. Purifying the DINOv2 target keeps the teacher decodable from the hidden state, while linear velocity prediction from the aligned representation and the denoiser's reliance on the aligned subspace both decrease (Figure 7(b)). Negative velocity R^2 reflects poor generalization of the linear probe, not the absence of reconstruction information. Generation quality is preserved while this coupling decreases, consistent with Section 3.1, where increasing it worsened generation. Together, these results support choosing a target by the content it asks the model to match, rather than by its reconstruction or class predictability alone.

C.5 Diagnostics Across Noise Levels

The supplementary noise, rank, and resampling checks retain the earlier protocol. It collects approximately 10^5 dense tokens from shuffled image batches and randomly splits them 80/20. Images can appear on both sides, so these scores measure held-out-token predictability. The conditional mean, full-layer effect, and subspace effects use separate sampling passes. A nonlinear REPA probe under this protocol gives $R^2 = 0.387$, compared with 0.343 for its linear counterpart.

Table 6 reports this protocol across high noise $[0.7, 1]$, mid-noise $[0.3, 0.7)$, and low noise $[0, 0.3)$. Presence uses DINOv2 as the probe target, except for raw MAE, where it uses MAE. The cosine column also compares against DINOv2, except for raw MAE. MAE-residual projectors output MAE-dimensional features, so their DINOv2 cosine is undefined.

REPA's aligned-subspace use is lowest at high noise. The predictable-only target becomes more velocity-predictive toward low noise, whereas the purified targets show the opposite trend. These are observations under the stated probes. They do not establish that the aligned subspace is the only route through which target information can affect the denoiser.

Table 6: **Diagnostics across noise levels.** ImageNet-1k/400K, seed 0, earlier token-split protocol. Use is the rank-256 aligned-subspace effect as a percentage of the full-layer effect. Probe targets are specified in the text.

Model	Noise	Presence R^2	Cosine	Velocity R^2	Use (%)
Unaligned	High	0.177	0.005	0.582	8.5
	Mid	0.196	0.008	0.451	13.9
	Low	0.196	0.006	0.333	15.0
Raw MAE	High	0.255	0.947	0.329	3.8
	Mid	0.386	0.960	0.275	17.7
	Low	0.485	0.970	0.269	22.7
REPA	High	0.293	0.513	0.126	2.1
	Mid	0.343	0.630	0.164	10.9
	Low	0.386	0.714	0.164	13.4
LAP-L (DINOv2)	High	0.290	0.482	0.112	1.4
	Mid	0.342	0.606	0.089	6.5
	Low	0.385	0.692	0.060	8.0
LAP-N (DINOv2)	High	0.301	0.445	0.081	0.8
	Mid	0.361	0.565	0.079	4.5
	Low	0.403	0.644	0.066	5.5
LAP-L (MAE)	High	0.247	—	0.190	3.0
	Mid	0.268	—	0.149	15.5
	Low	0.283	—	0.063	18.8
LAP-N (MAE)	High	0.258	—	0.129	2.2
	Mid	0.290	—	0.074	11.0
	Low	0.302	—	0.056	13.4
Predictable-only (DINOv2)	High	0.204	0.208	0.163	8.8
	Mid	0.210	0.219	0.184	11.3
	Low	0.217	0.220	0.270	16.5
REPA + prediction loss	High	0.259	0.501	0.628	2.8
	Mid	0.297	0.607	0.541	16.0
	Low	0.324	0.679	0.452	25.6

C.6 Reference Subspaces, Resampling, and Rank

The controls in this subsection use the earlier token-split protocol. A random subspace gives a reference for how much disruption an intervention of the same rank causes. For REPA, the rank-256 random-subspace effect in Table 7 is larger than the aligned-subspace effect in every noise bin. The unaligned model's untrained readout provides a second reference. It does not define the same directions as REPA's learned readout.

Table 7: **Random-subspace references.** Earlier protocol. Mean-fill use as a percentage of full-layer ablation. REPA brackets give the observed ranges over random subspace draws. These are not training-seed intervals.

Checkpoint	High	Mid	Low
REPA (400K)	13.8 [13.6, 14.0]	22.4 [21.0, 23.9]	30.1 [28.8, 31.5]
Unaligned (400K)	13.0	15.9	20.7

Resampling preserves the marginal distribution of the replaced component more closely than mean filling. Table 8 compares the two interventions at mid-noise. REPA and the purified-target models have smaller normalized effects under resampling, but the exact ordering of all methods is not preserved. For example, raw MAE and the predictable-only model change order. The robust ob-

servation is the small resampling effect for REPA and purified targets, not equivalence of the two estimators.

Table 8: **Intervention sensitivity.** Earlier protocol, ImageNet-1k/400K, seed 0, mid-noise, rank 256. Each estimator uses its corresponding full-layer normalization.

Model	Mean fill (%)	Resampling (%)
Unaligned	13.9	15.0
Raw MAE	17.7	6.5
LAP-L (MAE)	15.5	4.9
LAP-N (MAE)	11.0	1.4
REPA	10.9	2.0
LAP-L (DINOv2)	6.5	1.6
LAP-N (DINOv2)	4.5	1.1
Predictable-only (DINOv2)	11.3	10.6
REPA + prediction loss	16.0	14.6

The rank sweep in Table 9 tests whether the conclusion depends on removing 256 directions. Increasing rank generally removes more useful information. Comparisons should therefore be made at the same rank. REPA remains close to the unaligned reference at ranks 32 and 128, and below it at rank 256. The decreasing DINOv2 sequence holds at all three ranks. The MAE ordering does not: at rank 32, LAP-N (MAE) has the largest aligned-subspace effect among the MAE targets.

Table 9: **Rank sensitivity of mean-fill use.** Earlier protocol, mid-noise, 400K, seed 0. Values are percentages of full-layer ablation. Parentheses show the target-probe subspace control.

Model	r=32	r=128	r=256
Unaligned	0.8 (0.2)	4.4 (0.8)	13.9 (2.8)
Raw MAE	2.0 (1.0)	8.6 (7.1)	17.7 (16.4)
REPA	0.8 (0.3)	4.8 (1.8)	10.9 (8.7)
LAP-L (DINOv2)	0.6 (0.3)	2.9 (1.6)	6.5 (6.5)
LAP-N (DINOv2)	0.2 (0.3)	2.1 (1.9)	4.5 (8.8)
LAP-L (MAE)	3.0 (0.3)	10.4 (1.7)	15.5 (5.6)
LAP-N (MAE)	5.8 (0.5)	8.7 (1.9)	11.0 (8.7)
Predictable-only (DINOv2)	1.6 (0.1)	4.5 (0.5)	11.3 (2.2)
REPA + prediction loss	1.7 (1.2)	4.3 (3.7)	16.0 (7.3)

C.7 Component Readout and Easy-to-Match Content

The component diagnostic in Figure 8(b) uses the same 1,024 fit and 256 evaluation images as the other main-text probes. We fit the affine clean-latent-to-DINOv2 decomposition within batches of 64 images. No decomposition batch crosses the split boundary. We select 48 tokens per image for the component probes, giving 49,152 fit and 12,288 evaluation tokens. Readout cosines use all 256 tokens of each evaluation image. Thus, the targets remain batch-defined components, while the probes are tested on disjoint images.

The image-predictable component has higher presence than the residual (0.256 vs. 0.188), while the projector has higher cosine similarity to the residual (0.581 vs. 0.313). The paired differences have 95% image-bootstrap intervals of $[0.047, 0.086]$ for predictable-minus-residual presence and $[0.254, 0.283]$ for residual-minus-predictable cosine. These measurements distinguish availability from readout matching. They do not prove that the projector causally selects semantic content, because component geometry and scale can also affect cosine.

For comparison, a random-token split of the same component collection gives predictable/residual presence $0.443/0.263$. The earlier component protocol sampled 60K tokens across 20 batches and gave approximately $0.40/0.25$, with readout cosines $0.31/0.58$. Both protocols retain the ordering,

but token-held-out scores are higher. The earlier development-scale residual/predictable cosine pair is $0.52/0.42$.

C.8 Constant-Readout Evaluation

The constant-readout baseline C_0 of Section 4.3 measures how much alignment cosine a single fixed direction obtains without image-specific computation. We estimate it as the norm of the sample mean of unit-normalized tokens. Features come from $2{,}048$ training images drawn with a fixed seed, processed in consecutive batches of 64 so that the batchwise targets (mean removal and LAP-L) are formed exactly as in training, with one fixed VAE posterior sample per image shared by all targets. Table 10 reports estimates for dev-200 and ImageNet-1k with the corresponding purifiers. Raw MAE has a baseline of 0.933 on dev-200 and 0.929 on ImageNet-1k. Its logged training cosine of 0.957 is an illustrative comparison from different samples (Appendix B.5). For predictable-only DINOv2, the baseline is 0.802 on dev-200 and 0.733 on ImageNet-1k. Both permit high alignment scores without reading the hidden state. Under LAP-N, the smallest token norms across these samples are approximately 24.3 for MAE, 20.6 for DINOv2, and 8.9 for CLIP.

Per-position constant readouts. The shared-direction baseline is the strictly input-free one. A readout that is allowed one fixed direction per token position still reads nothing about the image, but it can exploit position-dependent structure in the target; its best expected cosine is $C_0^{\mathrm{pos}} = \frac{1}{N}\sum_i \|\mathbb{E}[\hat{y}^{(i)}]\|$, the position-wise analogue of Equation (6), and $C_0^{\mathrm{pos}} \geq C_0$ by the triangle inequality. Table 10 reports both. For the raw targets the two nearly coincide for MAE and CLIP, whereas raw DINOv2 tokens carry a position-dependent mean that a shared direction misses (0.16 against 0.31 on ImageNet-1k). The evaluated purified targets retain position-dependent mean structure: a per-position readout still reaches about 0.33 on LAP-N MAE, 0.21 on LAP-N DINOv2, and 0.10 on LAP-N CLIP. For the shared token-wise projector, positional information is available through the hidden state. The main text uses C_0 to isolate a shared direction, while the per-position baseline quantifies what a readout with access to position alone can achieve without image content. The tested target transformations do not explicitly subtract a position-specific mean across images. LAP-C subtracts one batch-mean token shared by all positions, while the iREPA-style operation standardizes each image across its positions (Appendix E.4). Neither operation generally forces the position-wise directional mean to vanish. Subtracting a frozen per-position mean is a further member of the purification family, with the token position as the predictor's only input; Appendix C.9 evaluates it as LAP-P.

Table 10: **Constant-readout baselines by dataset, encoder, and target.** Empirical estimates from $2{,}048$ images per dataset. C_0 is the best expected cosine of one direction shared by all tokens (Equation (6)); $C_0^{\mathrm{pos}} = \frac{1}{N}\sum_i \|\mathbb{E}[\hat{y}^{(i)}]\| \geq C_0$ is the best expected cosine of a readout with one fixed direction per token position, which uses the position but not the image. LAP-N uses the purifier trained on the same dataset ($k = 5$ for MAE, $k = 3$ for DINOv2 and CLIP); CLIP $k = 5$ on dev-200 gives 0.062 and 0.094. The predictable-only row is the batchwise affine prediction of DINOv2 from the clean latent. A dash denotes an unavailable measurement.

		C_0 (shared direction)				C_0^{pos} (one direction per position)			
Data	Encoder	Raw	LAP-C	LAP-L	LAP-N	Raw	LAP-C	LAP-L	LAP-N
Dev-200	MAE	0.933	0.019	0.020	0.011	0.945	0.416	0.437	0.329
	CLIP	0.607	0.088	0.088	0.067	0.620	0.176	0.169	0.110
	DINOv2	0.221	0.006	0.007	0.011	0.347	0.275	0.272	0.207
	AIMv2	0.663	0.041	0.041	–	0.701	0.224	0.221	–
	DINOv3	0.525	0.018	0.018	–	0.587	0.311	0.310	–
	DINOv2, predictable-only	0.802	–	–	–	0.804	–	–	–
ImageNet-1k	MAE	0.929	0.016	0.017	0.010	0.941	0.401	0.420	0.326
	CLIP	0.585	0.082	0.082	0.066	0.593	0.145	0.140	0.098
	DINOv2	0.162	0.005	0.005	0.011	0.306	0.265	0.263	0.205
	DINOv2, predictable-only	0.733	–	–	–	0.735	–	–	–

The predictable-only DINOv2 target is also easy to match. Its two full-scale runs start near cosine $0.82/0.81$ at step 500 and finish near $0.83/0.85$. Raw DINOv2 starts near 0.32 and finishes near $0.62/0.63$. Training-log cosines and held-out diagnostic cosines use different samples and should not be equated. In particular, the DINOv2 cosine in Table 6 for the predictable-only model is measured against raw DINOv2, rather than that model's own training target.

C.9 Position-Dependent Purification (LAP-P)

The per-position baseline above identifies shared target structure that LAP-C, LAP-L, and LAP-N can retain: a mean token that depends on position i but not on the image. A position-dependent predictor subtracts $m_i = \mathbb{E}[z^{(i)}]$, estimated once on $8{,}192$ training images and then frozen. We call this LAP-P. The population position means average to the global mean token. Empirically, LAP-P and LAP-C have the same shared-direction score at the displayed precision ($C_0 = 0.019$ for MAE), while the per-position score falls from 0.42 to 0.03 (Table 11). Position-dependent centering can also be applied after LAP-N by subtracting the per-position mean of its residual (LAP-N+P).

Table 11: **Position-dependent purification.** C_0 and C_0^{pos} from 2,048 dev-200 images (Appendix C.8); FID-DINOv2 on dev-200/50K and FID-50K on ImageNet-1k/400K, individual seeds and means. The dev-200 purifier for DINOv2 uses k=3. Full-scale LAP-P runs were made for MAE only. Bold marks the lowest FID per encoder and column, including ties at displayed precision.

				Dev-200 FID-DINOv2			ImageNet-1k FID-50K		
Encoder	Target	C_0	C_0^{pos}	Seed 0	Seed 1	Mean	Seed 0	Seed 1	Mean
MAE	Raw	0.933	0.945	1641.4	1643.2	1642.3	53.68	52.41	53.05
	LAP-C	0.019	0.416	1544.0	1552.6	1548.3	45.07	45.27	45.17
	LAP-L	0.020	0.437	1504.3	1512.3	1508.3	44.81	45.48	45.15
	LAP-N	0.011	0.329	1486.2	1482.6	1484.4	43.06	**41.53**	42.30
	LAP-P	0.019	0.027	1495.9	1502.8	1499.4	44.02	44.62	44.32
	LAP-N+P	0.012	0.025	**1471.3**	**1436.4**	**1453.8**	**42.33**	41.80	**42.07**
DINOv2	Raw	0.221	0.347	1551.2	1518.6	1534.9	42.14	**41.35**	41.75
	LAP-L	0.007	0.272	1546.0	**1487.8**	1516.9	**40.61**	42.33	**41.47**
	LAP-N	0.011	0.207	**1508.4**	1536.2	1522.3	41.90	42.33	42.11
	LAP-P	0.010	0.019	1518.9	1507.3	1513.1	–	–	–
	LAP-N+P	0.010	0.020	1512.1	1504.0	**1508.1**	–	–	–

Development scale. On MAE, LAP-P alone reaches 1499.4, better than LAP-C (1548.3) and LAP-L (1508.3) and close to LAP-N (1484.4), although its predictor uses no image content; LAP-N+P is the best development-scale MAE target (1453.8, 30.6 below LAP-N). On DINOv2 the four purified targets lie within 15 points of one another.

Full scale. The development-scale ordering does not persist. LAP-P reaches 44.32 FID-50K, 0.85 below LAP-C but 2.0 above LAP-N. LAP-N+P reaches 42.07 against 42.30 for LAP-N, a difference within seed variation that improves only one of the two matched seeds. Thus, position-dependent centering improves on LAP-C in this comparison, while its additional benefit on top of LAP-N is not resolved by these two seeds. We retain LAP-P as an ablation. Per-image spatial standardization in iREPA instead centers each image across its own positions, so it does not generally remove a position-dependent mean across images (Appendix E.4).

C.10 Clean-Latent Prediction Across Noise Levels

The raw-target utility scores in Table 28 use 30 shuffled batches of 32 images (245,760 tokens). Probes fit the first 80% of tokens and evaluate on the remaining 20%, without a second token shuffle. This boundary falls between images.

To ask what target features add beyond the noisy input, we compare two clean-latent probes:

$$\Delta R^2 = R^2(X \mid x_t, y) - R^2(X \mid x_t). \tag{60}$$

Table 12 shows that MAE adds more linear reconstruction information than DINOv2 or CLIP, including at high noise. This does not translate into the largest alignment benefit. At low noise, the hidden states of both unaligned and REPA models already predict clean-latent patches with $R^2 \approx 0.95$. Incremental probes use 16 batches of 32 images and the same contiguous 80/20 token split. This boundary can split one image. Separate posterior and noise draws give slightly different noisy-input baselines for each encoder. Small negative increments can arise from estimation and regularization.

Table 12: **Clean-latent prediction.** Incremental utility uses dev-200 data. Hidden-state scores use the earlier token-split protocol on 400K ImageNet-1k checkpoints. The baseline row shows the range across the three encoder evaluations. High, mid, and low denote the noise bins defined above.

		High	Mid	Low
Base range	$R^2(X \mid x_t)$	0.037–0.051	0.472–0.501	0.926–0.934
ΔR^2	MAE	0.655	0.266	0.013
	DINOv2	0.268	0.052	−0.001
	CLIP	0.114	0.026	−0.001
$R^2(X \mid h_t)$	Unaligned	0.264	0.676	0.949
	REPA	0.248	0.662	0.946

C.11 Seed Coverage and Uncertainty

Core generation comparisons use two independent training seeds. Slash-separated pairs give seeds 0 and 1. A lone generation value is a single evaluated run unless identified as a mean. Diagnostic coverage is specified separately. The range between two seeds describes the observed variation and is not a confidence interval.

The dev-200 REPA pair, $1551.2/1518.6$, gives a sample standard deviation $s = |1551.2 - 1518.6|/\sqrt{2} \approx 23.0$. Adding the repeat at 1553.3 gives $s \approx 19.5$. Even this one additional run changes the estimate. Variation can also depend on the configuration.

Under an independent, equal-variance approximation, the standard error of the difference between averages of n and m runs is $s\sqrt{1/n + 1/m}$. With only two runs per configuration, this approximation does not justify precise significance claims or equivalence tests. We therefore report individual values, compare consistent directions across seeds and metrics, and avoid ranking closely performing methods.

Evaluating another metric on the same generated images checks sensitivity to the feature space. Drawing new samples checks sampling variation. Evaluating intermediate checkpoints shows the training trajectory. None of these replaces independent training seeds. The Inception-space check and complete trajectories are in Appendix H.

D Controls for Predictive Supervision and Target Content

The controls address two possible confounds: the architecture used for auxiliary prediction, and the geometry or instability introduced by decomposing a target. All development values in this appendix are FID-DINOv2. Standard FID is named explicitly.

D.1 Auxiliary Prediction Controls

The velocity-prediction head r_ω is a timestep-conditioned MLP with one hidden layer of width 256 and SiLU activation. It receives the aligned representation a_t. The loss is uniformly weighted over noise times, and gradients pass through the head into the shared hidden state. Setting $\lambda_s = 0$ recovers REPA.

We also apply the same loss to a fixed random projection of h and to a separately learned projection. These controls test whether the degradation requires supervision of the aligned representation itself.

At dev-200/100K with $\lambda_s = 2$, the seed-0 scores are 1284.5 and 1277.4, compared with 1242.0 for supervision through the aligned representation and 1157.3 for REPA. Both controls reduce aligned-subspace use from 8.3% to about 4.1–4.3%, while leaving target presence and aligned-readout sufficiency near the REPA reference.

Thus, degradation is not unique to the aligned projection. The results are consistent with interference through shared parameters, but they do not isolate a particular gradient-level mechanism. A separate MAE sweep at dev-200/50K gives 1641.4, 1605.4, and 1649.7 for $\lambda_s = 0, 0.5, 2$. This single-run check has no monotonic benefit.

The main negative result replicates across two seeds. At dev-200/100K, auxiliary prediction gives $1242.0/1252.3$ against REPA's $1157.3/1161.2$. At full scale, it gives $879.5/877.1$ against $825.1/821.1$. Standard FID-50K likewise increases from 41.75 to 46.79 at seed means. The full sweep is in Table 27, and the standard metrics of this control are in Table 21.

D.2 The Affine Target Decomposition

Let $X_p \in \mathbb{R}^{M\times 16}$ contain the clean-latent patches from one batch, with $M = BN$ tokens. Append a column of ones to obtain $\widetilde{X}_p$. For encoder tokens $Z \in \mathbb{R}^{M\times D}$, ordinary least squares defines

$$\widehat{W} = \widetilde{X}_p^{+} Z, \qquad Z_{\mathrm{pred}} = \widetilde{X}_p \widehat{W}, \qquad Z_{\mathrm{res}} = Z - Z_{\mathrm{pred}}. \tag{61}$$

Aligning to Z_{res} defines LAP-L. It is the affine variant of the residual-target construction in Section 4.1, with coefficients fitted to the current batch rather than frozen before training. The fitted component has rank at most 17 across the batch tokens. With an intercept, the residual has zero batch mean and is orthogonal to the design matrix, up to numerical precision. These in-batch properties do not imply independence from local appearance or semantic purity.

To quantify the split, we use centered explained variance,

$$\rho_{\mathrm{lin}} = 1 - \frac{\|Z_{\mathrm{res}}\|_F^2}{\|Z - \mathbf{1}\bar{z}^\top\|_F^2},$$

where $\bar{z}$ is the mean token in the fitted batch. The recorded per-batch residual norms give approximately 2.7% on dev-200 and 2.1% at full scale. The batch mean is also removed, but does not count toward this centered score. A dataset-level fit gives 1.9% and 1.3%, respectively, measured on its fitting collection rather than held-out tokens.

Refitting the coefficients each batch also changes the target. Across batch recompositions, per-token target-direction cosine is about 0.85 on dev-200 and 0.76 at full scale. The per-batch and dataset-level coefficient matrices have cosine about 0.90 and 0.85. These checks quantify the variation rather than establish that it is negligible.

The shared mean token carries 4.9% of uncentered target energy on dev-200 and 2.6% at full scale. These energy fractions use a different denominator from centered explained variance. The distinction matters when comparing mean removal with linear residualization.

For Figure 2(a), the affine fit uses all 64 images in the first batch of the full-scale purifier-validation permutation. Two examples are selected for legibility. After the split, each component is centered within each image only for display. Maps show cosine to patch $(7, 7)$ on the zero-indexed token grid with a shared $[-1, 1]$ scale. These maps are illustrative, and the FID values summarize separate dataset-level generation evaluations.

D.3 Separating Rank, Instability, and Predictable Content

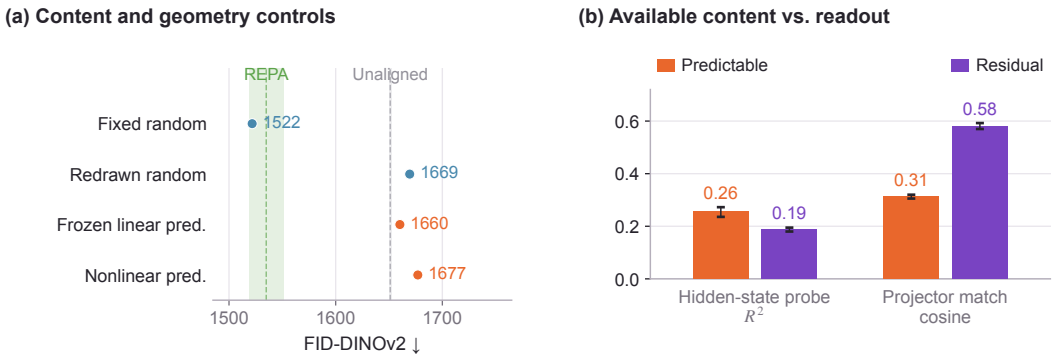


Figure 8: **Reconstruction content is easy to recover but ineffective for alignment**. (a) Dev-200/50K controls (single runs). Baseline lines and bands show two-seed means and ranges. (b) Predictable content is more decodable, while the projector matches the residual more closely (95% image-bootstrap intervals). One 400K REPA checkpoint, mid-noise, disjoint fit/evaluation images (Appendices C and D).

Figure 8(a) summarizes four single-run development controls. A fixed random rank-17 projection of DINOv2 gives 1521.7, close to the REPA seed range. A rank-matched projection can therefore remain effective. Because it is a projection of image features, this control retains image-dependent information.

Redrawing the projection each batch gives 1669.4, close to the unaligned baseline. Target instability can therefore remove the benefit. However, the frozen dataset-level reconstruction predictor gives 1660.2, and the nonlinear predicted component $d_\phi(X_p)$ gives 1677.0. Both reconstruction-based targets remain ineffective without per-batch coefficient fitting. These controls weaken explanations based solely on rank or refitting noise.

At full scale, the redrawn random control gives 937.9 in one run, close to the unaligned 945.7/947.4 pair. Its 100K–400K trajectory is 1273.8/1099.2/995.3/937.9. Unlike the predictable-only target, it does not show the same worse-than-unaligned endpoint. Mean-only removal and a linear split applied after LAP-N are reported in Table 28.

D.4 Where the Predictable-Only Target Becomes Harmful

At development scale, the predictable-only target loses the REPA benefit but remains near the unaligned model. At full scale, both predictable-only runs (1000.6/995.2) are worse than both unaligned runs (945.7/947.4). The mean proxy gap is about 51 points. Standard FID-50K agrees: 56.71 versus 53.92. The predictable-only target also has the lowest precision and recall in Table 21.

The guided check gives the same direction (29.92 vs. 27.61, seed 0). The claim of harm below the unaligned baseline is therefore scoped to the full-scale evaluations. The development controls establish loss of alignment benefit, not harm in every training setting.

E Purifier Training, Validation, and Ablations

LAP-L uses the batchwise affine fit in Appendix D. This appendix describes LAP-N's frozen nonlinear predictor, its validation and selection, and its interaction with other alignment transformations. Both variants subtract local predictions, but their fitting protocols and predictor inputs differ.

E.1 Architecture and Validation Criteria

The predictor operates on the 16×16 patch grid. It consists of a $k \times k$ convolution from 16 to 512 channels, GELU, a 1×1 convolution from 512 to 512 channels, GELU, and a final 1×1

convolution to the encoder dimension D. We fit it with mean-squared error for 20K steps using batch 256, AdamW at learning rate 10^{-3}, and a cosine learning-rate schedule. We use weight decay 10^{-4} and 500 warmup steps. A seed-0 permutation reserves 4,096 images for validation and fits the predictor on the remaining images. Fitting takes approximately two A100-hours per predictor. Locality refers to the latent-token grid, not a strict receptive-field bound in image pixels.

On held-out data, we define the explained-variance score

$$\rho = 1 - \frac{\mathbb{E}\|z - d_\phi(X_p)\|^2}{\mathbb{E}\|z - \bar{z}\|^2}. \tag{62}$$

This score measures prediction error relative to a constant mean-token predictor. It is an operational measure of removable local predictability, not a semantic-content estimate.

We assess class-information retention with linear probes fit on 21,000 images and evaluated on the 4,096 images held out from purifier fitting. We test both mean-pooled features and token features, averaging eight token logits per image for the latter. The fit and evaluation images are disjoint.

The implemented selection rule compares each residual to the linear-residual target and to the component removed by the predictor. Let a_{lin} be the pooled linear-residual accuracy and $n_v = 4096$. The tolerance is $\delta = 2\sqrt{a_{\text{lin}}(1 - a_{\text{lin}})/n_v}$. A receptive field is feasible when, for both pooled and token probes, its residual accuracy is at least the corresponding linear-residual accuracy minus δ, and exceeds the predicted-component accuracy by more than δ. We choose the largest feasible tested field. This tolerance is a selection heuristic, not a significance test, and the probes do not certify complete semantic preservation.

We separately monitor the training-to-validation gap in ρ to check prediction generalization. Table 13 reports the pooled accuracies and these gaps.

Table 13: **Purifier validation on dev-200.** Class accuracy is held-out pooled top-1 accuracy over 200 classes. Chance accuracy is 0.005. The last column is the training-to-validation gap in ρ. Bold marks the higher classification accuracy between the raw and purified targets in each row.

Encoder	k	ρ	Raw accuracy	Purified accuracy	ρ gap
MAE ViT-L	1	0.178	0.661	**0.693**	≤ 0.003
	3	0.314	0.661	**0.688**	≤ 0.003
	5	0.352	0.661	**0.687**	≤ 0.003
DINOv2 ViT-B	3	0.125	**0.798**	0.794	≤ 0.003
CLIP ViT-L	3	0.074	0.787	**0.791**	≤ 0.014
	5	0.084	0.787	**0.790**	≤ 0.014

All listed predictors pass the class-probe feasibility rule, and their training-to-validation gaps in ρ are below 0.014. At full ImageNet-1k scale, ρ is 0.344 for MAE at $k = 5$, 0.100 for DINOv2 at $k = 3$, and $0.063/0.079$ for CLIP at $k = 3/5$. MAE has the largest locally predictable fraction at both scales.

E.2 Visualizing the MAE Target Transformation

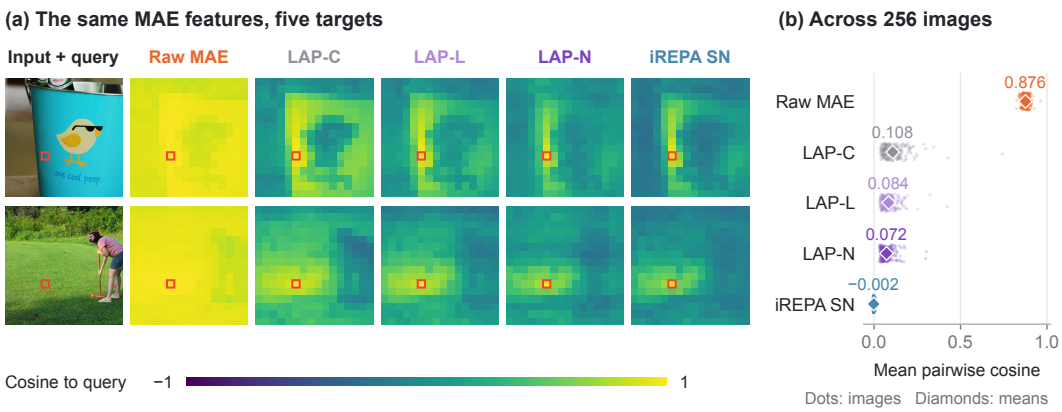


Figure 9: **How purification changes the MAE target.** (a) Cosine similarity to the fixed red-boxed query patch for the raw features, the three LAP variants, and the iREPA-style spatially normalized target (SN), on a shared scale. (b) Mean off-diagonal patch cosine over 256 held-out images. All LAP variants remove MAE's high common similarity, and the residual similarity decreases with purifier capacity; per-image spatial normalization gives a near-zero value in this evaluation.

Figure 9 uses the frozen full-scale MAE encoder and its $k = 5$ purifier. We take the first 256 images from the purifier's existing held-out permutation (seed 0), without selecting images by feature quality. The two displayed examples are chosen for image legibility and because the three variants visibly differ on them; their query is at row 10, column 5 of the 16×16 token grid, using zero-based indices, one of five fixed query positions recorded for every image. All aggregate measurements include the full 256-image subset and do not depend on this choice.

We compare raw tokens z with the three LAP targets, each formed as in training: LAP-C, $z - \bar{z}_B$, with the mean $\bar{z}_B$ taken over all images and tokens in consecutive batches of 64; LAP-L, the per-batch least-squares affine residual of z on the patchified clean latent over the same batches; and LAP-N, $z - d_\phi(X_p)$. LAP-L and LAP-N use a sampled clean VAE latent with posterior seed 916. Each heatmap shows cosine similarity between the query token and every patch token. We use a common $[-1, 1]$ color scale and do not subtract per-image means, since doing so would conceal the shared component being examined.

For each image, we also summarize similarity over all distinct token pairs:

$$s(y) = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{\langle y_i, y_j \rangle}{\|y_i\| \, \|y_j\|}, \qquad N = 256.$$

The image-averaged scores are 0.876 for raw MAE, 0.108 for LAP-C, 0.084 for LAP-L, and 0.072 for LAP-N. Thus, centering already accounts for most of the change in common similarity, and the image-dependent predictors reduce it further in order of capacity. This statistic describes target geometry and is not a semantic-quality or generation metric. The iREPA-style spatially normalized target, which standardizes each image across its own tokens per channel (Appendix E.4), scores -0.002. Spatial centering makes the unnormalized token mean zero, but does not in general make the mean of unit-normalized tokens zero. Thus, the near-zero pairwise cosine is an empirical observation, and this statistic alone does not identify which target transformation best supports generation. The two operations differ in what they remove, a per-image mean and scale versus locally predictable content, and their composition is evaluated in Table 2.

As a separate check, we fit a constant mean-token readout on a disjoint 512-image subset of the purifier's training split. Its evaluation cosine is 0.929 against raw MAE and approximately zero against LAP-L (-0.002) and LAP-N (-0.005). These values corroborate the loss of a strong shared direction after purification. This fresh evaluation uses its own image subset and should not be conflated with the trained projector's cosine or with the earlier constant-readout protocol in Appendix C. All measurements use existing frozen checkpoints and require no new diffusion training.

Where the shared direction lives in the trained model. Figure 10 applies the same query-patch visualization to the denoiser itself: the block-8 hidden state h_t and the projector output $a_t = g_\psi(h_t)$ at $t = 0.5$, one shared noise draw, for the 400K seed-0 checkpoints of the unaligned model, raw MAE, LAP-N (MAE), REPA (DINOv2), and LAP-L (DINOv2), on the same 256 held-out images. Table 14 gives the image-averaged pairwise cosine $s(\cdot)$ of h_t, of a_t, and of each model's own training target, together with the cosine between a_t and the projector's bias vector. The hidden states of all five models have similar statistics (0.62–0.76), including the unaligned model. Raw MAE produces high projector-output similarity (0.95) and a cosine of 0.74 between a_t and the projector's bias. After LAP-N, the bias cosine is -0.004 and projector-output similarity is much lower (0.14). These observations are consistent with a substantial shared component in the raw-MAE readout, as suggested by Section 4.3, but do not establish that it is a constant function or that the bias causes the generation gap.

Table 14: **Patch similarity inside the trained models.** Image-averaged mean pairwise cosine $s(\cdot)$ over 256 held-out images at $t = 0.5$ (block 8, 400K, seed 0), and the cosine between the projector output and the projector's bias vector.

Model	$s(h_t)$	$s(a_t)$	s(target)	$\cos(a_t, \text{bias})$
No alignment	0.618	0.648	–	untrained
Raw MAE	0.716	0.948	0.876	0.737
LAP-N (MAE)	0.764	0.137	0.072	−0.004
REPA (DINOv2)	0.690	0.465	0.281	0.192
LAP-L (DINOv2)	0.690	0.410	0.251	−0.007

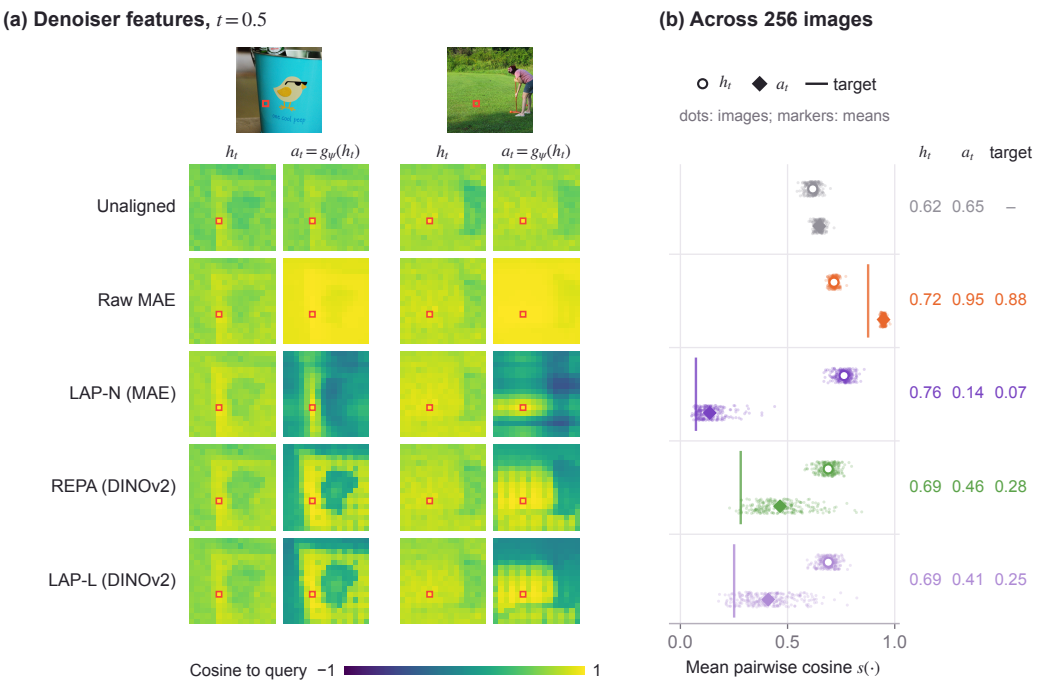


Figure 10: **The denoiser's features under raw and purified targets.** (a) Cosine to the red-boxed query patch for the block-8 hidden state h_t and the projector output a_t at $t = 0.5$, two held-out images, five 400K seed-0 models. (b) Mean pairwise cosine of h_t (circles), a_t (diamonds), and the model's own target (bars) over 256 images. Raw MAE has the highest projector-output similarity.

E.3 Receptive-Field Selection and the CLIP Check

The selection rule chooses the largest feasible tested k, without using diffusion-generation scores. MAE was evaluated at $k \in \{1, 3, 5\}$, and the rule selects $k = 5$. DINOv2 initially used only $k = 3$. Later predictors at $k = 5, 7$ give $\rho = 0.151, 0.168$, with class accuracy near 0.794 against raw accuracy 0.798. Both pass the same feasibility rule, so the initial $k = 3$ choice should not be read as the result of a complete receptive-field search.

The later DINOv2 comparison gives mean generation scores 1532.4 and 1547.5 for $k = 3$ and $k = 5$, using four seeds each. The available runs do not establish a benefit from increasing the receptive field. CLIP was initially run at the cheaper $k = 3$, although $k = 5$ also passes the criteria. Table 15 records both choices.

Table 15: **CLIP target transformations.** Dev-200/50K FID-DINOv2. Lower is better. Means are computed before rounding the individual seed values. Bold marks the lowest FID in each column.

Target	Seed 0	Seed 1	Mean
Raw CLIP	1548.9	1553.0	1550.9
LAP-L	**1517.1**	1513.9	**1515.5**
LAP-N, $k = 3$	1537.6	**1498.1**	1517.9
LAP-N, $k = 5$	1527.7	1516.2	1522.0

The three purified CLIP means lie within 6.5 points of one another. Each improves on raw CLIP by about 29–35 points, much less than the approximately 158-point MAE improvement. The two-seed results do not resolve which CLIP transformation is best. CLIP also illustrates why ρ is not a universal ranking rule: it has a smaller predictable fraction than DINOv2 but a slightly larger observed development-scale gain.

E.4 Interaction with iREPA-Style Operations

We test whether LAP-N and operations inspired by iREPA (Singh et al., 2026) provide complementary benefits. Our control replaces the MLP's first layer with a 3×3 convolution and its remaining two layers with 1×1 convolutions, preserving depth, intermediate width, and nonlinearities. It also standardizes target features across spatial tokens within each image and feature dimension. The original iREPA head is a single 3×3 convolution. Our depth-matched control therefore studies spatial mixing and normalization under our recipe, rather than reproducing the original iREPA architecture.

The design crosses raw versus LAP-N targets with an MLP versus the combined convolutional-head and spatial-normalization operations. Table 16 reports the individual values. Each cell uses two dev-200/50K runs. We also apply the operations to the predictable-only DINOv2 target.

Table 16: **LAP-N and iREPA-inspired operations.** FID-DINOv2 for seeds 0 and 1. The right column bundles a three-layer convolutional head and spatial normalization. Bold marks the lowest FID for each encoder, head, and training seed.

Target	MLP head	Conv head + spatial norm
Raw MAE	1641.4 / 1643.2	**1508.0** / 1496.9
LAP-N (MAE) (k=5)	**1486.2 / 1482.6**	1510.7 / **1484.4**
Raw DINOv2 (= REPA)	1551.2 / **1518.6**	**1476.6 / 1504.3**
LAP-N (DINOv2) (k=3)	**1508.4** / 1536.2	1514.3 / 1523.7
Predictable-only (DINOv2)	1679.9 / 1641.6	1648.6 / 1653.8

The combined operations improve raw MAE by about 140 points, but increase the LAP-N (MAE) mean by about 13 points. For DINOv2, they improve the raw and LAP-N means by about 44 and 3 points. Defining interaction as the raw-target improvement minus the LAP-N-target improvement gives about 153 points for MAE and 41 for DINOv2. The MAE result is the clearer indication of overlapping benefits. The smaller DINOv2 differences remain sensitive to seed variation.

Mean-token removal alone (LAP-C) improves MAE by about 94 points. The combined operations improve raw MAE by about 140 points, compared with about 158 for LAP-N. These comparisons are consistent with shared effects on easy-to-match content. They do not identify which operation causes the remaining improvement or prove a common mechanism.

On the predictable-only target, the combined operations give a mean of 1651.2. Spatial normalization alone gives 1665.4/1650.1, or 1657.8 on average. Both remain close to the unaligned mean of 1651.2. Thus, these operations do not recover a clear benefit from the predictable-only target in this development setting.

A later four-seed comparison gives 1506.0 for raw DINOv2 with the combined operations and 1532.4 for LAP-N (DINOv2) alone. This comparison does not resolve their ordering. We consequently treat the MAE interaction as the stronger observation and do not claim that LAP-N supersedes iREPA. These are dev-200/50K comparisons. At full scale, adding the iREPA-style operations to LAP-N (MAE) improves mean FID-50K from 42.30 to 41.32 (Table 2), so the interaction depends on the training setting.

E.5 A Check on Relational Structure

We additionally compare pairwise patch-token cosine matrices on 512 development images. For each image, center the tokens of the raw target, purified target, and clean latent separately, then form their cosine matrices. Let s_z, s_y, and s_X contain the upper-triangular off-diagonal entries of these matrices. Regress both s_z and s_y on $[\mathbf{1}, s_X]$ and compute the Pearson correlation between their residuals. We average that correlation across images. This removes linear dependence on the latent-patch similarity vector, not all appearance information.

The images are the last 512 entries in dataset order and were not explicitly excluded from purifier fitting. This is a descriptive relational-retention diagnostic, not held-out validation or iREPA's spatial-structure metric. The recorded DINOv2 retention scores are $0.97/0.96/0.96$ for $k = 3/5/7$. MAE scores range from 0.83 to 0.93 across its grid.

The larger structural change occurs for MAE, which also benefits more from purification. Under this diagnostic, structural change therefore does not track generation harm across the tested targets. This does not rule out a role for spatial structure. The measurement and the intervention probe different aspects of the representation.

F Robustness to Batch Size, Capacity, and Guidance

The base result is specific to SiT-B/2 trained with batch 64. We test larger batches and models to determine which conclusions extend beyond that setting. Table 17 separates the development proxy, full-scale proxy, standard FID, and guided FID into different blocks. Development comparisons use two-seed means. Full-scale extensions and guidance checks use seed 0.

Table 17: **Generation quality across batch size, model capacity, and guidance settings.** Metrics are separated into blocks because FID-DINOv2 and FID-50K are not numerically comparable. Lower is better. The development results average two seeds, while the full-scale and guidance checks use seed 0. Bold marks the lowest FID within each row.

Configuration	Unaligned	Raw MAE	REPA	LAP-N (MAE)
Dev-200/50K: FID-DINOv2 (two-seed means)				
SiT-B/2, batch 64	1651.2	1642.3	1534.9	**1484.4**
SiT-B/2, batch 256	1456.8	1421.5	**1226.7**	1228.9
SiT-L/2, batch 64	1477.9	1466.6	**1174.6**	1215.0
ImageNet-1k/400K: FID-DINOv2 (seed 0)				
SiT-B/2, batch 256	746.7	718.3	**623.7**	630.3
SiT-L/2, batch 64	740.9	678.1	**472.6**	520.2
SiT-L/2, batch 256	488.4	410.8	**305.6**	334.5
ImageNet-1k/400K: FID-50K (seed 0)				
SiT-B/2, batch 256	36.38	34.46	**27.96**	28.98
SiT-L/2, batch 64	38.44	33.97	**19.09**	24.09
SiT-L/2, batch 256	20.33	15.39	**9.64**	12.06
Guided FID-50K: CFG 1.5 (seed 0)				
SiT-B/2, batch 64	27.61	26.83	**18.02**	18.27

F.1 Larger Batch and Model

SiT-B/2, batch 256. LAP-N (MAE) remains close to REPA at both scales. At full scale, standard FID is 28.98 versus 27.96, a gap of 1.02. The corresponding proxy gap is 6.6. Raw MAE also improves over the unaligned model in this setting. Its lack of benefit in the base configuration should not be generalized to every training recipe.

SiT-L/2. LAP-N improves MAE at both tested batch sizes, but a gap to REPA remains. At batch 64, the standard FID values are 38.44 for unaligned, 33.97 for raw MAE, 24.09 for LAP-N (MAE), and 19.09 for REPA. At batch 256, the corresponding values are 20.33, 15.39, 12.06, and 9.64.

For a common baseline, define the fraction of the REPA gain recovered by LAP-N as

$$G = \frac{\mathrm{FID}_{\mathrm{unaligned}} - \mathrm{FID}_{\mathrm{LAP-N}}}{\mathrm{FID}_{\mathrm{unaligned}} - \mathrm{FID}_{\mathrm{REPA}}}. \tag{63}$$

LAP-N (MAE) recovers about 74% of the standard-FID gain on SiT-L/2 at batch 64 and 77% at batch 256. Proxy-based fractions are about 82% and 84%. These are descriptive summaries of single-seed runs, not confidence bounds.

The larger-model results also show a quality tradeoff. REPA has higher Inception Score than LAP-N (MAE) at both batch sizes (71.0 vs. 56.5, and 114.1 vs. 96.1). We therefore describe the result as substantial recovery, rather than parity across metrics.

F.2 Alignment Depth

The base SiT-B/2 recipe aligns at block 8. For SiT-B/2, Singh et al. (2026) report better REPA results at block 4, and REG and sREPA use depth 4 in their original settings. We compare block 4 with block 8 on ImageNet-1k at batch 64, using two seeds, the raw target and the best LAP variant from Table 1, and REG with and without LAP-L. Table 18 reports Inception FID-10K at 100K steps and available FID-50K evaluations at 400K steps. Block 4 improves each raw target with a reported 400K comparison: MAE 53.1 → 50.5, DINOv2 42.1 → 36.3, DINOv3 42.7 → 37.4, and REG 29.2 → 27.1. The MAE LAP-N gain persists (11.7 versus 10.8 FID). At 100K, AIMv2 and CLIP show larger LAP gains at block 4 than at block 8 (8.0 and 7.0 versus 4.3 and 4.5 FID-10K). For DINOv2, DINOv3, and REG, the block-4 LAP changes are small (+0.5, −0.4, and +0.2 FID-50K). The gap between purified MAE and DINOv2-based REPA is 0.2 at block 8 under the same evaluation draw and 2.5 at block 4. At 100K, their block-4 FIDs are close (71.5 versus 71.7).

Block-4 MAE, CLIP, and AIMv2 training uses H100 GPUs, DINOv3 and REG use A100s, and DINOv2 uses H100/A100 GPUs. Checkpoint resumption for CLIP and AIMv2 restarts the data order. The reported 400K samples use A100s for DINOv3 and REG and H100s for MAE and DINOv2. Block-8 hardware is described in Appendix I.

Table 18: **Alignment at block 4 versus block 8.** Left: Inception FID-10K of the 100K-step checkpoints; right: FID-50K at 400K steps. Seeds 0/1 with their mean in parentheses. The block-8 DINOv2 FID-50K values are those of the original evaluation draw, which the block-4 runs share, rather than the re-drawn values of Table 1; no block-8 100K evaluation is available for REG. Bold marks the lowest reported mean per teacher and evaluation budget. A dash denotes an unavailable measurement.

		FID-10K at 100K steps		FID-50K at 400K steps	
Teacher	Target	Block 4	Block 8	Block 4	Block 8
MAE	Raw	87.26 / 87.09 (87.18)	87.83 / 86.54 (87.19)	50.54 / 50.49 (50.51)	53.68 / 52.41 (53.05)
	LAP-N	71.73 / 71.20 (**71.46**)	72.91 / 72.63 (72.77)	38.88 / 38.72 (**38.80**)	43.06 / 41.53 (42.30)
AIMv2	Raw	80.95 / 80.94 (80.95)	82.41 / 82.52 (82.47)	–	48.08 / 47.87 (47.97)
	LAP-L	72.51 / 73.43 (**72.97**)	78.98 / 77.28 (78.13)	–	44.40 / 44.22 (**44.31**)
CLIP	Raw	80.04 / 80.17 (80.11)	79.33 / 79.93 (79.63)	–	44.73 / 44.74 (44.74)
	LAP-N	72.65 / 73.59 (**73.12**)	75.67 / 74.63 (75.15)	–	42.32 / 42.11 (**42.21**)
DINOv3	Raw	73.24 / 73.74 (73.49)	75.16 / 77.61 (76.38)	37.02 / 37.83 (37.43)	42.12 / 43.34 (42.73)
	LAP-L	70.32 / 71.93 (**71.13**)	76.88 / 72.60 (74.74)	36.46 / 37.62 (**37.04**)	43.23 / 40.92 (42.07)
DINOv2	Raw (REPA)	72.59 / 70.79 (**71.69**)	75.32 / 74.74 (75.03)	37.10 / 35.45 (**36.28**)	42.42 / 41.73 (42.07)
	LAP-L	72.33 / 72.09 (72.21)	74.38 / 76.12 (75.25)	36.93 / 36.64 (36.79)	40.86 / 42.63 (41.75)
DINOv2, REG	REG	64.01 / 63.33 (63.67)	–	27.49 / 26.61 (**27.05**)	28.66 / 29.82 (29.24)
	+LAP-L	63.73 / 62.19 (**62.96**)	–	27.80 / 26.66 (27.23)	28.48 / 28.85 (28.67)

F.3 Classifier-Free Guidance

The guidance check evaluates the 400K checkpoints of the base SiT-B/2, batch-64 recipe with CFG scale 1.5 applied at every sampling step (SDE-250, 50K samples, ADM evaluator), for both training seeds (Table 19). The MAE, CLIP, and AIMv2 gains persist under guidance in both seeds (-8.5, -1.6, and -2.6 FID on average), and LAP-N (MAE) recovers 99% of the REPA gain relative to the unaligned model ($27.70 \rightarrow 17.90$ against 17.82). For DINOv2, DINOv3, and REG the guided difference changes sign between the two seeds and the means lie within 0.2 FID of each other (-0.2, -0.2, and $+0.1$), consistent with the small unguided differences. Mean recall is higher with LAP for these three pairs. The MAE, DINOv2, and unaligned rows and seed 1 of AIMv2 were sampled on A100 GPUs, the CLIP, DINOv3, REG rows and seed 0 of AIMv2 on H100 GPUs; the two GPU types lay out the random stream differently, so within each seed the two members of a pair share hardware and noise draws, whereas rows on different GPU types do not.

Table 19: **Guided sampling.** FID-50K, Inception Score, precision, and recall at CFG scale 1.5 for the 400K checkpoints; two-seed means (the predictable-only control has one seed). Shaded rows use LAP; bold marks the better value within each raw/LAP pair.

Teacher	Target	FID ↓	IS ↑	Prec. ↑	Rec. ↑
–	No alignment	27.70	53.78	0.603	0.530
MAE	Raw	26.37	57.46	0.606	**0.532**
	LAP-N	**17.90**	**79.89**	**0.676**	0.529
AIMv2	Raw	22.32	68.58	0.638	0.528
	LAP-L	**19.68**	**77.35**	**0.654**	**0.545**
CLIP	Raw	19.75	76.23	0.654	0.536
	LAP-N	**18.12**	**82.83**	**0.661**	**0.542**
DINOv3	Raw	18.42	82.29	**0.656**	0.536
	LAP-L	**18.22**	**84.34**	**0.656**	**0.553**
DINOv2	Raw (REPA)	17.82	85.97	0.665	0.538
	LAP-L	**17.60**	**86.07**	**0.666**	**0.544**
DINOv2, REG	Predictable-only (one seed)	29.92	51.78	0.574	0.531
	REG	**15.19**	**110.55**	0.662	0.539
	+LAP-L	15.31	109.02	**0.663**	**0.549**

G An Encoder-Free Alignment Control

This experiment asks whether a residual target from the diffusion model's own teacher can provide guidance comparable to a pretrained encoder. We call the residual variant alignment to residual context (ARC) and compare it with an otherwise identical raw self-teacher target.

G.1 Construction and Evaluation

The student's block-8 readout is aligned to an EMA teacher's layer-10 hidden state. The teacher processes an independently noised copy of the same image at $t = 0.25$. For ARC, we remove the part of the teacher state predicted by local clean and noised patches using an affine fit. ARC-raw skips this removal. Both variants then standardize each feature dimension across batch tokens. Residualization is the manipulated variable.

Both variants use dev-200 and batch 64. The alignment coefficient increases linearly from zero to 0.5 over the first 10K steps. We evaluate the pre-specified 25K, 50K, 75K, and 100K checkpoints. A diagnostic on unaligned checkpoints gives residual variance fractions of 0.53–0.78 and DINOv2-probe enrichment factors of 1.67 at 10K and 1.15 at 50K. These are relative diagnostics of the self-teacher, not evidence that it contains as much useful guidance as DINOv2.

Table 20: **Encoder-free target comparison.** Dev-200 FID-DINOv2, single runs. A dash denotes an unavailable measurement. Bold marks the lowest FID at each training checkpoint.

Target	25K	50K	75K	100K
Unaligned	–	1652.4	1454.3	1355.1
Raw self-teacher	2555.0	**1622.5**	**1425.3**	**1307.5**
Residual self-teacher	**2537.3**	1623.6	1429.5	1313.1

G.2 What the Comparison Establishes

In Table 20, the residual-minus-raw difference ranges from -17.7 to $+5.6$ across the trajectory, without a consistent advantage for residualization. At 100K, the raw and residual variants improve on the unaligned model by 47.6 and 42.0 proxy points. These gains are only about 21–24% of the matched REPA gain.

This control provides no clear benefit from residualizing the self-teacher under the tested recipe. It does not establish that pretrained encoders are necessary: the self-teacher may simply provide too little useful structure. Other encoder-free regularizers can improve diffusion training (Wang & He, 2025). A stronger test would match the quality of the encoder-free target to the pretrained target before comparing residualization.

H Generation Evaluation and Additional Results

H.1 Standard Evaluation and Mechanism Controls

Standard evaluation generates 50K images from each 400K EMA checkpoint using the Euler–Maruyama SDE sampler with 250 steps and no classifier-free guidance. We use the ADM/guided-diffusion evaluation suite and its ImageNet-256 reference statistics to report FID, Inception Score, precision, and recall. Base comparisons evaluate both training seeds. Means and sample SDs use two decimals for FID/IS and three for precision/recall; a displayed zero SD can reflect rounding. Larger-model and guidance checks use the coverage stated in Appendix F.

The base DINOv2 rows use a fresh, matched reevaluation of all six raw, LAP-L, and LAP-N checkpoints, with sampling seed 20260916, two GPUs, and 64 samples per GPU batch. Initial noise, class labels, and sampler random streams match across these conditions. Their mean FIDs changed from $42.07/41.75/42.25$ to $41.75/41.47/42.11$, which we use for the base comparisons. The depth comparison in Appendix F.2 retains the matched original draw. Other conditions retain their original evaluations under the same sampler and metric protocol, so cross-condition comparisons outside this reevaluated group do not share the fresh sample draw. This reevaluation adds no training seeds.

Table 21: **All LAP variants under other recipes and tokenizers.** Protocol of Table 1, mean ± sample SD over two training seeds. The first two blocks give all LAP variants behind Tables 2 and 3 (MAE under REG and sREPA was trained with LAP-N only), and the last gives the two interventions of Section 3. Under the iREPA-style operations, LAP-C equals the recipe alone, because per-image spatial standardization already removes the batch-mean token (verified numerically on the training targets). Shaded rows use LAP. Individual seeds are in Tables 29 and 30. Bold marks the best mean within each teacher–recipe or tokenizer group and metric, including ties at displayed precision.

Setting	Target	FID ↓	IS ↑	Prec. ↑	Rec. ↑
REPA-family recipes, SD-VAE					
MAE, iREPA-style	Raw	43.06 ±0.62	33.23 ±0.66	0.491 ±0.004	0.617 ±0.006
	LAP-L	43.10 ±0.40	33.07 ±0.62	0.490 ±0.003	**0.627** ±0.005
	LAP-N	**41.32** ±0.28	**34.70** ±0.37	**0.502** ±0.002	0.626 ±0.001
MAE, VA-REPA	Raw	52.82 ±0.90	26.32 ±0.58	0.437 ±0.001	0.606 ±0.006
	LAP-C	43.56 ±0.15	32.33 ±0.01	0.489 ±0.000	0.618 ±0.000
	LAP-L	43.43 ±0.63	32.65 ±0.63	0.492 ±0.003	0.620 ±0.001
	LAP-N	**41.60** ±0.91	**34.06** ±0.53	**0.503** ±0.004	**0.622** ±0.004
MAE, REG	Raw	52.92 ±0.53	26.33 ±0.41	0.434 ±0.002	0.608 ±0.002
	LAP-N	**41.65** ±0.06	**34.31** ±0.16	**0.500** ±0.003	**0.626** ±0.002
MAE, sREPA	Raw	50.43 ±0.53	27.41 ±0.22	0.454 ±0.002	0.606 ±0.009
	LAP-N	**38.46** ±0.62	**36.89** ±0.30	**0.519** ±0.003	**0.622** ±0.002
DINOv2, iREPA-style	Raw	41.32 ±0.19	35.65 ±0.01	0.496 ±0.004	**0.635** ±0.003
	LAP-L	**40.55** ±0.39	**36.29** ±0.16	**0.503** ±0.003	0.632 ±0.004
	LAP-N	41.03 ±0.65	35.70 ±0.66	0.500 ±0.002	0.633 ±0.004
DINOv2, VA-REPA	Raw	40.71 ±0.14	36.27 ±0.24	0.500 ±0.002	0.635 ±0.001
	LAP-C	41.92 ±0.92	35.11 ±0.73	0.492 ±0.006	0.629 ±0.010
	LAP-L	**40.39** ±0.02	**36.54** ±0.04	**0.502** ±0.001	**0.637** ±0.006
	LAP-N	41.89 ±0.62	35.45 ±0.72	0.496 ±0.003	0.628 ±0.004
DINOv2, REG	Raw	29.24 ±0.82	57.08 ±2.10	0.559 ±0.003	0.610 ±0.001
	LAP-C	30.02 ±0.54	54.90 ±0.82	0.559 ±0.003	**0.619** ±0.007
	LAP-L	**28.67** ±0.26	57.49 ±0.07	**0.567** ±0.005	0.615 ±0.004
	LAP-N	29.51 ±0.19	**57.64** ±0.94	0.559 ±0.003	0.607 ±0.008
DINOv2, sREPA	Raw	38.13 ±0.01	39.55 ±0.01	0.515 ±0.005	0.635 ±0.004
	LAP-C	39.02 ±1.08	38.68 ±1.65	0.510 ±0.002	**0.640** ±0.002
	LAP-L	37.63 ±0.86	39.88 ±1.11	0.518 ±0.003	0.636 ±0.001
	LAP-N	**36.92** ±0.42	**41.11** ±0.46	**0.522** ±0.002	0.631 ±0.005
Other tokenizers, DINOv2, REPA					
REPA-E VAE	No alignment	32.30 ±0.04	40.61 ±0.09	0.584 ±0.002	0.604 ±0.005
	Raw	26.10 ±0.32	52.41 ±0.72	0.611 ±0.004	**0.620** ±0.001
	LAP-C	26.43 ±0.07	51.60 ±0.25	0.609 ±0.000	0.619 ±0.003
	LAP-L	25.69 ±0.41	53.11 ±0.25	**0.615** ±0.003	0.614 ±0.000
	LAP-N	**24.94** ±0.12	**55.13** ±0.41	0.614 ±0.003	0.617 ±0.005
Other tokenizers, MAE, REPA					
REPA-E VAE	Raw	33.74 ±0.45	39.32 ±0.88	0.574 ±0.005	**0.611** ±0.003
	LAP-N	**27.56** ±0.40	**47.43** ±0.44	**0.606** ±0.003	0.607 ±0.000
EQ-VAE	Raw	52.41 ±1.37	26.31 ±0.91	0.452 ±0.004	0.597 ±0.004
	LAP-N	**40.01** ±0.25	**35.24** ±0.36	**0.523** ±0.003	**0.608** ±0.005
Interventions of Section 3, DINOv2					
REPA + prediction loss (Section 3.1)		46.79 ±0.16	29.99 ±0.27	0.473 ±0.001	0.617 ±0.007
Predictable-only target (Section 3.2)		56.71 ±0.42	24.37 ±0.00	0.415 ±0.001	0.591 ±0.003

REPA-family baselines and LAP-C. The LAP-C, iREPA-style, VA-REPA, and CLIP rows of Tables 1 and 2 were trained after the initial protocol, with the base recipe unchanged and the component settings specified below. LAP-C subtracts the per-batch mean token (Appendix E.4). The iREPA-style rows use the depth-matched control of Appendix E.4: spatial standardization of the target within each image and a three-layer convolutional head in place of the MLP; they are not the original single-convolution iREPA head. VA-REPA (Yang et al., 2026) reweights the alignment loss by $w(t) = \sigma\big(k(\tau - t)\big)$ with the authors' defaults $\tau = 0.7$, $k = 20$, normalized by the batch sum of w, and the same $\lambda_a = 0.5$; we do not include its companion Stable Velocity Matching objective, so the row isolates the alignment schedule. REPA-E (Leng et al., 2025) tunes the VAE

end to end, which our budget does not cover; its frozen end-to-end-tuned VAE is evaluated in Table 3, described next. These runs were trained on H100 and A100 GPUs with the dataset staged on a local network file system, whereas the earlier rows used A100s; the recipe and evaluation are otherwise identical, and the rows use the original evaluation protocol, not the fresh sample draw of the DINOv2 reevaluation. Individual FIDs are in Table 29. The composition rows apply those operations or that schedule to a LAP target with no other change; the raw-target VA-REPA row for MAE and the DINOv2 composition rows were trained in the same wave. Table 2 reports the evaluated variant with the lowest two-seed mean FID within each recipe. Table 3 uses LAP-N for MAE with every tokenizer. The initial composition runs used each teacher's best variant under plain REPA; the other variants in Table 21 were trained later. Individual seeds: LAP-N (MAE) with the VA-REPA schedule $42.24/40.95$, with the iREPA-style operations $41.12/41.52$; raw MAE with the VA-REPA schedule $53.46/52.19$; LAP-L (DINOv2) with the iREPA-style operations $40.28/40.83$ and with the VA-REPA schedule $40.37/40.40$. **Frozen REPA-E VAE.** The REPA-E VAE rows of Tables 3 and 21 replace the SD-VAE by the end-to-end-tuned VAE released with REPA-E (Leng et al., 2025), used frozen: latents are encoded online during training and normalized with the released per-channel statistics, and samples are decoded with the same VAE. Everything else follows the base recipe ($\lambda_a = 0.5$, 400K steps, ADM evaluation). With the DINOv2-B target (Table 21), the unaligned seeds are $32.27/32.33$, the REPA seeds $25.88/26.33$, and the LAP-L seeds $25.98/25.39$, so REPA and LAP-L are within seed variation of each other on this tokenizer as well. With the MAE ViT-L target on the same frozen VAE (Table 3) (purifier refitted on its latents, $k = 5$, held-out $\rho = 0.357$), the raw-target seeds are $33.42/34.06$ and the LAP-N seeds $27.84/27.28$; these runs were trained with two-GPU data parallelism (per-GPU batch 32) and sampled on two GPUs (Appendix I).

Frozen EQ-VAE. The EQ-VAE rows of Table 3 replace the SD-VAE by the equivariance-regularized fine-tune of Kouzelis et al. (2025), frozen, using the 5-epoch OpenImages checkpoint on which that paper's SiT-B/2 results are based. Per-channel latent means and standard deviations were measured on $8{,}192$ training images (the released configuration's scalar scaling factor is the SD-VAE default and does not match the measured statistics); latents are encoded online and samples decoded with the same VAE; the released weights carry no declared license. With the MAE ViT-L target the LAP-N purifier was refitted on EQ-VAE latents ($k = 5$, held-out $\rho = 0.343$). Seeds: raw target $51.44/53.38$, LAP-N $40.19/39.83$; the unaligned seeds are $52.47/52.53$. These MAE runs were trained with data parallelism on H100s at the same global batch of 64 (two GPUs, and four GPUs for the last 100K and 250K steps of the two LAP-N seeds), resuming from checkpoints after interruptions, and sampled on two A100s (raw target) or four H100s (LAP-N) (Appendix I).

REG and sREPA. REG (Wu et al., 2025) prepends a noised DINOv2 CLS token to the latent token sequence, denoises it jointly with the image tokens (loss weight $\beta = 0.03$, the authors' default), and aligns the resulting 257-token sequence; the CLS token is projected by a linear layer followed by an affine LayerNorm and is discarded at sampling time. Our reimplementation follows the official code. sREPA (Xu et al., 2026) adds the mean squared error between the off-diagonal entries of the token self-similarity matrices of the ℓ_2-normalized projected features and of the target, with the authors' defaults $\lambda_{\text{proj}} = 1.0$ and $\lambda_{\text{struc}} = 2.0$; no official code was available, and our implementation was checked numerically against the paper's Algorithm 1. In the LAP arms the structural loss is computed on the LAP target; the CLS row of REG is not purified, because it has no co-located latent patch. For MAE, whose forward pass drops the class token, the REG target uses MAE's own post-LayerNorm CLS feature; unlike the DINOv2 CLS token it is nearly constant across images (mean pairwise cosine 0.98 vs. 0.03), which is consistent with REG giving raw MAE no gain. Both recipes use our alignment depth (block 8) and batch size rather than the authors' (depth 4, batch 256), so absolute FIDs differ from the published ones. Individual seeds: DINOv2 with REG $28.66/29.82$, with LAP-L $28.48/28.85$; DINOv2 with sREPA $38.14/38.13$, with LAP-L $37.03/38.24$; MAE with REG $53.29/52.54$, with LAP-N $41.70/41.61$; MAE with sREPA $50.06/50.80$, with LAP-N $38.90/38.02$.

Additional teachers. Six encoders were screened by C_0 on dev-200 (2,048 images, the protocol of Appendix C.8) before any training: AIMv2-L (Fini et al., 2025) 0.66, SigLIP 2-B (Tschannen et al., 2025) 0.57, DINOv3-B (Siméoni et al., 2025) 0.52, EVA-02-B (Fang et al., 2024) 0.44, PE-Core-B (Bolya et al., 2025) 0.44, and Web-SSL DINO-300M (Fan et al., 2025) 0.37. Each yields 256 patch tokens for a 256×256 input after class and register tokens are dropped, with its own input normalization. By a rule fixed before the runs, DINOv3 and the highest-C_0 candidate, AIMv2, were trained at full scale with the raw target and with LAP-L, two seeds each (Table 1); LAP-L needs

no per-teacher purifier. LAP-C and LAP-N runs, the latter with purifiers selected by the rule of Appendix E, were added afterwards to complete the table. DINOv3 weights are the timm port of the official LVD-1689M checkpoint; AIMv2 is released under a research-only license. Seeds: DINOv3 $42.12/43.34$, with LAP-L $43.23/40.92$; AIMv2 $48.08/47.87$, with LAP-L $44.40/44.22$.

Table 21 completes the base-setting results with the remaining LAP variants and the two mechanism interventions of Section 3. The auxiliary prediction loss improves velocity prediction but worsens generation relative to REPA on FID, Inception Score, precision, and recall. The predictable-only target is worse than the unaligned baseline on standard FID and has the lowest precision and recall of all base-setting runs.

H.2 Pre-Specified Encoder Sweep and an External Check of C_0

Motivation. The five teachers of Table 1 order the LAP-L gain by raw C_0, but C_0 and raw-target quality are correlated under the paper's own hypothesis, since a large shared direction is what makes a raw target easy to match without image-specific computation. Five points cannot separate the two. We therefore (i) test C_0 against the published results of Singh et al. (2026) and (ii) run a pre-specified sweep of eight further encoders.

External check. We measured C_0 (dev-200, protocol of Appendix C.8) for 37 encoders, including every encoder of Singh et al. (2026) that we could load (Table 22 lists those used below; the full screen is released with the code). Against their published SiT-XL/2 results at 100K steps (25 encoders; their appendix bar-chart labels, two of which differ slightly from their tables), raw C_0 has Spearman correlation 0.81 with REPA FID and 0.70 with the gain from REPA to iREPA (both $p < 10^{-3}$), but the partial rank correlation of the gain with C_0 given REPA FID is -0.13 ($p = 0.55$). Their SiT-B/2 400K table (16 encoders) gives 0.54, 0.49, and 0.23 ($p = 0.03$, 0.06, 0.38), and their SiT-L/2 table (8 encoders) 0.25, 0.24, and 0.10. The shared direction thus predicts which teachers align poorly and gain most from their spatial normalization, but in published data it cannot be separated from teacher quality. Their gain also includes a convolutional projector, which their ablation credits with 0.5–7.4 FID on its own.

Sweep design. Eight encoders were selected by a rule recorded internally before any of their generation results existed (`figs/data/prereg/prereg_sweep_sep23.md` and its amendment, released with the code): candidates were stratified by raw C_0 into four bins with two encoders each, preferring, within a bin, encoders off the C_0-versus-quality diagonal (using iREPA's REPA FID as the quality proxy) and then size variants of families already in Table 1; encoders above one billion parameters were excluded for cost. Each is trained with the raw target and with LAP-L, which needs no fitted purifier, for 100K steps on ImageNet-1k with the base recipe and two seeds, and evaluated with the Inception FID-10K protocol of Figure 5, so that the five teachers of Table 1 enter the same comparison through their 100K checkpoints. Pre-specified predictions: the LAP-L gain increases with raw C_0 (above 3 FID for $C_0 > 0.7$, below 1 FID for $C_0 < 0.35$); as a secondary predictor, the gain follows $\Delta C_0 = C_0(\text{raw}) - C_0(\text{LAP-L})$, the shared direction that LAP-L actually removes, which matters for CLIP-B/16, whose LAP-L target retains $C_0 = 0.25$; and PE-Core-L, the designated off-diagonal encoder (low C_0, poor published REPA FID), gains less than the median encoder despite its poor raw FID. B-size encoders were trained on V100 GPUs and the others on A100/H100 GPUs; both seeds of an encoder share a GPU type. **Result.** The eight added encoders contribute 32 runs (Table 22). Across the 13 encoders, the LAP-L gain has Spearman correlation 0.94 with raw C_0 (permutation $p < 10^{-4}$), 0.85 with ΔC_0, and 0.80 with the raw-target FID; C_0 and raw FID are themselves correlated (0.90). The partial rank correlation of the gain with C_0 given raw FID is 0.83, whereas that of the gain with raw FID given C_0 is -0.26: the association with the shared direction persists after controlling for raw-target FID. The pre-specified thresholds held for five of six encoders: the three encoders with $C_0 > 0.7$ gain 11.5, 10.7, and 4.5 FID, and two of the three with $C_0 < 0.35$ gain 0.7 and -0.2, while DINOv2-L gains 1.6. The off-diagonal test came out as predicted: PE-Core-L has a poor raw-target FID (82.0) yet no gain (0.0; $+1.7$ and -1.7 in the two seeds) and ranks 12th of 13. The same order holds within families: DINOv3-H+, -B, and -L (C_0 of 0.68, 0.53, and 0.48) gain 3.2, 1.6, and 0.7; the three DINOv2 sizes (0.22–0.25) gain between -0.2 and 1.6; MAE-B and MAE-L (0.94 and 0.93) gain 11.5 and 10.7. CLIP-B/16 retains $C_0 = 0.25$ after LAP-L yet gains 4.5, so the secondary predictor ΔC_0 did not improve on raw C_0. Seed-to-seed differences in the gain reach 6.7 FID for DINOv3-B and 3.4 for PE-Core-L, so individual gains below about 2 FID are within noise, and the evidence is the ordering, not any single encoder.

Table 22: **Encoder sweep at 100K steps.** Dev-200 C_0 of the raw and LAP-L targets, and Inception FID-10K of the 100K-step checkpoints with the raw and the LAP-L target; two-seed means, with the gain of each seed in the last column. Rows are ordered by raw C_0; † marks the five teachers of Table 1, evaluated at their 100K checkpoints. Bold marks the lower FID between raw and LAP-L targets within each row, including ties at displayed precision.

Encoder	C_0 raw	C_0 LAP-L	Raw	LAP-L	Gain	Gain by seed
MAE ViT-B	0.938	0.022	86.94	**75.48**	11.47	12.03 / 10.90
MAE ViT-L†	0.933	0.020	87.19	**76.53**	10.66	11.67 / 9.63
CLIP ViT-B/16	0.704	0.246	83.46	**78.92**	4.53	5.10 / 3.97
DINOv3 ViT-H+	0.682	0.033	80.47	**77.27**	3.20	4.13 / 2.28
AIMv2 ViT-L†	0.663	0.041	82.47	**78.13**	4.34	3.43 / 5.24
CLIP ViT-L†	0.607	0.088	79.63	**75.65**	3.98	4.05 / 3.90
PE-Spatial ViT-B	0.575	0.017	76.79	**75.13**	1.66	1.48 / 1.84
DINOv3 ViT-B†	0.525	0.018	76.38	**74.74**	1.64	−1.72 / 5.01
DINOv3 ViT-L	0.476	0.028	76.30	**75.63**	0.67	0.01 / 1.33
PE-Core ViT-L	0.403	0.030	**82.03**	**82.03**	0.00	1.68 / −1.68
DINOv2 ViT-L	0.254	0.009	76.05	**74.50**	1.56	2.72 / 0.39
DINOv2 ViT-S	0.244	0.009	74.83	**74.15**	0.68	0.54 / 0.82
DINOv2 ViT-B†	0.221	0.007	**75.03**	75.25	−0.22	0.94 / −1.38

Best-variant visualization. Figure 6 summarizes the purification gain for each encoder at 100K steps using FID-10K. For the five teachers in Table 1, we plot the best evaluated LAP variant by mean FID over two training seeds: MAE-L and CLIP-L use LAP-N, while AIMv2-L, DINOv3-B, and DINOv2-B use LAP-L. The eight additional encoders use LAP-L. Within each encoder, the raw and LAP targets share the same training and evaluation protocol. The dashed monotone quadratic curve is shown only as a descriptive trend, not a calibrated predictor. The reported Spearman correlation of 0.94 and partial rank correlation of 0.83 are computed from the matched fixed-LAP-L comparison across all 13 encoders in Table 22.

H.3 Checking the Development Metric

We also evaluate the development samples in InceptionV3 pool3 feature space. This check uses the same 10K generated images and development reference set as the proxy evaluation. It is therefore distinct from standard ImageNet-1k FID-50K.

Across the 13 recorded runs in Table 23, representing 11 configurations, the two feature spaces have Spearman rank correlation about 0.92. Both show a large improvement from purifying MAE, while raw MAE remains close to the unaligned model. They need not agree on the ordering of close methods. The predictable-only target remains near the unaligned baseline at this scale.

This check addresses feature-space sensitivity, not independent training replication. Re-evaluating the DINOv2 proxy with fresh sample draws changes individual scores by about 1–15 points, with a median change of 7.

Table 23: **Development results in two feature spaces.** Dev-200/50K, identical generated samples within each row. Rows are sorted by the Inception-space score. Each row is one evaluated checkpoint. Bold marks the lowest FID in each feature space among the listed checkpoints.

Model	FID-Inception (dev)	FID-DINOv2
LAP-N (MAE) (k=5)	**106.98**	**1486.2**
LAP-L (MAE)	107.55	1504.3
REPA, seed 1	109.94	1518.6
LAP-N (DINOv2) (k=3)	110.09	1508.4
LAP-N (CLIP) (k=3)	110.42	1537.6
REPA, seed 0	110.61	1551.2
LAP-L (CLIP)	110.72	1517.1
LAP-L (DINOv2)	111.87	1546.0
REPA, repeat	112.30	1553.3
Raw CLIP	114.64	1548.9
Raw MAE	122.44	1641.4
Unaligned	124.72	1652.4
Predictable-only (DINOv2)	125.01	1679.9

H.4 Complete Training Trajectories

The Inception FID-10K curves in Figure 5 use the ADM sampler and reference statistics of Table 1, with 10K generated samples per checkpoint. The scores use no guidance and are not numerically interchangeable with FID-50K. At 400K steps, they lie about 2–3 points above FID-50K and preserve the main ordering.

Inception FID across training. LAP-L improves MAE, AIMv2, and CLIP from the first evaluated checkpoint. At 100K steps, their two-seed mean FID-10K gains are 10.7, 4.3, and 4.0, respectively. LAP-N with MAE is below DINOv2-based REPA at 100K (72.8 vs. 75.0) and matches it at 400K (45.1 vs. 45.1), whereas raw MAE tracks the unaligned model throughout. These are gains per training step, not measured savings in total compute, which also includes fitting and evaluating the purifier (Appendix I).

Table 24: **Complete Inception FID-10K trajectories.** ImageNet-1k, SiT-B/2, batch 64, no guidance. Mean ± sample SD over two training seeds at every checkpoint. This table includes DINOv2 and the additional LAP variants omitted from the compact main-text curves.

Teacher	Target	100K	200K	300K	400K
No alignment	–	88.47 ±0.60	70.79 ±1.56	62.01 ±1.02	56.61 ±0.26
MAE ViT-L	REPA	87.19 ±0.91	69.10 ±1.03	60.87 ±1.34	55.54 ±1.20
MAE ViT-L	LAP-L	76.53 ±0.53	60.34 ±0.18	52.51 ±0.37	48.01 ±0.09
MAE ViT-L	LAP-N	72.77 ±0.19	56.55 ±0.51	49.18 ±0.47	45.05 ±1.32
AIMv2 ViT-L	REPA	82.47 ±0.08	64.33 ±0.17	56.03 ±0.08	50.73 ±0.27
AIMv2 ViT-L	LAP-L	78.13 ±1.20	60.28 ±0.12	52.41 ±0.19	47.19 ±0.34
CLIP ViT-L	REPA	79.63 ±0.42	60.94 ±0.41	52.36 ±0.09	47.43 ±0.09
CLIP ViT-L	LAP-L	75.65 ±0.53	57.60 ±0.87	50.05 ±1.14	45.33 ±0.99
CLIP ViT-L	LAP-N	75.15 ±0.74	57.81 ±0.57	50.05 ±0.30	44.85 ±0.57
DINOv3 ViT-B	REPA	76.38 ±1.73	58.37 ±0.57	49.98 ±0.57	45.70 ±0.96
DINOv3 ViT-B	LAP-L	74.74 ±3.03	57.67 ±1.74	49.35 ±1.63	44.85 ±1.67
DINOv2 ViT-B	REPA	75.03 ±0.41	57.90 ±0.37	49.18 ±0.32	45.05 ±0.62
DINOv2 ViT-B	LAP-L	75.25 ±1.24	57.63 ±1.50	49.20 ±1.46	44.82 ±0.95
DINOv2 ViT-B	LAP-N	76.61 ±0.76	58.33 ±1.18	50.04 ±0.33	45.02 ±0.34

Figure 5 compares REPA with one LAP variant for MAE, AIMv2, CLIP, and DINOv3. LAP-N is used for MAE and CLIP and LAP-L for AIMv2 and DINOv3, selected by the final FID in Table 1 and held fixed across checkpoints. Table 24 retains all five teachers, all evaluated LAP variants, and the unaligned reference. Table 25 lists the FID-DINOv2 proxy at all evaluated intermediate checkpoints for the original runs. LAP-N (MAE) has lower proxy values than REPA at the three intermediate steps in both matched-seed comparisons. The unaligned and raw-MAE curves have one available seed each, while the final main-table comparisons use two seeds.

Table 25: **FID-DINOv2 across training steps.** ImageNet-1k, SiT-B/2, batch 64. Rows use seed 0 unless marked seed 1. Unaligned and raw MAE have one intermediate-checkpoint trajectory each, with the other seed evaluated only at the endpoint. Bold marks the lowest FID per training step and seed among the reported runs.

Target / seed	100K	200K	300K	400K
Unaligned	1257.8	1078.9	997.0	945.7
Raw MAE, seed 1	1258.7	1077.3	998.6	933.9
Raw DINOv2 (= REPA)	1169.6	978.3	894.4	825.1
Raw DINOv2, seed 1	1160.4	973.0	872.9	821.1
LAP-L (DINOv2)	1157.3	963.0	**869.8**	**804.8**
LAP-L (DINOv2), seed 1	1180.6	987.5	893.4	827.8
LAP-N (DINOv2) (k=3)	1168.2	984.3	890.2	821.7
LAP-N (DINOv2), seed 1	1191.8	992.8	908.6	833.4
LAP-L (MAE)	1159.4	998.9	906.0	837.7
LAP-L (MAE), seed 1	1164.6	999.8	910.5	856.5
LAP-N (MAE) (k=5)	**1135.1**	**956.2**	879.7	821.3
LAP-N (MAE), seed 1	**1127.6**	**941.2**	**865.1**	**797.6**

H.5 Qualitative Samples

Figure 11 presents samples using matched sampling seeds across models. These examples illustrate the evaluated outputs. They do not replace the quantitative distribution-level metrics or establish perceptual equivalence between methods.

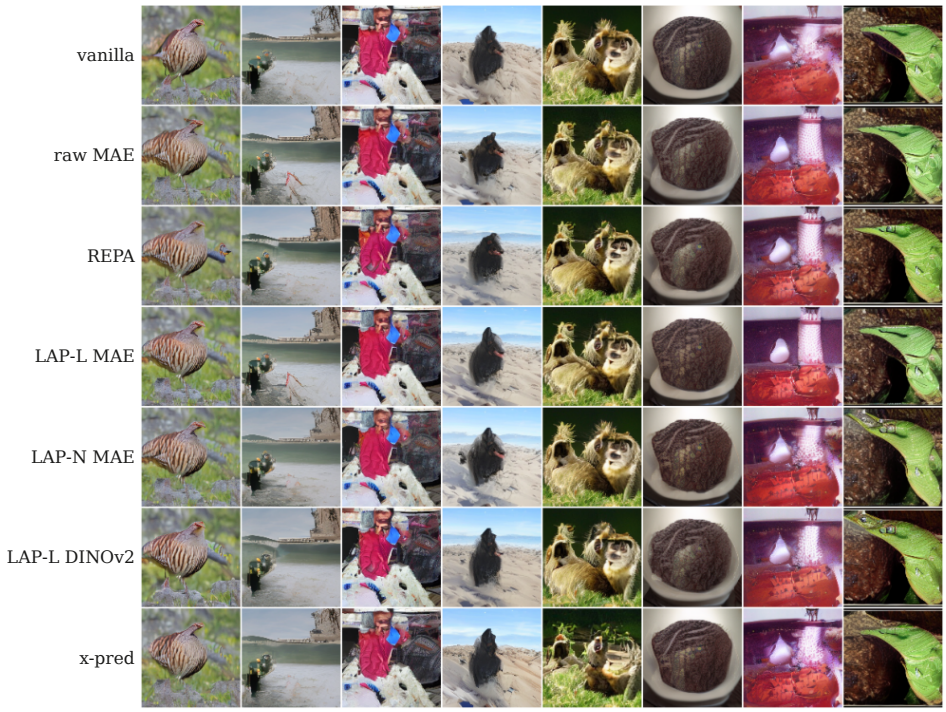


Figure 11: **Qualitative comparison of the 400K models.** SDE-250 sampling without guidance. Each column uses a matched sampling seed across models. Row labels "vanilla" and "x-pred" denote the unaligned model and the predictable-only DINOv2 target.

I Computational Resources

Runs use one A100 unless noted; the REPA-family baseline, CLIP, and mean-removal runs used one H100 or A100 each, with the cached dataset staged on a local network file system because the shared file system was I/O-bound at the time. The MAE runs on the EQ-VAE and REPA-E tokenizers (Table 3) encode latents online and were trained with data parallelism on two H100s at the same global batch of 64, with the final segments of the two EQ-VAE LAP-N seeds on four H100s; their 50K samples were drawn with the sample indices split across two A100s or four H100s. In the base batch-64 recipe, a 400K-step ImageNet-1k run takes approximately 18–36 hours, depending on the encoder. A 100K-step development run takes about nine hours, and a 50K-step run takes about three to four and a half hours. Fitting one purifier takes about two hours. During diffusion training, LAP-N also evaluates the frozen predictor once per batch in FP32 without gradients. The predictor is discarded for sampling. The experiments do not include matched wall-clock or peak-memory benchmarks, so per-step convergence gains do not establish an advantage in total compute.

Standard 50K-image sampling takes approximately eight to ten A100-hours per model at sampling batch 64. The subsequent ADM metric calculation runs on CPU. The original experiment set and first replication wave used approximately 750 A100-hours. The additional validation and robustness checks used approximately 1960 A100-hours and 60 V100-hours. These are project-record estimates and exclude exploratory runs. The image-disjoint validation of nine frozen checkpoints took about six minutes on one H100, including target extraction and probe fitting.

The full-scale SiT-L/2, batch-256 runs use four gradient-accumulation steps of 64 images. Optimizer steps, EMA updates, and checkpoints are counted after accumulation. Thus, the reported batch size and training-step count refer to the effective batch-256 updates.

J TWO AERIAL-SCENE DATASETS

The introduction motivates purification by domains in which reconstruction-based encoders are the ones available. We tested this motivation at development scale on two aerial-scene datasets, comparing the ImageNet encoders only; the outcome is consistent with the motivation but does not establish it, and we report it here so that the claim is not read as established.

Setup. NWPU-RESISC45 (Cheng et al., 2017), all 31,500 images in 45 classes at 256×256, and AID (Xia et al., 2017), 10,000 images in 30 classes downsampled from 600×600. Targets: DINOv2 as the off-domain reference, and ImageNet MAE raw and purified by LAP-N with a k=5 predictor fitted on the dataset's own latents. The recipe is the dev-200 protocol (SiT-B/2, batch 64, 50K steps, $\lambda_a = 0.5$), with four training seeds on RESISC45 and two on AID. We report Inception FID between 10K samples (SDE-250, no guidance) and 10K training images of the same dataset; we use Inception features to avoid evaluating with one of the alignment teachers. The same ADM evaluations also provide IS, precision, and recall, reported in Table 4. Means and sample SDs are computed from the unrounded per-seed metrics. 50K steps correspond to about 100 epochs of RESISC45 and, because AID is small, 320 epochs of AID.

Table 26: **Development-scale alignment on two aerial-scene datasets.** FID-10K against 10K training images of the same dataset; mean $\pm$ standard deviation over n training seeds. Lower is better. Shaded rows are LAP targets. Bold marks the lowest mean FID within each dataset.

Dataset	Alignment target	n	FID-10K
RESISC45 (45 classes)	No alignment	4	68.1 ± 5.1
	DINOv2	4	66.3 ± 3.7
	ImageNet MAE, raw	4	64.6 ± 4.4
	LAP-N (ImageNet MAE)	4	$\mathbf{61.5} \pm 2.3$
AID (30 classes)	No alignment	2	62.4 ± 2.1
	DINOv2	2	63.7 ± 1.3
	ImageNet MAE, raw	2	66.1 ± 1.5
	LAP-N (ImageNet MAE)	2	$\mathbf{60.1} \pm 3.3$

Outcome. Table 26 summarizes the FID results. On RESISC45, purified ImageNet MAE has the best mean (61.5 ± 2.3), about 6.6 below the unaligned model (68.1 ± 5.1; Welch $t = 2.4$), 4.9 below off-domain DINOv2 (66.3 ± 3.7; $t = 2.2$), and 3.2 below raw ImageNet MAE ($t = 1.3$). On AID, purification improves ImageNet MAE from 66.1 ± 1.5 to 60.1 ± 3.3 ($t = 2.4$), and purified ImageNet MAE is again the best target, but only 2.3 below the unaligned model ($t = 0.8$); DINOv2 does not improve on no alignment there. On both datasets, purified ImageNet MAE is the best aligned target and off-domain DINOv2 is not, which is in the direction of the introduction's motivation. LAP also achieves the highest mean IS and precision on both datasets, and the highest recall on RESISC45. On AID, LAP has slightly lower recall than no alignment (0.214 vs. 0.222). The margins over no alignment are at most about two standard deviations over two to four seeds, and the runs are short, so we report the motivation as suggestive rather than established, and say so in the limitations.

K RELATED WORK

K.1 WHERE AND WHEN REPRESENTATIONS GUIDE GENERATION

REPA aligns intermediate diffusion features with pretrained visual representations (Yu et al., 2025). Related approaches change where that guidance enters the pipeline. REPA-E jointly tunes the VAE and diffusion model (Leng et al., 2025), while representation autoencoders use pretrained encoder features as the generative representation space (Zheng et al., 2026). SRA and LayerSync derive guidance from the model's own representations or layers (Jiang et al., 2026; Haghighi et al., 2026). U-REPA extends alignment to U-Net architectures (Tian et al., 2025); its multi-resolution matching has no counterpart in the SiT backbone used here, so we do not compare against it.

Other methods change the timing or strength of supervision. HASTE combines early-stopped alignment with attention-map supervision (Wang et al., 2025b), and VA-REPA adapts alignment using the variance structure of flow matching (Yang et al., 2026). TARO studies timestep-adaptive alignment for video-to-audio generation (Ton et al., 2025). These approaches motivate studying when alignment helps, whereas our interventions ask which content within a fixed target is useful.

K.2 Transforming the Target or Matching Objective

The most directly related work is iREPA (Singh et al., 2026). It compares 27 visual encoders and argues that spatial structure is more predictive of generation performance than global classification accuracy. It also modifies target normalization and the projection head. Our question concerns locally predictable content within a fixed encoder. The interaction experiments in Appendix E test whether the corresponding operations and LAP-N offer additional benefits when combined. They do not establish that spatial structure and local predictability are mutually exclusive explanations.

PixelREPA diagnoses a regression shortcut in pixel-space image transformers and introduces a masked transformer adapter (Shin et al., 2026). Spectrum Matching studies spectral properties and applies a difference-of-Gaussians transformation to REPA targets (Ning et al., 2026). SARA and structural representation alignment modify how representation structure is matched (Chen et al., 2025; Xu et al., 2026). These works show that target processing and the matching objective can matter beyond encoder identity.

Guidance can also come from the generative pipeline. AHPA uses timestep-dependent hierarchical VAE priors (Min et al., 2026), and SRA 2 uses VAE representations (Wang et al., 2026). Representation entanglement introduces representation guidance within the generative architecture (Wu et al., 2025). Our method keeps the pretrained encoder fixed and changes the target supplied to the existing alignment loss.

K.3 Mechanistic Analysis and Distillation

AG-REPA uses causal layer interventions in audio flow matching and distinguishes stored information from contribution to the velocity prediction (Zhang et al., 2026). This is an important precedent for separating decodability from functional contribution. Our study focuses on target content and an auxiliary prediction intervention in image generation, rather than selecting alignment layers.

REED develops a theoretical framework for representation guidance and training curricula (Wang et al., 2025a). Our linear-Gaussian analysis has a narrower purpose: to illustrate why decodability and predictive contribution can differ. It does not replace a theory of diffusion optimization.

Knowledge-distillation methods such as task-aware layer-wise distillation and variational information distillation adapt what a student learns from a teacher (Liang et al., 2023; Ahn et al., 2019). Their objectives are not equivalent to the clean-latent prediction utility measured here. Likewise, linear concept-erasure methods remove specified decodable information under explicit linear criteria (Ravfogel et al., 2020; Belrose et al., 2023). LAP shares the idea of selective removal. Its affine and nonlinear local predictors define operational residuals, without guaranteeing removal of all reconstruction information.

Finally, representation regularization can help without an external encoder (Wang & He, 2025). The limited self-teacher result in Appendix G should therefore be read as a result for that construction, not a general argument against encoder-free guidance.

L Individual Runs for the Main Comparisons

This appendix keeps the numerical results behind the main comparisons in one place. Generation columns show individual training seeds. Diagnostic means are identified separately. Two runs describe replication but do not provide a well-calibrated confidence interval. A dash denotes an unavailable measurement.

L.1 Auxiliary Prediction Sweep

Table 27 lists the auxiliary prediction-loss runs of Section 3.1, and Figure 12 plots them. On dev-200, velocity sufficiency saturates by $\lambda_s = 0.5$ (0.50, against 0.52 at $\lambda_s = 2$), yet mean FID-DINOv2 keeps worsening, from 1159 to 1187 and then 1247. Reliance on the aligned subspace is unchanged up to $\lambda_s = 0.5$ (8.3% and 7.6% of the full-layer effect) and rises to 40% only at $\lambda_s = 2$. At full scale it rises from 11% to 15%, with seeds at 18.9% and 11.5%, while FID-50K worsens in both seeds. The degradation therefore does not require the denoiser to rely more on the supervised directions. For reference, the unmodified REPA model's aligned subspace is decodable (presence $R^2 = 0.207$), but ablating it increases velocity error by 12.2% of the full-layer effect, less than rank-matched random subspaces at 22.7% (Figure 12(a)).

Table 27: **Auxiliary prediction-loss sweep.** Diagnostics are two-seed means. Dev-200 retains the earlier token split, while full-scale diagnostics use disjoint images. Proxy columns give individual FID-DINOv2 scores. At $\lambda_s = 2$, use is $37.6/42.8\%$ on dev-200 and $18.9/11.5\%$ at full scale (seeds 0/1). Standard FID-50K values are in Table 29. Bold marks the lowest generation proxy per scale and training seed. Diagnostic columns are not ranked as generation performance.

λ_s	Velocity R^2	Align. cosine	Use (%)	Proxy seed 0	Proxy seed 1
Dev-200, 100K steps					
0	0.16	0.65	8.3	**1157.3**	**1161.2**
0.5	0.50	0.63	7.6	1167.2	1207.1
2	0.52	0.61	40.2	1242.0	1252.3
ImageNet-1k, 400K steps					
0	0.11	0.64	11.4	**825.1**	**821.1**
2	0.53	0.61	15.2	879.5	877.1

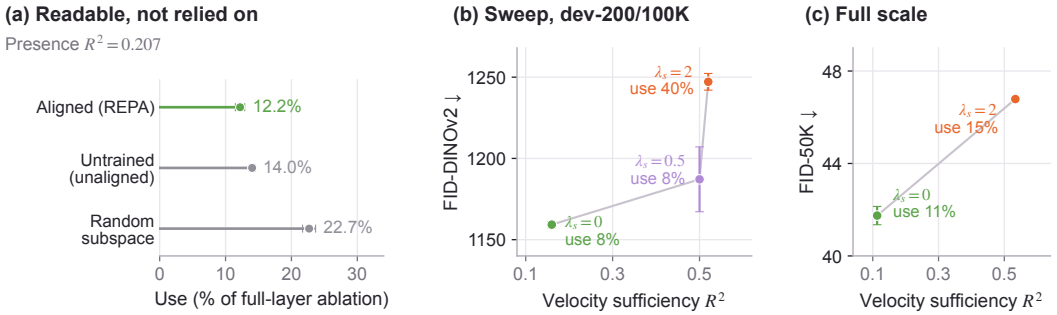


Figure 12: **Readability, reliance, and the auxiliary prediction loss.** (a) Rank-256 mean-fill ablations of the REPA model as a percentage of the full-layer effect (ImageNet-1k/400K, seed 0, mid-noise, disjoint fit/evaluation images, 95% image-bootstrap intervals). (b) The auxiliary prediction loss on dev-200/100K: velocity sufficiency saturates by $\lambda_s = 0.5$ while FID-DINOv2 keeps worsening, and reliance on the aligned subspace ("use") rises only at $\lambda_s = 2$. These diagnostics use the earlier token split. (c) The same loss at full scale, with standard FID-50K and disjoint-image diagnostics. Points are two-seed means and bars span the seed values. Values are in Table 27.

L.2 Encoder Targets and Purification

Table 28 separates target prediction utility from generation results. The utility column is defined only for the raw encoders that were probed. The development and full-scale columns use different datasets and training lengths, so their proxy values should be compared within each pair of columns.

Table 28: **Encoder comparisons and target-transformation ablations.** Utility is the clean-latent probe R^2 for the five raw encoders. Generation columns give individual FID-DINOv2 scores, including the development-scale purification ladder in Figure 1(b) and encoder comparison in Figure 3. The iREPA-style and VA-REPA rows are the REPA-family baselines of Table 2 (dev-200 values from Table 16). The centered variance explained by the DINOv2 linear split is approximately 2.7% at development scale and 2.1% at full scale. Standard FID values for Figure 2(b) are in Table 29. Bold marks the lowest generation score in each column among the listed configurations.

Alignment target	Utility R^2	Dev-200/50K		ImageNet-1k/400K	
		Seed 0	Seed 1	Seed 0	Seed 1
No alignment	–	1652.4	1649.9	945.7	947.4
Raw MAE	0.72	1641.4	1643.2	946.2	933.9
LAP-C (MAE), mean removal	–	1544.0	1552.6	849.4	839.8
MAE, iREPA-style ops	–	1508.0	1496.9	832.2	816.6
LAP-L (MAE)	–	1504.3	1512.3	837.7	856.5
LAP-N (MAE)	–	1486.2	1482.6	821.3	797.6
LAP-N (MAE), then linear	–	1512.2	1468.6	–	–
LAP-P (MAE), per-position mean	–	1495.9	1502.8	832.2	834.8
LAP-N (MAE) + P	–	**1471.3**	**1436.4**	814.3	800.2
LAP-N (MAE) + iREPA-style ops	–	1510.7	1484.4	**788.7**	802.1
MAE, VA-REPA schedule	–	–	–	949.4	924.0
LAP-N (MAE) + VA-REPA schedule	–	–	–	802.3	**782.0**
DINOv2 (REPA)	0.33	1551.2	1518.6	825.1	821.1
DINOv2, iREPA-style ops	–	1476.6	1504.3	811.8	805.5
DINOv2, VA-REPA schedule	–	–	–	806.4	810.7
LAP-L (DINOv2) + iREPA-style ops	–	–	–	796.1	802.2
LAP-L (DINOv2) + VA-REPA schedule	–	–	–	794.6	799.0
LAP-L (DINOv2)	–	1546.0	1487.8	804.8	827.8
LAP-N (DINOv2)	–	1508.4	1536.2	821.7	833.4
LAP-C (DINOv2), mean removal	–	–	–	819.4	834.8
LAP-P (DINOv2)	–	1518.9	1507.3	–	–
LAP-N (DINOv2) + P	–	1512.1	1504.0	–	–
Predictable-only (DINOv2)	–	1679.9	1641.6	1000.6	995.2
CLIP	0.18	1548.9	1553.0	853.3	852.1
LAP-L (CLIP)	–	1517.1	1513.9	824.0	831.9
LAP-N (CLIP), k=3	–	1537.6	1498.1	837.6	818.8
LAP-C (CLIP), mean removal	–	–	–	825.0	824.0
I-JEPA	0.29	1529.4	1515.7	–	–
MoCo v3	0.44	1584.2	1598.0	–	–

L.3 STANDARD FID BY SEED

Tables 29 and 30 report the standard FID-50K runs behind Tables 1 to 3 and 21. These values use the same base SiT-B/2, batch-64, 400K recipe. The larger-setting single-seed evaluations are listed separately in Table 17.

Table 29: **Standard FID-50K by training seed.** Individual values behind Tables 1 to 3 and 21, continued in Table 30. All models use the ADM protocol with SDE-250 and no guidance. Across both tables, bold marks the lowest FID for each teacher, recipe, tokenizer, and training seed.

Model	FID seed 0	FID seed 1
No alignment	53.95	53.90
Raw MAE	53.68	52.41
LAP-C (MAE), mean removal	45.07	45.27
MAE, iREPA-style ops	43.50	42.62
LAP-L (MAE)	44.81	45.48
LAP-N (MAE)	43.06	**41.53**
LAP-N (MAE) + VA-REPA schedule	**42.24**	**40.95**
DINOv2 (REPA)	42.14	**41.35**
DINOv2, iREPA-style ops	41.45	41.18
DINOv2, VA-REPA schedule	40.81	40.61
LAP-L (DINOv2)	**40.61**	42.33
LAP-N (DINOv2)	41.90	42.33
REPA + prediction loss	46.67	46.90
Predictable-only (DINOv2)	57.00	56.42
Raw CLIP	44.73	44.74
LAP-L (CLIP)	**42.08**	43.30
LAP-N (CLIP)	42.32	**42.11**
MAE, VA-REPA schedule	53.46	52.19
LAP-N (MAE) + iREPA-style ops	**41.12**	**41.52**
LAP-L (DINOv2) + iREPA-style ops	**40.28**	40.83
LAP-L (DINOv2) + VA-REPA schedule	**40.37**	**40.40**
LAP-C (DINOv2), mean removal	41.80	43.49
LAP-C (CLIP), mean removal	42.39	42.80
LAP-P (MAE), per-position mean	44.02	44.62
LAP-N (MAE) + P	**42.33**	41.80
No alignment, REPA-E VAE	32.27	32.33
DINOv2 (REPA), REPA-E VAE	25.88	26.33
LAP-L (DINOv2), REPA-E VAE	25.98	25.39

Table 30: **Standard FID-50K by training seed (continued).** REG, sREPA, the two additional teachers, and the remaining LAP variants; same protocol and notation as Table 29. Across both tables, bold marks the lowest FID for each teacher, recipe, tokenizer, and training seed.

Model	FID seed 0	FID seed 1
REG (DINOv2)	28.66	29.82
LAP-L (DINOv2) + REG	**28.48**	**28.85**
sREPA (DINOv2)	38.14	38.13
LAP-L (DINOv2) + sREPA	**37.03**	38.24
REG (MAE)	53.29	52.54
LAP-N (MAE) + REG	**41.70**	**41.61**
sREPA (MAE)	50.06	50.80
LAP-N (MAE) + sREPA	**38.90**	**38.02**
Raw DINOv3	**42.12**	43.34
LAP-L (DINOv3)	43.23	**40.92**
Raw AIMv2	48.08	47.87
LAP-L (AIMv2)	44.40	**44.22**
LAP-L (MAE) + iREPA-style ops	43.38	42.81
LAP-C (MAE) + VA-REPA schedule	43.45	43.67
LAP-L (MAE) + VA-REPA schedule	42.98	43.88
LAP-N (DINOv2) + iREPA-style ops	41.49	**40.57**
LAP-C (DINOv2) + VA-REPA schedule	41.27	42.56
LAP-N (DINOv2) + VA-REPA schedule	42.33	41.44
LAP-C (DINOv2) + REG	29.64	30.40
LAP-N (DINOv2) + REG	29.37	29.64
LAP-C (DINOv2) + sREPA	38.26	39.78
LAP-N (DINOv2) + sREPA	37.21	**36.62**
LAP-C (DINOv2), REPA-E VAE	26.48	26.38
LAP-N (DINOv2), REPA-E VAE	**25.03**	**24.85**
LAP-C (DINOv3)	43.79	42.03
LAP-N (DINOv3)	43.10	43.44
LAP-C (AIMv2)	45.59	44.34
LAP-N (AIMv2)	**43.98**	45.14
Raw MAE, REPA-E VAE	33.42	34.06
LAP-N (MAE), REPA-E VAE	**27.84**	**27.28**
No alignment, EQ-VAE	52.47	52.53
Raw MAE, EQ-VAE	51.44	53.38
LAP-N (MAE), EQ-VAE	**40.19**	**39.83**