Is a Linear Probe Evidence of a Linear Representation?

Nadir Kutluozen[*]
Cold Spring Harbor Laboratory

Christian Internò[*]
Bielefeld University

David Klindt
Cold Spring Harbor Laboratory

[*] *Equal contribution*

Abstract

The *linear representation hypothesis* holds that pretrained vision models encode generative factors as linearly decodable directions in feature space, and comes with a standard methodology: train a linear probe, declare a factor *represented* if accuracy is high. We show this test is too permissive. Asymptotically, any sufficiently rich feature map, including a random one, admits a linear readout, so high probe accuracy is consistent with two very different geometries: a true linear world model, where the latent-aligned subspace dominates the representation, or a kernel machine, where the latents lie linearly inside a much larger curved manifold. This distinction matters because the curved structure orthogonal to the probe is relevant for out-of-distribution generalization, robustness, and compositional transfer. We propose three asymmetric diagnostics that distinguish the two cases: reverse predictivity, spectral concentration, and out-of-distribution generalization. We validate them in a fully observed simulation where ground truth is known. Applied to six pretrained vision transformers on SVG-World, our deterministic generative benchmark with paired visual styles, and to two of them on CelebA, they show pretraining produces structure beyond a random kernel but well short of a linear world model. On dSprites, 3D Shapes and MPI3D, forward decodability and latent-aligned variance dissociate. Thus, forward decoding alone does not show that the probed factors align with the dominant geometry of the representation. In summary, our work cautions about the success of linear probes as opposed to truly linear and low dimensional representations.

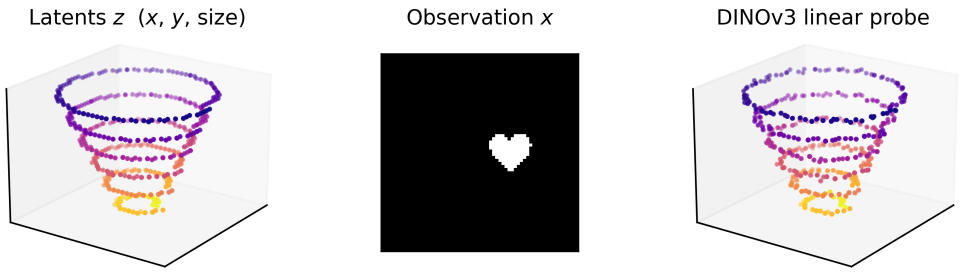


Figure 1: **Forward Linear Decoding.** A 3D latent trajectory z (left) generates images x (middle) through a renderer [1]. A linear probe on DINOv3-B/16 features recovers the trajectory in latent space (right), inverting the rendering pipeline with a linear map. We ask when recovery is evidence of a linear world model (Def. 1) rather than a generic property of any sufficiently rich feature map.

Correspondence: `nadirkutluozen.nk@gmail.com`, `christian.interno@uni-bielefeld.de`.
Code at `https://github.com/klindtlab/linear-probe-nullspace`

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

1 Introduction

A recurring claim in representation learning is that pretrained models internally organize the world into linearly accessible factors of variation [2]. This *linear representation hypothesis* has been invoked to explain phenomena ranging from analogies in word embeddings [3, 4] to mechanistic interpretability of large language models [5, 6], and it underwrites an increasingly popular methodology: train a linear probe and declare a latent variable "represented" when R^2 is high. The implicit inference is: *if z is linearly decodable from $f(x)$, then f has learned z as an axis of its representation.* We formalize the strongest version of this hypothesis, namely, that z spans the dominant directions of f's variance, as a *linear world model* of z (Def. 1).

Testing this hypothesis is more nuanced than it appears. Rich nonlinear feature maps, including random ones, are universal function approximators, so with enough features almost any sufficiently smooth target can be linearly read out from them [7]. The infinite-width limit at initialization is itself a Gaussian process [8, 9], and the neural tangent kernel [10] extends this to wide networks during training. A high forward R^2 is therefore consistent with two distinct hypotheses: (*H1*) f is a linear world model of z (Def. 1), or (*H2*) the model is a high-dimensional feature map in which z lies linearly but carries a small share of the variance. Empirically, probes are known to recover signal from models that lack the structure [11, 12, 13, 14]; we extend this concern from the probe to the encoder itself. Probe success is also subject to a failure mode under *superposition*, where linear representation is necessary but not sufficient for recovery [15, 16].

Contributions. We **(i)** formalize the notion of a *linear world model* (Def. 1) and propose three asymmetric diagnostics, reverse predictivity, spectral concentration, and out-of-distribution (OOD) generalization, each of which can fail for a kernel-machine map that satisfies forward decoding, plus a ceiling that makes reverse predictivity comparable across models (§3); **(ii)** validate them in a fully observed simulation where 2D latents are lifted by a frozen random Fourier map into random or trained MLP features, recovering the asymptotic random-features prediction for the random-init baseline and the Def. 1 regime for the trained, demonstrably nonlinear counterpart (§4); **(iii)** introduce SVG-World, a deterministic testbed with a 32-dim latent layout shared between two rendering styles (AquaWorld, WestWorld), and apply the diagnostics to six pretrained vision transformers from three training paradigms (DINOv3 S/B/L, DINOv2, DeiT, CLIP; DINOv3-B/16 and CLIP-B/16 also on dSprites, 3D Shapes, MPI3D and CelebA), finding that DINOv3-B/16 sits partway between the simulation's two regimes with style-invariant alignment concentrated in the top $\sim$32–64 PCs (§5–5.4).

2 Related Work

Linear Representations and World Models. From word-embedding analogies [3] and earlier distributed representations [4], through vision probes [2], to recent mechanistic interpretability of LLMs [5, 6], a line of work claims that high-level concepts are encoded as linear directions; sparse autoencoders [17, 18] add that linearity plus sparsity recovers the "atoms" of the representation under superposition. *World model* has been used with varying formality, from latent dynamics models for RL [19, 20] to modern video generators [21], and the hope that pretrained models instantiate one [22] is often operationalized by linear probing on factors believed to be relevant.

Identifiability, Disentanglement, and High-Dimensional Representations. Nonlinear ICA [23, 24] and linear identifiability up to affine transforms [25, 26] give formal conditions for recovering latents; classic work shows disentanglement is unidentifiable without inductive biases or supervision [27, 28], and Higgins et al. [29] propose a symmetry-based definition. Our reverse-predictivity diagnostic [30] is closely related to the covariance-alignment arguments of this literature. Stringer et al. [31] show that neural responses follow a power-law spectrum, which the strict "variance-concentration" reading of Def. 1 cannot satisfy (§6).

Random Features and Kernels. Rahimi and Recht [7] show that random feature maps approximate RKHS kernels, so arbitrarily complex targets become linearly recoverable with enough dimensions; the neural tangent kernel [10] extends this to wide networks at initialization, and later work relates the kernel eigenspectrum to learnability and generalization [32, 33]. Sprekeler et al. [34] make the same point for nonlinear ICA: random polynomial features with slow feature analysis recover sources without learning the feature map. All predict an asymptotic equivalence of trained and random feature maps for forward linear decoding; characterizing the finite-d regime is part of our contribution.

Generalization-Based Diagnostics. A representation that genuinely encodes a factor should support compositional generalization [35, 36] and extrapolation beyond the training support. Schott et al. [37] report that pretrained representations often fail to generalize compositionally even within one domain, so within-support forward decoding can mislead. The cross-condition generalization performance of Bernardi et al. [38] operationalizes a related idea in neuroscience, i.e., transfer of linear separability; we adopt a related framing in §5.3, with two paired visual styles as the conditions.

3 Theory: When Is Linear Decoding Evidence of a Linear World Model?

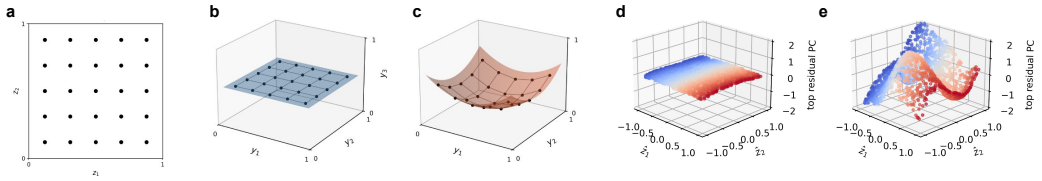


Figure 2: **Linear World Model vs. Kernel Machine. (a)** 2D latent grid. **(b, c)** Cartoons: a linear world model lifts the grid to a flat plane (Def. 1); a kernel-machine map might lift it to a curved surface. Forward R^2 is high in both; reverse R^2 separates them (Prop. 2). **(d, e)** Empirical realization at h=1024: horizontal axes are forward-decoded $\hat{z}_1, \hat{z}_2$; vertical is the top variance direction in the decoder's null space, residualized against $\hat{z}$. Trained is a plane, random a curved manifold.

We begin with a formal definition, recall the asymptotic argument that makes forward decoding a weak diagnostic in the limit, and close with three asymmetric (dimension independent) tests.

3.1 Setup and Definition

Let $\mathcal{X}$ be a space of inputs and $z : \mathcal{X} \to \mathbb{R}^k$ the latents (of the *data generating process*). A frozen feature map $f : \mathcal{X} \to \mathbb{R}^d$ produces embeddings on which we run linear probes.

Definition 1 (Linear world model)**.** *We say that f* implements a linear world model of z with residual level $\sigma^2 \geq 0$ *if, with $\tilde{z} = \mathrm{Cov}(z)^{-1/2}(z - \mathbb{E}[z])$ the whitened latents, there exist $W \in \mathbb{R}^{d\times k}$ and $\mu \in \mathbb{R}^d$ such that*

$$f(x) = W\tilde{z}(x) + \mu + \varepsilon(x), \qquad \mathbb{E}\big[\varepsilon(x)\,\tilde{z}(x)^\top\big] = 0, \qquad \mathbb{E}\Big[\|\varepsilon(x)\|^2\Big] \leq \sigma^2\,\|W\|_F^2\,. \tag{1}$$

That is, f is, up to a bounded orthogonal residual, an affine map of z. The residual ε is the nonlinear part of f, for example curvature in z or variation due to other factors. Since $\mathrm{Cov}(\tilde{z}) = I$, $\|W\|_F^2$ is the variance of f carried by the latents, so σ^2 is the ratio of residual to latent-aligned variance (unit-free), and $\sigma^2 = 0$ means f is exactly an affine image of $\tilde{z}$. As with linear probes, we call the map linear: μ drops out of every statistic below, all computed on centered data. Throughout, "linear world model" means only Def. 1, variance-dominant linear alignment with a specified factor set, not a general account of how f represents the world. This is strictly stronger than "z is linearly decodable from f": it requires that *most* of f's variance is aligned with z, not merely that z lies inside the span of f. Def. 1 is one end of an axis whose other end is a rich feature map that merely admits a linear readout of z (Prop. 1); the diagnostics locate f on this axis for a specified factor set.

3.2 Forward Decodability Is Asymptotically Free

The following statement is a direct consequence of classical random-features theory [7]; we state it informally and defer the technical version to App. A.

Proposition 1 (Forward decodability from rich random features)**.** *Let $\kappa(\cdot,\cdot)$ be a bounded continuous kernel on $\mathcal{X}$ whose RKHS $\mathcal{H}_\kappa$ contains every coordinate z_j of the ground-truth latents. Let $f_{\mathrm{rand}}(x) \in \mathbb{R}^d$ be a random feature map such that $f_{\mathrm{rand}}(x)^\top f_{\mathrm{rand}}(x') \to \kappa(x,x')$ as $d \to \infty$. Then for any $\epsilon > 0$ and $\delta \in (0,1)$ there exists $d_{\epsilon,\delta} \in \mathbb{N}$ such that for all $d \geq d_{\epsilon,\delta}$, with probability at least $1-\delta$ over the draw of f_{rand}, there exists $W \in \mathbb{R}^{k\times d}$ with*

$$\mathbb{E}_x\Big[\|W f_{\mathrm{rand}}(x) - z(x)\|^2\Big] \leq \epsilon. \tag{2}$$

Prop. 1 has an asymptotic consequence: forward linear decodability is, in the limit, a property of having enough random features rather than of having learned anything about z. Whether the asymptotic regime dominates at the feature dimensions used in practice (d on the order of 10^2 to 10^3) is empirical, and we test it directly in §4 and §5.1.

3.3 Three Asymmetric Diagnostics

We propose three tests, each of which exploits a different asymmetry between Def. 1 and the mere existence of a linear decoder.

(D1) Reverse predictivity [30]. Under Def. 1, the latent-aligned subspace accounts for a large share of feature variance. We therefore measure

$$R^2_{\mathrm{rev}} \;=\; 1-\frac{\mathbb{E}_x \left\| f(x) - W_{\mathrm{rev}} z(x) - \mu \right\|^2}{\mathbb{E}_x \left\| f(x) - \mathbb{E} f \right\|^2}, \qquad W_{\mathrm{rev}} = \arg\min_{W,\mu} \mathbb{E}_x \left\| f(x) - W z(x) - \mu \right\|^2 . \tag{3}$$

Proposition 2 (Reverse-predictivity asymmetry)**.** *Suppose $f(x) = Az(x)+\mu+\varepsilon(x)$ with $\mathbb{E}[\varepsilon z^\top] = 0$. Then*

$$R^2_{\mathrm{rev}} \;=\; \frac{\mathrm{tr}(A\,\mathrm{Cov}(z)A^\top)}{\mathrm{tr}(A\,\mathrm{Cov}(z)A^\top) + \mathrm{tr}\,\mathrm{Cov}(\varepsilon)}. \tag{4}$$

In particular, if ε is dominated by a high-dimensional z-uncorrelated nonlinear component (the kernel-machine regime), R^2_{rev} is small even when forward $R^2 \approx 1$.

Proposition 3 (Ceiling)**.** *For any f, let $\lambda_1 \geq \cdots \geq \lambda_d$ be the eigenvalues of $\mathrm{Cov}(f)$ and $r = \mathrm{rank}\,\mathrm{Cov}(z)$. Then*

$$R^2_{\mathrm{rev}} \;\leq\; C_r \;=\; \frac{\sum_{i\leq r}\lambda_i}{\sum_{i\leq d}\lambda_i}, \tag{5}$$

with equality iff the range of W_{rev} lies in a leading r-dimensional principal subspace of f and the reverse-probe residual $f - W_{\mathrm{rev}} z$ has no variance in that range.

The proof is in App. A. We use $r = \mathrm{rank}\,\mathrm{Cov}(z)$ rather than k so that constant or redundant latent coordinates do not loosen the bound. The ratio R^2_{rev}/C_r makes intermediate values interpretable: it is the fraction of the attainable share that the factors take, comparable across encoders with different spectra. For a scalar factor, $C_1 = \lambda_1 / \mathrm{tr}\,\mathrm{Cov}(f)$ bounds even a perfectly isolated factor. Since C_r depends on the spectrum, we also report the participation ratio $\mathrm{PR} = (\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$.

(D2) Spectral concentration. Under Def. 1 with small σ^2, the latent-aligned subspace is spanned by the leading principal directions of f. D2 asks whether this is the case, independently of how much variance those directions carry. Project f onto its top m principal components and rescale each to unit variance (PCA whitening, fitted on training rows only), and let $R^2_{\mathrm{rev}}(m)$ be the reverse R^2 of the result. Because the whitened components have equal variance, $R^2_{\mathrm{rev}}(m)$ is the average, over the top m principal components, of the fraction of each component's variance that is linear in z. We report the scalar $\mathrm{SC} = R^2_{\mathrm{rev}}(r)$, with $r = \mathrm{rank}\,\mathrm{Cov}(z)$ as in Prop. 3: the extent to which the r leading directions of f are linear functions of the latents. A strict linear world model has $\mathrm{SC} \approx 1$. A kernel machine, whose leading directions are dominated by z-uncorrelated variation, has low SC even when forward R^2 is high. For $m > r$ the curve falls for any f, at least as r/m once the extra components are residual, so the informative region is $m \approx r$; the full curve shows where in the spectrum the latent-aligned variance lives. Unlike subspace-similarity measures such as CKA [39], SC needs no choice of basis for the latent-aligned subspace, only its location in f's spectrum.

(D3) Out-of-distribution generalization. Linear decodability that holds *only within training support* is consistent with overfitting a rich nonlinear kernel on that support. A genuine linear world model should extrapolate [40]. We test this by training linear probes on one visual-style condition and evaluating on another condition with similar latents; the encoder's positional directions should transfer if it has actually learned position as a factor rather than as a style-conditional pattern [35, 37, 38].

Summary. For a linear world model with small σ^2, all three diagnostics are close to their maximum: R^2_{rev} and SC are near 1, and a probe trained in one condition transfers to another with little loss. For a rich kernel-machine map, forward R^2 can be high while all three are low. The diagnostics are therefore necessary conditions: low scores rule out a linear world model of z, high scores do not

establish one. Like Def. 1, they are linear, so a factor encoded faithfully along a curved manifold also scores low. The rest of the paper asks where pretrained vision models fall between these two cases, for a specified factor set.

4 Controlled Validation in a Fully Observed Simulation

Before turning to vision foundation models, we verify that the diagnostics of §3.3 behave as predicted in a setting where ground truth is known and the random-features prediction can be tested directly.

Setup. Latents $z \sim \mathrm{Unif}([-1,1]^2)$ are first lifted by a frozen random Fourier map, $x_j = \cos(w_j^\top z + b_j)$ for $j = 1, \ldots, 256$, with $w_j \sim \mathcal{N}(0, 3^2 I_2)$ and $b_j \sim \mathrm{Unif}[0, 2\pi]$ drawn once and never trained [7], then mapped to features $f(z) \in \mathbb{R}^h$ by a two-layer ReLU MLP, $f(x) = W_2\,\mathrm{ReLU}(W_1 x + b_1) + b_2$ [c.f., 34]. We compare two regimes at matched hidden width h: (i) *random init*, fresh MLP weights, no training; (ii) *trained*, the same architecture trained end-to-end with a linear readout Uf to recover z using AdamW (learning rate $2 \cdot 10^{-3}$, 2000 steps, batch 256), weight decay $\lambda \in \{0, 1\}$. We hypothesize that weight decay encourages linear world models: reading z off a low-variance direction of f needs a large readout row, so penalizing $\|U\|$ pushes the z-aligned subspace toward the leading directions. The network never sees z, and it is nonlinear: at $h = 1024$ (5 seeds, 2000 test points) it visits 1988 ReLU activation regions on average and its Jacobian varies across inputs, where an affine network has one region and a constant Jacobian (App. B).

Result 1: Forward Decoding Alone Does Not Separate the Two. Fig. 3 (leftmost panel) shows the asymptotic prediction of Prop. 1 directly: random-init forward R^2 rises smoothly with width, reaching ≈ 1 by $h \approx 100$. By that width, the random feature map is forward-decodable as well as the trained model. Forward R^2 alone cannot distinguish the two beyond very small widths.

Result 2: The Asymmetric Diagnostics Open a Robust Gap. Reverse R^2 and the top-2 share separate trained from random at every width, and OOD at small and moderate widths (Fig. 3, right three panels). The trained-saturated curve (weight decay $\lambda = 1$) sits at 1 on all three; random sits well below it on reverse R^2 and the top-2 share, and approaches it on OOD only at large widths. Prop. 2 predicts random's reverse R^2 should approach 0 as the z-orthogonal residual dimension grows; the empirical curve plateaus near 0.2 rather than decaying, an artifact of finite-width ReLU MLPs not realizing the strict z-orthogonality the proposition assumes. The middle curve ($\lambda = 0$, supervised but unregularized) is substantively worse than the saturated case on reverse R^2 and the top-2 share, sagging to ~ 0.7 at intermediate widths, and is indistinguishable from the saturated case on forward and OOD. Consistent with this hypothesis, at $h = 1024$ (5 seeds), reverse R^2 is 0.9935 with $\lambda = 1$ vs. 0.7841 with $\lambda = 0$, at identical forward R^2 of 1.000. *We conclude that satisfying Def. 1 is a tunable property of training pressure, not just a binary trained-vs-not property.*

Result 3: A Linear World Model Is Visible in Feature Space. Fig. 2(d, e) plots random and weight-decay ($\lambda = 1$) models at $h = 1024$ against the forward-decoded latents and the largest residual direction. For the trained model the points lie on a flat plane, an affine image of the latent square, as Def. 1 predicts. For the random model they lie on a saddle: z is linearly decodable, but

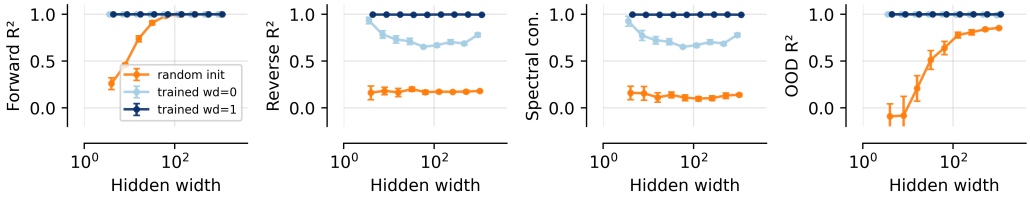


Figure 3: **Diagnostics on the simulation across hidden width h.** Random init (orange) reproduces the asymptotic random-features prediction: forward $R^2 \to 1$ as h grows, while reverse R^2 and the top-2 share stay low and OOD R^2 catches up only at large widths. Trained with weight decay $\lambda = 1$ (dark blue), all four diagnostics sit flat at ≈ 1 across widths. Without weight decay (light blue), reverse R^2 and the top-2 share dip at intermediate widths: the linear-world-model condition is sensitive to training pressure even when forward R^2 saturates. The third panel is the share of variance in the top two PCs that is linear in z, without whitening (App. B). Error bars over 3 seeds (initialization and batch order).

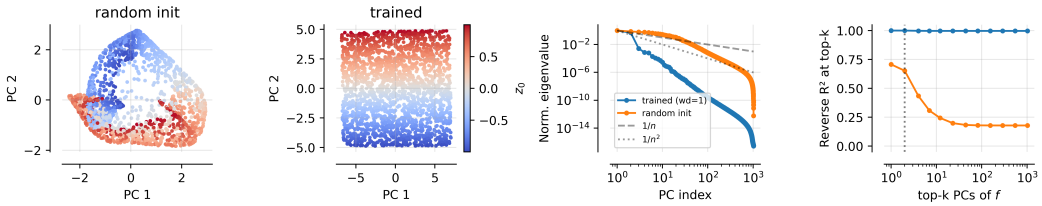


Figure 4: **Spectral diagnostics on the simulation** ($h = 1024$; random init vs. trained with $\lambda = 1$). (a, b) Test points on the top-2 PCs of f, colored by z_0. Random folds the latent square into a non-axis-aligned shape (top-2 subspace not z-aligned); trained gives an axis-aligned image, z_0 along PC 2 and z_1 along PC 1. (c) Normalized eigenspectrum of f (log-log): trained drops by several orders of magnitude after PC 2, random decays smoothly over hundreds of PCs. (d) Reverse R^2 on the top m PCs of f, without whitening. Trained stays at 1: its top two PCs are linear in z and carry almost all the variance, so later PCs barely change the ratio. Random falls from ≈ 0.7 at $m = 1$ to a plateau at its full $R^2_{\text{rev}} \approx 0.2$ as PCs not linear in z are added.

the feature map also bends the latent square along directions the probe ignores. This curvature is the residual variance that lowers R^2_{rev} in Prop. 2.

Result 4: The Spectral Picture. Fig. 4 compares the spectra of the random model and the model trained with $\lambda = 1$. The trained covariance is nearly rank two: the top two components carry 99.57% of the variance (shares 0.5657 and 0.4300, against 0.1384 and 0.1151 at random init), the eigenvalues drop by several orders of magnitude after the second (panel c), and the projection onto the top two PCs is an axis-aligned image of the latent square (panel b). The random model's spectrum decays smoothly over hundreds of components, and its top-two projection folds the latent square onto itself (panel a). Panel (d) shows the reverse R^2 after projecting f onto its top m PCs, without whitening. For the trained model it stays at 1 for every m, because the leading PCs are linear in z and the later ones carry almost no variance. For the random model it falls from ≈ 0.7 at $m = 1$ and levels off at ≈ 0.2, its full R^2_{rev}: each added PC brings variance that is not linear in z, which is the dilution described by Prop. 2.

Takeaway for the Empirical Sections. The simulation verifies that the diagnostics behave as Def. 1 and Prop. 2 predict, and that forward R^2 alone is insufficient to discriminate the two regimes once d is moderately large. The remaining empirical question is where DINOv3 sits on this scale.

5 SVG-World: A Controllable Two-Style Benchmark

To test for the presence of linear representations in real models, we need a *data generating process* (DGP), i.e., a simulation with fully specified latent variables. More precisely, we want a data source with (a) a small, enumerable set of continuous latents, (b) deterministic rendering, (c) enough visual richness that frozen vision models produce nontrivial features, (d) multi-object scenes for binding tests, and (e) a paired second visual style with the same latent layout, so that linear probes can be tested for style-invariance. Existing disentanglement benchmarks [1, 41] satisfy (a)–(c) but not (d) or (e). SVG-World is built to cover all five.

Pipeline. Each scene is rendered from a 32-dim latent vector $z \in [0, 1]^{32}$, of which 20 dimensions are active: 4 people each contribute a position (x, z), a discrete facing direction, and a skin-hue value, and 2 animals each contribute a position (x, z). The active latent layout is identical across both worlds (Fig. 5); only the rendering differs, so the same probe can test whether the directions that decode a latent are stable under a change of visual style (§5.3). The rasteriser is `cairosvg` at 224×224. We release the AquaWorld and WestWorld splits ($N = 10{,}000$ each; `github.com/klindtlab/linear-probe-nullspace`);

Empirical Scope. For all empirical sections that follow we report metrics on the 12 position latents (8 person-position dims + 4 animal-position dims). A precursor benchmark with 15 single-object latents in continuous color space (SVG-Faces) is included as a consistency check in App. H.

5.1 DINOv3 Forward Decoding

Setup. We extract frozen features from DINOv3-B/16 [42, 43] on each of the AquaWorld and WestWorld splits ($N = 10{,}000$ each, $50/50$ train/test), in two feature modes: the CLS token

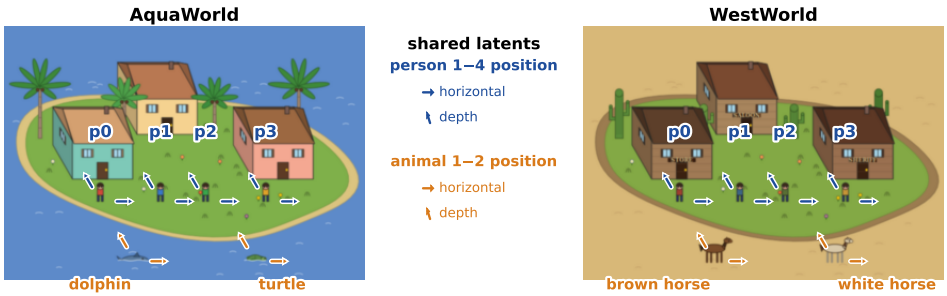


Figure 5: **SVG-World, a deterministic two-style benchmark with a shared latent layout.** Both panels show the canonical scene at $z = 0$; arrows mark each object's two position latents ($+x$ horizontal, $+z$ depth, foreshortened by the oblique projection). AquaWorld and WestWorld share the same 32-dim layout (4 people with (x, z) position, facing direction and skin hue; 2 animals with (x, z) position) but differ in rendering, which enables the cross-domain transfer test of §5.3.

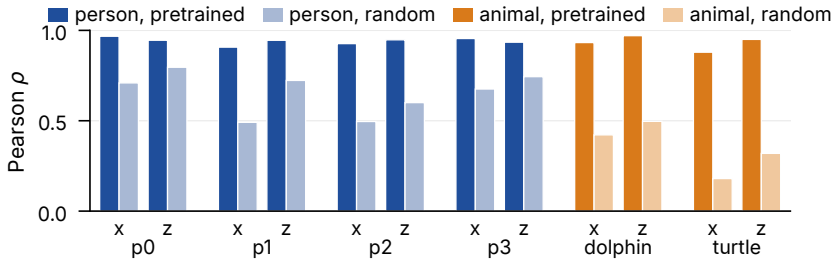


Figure 6: **Forward Decodability Separates Pretrained from Random-Init at d=768.** Per-latent ρ on CLS features; per-latent R^2 in Table 4 (App. D). Pretrained averages $\rho = 0.94$ ($R^2 = 0.88$); random averages 0.55 (0.33) ($\Delta\rho = 0.38$, rising to 0.58 on animals).

($d = 768$) and the mean of patch tokens ($d = 768$). For each mode we fit ridge regression from $f(x)$ to each latent and report test ρ and R^2 per latent, averaged over the two worlds. As a baseline we use *random-init* DINOv3: the same architecture with freshly initialized weights, no training, and the same probing protocol. Implementation details (checkpoint, ridge penalty, seeds) are in App. D.

Result. Pretrained DINOv3 recovers all 12 position latents with mean $\rho \approx 0.94$ ($R^2 = 0.88$) (CLS) and 0.91 (0.82) (mean-pooled patches), the kind of evidence routinely cited for the linear representation hypothesis (Table 4). Mean-pooling helps small foreground objects (animal R^2 0.87 → 0.91) and hurts large ones (R^2 0.89 → 0.77), as expected when patch pooling preserves spatial information that CLS aggregates away. Random-init partially decodes person positions ($R^2 = 0.41$) but loses animals almost completely ($R^2 = 0.17$): the pretraining contribution is concentrated on small foreground objects where feature richness alone does not suffice. These values use a fixed ridge penalty, which penalizes the lower-variance random features more strongly; with a cross-validated penalty random init reaches $R^2 = 0.66$ (Table 5), but the gap on animals remains.

Pretraining, Not Feature Richness, Drives Forward Decoding. Figure 6 visualizes the per-latent CLS comparison from Table 4. The asymptotic random-features prediction does not dominate at $d = 768$: pretrained recovers most position latents at $\rho > 0.9$ (all at $R^2 \geq 0.76$), while random-init is consistently below, and goes below 0.4 ($R^2 \leq 0.31$) on the animal positions. Per §4, however, this forward gap does not by itself establish that DINOv3 satisfies Def. 1. The asymmetric diagnostics in §§5.2 and 5.3 test the structural claim directly.

5.2 Reverse Predictivity and Spectral Concentration

We instantiate (D1) reverse predictivity as defined in Prop. 2 on DINOv3-B/16 features. On raw CLS features, pretrained DINOv3 reaches $R^2_{\text{rev}} = 0.18$, vs. 0.02 for random init; on mean-pooled patches, 0.28 vs. 0.02 (Fig. 7, dashed lines). Over five random initializations, the CLS gap is 0.187 [0.182, 0.191] in AquaWorld and 0.143 [0.139, 0.148] in WestWorld (95% paired scene-bootstrap intervals). The pretrained-vs-random ratio is approximately 10×, much larger than the forward ratio (∼ 1.4× with a cross-validated penalty, App. D). *The forward gap therefore understates how much*

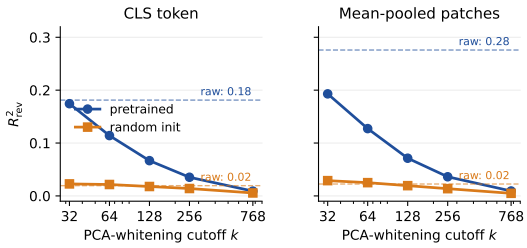


Figure 7: **Reverse Predictivity (D1) and Spectral Concentration (D2) on SVG-World.** R^2_{rev} vs. PCA-whitening cutoff k. Solid: top-k PCs whitened to unit variance. Dashed: raw features. Pretrained's latent-aligned variance concentrates in the top $\sim$32–64 PCs; random's is diffuse. The curve falls with k for any representation (App. E); what separates the two is its value at small k.

more latent-aligned variance pretrained has, while the reverse gap highlights it. Against the ceiling of Prop. 3 the separation is sevenfold: on CLS features in AquaWorld ($R^2_{\text{rev}} = 0.212$, $C_{12} = 0.609$), pretrained reaches 0.35 of the attainable share and random init 0.05. Standardizing features per dimension before fitting the reverse probe leaves the numbers essentially unchanged (App. E), so the gap is not an artifact of variance scaling along individual feature dimensions.

Spectral Concentration (D2). The (D2) diagnostic asks where in the spectrum of f the latent-aligned variance lives. We compute $R^2_{\text{rev}}(k)$, the reverse R^2 after PCA-whitening features to their top k principal components (every PC rescaled to unit variance), for $k \in \{32, 64, 128, 256, 768\}$. Figure 7 shows the result. Pretrained's R^2_{rev} is largest at $k \approx 32$: 0.17 on CLS and 0.19 on patches. It decays monotonically as k grows, falling below 0.04 at $k = 256$ and to 0.01 at full whitening ($k = 768$). Random init's curve is approximately flat at 0.02 across all k. Pretrained DINOv3-B/16 needs its top 17 principal components to collect 90% of its latent-predictable variance, against 55–59 for its random initializations (App. D).

Interpretation. DINOv3's self-distillation objective applies no explicit pressure to make f an affine image of any given factor set z; the encouragement of linear representations through weight decay in the simulation (§4, Result 2) has no obvious analog in the SSL loss. Thus, we do not expect DINOv3 to reach the strict Def. 1 regime ($R^2_{\text{rev}} \to 1$). What we do see is that pretraining produces clear spectral concentration of the moderate latent-aligned variance that exists, consistent with the spectral picture of the trained simulation in Fig. 4 but at a much lower absolute level. DINOv3 is therefore best read as occupying the middle of the kernel-machine to linear-world-model axis: well above a random-feature null on direction-alignment and spectral structure, but well below the Def. 1 regime.

5.3 Out-of-Distribution Generalization Across Visual Styles

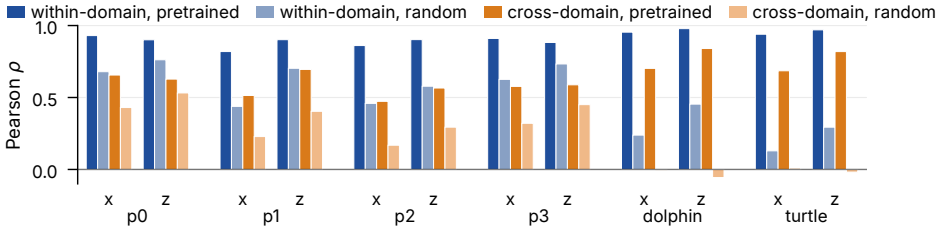


Figure 8: **Asymmetric Diagnostic D3: Cross-Domain Transfer.** Per-latent test ρ on DINOv3-B/16 mean-pooled patch features; cross-domain values average both transfer directions. The pretrained-vs-random gap survives the style change ($\Delta\rho = 0.41$ within, 0.42 across domains). Animal positions transfer best (pretrained $\rho = 0.68$–0.84 cross-domain; random near zero).

We test (D3) by training a ridge probe on one world's features and evaluating it on the other's, with features mean-centered per world so what is tested is the transfer of *directions*, not feature means. Figure 8 reports per-latent ρ on DINOv3-B/16 mean-pooled patch features for both pretrained and random-init weights, in both within- and cross-domain conditions, averaged across both transfer directions (AquaWorld$\to$WestWorld and the reverse).

Within-Domain vs. Cross-Domain. Pretrained averages $\rho = 0.91$ within-domain vs. 0.51 for random ($\Delta\rho = 0.41$). Cross-domain, the absolute values both fall (pretrained 0.65, random 0.23), but the gap between them does not shrink: $\Delta\rho = 0.42$ (with a cross-validated penalty the within-domain gap is 0.22, so the gap grows under the style change). Pretraining's contribution is therefore style-

invariant: the encoder's positional directions transfer across the AquaWorld $\leftrightarrow$ WestWorld stylistic gap as cleanly as they exist within a single world. Animal-position directions transfer particularly cleanly (cross-domain pretrained $\rho \approx 0.76$), while random-init falls to zero or negative on the same dims, despite both representations being trained or initialized identically across the two visual styles. We report ρ because feature scales differ between the two styles, so uncalibrated transfer R^2 is not meaningful (it falls to -58.8). Rescaling the probe output on half of the target style gives transfer R^2 of 0.16 and 0.31 in the two directions for pretrained, against 0.01 and 0.02 for random init.

Reading Alongside Reverse Predictivity. (D1), (D2) and (D3) measure complementary properties of the encoder. Reverse R^2 asks whether z accounts for a large share of f's *variance*; spectral concentration asks where in f's spectrum that share lives; cross-domain transfer asks whether the directions f uses to encode z are style-conditional or style-shared. DINOv3 sits between a random-feature null on all three and the linear-world-model regime occupied by the trained simulation in §4. Style-invariance is, however, the strongest signal: the size of the pretrained-vs-random gap survives a substantial change of rendering essentially undiminished, which is hard to reconcile with the kernel-machine hypothesis (H2, §1) but very natural under the latent-alignment hypothesis (H1). The next section tests whether these findings hold beyond DINOv3-B/16 and SVG-World.

5.4 Beyond SVG-World: Other Encoders, Benchmarks and Real Photographs

Setup. We repeat the analysis with other encoders and datasets. On SVG-World we compare six vision transformers from three training paradigms: self-distillation (DINOv3 S/16, B/16, L/16 [42]; DINOv2-B/14 [44]), supervised classification (DeiT-B/16 [45]) and image–text contrastive learning (CLIP-B/16 [46]), each against a random initialization of the same architecture. DINOv3-B/16 and CLIP-B/16 are also evaluated on three standard disentanglement benchmarks, dSprites [1], 3D Shapes [41] and MPI3D-realistic [47], and on CelebA photographs [48] (App. D). On these datasets we report forward R^2, reverse predictivity (D1) and spectral concentration (D2). The cross-style test (D3) requires two renderings of the same latent layout, which only SVG-World provides; on the benchmarks we instead hold out half of the joint combinations of two factors.

Encoders on SVG-World. All six transformers have higher R^2_{rev} than their random initializations, by 0.145 to 0.235 (95% intervals above zero). A ResNet-50 [49] shows a gap of 0.105 (App. D). Within DINOv3, larger models decode better but carry less latent-aligned variance: from S/16 to L/16, forward R^2 rises from 0.855 to 0.914 while R^2_{rev} falls from 0.257 to 0.169.

Controlled Benchmarks. On dSprites and MPI3D the pretrained encoder decodes the factors better than its random initialization (forward R^2 0.42 vs. 0.16 and 0.85 vs. 0.61) but has a lower R^2_{rev}; on 3D Shapes forward R^2 is tied. The random initializations on these datasets are nearly low-dimensional (participation ratio at most 3.8, against 8.7 to 16.6 pretrained), so most of their variance lies in the top r components and their ceiling is close to 1. The comparisons also depend on how the factors are encoded: four of the six encoder–benchmark pairs keep the sign of the gap under all three encodings in App. D, two do not.

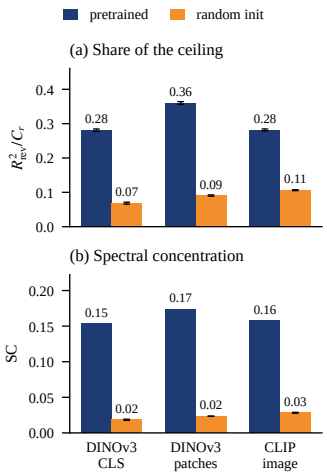


Figure 9: **CelebA.** (a) Share of the ceiling, R^2_{rev}/C_{40}, and (b) spectral concentration; pretrained vs. random init. Error bars: 95% bootstrap for pretrained in (a), s.d. for random (Table 9).

Real Photographs. On CelebA, pretrained models reach 0.28 to 0.36 of their ceiling against 0.07 to 0.11 at random initialization (Fig. 9, Table 9), close to the SVG-World values (0.35 vs. 0.05, §5.2). Spectral concentration separates them as clearly (0.15 to 0.17 vs. 0.02 to 0.03). CLIP's raw R^2_{rev} exceeds that of DINOv3 CLS (0.192 vs. 0.170), but both reach 0.28 of the ceiling.

In summary, on SVG-World and CelebA pretraining raises the latent-aligned share well above random initialization, yet the best encoders reach only about a third of their ceiling, far from the linear-world-model regime of §4; on the controlled benchmarks the comparison depends on encoder, dataset and factor encoding.

6 Discussion

Two Pictures Behind One Probe. A successful linear probe is compatible with two quite different representations. In the first, the representation is essentially a low-dimensional subspace: its main directions of variance are affine in the latents, with little left over. In the second, the latents are one linear readout among many from a rich nonlinear expansion of the input (a kernel machine or random-feature model), and most of the variance is unrelated to them. Random-features theory makes the second picture forward-decodable for almost any smooth target once it is wide enough, so a probe alone cannot separate the two; (D1) reverse predictivity [30], (D2) spectral concentration and (D3) transfer across conditions can, because they ask about the representation, not the readout.

Pretrained Encoders Fall in Between. Across the encoders and datasets tested, wherever the comparison is unambiguous, pretrained representations fit neither picture. They carry far more latent-aligned variance than their random initializations, in the leading principal directions, and on SVG-World their position-decoding directions stay partially stable across visual styles. Yet the probed factors take at most about half of what any linear image of them could reach (0.53 at most), and within one model family larger encoders decode the factors better while devoting a smaller share of their variance to them. Pretraining thus adds latent-aligned structure to a rich feature map rather than replacing it with a low-dimensional one.

Should We Expect Linear World Models? Probably not, and the simulation suggests why: a network trained to report the latents became a linear world model only when its readout was penalized, which makes low-variance encodings of the latents costly; without the penalty it decoded as well but kept substantial variance off the latent subspace. Self-supervised objectives exert no comparable pressure towards an externally chosen factor set, and a general-purpose encoder serving many downstream tasks has good reason to represent more than any one; power-law spectra in visual cortex [31] point the same way for biological representations. The strict notion of Def. 1 is thus best read as one end of a scale, not a property to expect of pretrained models; the useful empirical question is how much of the representation a given factor set occupies, and where. Objectives explicitly favoring compressed or bottlenecked codes [50, 51] are the natural place to look for encoders closer to that end.

What This Means for Probe-Based Claims. Much of interpretability, and many claims that models contain world models, rest on the first picture: a probed direction is one of the axes along which the model organizes its inputs. Closer to the second picture, a probe can be accurate while the structure it ignores governs behavior out of distribution, under composition, or when the probed direction is used for an intervention. Like Muzellec and Kar [30], we therefore suggest reporting, beside probe accuracy, reverse predictivity against its ceiling, where the latent-aligned variance sits in the spectrum, and transfer across conditions. The converse failure also occurs: under superposition, a representation linear in its factors can still defeat a linear probe [15, 16]. Probe accuracy is thus neither necessary nor sufficient evidence for a linear representation, and the two failure modes call for different diagnostics.

Scope and Limitations. The diagnostics test whether a specified factor set carries the dominant directions of variance of a representation (Def. 1). Low scores rule out a linear world model of those factors, not other structure, and our conclusions hold for the encoders and factor sets tested. Reverse predictivity depends on both: intermediate values must be read against the ceiling of Prop. 3, and for two of six encoder–benchmark pairs the sign of the comparison depends on the factor encoding (§5.4). Being linear and variance-based, reverse predictivity and spectral concentration underweight factors on sparse, low-variance or superposed directions, which transfer tests and a dictionary basis such as a sparse autoencoder detect better, and score a factor encoded faithfully along a curved manifold as low, which kernel-PCA or intrinsic-dimension controls would separate. CelebA extends the results to annotated photographs; open scenes without a specified factor set remain untested. Probes in §5 and on CelebA use a fixed ridge penalty (cross-validated elsewhere), and the simulation uses a supervised loss rather than the self-supervised losses behind the encoders studied.

Future Work. Measuring how fast the forward gap between pretrained and random encoders closes as the feature dimension grows (Prop. 1), for larger encoders and concatenated patch tokens, would separate feature richness from learned structure. The diagnostics apply unchanged to language models, including prompt-conditioned representations, and to published probes such as board-state [52, 53] or space and time [54] directions, once their factor sets are specified. For steering and interventions [55, 56], we expect directions from representations that fail the diagnostics to work in distribution but degrade under composition and shift, which is directly testable.

Acknowledgments and Disclosure of Funding

We would like to thank Habon Issa, Hazel Nam, Sunny Liu, Junze Zhou, Sonia Joseph and Liv Gorton for insightful discussions and feedback. Christian Internò acknowledges funding from the Honda Research Institute Europe. David Klindt acknowledges the CSHL GPU cluster with assistance from the US National Institutes of Health Grant S10OD028632-01. This work was supported by a Goodfire Research Grant and conducted using Silico, Goodfire's interpretability research agent.

References

[1] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing Sprites dataset. `https://github.com/deepmind/dsprites-dataset/`, 2017. (Cited on pages 1, 6, and 9.)

[2] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR Workshop*, 2017. (Cited on page 2 and 2.)

[3] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751, 2013. (Cited on page 2 and 2.)

[4] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 1986. (Cited on page 2 and 2.)

[5] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*, 2024. (Cited on page 2 and 2.)

[6] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. (Cited on page 2 and 2.)

[7] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NeurIPS*, 2007. (Cited on pages 2, 2, 3, and 5.)

[8] Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995. Published 1996, Springer Lecture Notes in Statistics 118. (Cited on page 2.)

[9] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *ICLR*, 2018. (Cited on page 2.)

[10] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. (Cited on page 2 and 2.)

[11] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *EMNLP-IJCNLP*, 2019. (Cited on page 2.)

[12] Yonatan Belinkov. Probing classifiers: promises, shortcomings, and advances. *Computational Linguistics*, 2022. (Cited on page 2.)

[13] Benjamin A. Levinstein and Daniel A. Herrmann. Still no lie detector for language models: probing empirical and conceptual roadblocks. *Philosophical Studies*, 182(7):1539–1565, February 2024. ISSN 1573-0883. doi: 10.1007/s11098-023-02094-3. URL `http://dx.doi.org/10.1007/s11098-023-02094-3`. (Cited on page 2.)

[14] Christian Internò, Jumpei Yamaguchi, Loren Amdahl-Culleton, Markus Olhofer, David Klindt, and Barbara Hammer. The observer effect in world models: Invasive adaptation corrupts latent physics, 2026. URL `https://arxiv.org/abs/2602.12218`. (Cited on page 2.)

[15] Vitória Barin Pacela, Shruti Joshi, Isabela Camacho, Simon Lacoste-Julien, and David Klindt. Stop probing, start coding: Why linear probes and sparse autoencoders fail at compositional generalisation, 2026. (Cited on pages 2 and 10.)

[16] David Klindt, Charles O'Neill, Patrik Reizinger, Harald Maurer, and Nina Miolane. From superposition to sparse codes: interpretable representations in neural networks. *arXiv preprint arXiv:2503.01824*, 2025. (Cited on pages 2 and 10.)

[17] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024. (Cited on page 2.)

[18] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. (Cited on page 2.)

[19] David Ha and Jürgen Schmidhuber. World models. In *NeurIPS*, 2018. (Cited on page 2.)

[20] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020. (Cited on page 2.)

[21] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. OpenAI technical report. (Cited on page 2.)

[22] Yann LeCun. A path towards autonomous machine intelligence, 2022. Open Review preprint. (Cited on page 2.)

[23] Aapo Hyvärinen, Hiroaki Sasaki, and Richard E. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *AISTATS*, 2019. (Cited on page 2.)

[24] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. In *AISTATS*, 2020. (Cited on page 2.)

[25] Geoffrey Roeder, Luke Metz, and Diederik P. Kingma. On linear identifiability of learned representations. In *ICML*, 2021. (Cited on page 2.)

[26] Emanuele Marconato, Sebastien Lachapelle, Sebastian Weichwald, and Luigi Gresele. All or none: Identifiable linear properties of next-token predictors in language modeling. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan, editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 4123–4131. PMLR, 03–05 May 2025. URL `https://proceedings.mlr.press/v258/marconato25a.html`. (Cited on page 2.)

[27] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999. (Cited on page 2.)

[28] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019. (Cited on page 2.)

[29] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv:1812.02230*, 2018. (Cited on page 2.)

[30] Sabine Muzellec and Kohitij Kar. Reverse predictivity: Going beyond one-way mapping to compare artificial neural network models and brains. *bioRxiv*, pages 2025–08, 2025. (Cited on pages 2, 4, 10, and 10.)

[31] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571: 361–365, 2019. (Cited on pages 2 and 10.)

[32] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *ICML*, 2020. (Cited on page 2.)

[33] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12:2914, 2021. (Cited on page 2.)

[34] Henning Sprekeler, Tiziano Zito, and Laurenz Wiskott. An extension of slow feature analysis for nonlinear blind source separation. *Journal of Machine Learning Research*, 15:921–947, 2014. (Cited on pages 2 and 5.)

[35] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018. (Cited on pages 3 and 4.)

[36] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *ICLR*, 2020. (Cited on page 3.)

[37] Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter V. Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. In *ICLR*, 2022. (Cited on pages 3, 4, and 25.)

[38] Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jerome Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4): 954–967, 2020. (Cited on pages 3 and 4.)

[39] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019. (Cited on page 4.)

[40] Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. *Advances in Neural Information Processing Systems*, 36:25112–25150, 2023. (Cited on page 4.)

[41] Chris Burgess and Hyunjik Kim. 3D Shapes Dataset. `https://github.com/deepmind/3d-shapes`, 2018. (Cited on pages 6 and 9.)

[42] 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. (Cited on pages 6 and 9.)

[43] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. (Cited on page 6.)

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. (Cited on page 9.)

[45] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pages 10347–10357, 2021. (Cited on page 9.)

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pages 8748–8763, 2021. (Cited on page 9.)

[47] Muhammad Waleed Gondal, Manuel Wüthrich, Đorđe Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: A new disentanglement dataset. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 15714–15725, 2019. (Cited on page 9.)

[48] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. (Cited on page 9.)

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. (Cited on page 9.)

[50] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. (Cited on page 10.)

[51] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. (Cited on page 10.)

[52] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ICLR*, 2023. (Cited on page 10.)

[53] Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *BlackboxNLP Workshop, EMNLP*, 2023. (Cited on page 10.)

[54] Wes Gurnee and Max Tegmark. Language models represent space and time. *ICLR*, 2024. (Cited on page 10.)

[55] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=aLLuYpn83y`. (Cited on page 10.)

[56] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2025. URL `https://openreview.net/forum?id=2XBPdPIcFK`. (Cited on page 10.)

[57] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NeurIPS*, 2008. (Cited on page 16 and 16.)

[58] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017. (Cited on page 16 and 16.)

[59] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *NeurIPS*, 2017. (Cited on page 16.)

Appendix Contents

A Full Theoretical Statements and Proofs **16**

A.1 Proposition 1 (Forward Decodability from Rich Random Features) 16

A.2 Proposition 2 (Reverse-Predictivity Asymmetry) 16

A.3 Proposition 3 (Ceiling) 17

B Simulation Details **18**

B.1 Specification 18

B.2 Which Spectral Statistic Fig. 3 Shows 18

B.3 Non-Degeneracy Audit 18

C SVG-World Dataset Details **19**

D Experimental Details **19**

D.1 Encoders on SVG-World 20

D.2 Controlled Benchmarks 20

D.3 CelebA 21

E Reverse Predictivity: Ablations **22**

F Cross-Domain Transfer: Per-Direction Results **22**

G Binding with Multiple Objects **23**

H SVG-Faces: A Single-Object Precursor Benchmark **23**

A Full Theoretical Statements and Proofs

This appendix gives the formal version of Prop. 1 and the proofs of Props. 1, 2 and 3.

A.1 Proposition 1 (Forward Decodability from Rich Random Features)

Formal statement. Let $\mathcal{X}$ be a compact metric space with data distribution $p_{\mathcal{X}}$, and let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a continuous, bounded, positive-definite kernel with reproducing kernel Hilbert space $\mathcal{H}_\kappa$. Suppose each latent coordinate satisfies $z_j \in \mathcal{H}_\kappa$ with $\|z_j\|_{\mathcal{H}_\kappa} \le R$. Let $\mathbb{P}_\omega$ be a distribution over parameters ω and $\psi_\omega : \mathcal{X} \to \mathbb{R}$ feature functions such that

$$\mathbb{E}_\omega[\psi_\omega(x)\,\psi_\omega(x')] \;=\; \kappa(x,x') \qquad \text{for all } x, x' \in \mathcal{X}, \tag{6}$$

with $\sup_x |\psi_\omega(x)| \le K$ almost surely. Draw $\omega_1, \dots, \omega_d$ i.i.d. from $\mathbb{P}_\omega$ and set $f_{\mathrm{rand}}(x) = d^{-1/2}\,(\psi_{\omega_1}(x), \dots, \psi_{\omega_d}(x))^\top \in \mathbb{R}^d$. Then for any $\epsilon > 0$ and $\delta \in (0,1)$ and all

$$d \;\ge\; d_{\epsilon,\delta} \;=\; \frac{k\,K^2 R^2}{\epsilon\,\delta}, \tag{7}$$

with probability at least $1-\delta$ over $\omega_1, \dots, \omega_d$ there exists $W \in \mathbb{R}^{k \times d}$ with

$$\mathbb{E}_{x \sim p_{\mathcal{X}}}\Big[\|W f_{\mathrm{rand}}(x) - z(x)\|^2\Big] \;\le\; \epsilon. \tag{8}$$

Proof. Fix $j \in \{1, \dots, k\}$. Because κ has the integral representation (6), every $g \in \mathcal{H}_\kappa$ can be written as $g(x) = \mathbb{E}_\omega[\beta(\omega)\,\psi_\omega(x)]$ for some $\beta \in L^2(\mathbb{P}_\omega)$, and $\|g\|_{\mathcal{H}_\kappa}$ is the smallest $\|\beta\|_{L^2(\mathbb{P}_\omega)}$ that achieves this [57, 58]. Choose such a β_j for z_j, so that

$$z_j(x) \;=\; \mathbb{E}_\omega[\beta_j(\omega)\,\psi_\omega(x)]\,, \qquad \mathbb{E}_\omega\big[\beta_j(\omega)^2\big] \;\le\; R^2. \tag{9}$$

Let $w_j = d^{-1/2}\,(\beta_j(\omega_1), \dots, \beta_j(\omega_d))^\top$, so that

$$\hat{z}_j(x) \;:=\; w_j^\top f_{\mathrm{rand}}(x) \;=\; \frac{1}{d}\sum_{i=1}^{d} \beta_j(\omega_i)\,\psi_{\omega_i}(x). \tag{10}$$

For fixed x, $\hat{z}_j(x) - z_j(x)$ is an average of d i.i.d. zero-mean terms, so

$$\mathbb{E}_{\omega_{1:d}}\big[(\hat{z}_j(x) - z_j(x))^2\big] \;\le\; \frac{1}{d}\,\mathbb{E}_\omega\big[\beta_j(\omega)^2\,\psi_\omega(x)^2\big] \;\le\; \frac{K^2R^2}{d}. \tag{11}$$

Stacking the w_j as the rows of W, summing over j and exchanging the expectations over x and $\omega_{1:d}$ gives

$$\mathbb{E}_{\omega_{1:d}}\Big[\mathbb{E}_x\Big[\|W f_{\mathrm{rand}}(x) - z(x)\|^2\Big]\Big] \;\le\; \frac{k\,K^2R^2}{d}. \tag{12}$$

By Markov's inequality, the inner expectation exceeds ϵ with probability at most $kK^2R^2/(d\,\epsilon)$, which is at most δ for $d \ge d_{\epsilon,\delta}$. □

Remarks. When the β_j are bounded, a concentration inequality improves the dependence on δ to $\log(1/\delta)$ [57], and Bach [58] and Rudi and Rosasco [59] give sharper rates under spectral assumptions on κ. For our purposes the content is qualitative: for targets in $\mathcal{H}_\kappa$, the number of random features needed for ϵ-accurate linear decoding is finite, and grows linearly in the number of latents and in their squared RKHS norm.

A.2 Proposition 2 (Reverse-Predictivity Asymmetry)

Statement (restated). Suppose $f(x) = Az(x) + \mu + \varepsilon(x)$ with $\mathbb{E}[\varepsilon(x)\,z(x)^\top] = 0$. Then the optimal reverse probe achieves

$$R^2_{\mathrm{rev}} \;=\; \frac{\mathrm{tr}(A\ \mathrm{Cov}(z)\,A^\top)}{\mathrm{tr}(A\ \mathrm{Cov}(z)\,A^\top) + \mathrm{tr}\,\mathrm{Cov}(\varepsilon)}. \tag{13}$$

This quantity can be arbitrarily small even when forward $R^2 = 1$.

Proof of (13)**.** Without loss of generality assume $\mathbb{E}[z] = 0$; the constant μ is absorbed into the intercept of the reverse probe. The least-squares reverse probe is

$$W_{\mathrm{rev}} \;=\; \mathbb{E}[f\, z^\top]\; \mathrm{Cov}(z)^{+}, \tag{14}$$

where $^+$ denotes the Moore–Penrose pseudoinverse. The pseudoinverse matters in practice: on SVG-World twelve of the 32 stored latent coordinates are constant, so $\mathrm{Cov}(z)$ is singular. Substituting $f = Az+\varepsilon$ and using $\mathbb{E}[\varepsilon\, z^\top] = 0$ gives $\mathbb{E}[f\, z^\top] = A\ \mathrm{Cov}(z)$, so $W_{\mathrm{rev}}\, z = A\ \mathrm{Cov}(z)\, \mathrm{Cov}(z)^{+} z = A\, P z$, with P the orthogonal projector onto the range of $\mathrm{Cov}(z)$. Since centered z lies in that range almost surely, $Pz = z$, so $W_{\mathrm{rev}}\, z = Az$ and the residual is $f - W_{\mathrm{rev}}\, z = \varepsilon$. Hence

$$\mathbb{E}\Big[\|f - W_{\mathrm{rev}}\, z\|^2\Big] \;=\; \mathrm{tr}\,\mathrm{Cov}(\varepsilon). \tag{15}$$

For the denominator, the cross term vanishes by $\mathbb{E}[\varepsilon\, z^\top] = 0$, so

$$\mathbb{E}\Big[\|f - \mathbb{E}f\|^2\Big] \;=\; \mathrm{tr}(A\ \mathrm{Cov}(z)\, A^\top) + \mathrm{tr}\,\mathrm{Cov}(\varepsilon). \tag{16}$$

Combining,

$$R^2_{\mathrm{rev}} \;=\; 1 - \frac{\mathrm{tr}\,\mathrm{Cov}(\varepsilon)}{\mathrm{tr}(A\ \mathrm{Cov}(z)\, A^\top) + \mathrm{tr}\,\mathrm{Cov}(\varepsilon)} \;=\; \frac{\mathrm{tr}(A\ \mathrm{Cov}(z)\, A^\top)}{\mathrm{tr}(A\ \mathrm{Cov}(z)\, A^\top) + \mathrm{tr}\,\mathrm{Cov}(\varepsilon)}. \qquad \square$$

Asymmetry between forward and reverse R^2. Let $z \in \mathbb{R}^k$ with $\mathrm{Cov}(z) = I_k$, and let $\eta(x) \in \mathbb{R}^M$ satisfy $\mathrm{Cov}(\eta) = s^2 I_M$ and $\mathbb{E}[\eta\, z^\top] = 0$, for example a nonlinear lift that is uncorrelated with z. Define $f(x) = (z(x)^\top, \eta(x)^\top)^\top \in \mathbb{R}^{k+M}$. Then $f = Az + \varepsilon$ with $A = (I_k\ \ 0)^\top$ and $\varepsilon = (0, \eta^\top)^\top$. The forward probe $(I_k\ \ 0)$ recovers z exactly, so forward $R^2 = 1$, while (13) gives

$$R^2_{\mathrm{rev}} \;=\; \frac{k}{k + s^2 M} \;\longrightarrow\; 0 \qquad (M \to \infty). \tag{17}$$

This is the kernel-machine regime: z is exactly linearly recoverable but carries a vanishing share of the variance of f. $\square$

Remark. In the simulation of §4, the network trained with weight decay reaches R^2_{rev} close to 1 and the unregularized one does not, consistent with the hypothesis that penalizing the readout suppresses variance of the η type. Pretrained encoders lie between the two regimes (§§5.2–5.4).

A.3 Proposition 3 (Ceiling)

Statement (restated). For any f, let $\lambda_1 \geq \cdots \geq \lambda_d$ be the eigenvalues of $\mathrm{Cov}(f)$ and $r = \mathrm{rank}\,\mathrm{Cov}(z)$. Then

$$R^2_{\mathrm{rev}} \;\leq\; C_r \;=\; \frac{\sum_{i \leq r} \lambda_i}{\sum_{i \leq d} \lambda_i}, \tag{18}$$

with equality iff the range of W_{rev} lies in a leading r-dimensional principal subspace of f and the reverse-probe residual $f - W_{\mathrm{rev}} z$ has no variance in that range.

Proof of (18)**.** Without loss of generality assume $\mathbb{E}[z] = 0$ and $\mathbb{E}[f] = 0$. Let $\hat{f} = W_{\mathrm{rev}}\, z$ be the optimal linear predictor of f from z and $e = f - \hat{f}$ the residual. The normal equations give $\mathbb{E}[e\, z^\top] = 0$, hence $\mathbb{E}[e\, \hat{f}^\top] = 0$ and

$$\Sigma \;:=\; \mathrm{Cov}(f) \;=\; S + \mathrm{Cov}(e), \qquad S \;:=\; \mathrm{Cov}(\hat{f}) \;=\; W_{\mathrm{rev}}\ \mathrm{Cov}(z)\, W_{\mathrm{rev}}^\top. \tag{19}$$

Both terms are positive semidefinite, so $S \preceq \Sigma$, and $\mathrm{rank}\, S \leq \mathrm{rank}\,\mathrm{Cov}(z) = r$. As in the proof of (13),

$$R^2_{\mathrm{rev}} \;=\; 1 - \frac{\mathrm{tr}\,\mathrm{Cov}(e)}{\mathrm{tr}\,\Sigma} \;=\; \frac{\mathrm{tr}\, S}{\mathrm{tr}\,\Sigma}. \tag{20}$$

Let P be the orthogonal projector onto $\mathrm{range}(S)$, of rank at most r. Since $PSP = S$ and $S \preceq \Sigma$,

$$\mathrm{tr}\, S \;=\; \mathrm{tr}(PSP) \;\leq\; \mathrm{tr}(P\, \Sigma\, P) \;\leq\; \sum_{i \leq r} \lambda_i, \tag{21}$$

where the last step is Ky Fan's maximum principle: over orthogonal projectors of rank at most r, $\mathrm{tr}(P\,\Sigma\,P)$ is maximized by a projector onto a leading r-dimensional eigenspace of Σ. Dividing by $\mathrm{tr}\,\Sigma = \sum_{i \le d} \lambda_i$ gives $R^2_{\mathrm{rev}} \le C_r$.

Equality in the first inequality holds iff $\mathrm{tr}(P\ \mathrm{Cov}(e)\ P) = 0$, that is, iff the residual has no variance in $\mathrm{range}(S)$. Equality in the second holds iff this range lies in a leading r-dimensional principal subspace of Σ (any such subspace, if eigenvalues are tied). □

Remark. The bound depends on f only through its spectrum, so R^2_{rev}/C_r is the fraction of the variance attainable by a rank-r linear image of the latents that the specified factors actually take. For a scalar factor, $C_1 = \lambda_1/\,\mathrm{tr}\,\Sigma$, so even a perfectly isolated factor cannot exceed the top-eigenvalue share. The ceiling is generally not attained, so the ratio normalizes against a bound rather than an achievable value, and we report it next to raw R^2_{rev}, not instead of it. It is a population statement; held-out estimates of the ratio can exceed 1 by sampling error.

B Simulation Details

B.1 Specification

Latents $z \sim \mathrm{Unif}([-1,1]^2)$ are lifted by a frozen random Fourier map, $x_j = \cos(w_j^\top z + b_j)$ for $j = 1, \ldots, 256$, with $w_j \sim \mathcal{N}(0, 3^2 I_2)$ and $b_j \sim \mathrm{Unif}[0, 2\pi]$ drawn once from a fixed seed and never trained; a linear map of z explains 0.2452 of the variance of this lift. The network is $f(x) = W_2\,\mathrm{ReLU}(W_1 x + b_1) + b_2$ with $W_1 \in \mathbb{R}^{h \times 256}$ and $W_2 \in \mathbb{R}^{h \times h}$, so h is both the hidden and the feature width, with PyTorch's default initialization. It is trained jointly with a linear readout $\hat{z} = Uf$ under mean-squared error with AdamW (learning rate $2 \cdot 10^{-3}$, 2000 steps, batch 256, weight decay $\lambda \in \{0, 1\}$ on all parameters). We use $8{,}000$ training and $2{,}000$ test points and widths $h \in \{4, 8, \ldots, 1024\}$. Probes are ridge regressions with penalty 10^{-3}; for OOD R^2 the probe is trained on $\|z\| < 0.8$ and tested on $\|z\| \ge 0.8$. Fig. 3 averages 3 seeds; Table 1 and the audit below use 5.

B.2 Which Spectral Statistic Fig. 3 Shows

The spectral-concentration panel of Fig. 3 plots the share of total feature variance that lies in the top $r = 2$ principal components of f and is linear in z, computed without whitening. This is not the whitened statistic SC defined in §3.3, and the two behave differently here (Table 1). Under the whitened definition both trained networks stay close to 1 at every width, so the dip of the unregularized network disappears, while random initialization rises with width, from 0.16 at $h = 4$ to 0.31 at $h = 64$ and 0.71 at $h = 1024$. In the simulation, whitened SC therefore separates trained from random networks clearly at small and moderate widths but only partly at $h = 1024$; on SVG-World and CelebA it separates every pretrained transformer from its random initialization by a factor of five or more (Tables 5 and 9). Fig. 4(d) projects onto the top m PCs without whitening.

Table 1: **Simulation diagnostics at** $h = 1024$ (mean over 5 seeds). Unwhitened top-2: share of total variance in the top two PCs that is linear in z, the statistic plotted as spectral concentration in Fig. 3. SC: whitened definition of §3.3. This run is independent of the one reported in §4; reverse R^2 differs by at most 0.003.

Condition	Fwd R^2	R^2_{rev}	C_2	Unwhitened top-2	SC	OOD R^2
Trained, $\lambda = 1$	1.000	0.996	0.996	0.995	1.000	1.000
Trained, $\lambda = 0$	1.000	0.786	0.797	0.781	0.983	0.996
Random init	1.000	0.213	0.258	0.181	0.713	0.892

B.3 Non-Degeneracy Audit

At $h = 1024$ (5 seeds, 2000 held-out points), the network trained with $\lambda = 1$ visits 1988.2 distinct ReLU activation regions on average (minimum 1983); 93.67% of units are active on between 5% and 95% of inputs; the dead fraction is 0.0359 and the mean active fraction 0.2683; and the input–output Jacobian has a coefficient of variation of 0.6573 across test points. An affine network would have one region and a constant Jacobian; no seed at any width did. As a calibration, forcing every pre-activation

to be positive, which makes the network affine on its data, gives 1 region, active fraction 1.000 and Jacobian coefficient of variation 2.4×10^{-15}, so the audit detects the failure it is meant to rule out. High reverse R^2 alone would not establish nonlinearity: an affine map that projects x onto the two directions of its latent-predictable component reaches $R^2_{\text{rev}} = 0.8992$ (trained network in the same run: 0.9936), so the argument rests on the activation statistics above.

Spectrum. On a linear scale, the top two eigenvalues of $\mathrm{Cov}(f)$ carry 99.57% of the variance (0.5657 and 0.4300, against 0.1384 and 0.1151 at random initialization). Later eigenvalues are small but nonzero, so the covariance is nearly, not exactly, rank two.

C SVG-World Dataset Details

Table 2 summarizes the dataset. The latent vector has 32 dimensions, of which 20 are active: for each of the four people, an (x, z) position, a facing direction and a skin hue; for each of the two animals, an (x, z) position. Facing is discretized inside the renderer as $\lfloor 3.999\, z \rfloor$, so it takes four values and its R^2 is bounded below 1 even for a perfect probe. The remaining 12 dimensions are shirt colors fixed by the renderer and are excluded from all probes.

Table 2: SVG-World dataset card.

Field	Value
Latent vector	$z \in [0, 1]^{32}$; 20 active, 12 fixed by the renderer
Person $\times 4$	(x, z) position in a fixed lane; facing $\in$ {S, E, N, W}; skin hue $0.05 + 0.06\, z$ (HSV)
Animal $\times 2$	(x, z) position in a fixed lane (AquaWorld: dolphin, turtle; WestWorld: brown and white horse)
Fixed (12 dims)	shirt HSV per person; excluded from all probes
Sampling	$z \sim \mathcal{U}[0, 1]^{32}$ i.i.d., seed 42
Occlusion	objects drawn back to front, so they may overlap
Rendering	SVG $\rightarrow$ `cairosvg`; one layout, two styles
Rasterization	600×420 SVG padded to 600×600, resized to 224×224 RGB
Splits	AquaWorld, WestWorld; $N = 10{,}000$ each; 50/50 train/test
Probed in the main text	the 12 position latents

D Experimental Details

This appendix gives the encoders, the probing protocols, compute, and the full results behind §5.4.

Encoders. All checkpoints are HuggingFace models pinned to a fixed revision (Table 3). We take one pooled vector from the final block, before any task or projection head. Each random control is the same architecture built from its configuration with default initialization; DINOv3-B/16 uses seeds 42–46 and the other encoders seed 42 unless stated otherwise. Inputs are 224×224, preprocessed as specified in each model's pinned preprocessor configuration. For ResNet-50, the random control keeps its untrained BatchNorm running statistics; calibrating them on the evaluation images would no longer be a data-free control, so we report the uncalibrated control and keep ResNet-50 out of all counts.

Table 3: Encoders, checkpoints and readouts.

Encoder	Checkpoint	Readout (d)
DINOv3-S/16	`facebook/dinov3-vits16-pretrain-lvd1689m`	CLS (384)
DINOv3-B/16	`facebook/dinov3-vitb16-pretrain-lvd1689m`	CLS or mean patch (768)
DINOv3-L/16	`facebook/dinov3-vitl16-pretrain-lvd1689m`	CLS (1024)
DINOv2-B/14	`facebook/dinov2-base`	CLS (768)
DeiT-B/16	`facebook/deit-base-patch16-224`	CLS (768)
CLIP-B/16	`openai/clip-vit-base-patch16`	pooled CLS, before projection (768)
ResNet-50	`microsoft/resnet-50`	global average pool (2048)

Probes. All probes are ridge regressions with an intercept. Two protocols are used. The DINOv3-B/16 analyses of §§5.1–5.3 (Table 4, Figs. 6, 7 and 8, and App. E) use a fixed penalty $\alpha = 1$ and one 50/50 split (seed 42) with the same indices in both worlds. All intervals in §5.2 and all results in §5.4 use the revised protocol: α chosen by 5-fold cross-validation inside the training half over $[10^{-3}, 10^{5}]$, five repeated 50/50 splits over scene identifiers (the two styles share each split), float64, and held-out reverse $R^2 = 1 - \sum_j \left\| f_j - \hat{f}_j \right\|^2 / \sum_j \left\| f_j - \bar{f}_j \right\|^2$ on the test half. Point estimates are on the shared split; intervals are paired bootstraps over held-out units with 10,000 resamples, made hierarchical over units and control initializations where there are five controls. For DINOv3-B/16 on SVG-World, moving from the first protocol to the second changes R^2_{rev} by at most 0.0042 in every condition. It does change forward R^2 for random initializations: their features have 14 to 29 times less total variance than pretrained features, so a fixed $\alpha = 1$ regularizes them much more strongly. With cross-validated α, random-init DINOv3-B/16 reaches forward $R^2 = 0.66$ on CLS features (Table 5), against 0.33 under the fixed penalty (Table 4); pretrained forward R^2 changes little (0.91 against 0.88). Cross-style probes are fitted on the training half of the source world and evaluated on the test half of the target world, with features centered on each world's own mean. For spectral concentration, PCA whitening is fitted on training rows only.

Factor encodings. On dSprites, 3D Shapes and MPI3D-realistic we draw 20,000 factor-balanced images (each factor's marginal is uniform, every factor combination appears at most once; sample seed 1234, with seeds 7 and 2024 as replicates) and split them 50/50 by factor combination. The main results use the *canonical* encoding: ordinal factors min-max scaled to $[0, 1]$, cyclic factors as a $(\sin, \cos)$ pair over the period of each dataset's stored angle grid, categorical factors excluded. Two alternatives test sensitivity to this choice: *ordinal*, every factor as its min-max-scaled index with no trigonometric encoding, and *with one-hot*, the canonical block plus one-hot columns for the categorical factors. The factor rank r is the rank of the encoded target block on the training half. The checkerboard test holds out half of the joint combinations of the two factors with the most values (those with odd index sum), so every marginal value is seen in training but half of the combinations are not.

Compute. Feature extraction runs on a single NVIDIA H100 GPU and dominates the cost: DINOv3-B/16 processes about 1,100 images per second and DINOv3-L/16 about 330, so a pretrained and random pair on 20,000 images takes under three minutes. The full CelebA run takes 20–30 minutes; the simulation and all probes run on CPU.

D.1 Encoders on SVG-World

Table 4 gives the per-latent forward decoding behind §5.1 and Fig. 6 (fixed-penalty protocol).

Table 5 reports the seven-encoder panel behind §5.4, under the revised protocol and averaged over the two styles. Every transformer has a reverse-predictivity gap whose interval excludes zero, and for every transformer pretraining raises spectral concentration by a factor of five or more. The cross-style correlation separates most encoders from their controls but not all: CLIP's random initialization transfers almost as well as the pretrained model (0.31 against 0.34), and ResNet-50's transfers better. On AquaWorld CLS features, pretrained DINOv3-B/16 needs the top 17 principal components to contain 90% of its latent-predictable variance, against 55–59 for its five random initializations.

D.2 Controlled Benchmarks

Table 6 gives the full results behind §5.4. For DINOv3-B/16 on dSprites and 3D Shapes, the random initialization has a much higher R^2_{rev} than the pretrained encoder, and normalizing by the ceiling does not change this, because the two ceilings are similar (0.81 and 0.82 on dSprites). These random networks are nearly one-dimensional (participation ratio 1.9 and 3.8), and most of that variance is linear in the factors. On renders of a single object against a plain background, simple factors such as position, scale and color account for much of the pixel variance: on raw 32×32 pixels, R^2_{rev} is 0.84 on 3D Shapes, against 0.17 on SVG-World (Table 8). A random encoder that stays close to its input can therefore score highly on D1 without having learned anything, which is why the diagnostics are necessary rather than sufficient conditions. On MPI3D-realistic the ceilings differ (pretrained 0.35, random 0.96), and after normalization the pretrained encoder is ahead (0.53 against 0.42). CLIP's

Table 4: **DINOv3-B/16 Forward Decoding on SVG-World, per-dim within-domain.** Pearson ρ and R^2 from a ridge probe trained on each world separately, averaged over AquaWorld and WestWorld. *Pre*: pretrained. *Rand*: freshly initialized, no training. The pretraining contribution is clearest on small foreground objects (animal $\Delta\rho = 0.58$ on CLS, 0.68 on patches). Position latents only; fixed ridge penalty $\alpha = 1$.

	CLS token				Mean-pooled patches			
	ρ		R^2		ρ		R^2	
latent	pre	rand	pre	rand	pre	rand	pre	rand
`p0_x`	0.97	0.71	0.94	0.47	0.93	0.68	0.85	0.41
`p0_z`	0.95	0.80	0.89	0.61	0.90	0.76	0.80	0.55
`p1_x`	0.91	0.49	0.82	0.21	0.82	0.44	0.65	0.16
`p1_z`	0.95	0.72	0.89	0.50	0.90	0.70	0.80	0.45
`p2_x`	0.93	0.50	0.86	0.20	0.86	0.46	0.71	0.16
`p2_z`	0.95	0.60	0.90	0.32	0.90	0.58	0.79	0.25
`p3_x`	0.95	0.68	0.91	0.42	0.91	0.63	0.81	0.30
`p3_z`	0.93	0.74	0.87	0.52	0.88	0.73	0.76	0.42
`dolphin_x`	0.93	0.42	0.86	0.25	0.95	0.24	0.89	0.07
`dolphin_z`	0.97	0.50	0.94	0.31	0.98	0.45	0.96	0.25
`turtle_x`	0.88	0.18	0.76	0.03	0.94	0.13	0.86	0.02
`turtle_z`	0.95	0.32	0.90	0.09	0.97	0.29	0.94	0.06
person mean	**0.94**	**0.65**	**0.89**	**0.41**	**0.89**	**0.62**	**0.77**	**0.34**
animal mean	**0.93**	**0.35**	**0.87**	**0.17**	**0.96**	**0.28**	**0.91**	**0.10**
all mean	**0.94**	**0.55**	**0.88**	**0.33**	**0.91**	**0.51**	**0.82**	**0.26**

Table 5: **Encoders on SVG-World** (12 position latents, $r = 12$; mean over the two styles; random control seed 42). Gap: R^2_{rev} pretrained minus random, with 95% paired bootstrap interval. SC: spectral concentration $R^2_{\text{rev}}(r)$. $\rho_\times$: mean cross-style correlation over both transfer directions. Each cell: pretrained / random.

Encoder (readout)	Fwd R^2	R^2_{rev}	Gap	R^2_{rev}/C_r	SC	$\rho_\times$
DINOv3-S/16 (CLS)	0.86 / 0.45	0.257 / 0.023	0.235 [0.229, 0.242]	0.38 / 0.04	0.27 / 0.03	0.47 / 0.09
DINOv3-B/16 (CLS)	0.91 / 0.66	0.188 / 0.023	0.164 [0.160, 0.169]	0.31 / 0.04	0.26 / 0.03	0.46 / 0.10
DINOv3-B/16 (patches)	0.93 / 0.59	0.284 / 0.024	0.259 [0.253, 0.264]	0.47 / 0.05	–	0.57 / 0.15
DINOv3-L/16 (CLS)	0.91 / 0.74	0.169 / 0.024	0.145 [0.139, 0.151]	0.26 / 0.05	0.24 / 0.02	0.48 / 0.06
DINOv2-B/14 (CLS)	0.91 / 0.58	0.224 / 0.028	0.196 [0.190, 0.202]	0.35 / 0.05	0.26 / 0.04	0.34 / 0.07
DeiT-B/16 (CLS)	0.89 / 0.55	0.186 / 0.020	0.165 [0.159, 0.170]	0.27 / 0.04	0.23 / 0.02	0.26 / 0.02
CLIP-B/16 (pooled)	0.87 / 0.45	0.174 / 0.020	0.153 [0.148, 0.158]	0.27 / 0.04	0.21 / 0.02	0.34 / 0.31
ResNet-50 (pool)	0.93 / 0.95	0.325 / 0.220	0.105 [0.100, 0.111]	0.55 / 0.49	0.40 / 0.35	0.27 / 0.46

random initialization on dSprites is not low-dimensional (participation ratio 18.7), and there the pretrained encoder is ahead on both raw and normalized R^2_{rev}. The checkerboard test follows forward R^2 closely: pretrained ahead on dSprites and MPI3D-realistic, tied on 3D Shapes.

The R^2_{rev} gaps are stable across control initializations (hierarchical intervals exclude zero in every cell) and across dataset samples (no gap moves by more than 0.0204 over three samples), but not across factor encodings (Table 7). Of the six transformer–benchmark pairs, four keep the sign of the gap under all three encodings; DINOv3-B/16 on 3D Shapes and CLIP-B/16 on MPI3D-realistic do not. A reverse-predictivity comparison is therefore defined only for a fully specified factor set and its encoding, and should not be read as a score of encoder quality.

D.3 CelebA

We sample 20,000 images from the CelebA training partition (seed 0) and use its 40 binary attributes, coded in $\{0, 1\}$, as the factor set ($r = 40$). DINOv3-B/16 uses the checkpoint above; CLIP-B/16 is the OpenAI ViT-B/16 model in `open_clip`, whose image embedding we probe. Random controls use seeds 0–4. Probes use a fixed $\alpha = 1$ and one 50/50 split (seed 42); intervals are percentile bootstraps

Table 6: **Controlled benchmarks** (canonical encoding, sample seed 1234; pretrained / mean of five random initializations). D3: R^2 on held-out checkerboard combinations, against random seed 42 only. ResNet-50 uses the data-free control and is reported separately.

Dataset (r)	Encoder	Fwd R^2	R^2_{rev}	C_r	R^2_{rev}/C_r	SC	PR	D3
dSprites (5)	DINOv3-B/16	0.42 / 0.16	0.09 / 0.61	0.81 / 0.82	0.11 / 0.74	0.08 / 0.41	4.4 / 1.9	0.42 / 0.17
	CLIP-B/16	0.30 / 0.04	0.17 / 0.09	0.67 / 0.46	0.25 / 0.20	0.12 / 0.14	6.1 / 18.7	0.31 / 0.04
3D Shapes (8)	DINOv3-B/16	0.96 / 0.96	0.11 / 0.67	0.70 / 0.92	0.15 / 0.72	0.26 / 0.24	8.7 / 3.8	0.95 / 0.96
	CLIP-B/16	0.96 / 0.97	0.15 / 0.90	0.61 / 1.00	0.25 / 0.91	0.17 / 0.32	16.6 / 2.3	0.95 / 0.97
MPI3D-real. (4)	DINOv3-B/16	0.85 / 0.61	0.19 / 0.40	0.35 / 0.96	0.53 / 0.42	0.26 / 0.27	19.8 / 3.1	0.85 / 0.61
	CLIP-B/16	0.76 / 0.51	0.14 / 0.09	0.35 / 0.97	0.41 / 0.09	0.22 / 0.19	20.8 / 2.6	0.77 / 0.51
dSprites	ResNet-50	0.24 / 0.19	0.25 / 0.54	0.71 / 0.76	0.35 / 0.72	0.25 / 0.46	6.7 / 3.1	0.24 / 0.19
3D Shapes	ResNet-50	0.88 / 0.99	0.08 / 0.39	0.62 / 0.97	0.13 / 0.40	0.19 / 0.39	13.3 / 1.2	0.87 / 0.99
MPI3D-real.	ResNet-50	0.79 / 0.68	0.20 / 0.86	0.37 / 0.99	0.53 / 0.87	0.30 / 0.29	19.2 / 1.1	0.79 / 0.67

Table 7: **Sensitivity to the factor encoding.** R^2_{rev} gap (pretrained minus mean of five random initializations), sample seed 1234.

Dataset	Encoder	Canonical	Ordinal	With one-hot
dSprites	DINOv3-B/16	-0.516	-0.398	-0.287
	CLIP-B/16	$+0.077$	$+0.097$	$+0.128$
3D Shapes	DINOv3-B/16	-0.559	$+0.075$	-0.094
	CLIP-B/16	-0.754	-0.073	-0.492
MPI3D-realistic	DINOv3-B/16	-0.213	-0.344	-0.478
	CLIP-B/16	$+0.057$	-0.213	-0.418
dSprites	ResNet-50	-0.297	-0.332	-0.274
3D Shapes	ResNet-50	-0.316	-0.010	$+0.023$
MPI3D-realistic	ResNet-50	-0.661	-0.639	-0.586

over held-out images with 1,000 resamples. Because the attributes are binary, recoding them (for example as ± 1) is an affine map and leaves every reverse statistic unchanged, so we vary the feature normalization instead: pretrained exceeds random on R^2_{rev} for raw, per-dimension standardized and PCA-whitened features alike (Table 9).

E Reverse Predictivity: Ablations

Standardizing each feature dimension before fitting the reverse probe leaves the aggregate R^2_{rev} of DINOv3-B/16 on SVG-World within 0.01 of its raw value: pretrained CLS goes from 0.181 to 0.175, random from 0.019 to 0.020. The pretrained-vs-random gap is therefore not an artifact of the scale of individual feature dimensions. The whitening curve of Fig. 7 on CLS features, at $m \in \{32, 64, 128, 256, 768\}$, gives $R^2_{\text{rev}}(m) = 0.174, 0.114, 0.067, 0.035, 0.009$ for the pretrained encoder and $0.023, 0.022, 0.018, 0.014, 0.006$ for random initialization. Evaluated at the factor rank $m = r = 12$, the curve gives 0.26 pretrained against 0.03 random (Table 5).

The whitened curve falls for $m > r$ for any representation with nonzero residual, because whitening lifts residual directions to unit variance however small they are; for an exact linear world model of three latents with small isotropic noise it exceeds 0.95 at $m = 3$ and falls below 0.30 at $m = 64$. The decline with m is therefore not evidence against a linear world model. What separates encoders is the value near $m = r$ and how many components are needed to collect the latent-predictable variance.

F Cross-Domain Transfer: Per-Direction Results

Fig. 8 averages the two transfer directions (AquaWorld→WestWorld and WestWorld→AquaWorld). Under the revised protocol the directions differ. On DINOv3-B/16 CLS features, the mean correlation between the transferred probe's predictions and the target latents is 0.381 in one direction and 0.533 in the other, against 0.074 and 0.099 for random initialization (means over five initializations). Transfer R^2 without calibration is not meaningful, because feature scales differ between the two styles and per-world centering removes only the mean difference (-58.8 for the pretrained encoder, around -2×10^4 at random initialization). After rescaling the probe output on half of the target

Table 8: **Pixel-space control.** Diagnostics computed on raw images downsampled to $32 \times 32 \times 3$ ($d = 3072$), canonical encoding.

Dataset	Fwd R^2	R^2_{rev}	C_r	R^2_{rev}/C_r
SVG-World (mean of styles)	0.995	0.166	0.412	0.42
dSprites	0.496	0.213	0.489	0.44
3D Shapes	0.998	0.836	0.885	0.94
MPI3D-realistic	0.636	0.389	0.676	0.58

Table 9: **CelebA** (20,000 images; 40 binary attributes, $r = 40$). Pretrained: 95% bootstrap interval over held-out images; random: mean $\pm$ s.d. over five initializations. Std.: R^2_{rev} after standardizing each feature dimension. SC: spectral concentration, R^2_{rev} on the top r whitened PCs. PR: participation ratio.

Encoder	Features	Weights	Fwd R^2	R^2_{rev}	Std.	C_r	R^2_{rev}/C_r	SC	PR
DINOv3-B/16	CLS	pre	0.391	0.170 [0.168, 0.173]	0.160	0.606	0.281	0.153	48.2
		rand	0.069	0.067 ± 0.003	0.060	0.980	0.069	0.018	2.5
DINOv3-B/16	patches	pre	0.412	0.245 [0.242, 0.248]	0.226	0.680	0.360	0.174	32.1
		rand	0.114	0.090 ± 0.002	0.081	0.994	0.091	0.023	1.7
CLIP-B/16	image	pre	0.411	0.192 [0.191, 0.195]	0.152	0.694	0.277	0.155	45.6
		rand	0.115	0.106 ± 0.001	0.097	0.996	0.106	0.028	1.8

world, transfer R^2 is 0.164 and 0.313 for the pretrained encoder, against 0.011 and 0.018 for random initialization (means over the same five initializations).

G Binding with Multiple Objects

With several objects in a scene, a linear readout must keep each object's attributes paired with that object. SVG-World largely sidesteps this: each person and animal moves within its own fixed lane, so identity is given by location, and each person's position is decoded separately with $R^2 \geq 0.82$ on CLS features (Table 4). The two-face regime of the SVG-Faces precursor (App. H) tests binding directly, since two faces with independently sampled attributes appear anywhere on the canvas and are ordered left to right by their x-coordinate. Here decoding is far weaker than with one face (Fig. 10): per-latent R^2 stays below about 0.45 even for the pretrained encoder. Random initialization matches the pretrained encoder on face position, size and skin color, and fails on eye and mouth geometry and eye color, which pretraining recovers only partially. Neither encoder binds attributes to objects reliably under a single linear readout.

H SVG-Faces: A Single-Object Precursor Benchmark

SVG-Faces was the first testbed for this project. Each image shows one smiley face controlled by 15 continuous latents (Fig. 11): face radius and center (x, y); eye radius, spacing and vertical offset; mouth width, vertical offset and curvature; and skin and eye color in HSV. A two-face regime with 30 latents is used for the binding test of App. G. The main results use SVG-World, which adds multiple objects and a second visual style; SVG-Faces is included as a consistency check with a different pipeline and a single object.

Forward decoding. A linear probe on DINOv3-B/16 recovers all nine geometric latents with $\rho \geq 0.98$ (Fig. 6). Color is harder: skin saturation and value reach $\rho \geq 0.97$, skin hue 0.93, and eye color 0.82–0.91, with eye hue lowest. The hue scatter plots bend at both ends because hue is circular and a linear probe cannot represent the wrap from 1 back to 0. Decoded latents re-rendered through the SVG pipeline give faithful reconstructions in distribution.

Pretrained vs. random. The comparison with random initialization splits the latents into two groups (Fig. 13). Random initialization decodes face position and radius and skin color as well as the pretrained encoder, with R^2 above 0.85. It fails on the small parts, eye and mouth geometry and eye color, where its R^2 is at most 0.23 and mostly near zero, while the pretrained encoder reaches

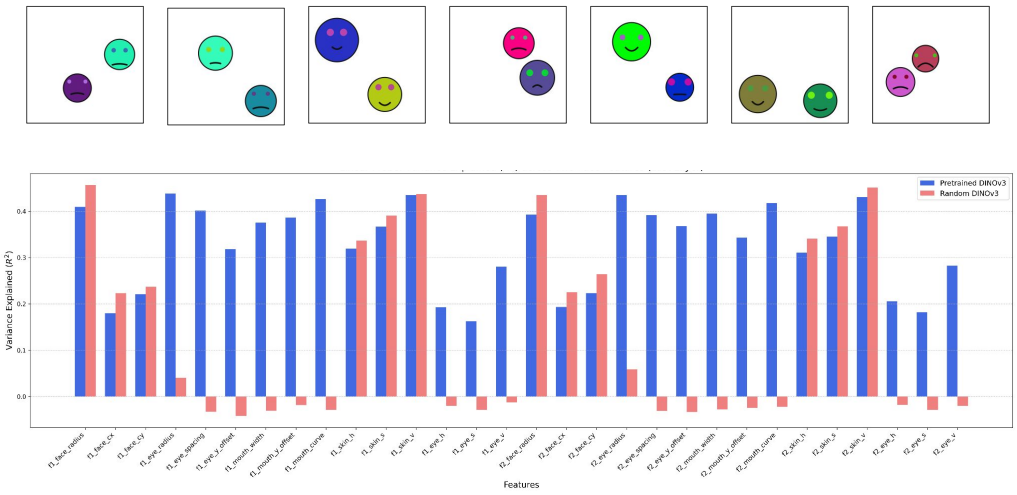


Figure 10: **Two-face regime of SVG-Faces.** (a) Example renders; each scene contains two faces with independently sampled latents, ordered left to right by x-coordinate. (b) Per-latent test R^2 for pretrained and random-init DINOv3-B/16 (30 latents, 15 per face).

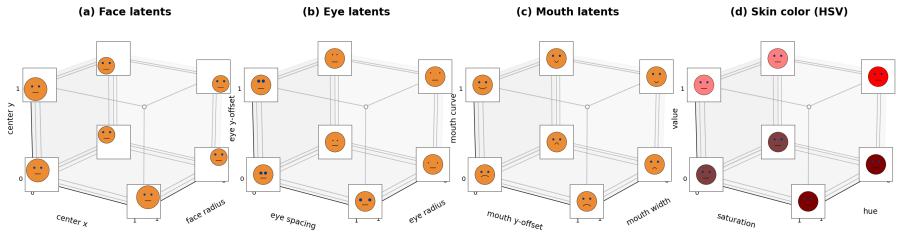


Figure 11: **SVG-Faces.** Each panel varies three latents of the single-face regime while the others are held fixed: **(a)** face center (x, y) and radius, **(b)** eye spacing, vertical offset and radius, **(c)** mouth vertical offset, width and curvature, and **(d)** skin hue, saturation and value. Eye color (HSV, not shown) completes the 15 latents. The renderer is deterministic.

0.67–0.99. Resolved by layer, the pretrained advantage on these latents builds up gradually over the transformer blocks.

Reverse predictivity. With the unwhitened reverse probe, aggregate forward R^2 is 0.92 for the pretrained encoder against 0.39 for random initialization, and aggregate R^2_{rev} is 0.47 against 0.34 (Fig. 14). The pattern differs from SVG-World: here the reverse gap is smaller than the forward gap in both absolute and relative terms, whereas on SVG-World the reverse ratio is much larger than the forward ratio (§5.2). One plausible reason is that a single large face, whose position, size and color are among the latents, accounts for most of the pixel variation, so even random features carry a large latent-aligned share, as on the controlled benchmarks (Table 8). We have not tested this explanation.

Generalization within one style. With a single visual style, the natural out-of-distribution test restricts the range of one latent during training. Fig. 15 reports six latents in three regimes: IID (random split), CompGen (train on one half of the latent's range, test on the other half) and Extrapolate

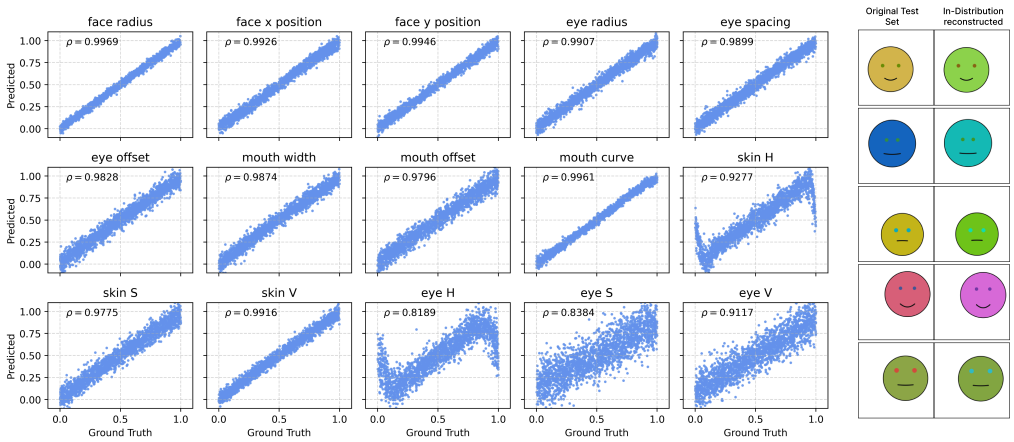


Figure 12: **DINOv3-B/16 forward decoding on SVG-Faces.** (a) Predicted vs. ground-truth values on held-out images for the 15 single-face latents. (b) Round-trip reconstructions: latents decoded from a held-out image and re-rendered through the SVG pipeline.

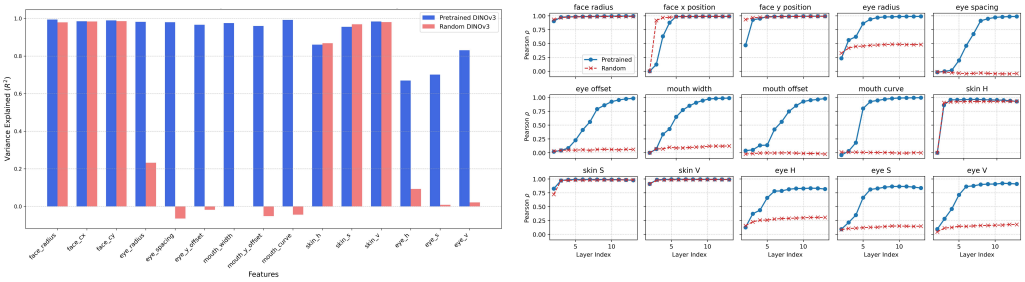


Figure 13: **SVG-Faces: pretrained vs. random initialization.** *Left:* per-latent test R^2 on the final block. *Right:* Pearson ρ per transformer block. Random initialization matches the pretrained encoder on face position, radius and skin color, and fails on eye and mouth geometry and eye color.

(train within a sub-range, test outside it). On geometry the pretrained encoder degrades less than random initialization; the clearest case is face radius under extrapolation ($R^2 = 0.96$ against 0.20). Skin value behaves the same way. Skin hue fails for both encoders outside the IID split, with negative R^2, and on skin saturation random initialization is ahead in every regime. Pretrained representations thus extrapolate better on geometry but not on color, in line with reports that pretrained representations often fail to generalize compositionally within a domain [37]. The cross-style test of §5.3 replaces this range restriction on SVG-World.

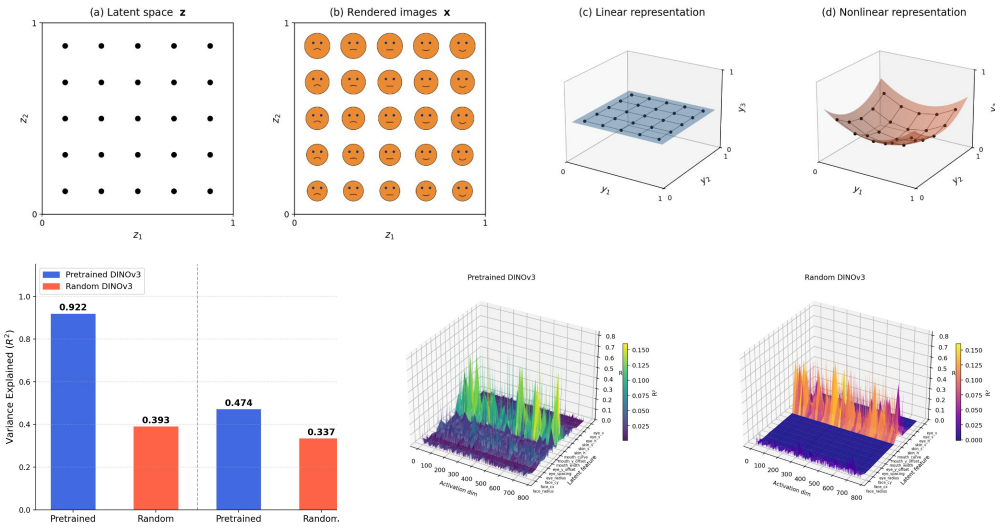


Figure 14: **SVG-Faces: reverse predictivity on DINOv3-B/16.** (a) Conceptual setup. (b) Aggregate forward and reverse R^2 for pretrained and random initialization. (c) Contribution of each (feature dimension, latent) pair to reverse R^2.

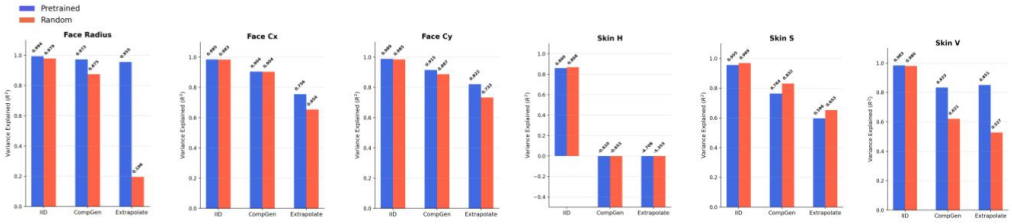


Figure 15: **SVG-Faces: generalization within one style.** Per-latent test R^2 for six latents in three regimes: **IID** (random split), **CompGen** (train on one half of a latent's range, test on the other half) and **Extrapolate** (train within a sub-range, test outside it). The pretrained encoder extrapolates better on face radius and skin value; both encoders fail on skin hue outside the IID split, and random initialization is ahead on skin saturation.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, the abstract and introduction accurately reflect the paper's main claims and contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, the paper discusses the main limitations of the work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Yes, the paper includes the theory assumptions and provides proofs in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Yes, the paper gives enough experimental details to reproduce the main results.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Yes, the code and data are included in the submitted zip file with the materials needed to reproduce the main results.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Yes, the paper gives the main experimental settings, including data splits, model details, probes, baselines, and appendix details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Yes, the simulation results report error bars, for example in Figure 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Yes, the paper includes compute-resource details in the experimental appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes, the work follows the NeurIPS Code of Ethics and uses synthetic data without human subjects or private data.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This is a foundational diagnostic study using synthetic data, with no direct deployment or human-subject setting.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk data or models; the released asset is synthetic SVG data.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes, the paper cites the existing models, datasets, and prior work used in the experiments.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Yes, SVG-World is introduced as a new asset and documented with dataset and pipeline details.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core methods do not use LLMs as an important or non-standard component of the research.

    Guidelines:

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