LOSS-INVARIANT PROJECTIONS AS PASSIVE PROBES OF LEARNED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned feature representations in neural networks often contain structure beyond that directly used by the final task output. We study such structure using *passive probes*: fixed, untrained, property-independent projections that remain unchanged while the representation evolves during training. We motivate this approach through prediction on $\mathcal{S}^2$, where equivalent vector and Hermitian parameterizations reveal an additional loss-invariant trace coordinate. This motivates a general construction in which fixed random projections serve as observers of learned features. Because the observer is loss-invariant and independent of the property being studied, changes in accessibility reflect changes in the representation relative to the fixed observer rather than adaptation of the observer. We show that ensembles of passive probes can have direct relationships to task-relevant information such as target alignment. Across surface-normal estimation, image classification, and image inpainting, we observe different changes in accessibility under our constructions: eventual difficulty becomes increasingly accessible in surface-normal estimation and image inpainting which are regression tasks, whereas its accessibility remains near its initial level in image classification. Comparisons with learned linear probes further show that recoverability and passive accessibility can evolve differently during training. These results show how passive probes can separately characterize changes in representation geometry and the accessibility of eventual task difficulty.

1 INTRODUCTION

Learned representations are commonly analyzed by asking what properties can be recovered from them or how their geometry changes across layers, models, or training. A standard approach to the first question is *probing*: an auxiliary predictor is trained on a frozen representation to recover a property of interest (Alain & Bengio, 2017; Conneau et al., 2018). Linear evaluation of frozen features is likewise widely used to evaluate learned visual representations (Chen et al., 2020). These approaches provide useful measures of *recoverability*: whether an observer adapted to the representation can extract a target property from it.

The interpretation of probe performance, however, depends partly on the observer being fitted. Probe capacity can permit memorization that obscures what should be attributed to the representation (Hewitt & Liang, 2019), while alternative formulations have emphasized sample efficiency, description length, and the inductive bias provided by a representation (Voita & Titov, 2020; Immer et al., 2022). More broadly, different probing methodologies operationalize different questions about a representation (Belinkov, 2022). These observations do not make learned probes unsuitable; rather, they make explicit that their measurements depend on an observer fitted to the property of interest.

A complementary line of work compares representations through their geometry. Representational similarity analysis compares patterns of pairwise dissimilarity (Kriegeskorte et al., 2008), while methods based on canonical correlation analysis, such as SVCCA (Raghu et al., 2017), compare representations under transformations of their feature coordinates. Centered kernel alignment (CKA) provides a related similarity measure with specified invariances and has proved effective for comparing neural representations across networks and initializations (Kornblith et al., 2019). Such methods characterize important aspects of representation geometry, but do not ask what becomes accessible to an observer that itself remains unchanged during learning. More generally, different representa-

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

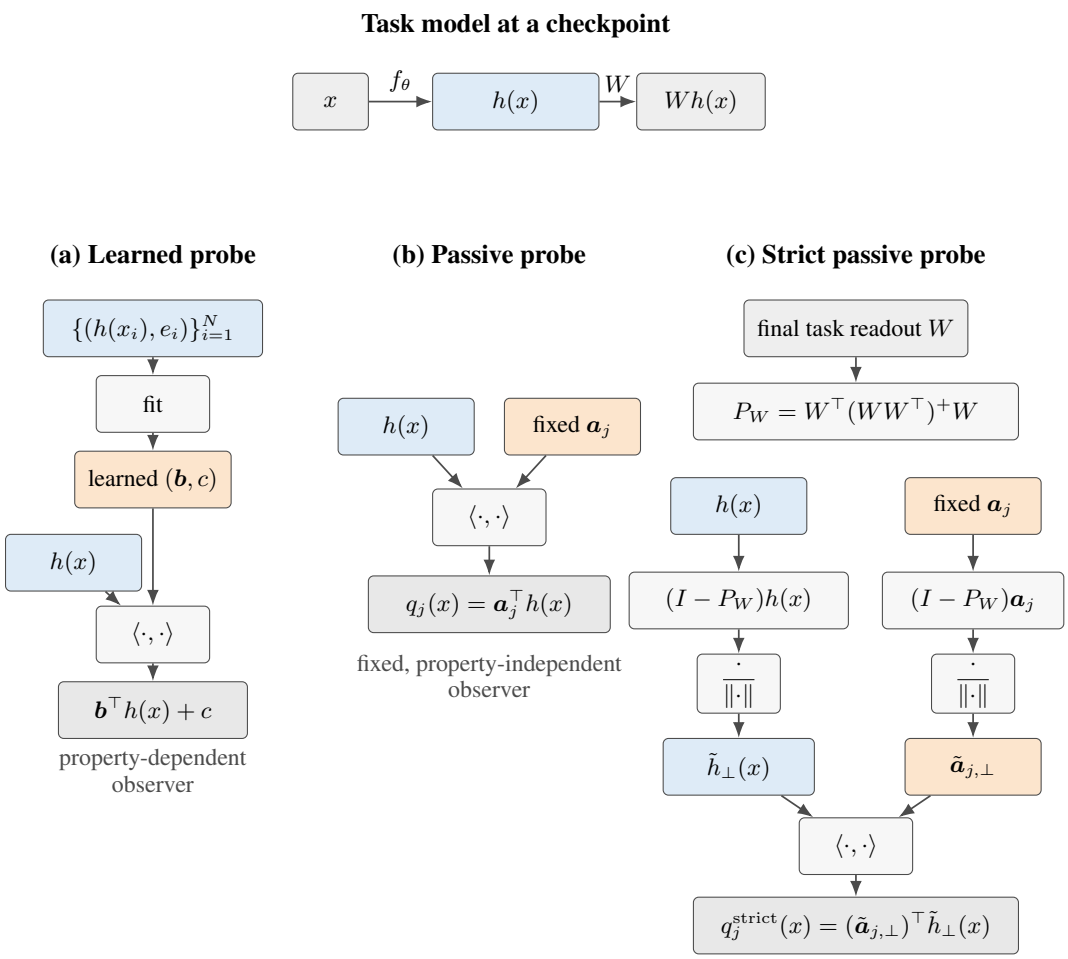

Figure 1: **Learned and passive observers of a changing representation.** At a checkpoint, f_θ maps input x to representation $h(x)$, and the task readout W produces prediction $Wh(x)$. **(a)** A learned probe $(\boldsymbol{b}, c)$ is fitted from representation–property pairs $\{(h(x_i), e_i)\}_{i=1}^N$, where e_i is the property of interest. **(b)** A passive probe instead uses a fixed, property-independent direction $\boldsymbol{a}_j$, sampled once and never optimized, with $q_j(x) = \boldsymbol{a}_j^\top h(x)$. **(c)** The strict construction uses the final task readout W to define the row-space projector $P_W = W^\top (WW^\top)^+ W$, projects both $h(x)$ and $\boldsymbol{a}_j$ with $I - P_W$, normalizes the resulting vectors, and computes their inner product.

tional similarity measures encode different notions of equivalence and can differ in their sensitivity to changes in the underlying representation (Williams et al., 2021; Ding et al., 2021). Such methods characterize important aspects of representation geometry, but do not ask what becomes accessible to an observer that itself remains unchanged during learning.

This leaves a complementary observational question: *what becomes accessible to a fixed observer as the representation changes during learning?* We study this question using *passive probes*: fixed, property-independent scalar projections sampled independently of the property of interest and held fixed across the training trajectory. Unlike a learned probe, a passive probe is not optimized to recover the target. Consequently, changes in its relationship to that target reflect changes in the representation relative to the fixed observer rather than adaptation of the observer. We refer to this quantity as *passive accessibility*, distinguishing it from recoverability after fitting a learned probe.

We additionally introduce a strict construction that removes the row space of the final task readout from the passive direction; see fig. 1. Null-space projections have previously been used to remove linearly recoverable properties from representations and to study their behavioral relevance (Ravfogel et al., 2020; Elazar et al., 2021). The role of the null space here is different: we do not learn a

direction from the property and then remove it. Instead, the passive direction is property-independent and is projected into the null space of the *task readout*. The resulting scalar coordinate is therefore task-readout-invariant; when the loss depends only on that readout, it is also loss-invariant.

Our construction arises naturally from an apparently redundant degree of freedom in an equivalent parameterization of prediction on the sphere $\mathcal{S}^2$. This motivating example, presented in section 2.1, leads to a general framework for tracking two aspects of learned representations: changes in their geometry and changes in the accessibility of a property along fixed scalar coordinates. These quantities need not evolve together. Likewise, increasing recoverability by an adapted linear observer need not imply increasing accessibility to an observer that remains fixed.

Our contributions are summarized as follows:

(i) We introduce *passive probes*: fixed, untrained, and property-independent scalar observers that track changes in accessibility without adapting the observer as the representation evolves.

(ii) We develop a "strict" probe construction by restricting passive directions to the null space of the final task readout, and derive relationships between passive-probe ensembles and target alignment.

(iii) Across surface-normal estimation, image classification, and image inpainting, we find distinct trajectories of passive accessibility to eventual task difficulty which differ between regression and classification settings. Comparisons with learned linear probes further show that linear recoverability and passive accessibility can evolve differently during training.

2 METHODOLOGY

2.1 MOTIVATING EXAMPLE: A LOSS-INVARIANT VARIABLE IN $\mathcal{S}^2$ PREDICTION

Consider the task of predicting a point on the 3D unit sphere $\mathcal{S}^2$. The simplest learning representation is a vector $\boldsymbol{r} \in \mathbb{R}^3$ which is subsequently normalized to be on the sphere. We can alternatively represent the point indirectly by the vector $\boldsymbol{\nu} \in \mathbb{R}^4$ representing the four unique elements of a 2×2 Hermitian matrix

$$\mathcal{H} = \begin{pmatrix} \nu_1 & \nu_2 + \nu_3 i \\ \nu_2 - \nu_3 i & \nu_4 \end{pmatrix} . \tag{1}$$

From $\mathcal{H}$, the point on $\mathcal{S}^2$ is obtained by extracting the eigenvector of $\mathcal{H}$ corresponding to its smallest eigenvalue. This vector is identified as an element of $\mathbb{CP}^1$ which is isomorphic to $\mathcal{S}^2$ (its bijection is given by the stereographic projection). Although the two representations appear quite different, they are known to be algebraically equivalent. More directly,

$$\boldsymbol{r} = \left(-\nu_2,\, \nu_3,\, \frac{\nu_4 - \nu_1}{2} \right), \qquad \|\boldsymbol{r}\| = \frac{\lambda_2 - \lambda_1}{2} . \tag{2}$$

where λ_i are the eigenvalues of $\mathcal{H}$ with $\lambda_1 \leq \lambda_2$. Furthermore, their learning can be shown to be identical under matched parameterizations (Appendix A.1), with coincident degenerate conditions. The primary difference therefore lies in their dimensionality. The Hermitian representation contains an additional degree of freedom, its trace $\tau = Tr(\mathcal{H}) = \nu_1 + \nu_4$, which is invariant to the loss (Appendix A.2). Despite receiving no gradient information, we observe that τ has the potential to indirectly amass task-relevant information (such as task difficulty) through the learning process. This is because the underlying representation from which it is computed updates during training. This may make the Hermitian representation more favorable over direct prediction of $\boldsymbol{r}$. By eq. (2), $\boldsymbol{r}$ and τ can be computed as fixed linear projections of the representation $\boldsymbol{\nu}$, with the former being task-specific and the latter being loss-invariant. Using this geometric setup as motivation, we analyze the more general case of arbitrary linear projections of learned representations.

2.2 LINEAR PROJECTIONS OF FEATURE REPRESENTATIONS

Let $h(x) \in \mathbb{R}^d$ be a learned feature representation depending on data input x, and let $W \in \mathbb{R}^{n \times d}$ be the learned weight matrix corresponding to the final linear layer of a model (see fig. 1, top). Then, model prediction $p(x)$ is simply given by

$$p(x) = W h(x). \tag{3}$$

We can sample normally distributed linear probes $\boldsymbol{a}_j$ such that

$$q_j(x) = \boldsymbol{a}_j^\top h(x). \tag{4}$$

Note that $\boldsymbol{a}_j$ is independently constructed, never optimized, and remains fixed throughout training (i.e. $\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_j} = 0$). Changes in q_j therefore solely arise from changes in the representation $h(x)$ rather than adaptation of the probe.

If $q_j(x)$ is constructed from $h(x)$ like $p(x)$, then why would it be able to contain any new task-relevant information not already present in $p(x)$? This is enabled by two main mathematical facts. First, the training loss is applied to $p(x)$ which only ever sees $h_\parallel(x)$, the component of $h(x)$ that lies in the row span of W. $p(x)$ never sees $h_\perp(x)$, the component of $h(x)$ that lies in the nullspace of W (i.e. $Wh_\perp(x) = 0$). On the other hand, q_j is derived from both components of $h(x)$, giving it access to information discarded by W. This is mathematically described by

$$p(x) = Wh_\parallel(x) + Wh_\perp(x) = Wh_\parallel(x) \tag{5}$$

$$q_j(x) = \boldsymbol{a}_j^\top h_\parallel(x) + \boldsymbol{a}_j^\top h_\perp(x). \tag{6}$$

Thus, q_j can be seen as an attempt at recovering more information from the learned representation. Note that $n < d$ for most real-world model architectures, so the dimensionality of $h_\perp$ is assumed nonzero. Second, for tasks specifically utilizing an L_2 squared loss (e.g. simple regression tasks)[1], we can define the conditional mean and variance $\mu(x) = \mathbb{E}[\mathcal{Y} \mid x]$ and $\Sigma(x) = \mathrm{Cov}(\mathcal{Y} \mid x)$ respectively of the target variable $\mathcal{Y}$, and derive

$$\mathbb{E}\big[\|p(x) - \mathcal{Y}\|^2 \mid x\big] = \|p(x) - \mu(x)\|^2 + Tr(\Sigma(x)), \tag{7}$$

$$\nabla_{p(x)}\mathbb{E}\big[\|p(x) - \mathcal{Y}\|^2 \mid x\big] = 2(p(x) - \mu(x)). \tag{8}$$

The best prediction is therefore $p^*(x) = \mu(x)$. The loss distribution depends on both the conditional mean and variance. However, the gradient only depends on the conditional mean. In other words, the conditional variance, which helps characterize the loss distribution, does not ultimately affect the best prediction. This is in contrast to other loss functions (e.g. cross entropy loss) which may encourage the prediction to incorporate multiple moments of the loss distribution. Both mathematical arguments establish why we would not expect the loss-driven $p(x)$ to contain all the information present in $h(x)$, and the first argument motivates the construction of q_j to potentially capture some of that missing information. However, it is important to clarify that there is no guarantee that $h(x)$ actually contains any additional task-relevant information than what is exposed to $p(x)$ or that this additional information is available via linear probes $\boldsymbol{a}_j$. Our analyses and findings only lend empirical evidence to the potential accessibilty of additional task information via random projections.

2.3 Analysis Framework

The theoretical effectiveness of the probes depends entirely on the amount of information available in the feature maps. To further our analysis of this information, we formally define the following quantities

$$P_W = W^\top \left(WW^\top\right)^+ W \tag{9}$$

$$h_\perp(x) = (I - P_W)h(x) \qquad \boldsymbol{a}_{j,\perp} = (I - P_W)\boldsymbol{a}_j, \tag{10}$$

where $(\cdot)^+$ denotes the Moore–Penrose pseudoinverse. P_W is the projector onto the row space of W and both $h_\perp$ and $\boldsymbol{a}_{j,\perp}$ lie in the nullspace of W. These quantities isolate the information present in the probe that is linearly distinct from the information in $p(x)$ and is a useful alternative quantity for comparison in our analysis (see fig. 1 and section C).

Next, we attempt to quantify the accessibility of information via the probes (i.e. their potential utility) using metrics from a trained model and a reference dataset. This requires defining the task-relevant information we hope to access. For this paper, we focus on observing task difficulty as

[1]The same first-moment argument applies to regression constrained to a sphere (i.e. cosine loss). For unit target $\mathcal{Y}$ and unit prediction $\boldsymbol{p}$, the conditional expected loss is $1 - \boldsymbol{p}^\top \mathbb{E}[\mathcal{Y} \mid x]$, so its optimum is the direction of the conditional mean when that mean is nonzero. This does not hold for angular loss.

defined by the relevant loss function $\mathcal{L}$ (see section 3.1). Note that the following are only diagnostic statistics and not predictive metrics. They can be calculated with either the regular or strict quantities.

Given a fixed dataset of N samples $\mathbf{x} = \{x_1, \ldots, x_N\}$, we can compute the error vector

$$\mathbf{e} = \{\mathcal{L}(p(x_1), y_1), \ldots, \mathcal{L}(p(x_N), y_N)\} \tag{11}$$

from each prediction and target y_i and use it to quantify task difficulty. Applying the matrix operator $C_N = I_N - \mathbf{1}_N \mathbf{1}_N^\top / N$, we can center error vector as $\bar{\mathbf{e}} = C_N \mathbf{e}$.

Linear kernel alignment First, the learned features of the dataset are stacked into a matrix

$$H = \begin{bmatrix} h(x_1)^\top \\ \vdots \\ h(x_N)^\top \end{bmatrix} \in \mathbb{R}^{N \times d}. \tag{12}$$

The Gram entry $K_{i\ell}$ measures the similarity of the selected features for examples x_i and x_ℓ and is computed for the dataset as

$$K = HH^\top \qquad K_c = C_N K C_N. \tag{13}$$

Using this similarity, we can define the (normalized) linear alignment metric

$$A_K = \frac{\bar{\mathbf{e}}^\top K_c \bar{\mathbf{e}}}{\|\bar{\mathbf{e}}\|^2 \operatorname{tr}(K)}. \tag{14}$$

This is the fraction of centered probe-ensemble energy aligned with eventual difficulty. It is defined when both denominator factors are nonzero and lies in the range $[0, 1]$. Note that A_K is computed independent of the probes $\boldsymbol{a}_j$ and is thus a property of the learned features and error vector. The similarity kernel is linear, so finite random projections can approximate A_K and access the information it represents.

Nonlinear kernel alignment To capture nonlinear relationships with task difficulty, we utilize Gaussian similarity as a radial kernel. Let $s_j > 0$ be the standard deviation of $\mathbf{q}_j = H\boldsymbol{a}_j$, and form

$$G_{i\ell} = \frac{1}{m} \sum_{j=1}^{m} \exp\left(-\frac{(q_j(x_i) - q_j(x_\ell))^2}{2s_j^2} \right), \qquad G_c = C_N G C_N \tag{15}$$

$$A_G = \frac{\bar{\mathbf{e}}^\top G_c \bar{\mathbf{e}}}{\|\bar{\mathbf{e}}\|^2 \operatorname{tr}(G_c)}. \tag{16}$$

A_G measures whether examples that are similar under the random projection kernels have systematically similar centered difficulty. It is similarly only defined for nonzero denominator and lies in range $[0, 1]$. At the population level, it detects variation in mean difficulty along q_j, including non-monotonic variation. Unlike A_K, its value depends on the actual probed values and the number of probes because G is an average over m scalar Gaussian kernels in probed coordinate space. Increasing m reduces the Monte Carlo variation of this ensemble statistic.

Statistical Measures of Correlation The previous alignment metrics characterize the accessibility of task difficulty to random projections. To better contextualize them, we also compute some standard measures of correlation to analyze the general relationship between learned features and task difficulty. With $\bar{\mathbf{q}}_j = C_N \mathbf{q}_j$, we calculate the squared Pearson correlation for linear association as

$$\rho^2_{Pearson,j} = \frac{(\bar{\mathbf{q}}_j^\top \bar{\mathbf{e}})^2}{\|\bar{\mathbf{q}}_j\|^2 \|\bar{\mathbf{e}}\|^2}, \tag{17}$$

For monotonic association, we calculate the absolute value of the Spearman correlation as

$$|\rho_{\text{Spearman},j}| = \left| \frac{\sum_{i=1}^{N} (r_{j,i} - \bar{r}_j)(s_i - \bar{s})}{\sqrt{\sum_{i=1}^{N} (r_{j,i} - \bar{r}_j)^2} \sqrt{\sum_{i=1}^{N} (s_i - \bar{s})^2}} \right|, \tag{18}$$

where $r_{j,i}, s_i$ are the indices of the sorted values of $\mathbf{q}_j$ and $\mathbf{e}$ respectively and $\bar{r}_j, \bar{s}$ are the mean values. Finally, for a one-dimensional predictor $\hat{e}_{j,i} = g_j(q_j(x_i))$ fitted only on the calibration split, we compute the coefficient of determination

$$R_j^2 = 1 - \frac{\sum_{i \in \text{test}} (e_i - \hat{e}_{j,i})^2}{\sum_{i \in \text{test}} (e_i - \bar{e}_{\text{test}})^2}. \tag{19}$$

This tests whether a single fixed coordinate supports reproducible nonlinear prediction of difficulty on unseen observations.

3 Experiments and Results

We study passive accessibility during training in three settings: surface-normal estimation on NYUv2, image classification on CIFAR-100, and image inpainting on CIFAR-100. For each task, we train five independently seeded models and evaluate 256 fixed Gaussian passive probes at a predetermined sequence of checkpoints.

At every checkpoint, we compare the passive coordinates $q_j(x)$ with the corresponding task-difficulty variable defined by the final trained model. Our primary analysis reports both raw probes and the task-invariant strict probes defined in Appendix C. We report linear alignment A_K (eq. (14)), nonlinear alignment A_G (eq. (16)), squared Pearson correlation (eq. (17)), absolute Spearman correlation (eq. (18)), and held-out calibrated R^2 (eq. (19)). The first four quantities are evaluated directly on the test set. For calibrated R^2, the scalar calibration is fitted on the calibration split and evaluated on the held-out test split. Task performance is reported on the same test data. Appendix B.4 specifies probe sampling, the fixed final readout subspace, and the calibration and test evaluation procedures.

3.1 Tasks and experimental setup

Surface-normal estimation We train the baseline-normal architecture of DSINE (Bae & Davison, 2024) on the NYUv2 dataset (Silberman et al., 2012). At each valid pixel x, the model predicts a unit surface normal $\hat{n}_\theta(x)$ and is trained with the cosine loss

$$\ell_{\text{normal}}(x) = 1 - \hat{n}_\theta(x)^\top n^*(x), \tag{20}$$

where $n^*(x)$ is the ground-truth unit normal. We probe the 128-dimensional representation immediately preceding the final convolutional output. Task difficulty is the angular error of the final model, $e(x) = \arccos\big(\hat{n}_\theta(x)^\top n^*(x)\big)$.

Models are trained for 20 epochs. The evaluation uses separate calibration and test sets containing 327 images each, with 83,712 sampled valid pixels in each split. Full model, training, and data details are provided in Appendix B.1.

CIFAR-100 classification We train a vision transformer (Dosovitskiy et al., 2021) from scratch on CIFAR-100 (Krizhevsky, 2009). For image x with label y, the training objective is cross entropy,

$$\ell_{\text{cls}}(x, y) = -\log \pi_\theta(y \mid x), \tag{21}$$

where $\pi_\theta(y \mid x)$ is the probability assigned to the correct class. We probe the 192-dimensional CLS representation immediately preceding the 100-class linear layer. Eventual difficulty $e(x, y)$ is the negative log likelihood in eq. (21) evaluated under the final model θ.

We train on 45,000 images, reserve 5,000 images for calibration, and use the official 10,000-image test set only for evaluation. Models are trained for 300 epochs. Architecture, optimization, augmentation, and checkpoint details are provided in Appendix B.2.

CIFAR-100 inpainting We adopt the vision transformer used for the classification task for image inpainting. The central 16×16 region of each 32×32 image is masked, corresponding to 16 spatial patches. A shared linear reconstruction head predicts the RGB values of each masked 4×4 patch from its spatial-token representation. We probe the 192-dimensional representation immediately preceding this head.

Table 1: **Passive accessibility during training.** Entries report initialization → final checkpoint, with $(\Delta \pm \mathrm{SD}_\Delta)$ shown below. Initial and final values are across-seed means; Δ is computed within each seed before aggregation over five independently trained models. Probe metrics are reported in units of 10^{-3}. Calibrated R^2 uses a scalar calibration fitted on CAL and evaluated on TEST. Task performance is evaluated on TEST and is shared by the raw and strict probe families.

Task	Family	Probe metrics ($\times 10^{-3}$)					Performance
		A_K	A_G	Pearson^2	|Spearman|	cal. R^2	
Surface normals	Raw	23.60 → 30.87 $(+7.27 \pm 5.28)$	14.55 → 21.57 $(+7.01 \pm 1.19)$	18.24 → 25.81 $(+7.56 \pm 3.10)$	131.26 → 191.87 $(+60.61 \pm 11.05)$	27.91 → 64.28 $(+36.37 \pm 3.98)$	75.58 → 19.20 (-56.38 ± 21.70) deg
	Strict	16.79 → 16.00 (-0.79 ± 2.04)	9.33 → 14.78 $(+5.45 \pm 1.33)$	13.79 → 15.60 $(+1.81 \pm 1.50)$	110.84 → 154.95 $(+44.11 \pm 8.80)$	17.13 → 51.76 $(+34.63 \pm 4.61)$	
Classification	Raw	2.47 → 2.37 (-0.11 ± 0.37)	1.41 → 1.39 (-0.03 ± 0.22)	1.88 → 2.31 $(+0.43 \pm 0.38)$	69.70 → 46.78 (-22.92 ± 3.97)	$-1.07 \rightarrow -0.23$ $(+0.84 \pm 0.47)$	1.00 → 70.62 $(+69.62 \pm 0.56)$ %
	Strict	2.28 → 2.02 (-0.25 ± 0.53)	1.39 → 1.26 (-0.13 ± 0.37)	1.87 → 2.12 $(+0.25 \pm 0.59)$	69.27 → 44.20 (-25.06 ± 7.40)	$-1.23 \rightarrow -0.91$ $(+0.32 \pm 0.62)$	
Inpainting	Raw	0.62 → 4.50 $(+3.88 \pm 0.87)$	0.61 → 2.97 $(+2.36 \pm 0.55)$	0.68 → 4.84 $(+4.16 \pm 0.92)$	26.94 → 76.69 $(+49.75 \pm 8.05)$	0.81 → 5.34 $(+4.54 \pm 1.11)$	5.71 → 17.31 $(+11.60 \pm 0.01)$ dB
	Strict	0.63 → 9.35 $(+8.73 \pm 1.61)$	0.61 → 4.91 $(+4.30 \pm 0.79)$	0.70 → 8.60 $(+7.90 \pm 1.29)$	27.79 → 106.17 $(+78.38 \pm 7.75)$	0.84 → 10.26 $(+9.42 \pm 1.61)$	

For image x and masked patch p, the reconstruction loss is

$$\ell_{\mathrm{inp}}(x,p) = \frac{1}{48} \left\| \hat{y}_\theta(x,p) - y(x,p) \right\|_2^2, \tag{22}$$

where $y(x,p) \in [0,1]^{48}$ contains the RGB values of the target patch. Training minimizes this error over the 16 masked patches. Eventual difficulty $e(x,p)$ is the per-patch reconstruction error in eq. (22) evaluated under the final model θ.

We use 45,000 images for training, 5,000 for calibration, and the official 10,000-image test set for evaluation. Each calibration and test image contributes all 16 masked patches, giving 80,000 calibration observations and 160,000 test observations. Models are trained for 100 epochs. Architecture, masking, optimization, and checkpoint details are provided in Appendix B.3.

Architectures, optimization settings, data splits, augmentation, masking, and checkpoint schedules are specified in Appendix B.

3.2 Passive accessibility during training

Table 1 summarizes passive accessibility at initialization and at the final training checkpoint, and corresponding trajectories across all evaluated checkpoints are reported in fig. 2.

The three tasks exhibit distinct patterns. For surface-normal estimation, passive accessibility increases under several measures, including under the strict probe construction. In particular, strict A_G and held-out calibrated R^2 increase substantially, whereas strict A_K changes little. Thus, the nonlinear and calibrated measures reveal a systematic change that is weak under linear alignment.

For CIFAR-100 classification, task performance improves substantially while most passive accessibility measures remain near their initialization levels under both raw and strict probes. Absolute Spearman correlation instead decreases. Improved task performance therefore does not by itself imply increased accessibility of eventual difficulty to a fixed, property-independent scalar observer.

For CIFAR-100 inpainting, passive accessibility increases under both raw and strict probes, with positive endpoint changes in A_K, A_G, Pearson^2, absolute Spearman correlation, and held-out calibrated R^2. Together, the three tasks show that passive accessibility can develop substantially during training, but need not do so simply because task performance improves. It is interesting to note that in our experiments, only the regression loss tasks (which Section 2 mentioned does not require incorporation of further information on the loss distribution) provided a loss landscape such that task difficulty was accessible through the learned representations.

These task-level patterns also hold when feature normalization and removal of the final task-readout row space are applied separately, as shown by the unit-normalized and readout-orthogonal controls

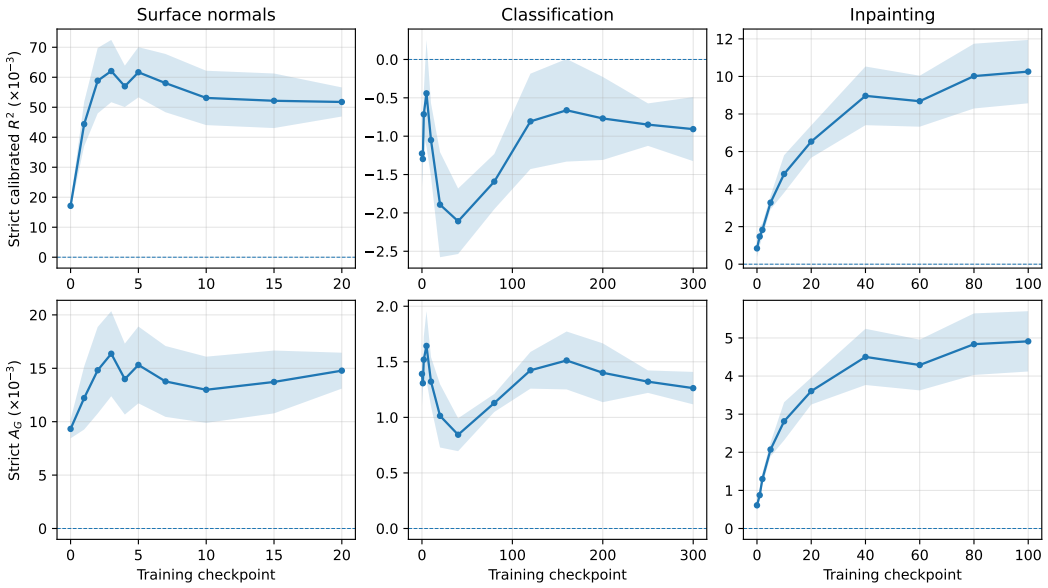

Figure 2: **Strict passive accessibility across training.** Calibrated R^2 (top) and A_G (bottom) for strict passive probes at each evaluated training checkpoint, averaged over five independent training seeds; shading denotes ± 1 standard deviation across seeds. Surface-normal estimation shows a strong early increase in both measures followed by an elevated but non-monotonic trajectory, while inpainting shows a sustained increase across training. Classification exhibits intermediate variation but no sustained initial-to-final increase in either measure. Despite their different constructions, the two measures therefore show consistent task-level patterns. In particular, A_G requires no fitted mapping from the passive coordinates to eventual difficulty, whereas calibrated R^2 uses calibration-set bin means.

in Appendix C. The calibrated-R^2 endpoint conclusions are also stable across $B \in \{5, 10, 20, 40\}$ calibration quantile bins (Appendix D), including the primary choice $B = 20$.

3.3 Passive accessibility and learned recoverability

Passive and learned probes ask different questions. A passive probe holds the observer fixed throughout training, whereas a learned probe measures how well eventual difficulty can be recovered after adapting an observer to the representation at each checkpoint. The two measurements therefore need not evolve together.

For the learned comparison, we fit a ridge regression from the raw representation to the same eventual-difficulty target used in the passive analysis. Regularization is selected using calibration data only, and the fitted probe is evaluated on held-out test observations. Both analyses use the raw representation and the same CAL/TEST observations. Strict passive probes serve as a separate task-readout-invariant control and are not included in this representation-matched comparison. Appendix E provides the feature-standardization, cross-validation, regularization selection, and matched evaluation protocol.

Table 2 compares the two observers using squared Pearson correlation, absolute Spearman correlation, and calibrated R^2. Learned-probe correlations are computed directly from its scalar TEST predictions. For both observers, calibrated R^2 uses the same 20-bin scalar calibration fitted on CAL and evaluated on TEST. As in Table 1, values are reported in units of 10^{-3}.

The clearest separation occurs in classification. Learned recoverability increases strongly across all three measures, whereas passive accessibility changes little overall: passive Pearson^2 and calibrated R^2 remain near their initialization levels, while absolute Spearman correlation decreases. Thus, substantial growth in linear recoverability need not be accompanied by greater accessibility to a fixed, property-independent scalar observer.

Table 2: **Passive accessibility versus learned recoverability.** Passive probes are fixed, property-independent scalar observers, whereas learned ridge probes are fitted to eventual difficulty separately at each checkpoint. Both use the raw representation and the same CAL/TEST observations. Entries report initialization → final checkpoint, with $(\Delta \pm \mathrm{SD}_\Delta)$ shown below, across five training seeds. All metrics are reported in units of 10^{-3}.

Task	**Observer**	**Metrics** ($\times 10^{-3}$)		
		Pearson^2	**$\lvert$Spearman$\rvert$**	**cal. R^2**
Surface normals	Passive	18.24 → 25.81 $(+7.56 \pm 3.10)$	131.26 → 191.87 $(+60.61 \pm 11.05)$	27.91 → 64.28 $(+36.37 \pm 3.98)$
	Learned	93.01 → 186.94 $(+93.94 \pm 13.28)$	329.68 → 517.11 $(+187.43 \pm 7.68)$	97.17 → 189.82 $(+92.65 \pm 9.25)$
Classification	Passive	1.88 → 2.31 $(+0.43 \pm 0.38)$	69.70 → 46.78 (-22.92 ± 3.97)	$-1.07 \rightarrow -0.23$ $(+0.84 \pm 0.47)$
	Learned	10.86 → 188.56 $(+177.70 \pm 6.55)$	173.96 → 511.53 $(+337.57 \pm 14.12)$	5.29 → 188.76 $(+183.48 \pm 9.82)$
Inpainting	Passive	0.68 → 4.84 $(+4.16 \pm 0.92)$	26.94 → 76.69 $(+49.75 \pm 8.05)$	0.81 → 5.34 $(+4.54 \pm 1.11)$
	Learned	61.56 → 200.91 $(+139.34 \pm 5.01)$	310.01 → 615.08 $(+305.07 \pm 11.01)$	64.33 → 200.65 $(+136.31 \pm 4.60)$

Surface-normal estimation and inpainting show a complementary pattern. In both tasks, passive accessibility and learned recoverability increase during training, although the learned changes are larger across the matched measures. Together, the three tasks show that linear recoverability and passive accessibility can evolve differently during training. They therefore characterize empirically separable properties of the learned representation.

4 Conclusion

Passive probes provide a framework for examining how task-relevant information becomes accessible during learning without explicit fitting of the probe to the property of interest. Their relationship to representation geometry supports the interpretation of these measurements, including probing the null space of the final task output. Across the three tasks considered, the findings reveal different levels of accessibility of eventual difficulty from its fixed, untrained directions. The contrasting behavior of classification and the two regression tasks indicates that this relationship depends on the learning setting. Passive probes differ in information content from learned probes and can thus provide complementary evidence about the organization of learned representations. Further work is needed to establish how the task, training objective, and architecture contribute to the observed differences.

AI use statement

In this work, we used generative AI tools to assist in critically evaluating and refining an author-developed conceptual framework and mathematical derivations, checking mathematical claims, refining hypotheses, designing aspects of the experimental and analysis methodology, implementing and debugging analysis code, and interpreting experimental results. The core research idea, including the initial identification of the loss-invariant degree of freedom, the relevant equivalence, and the foundational mathematical formulation motivating this work, was developed by the authors. Generative AI was subsequently used to stress-test these ideas, explore their implications and generalizations, identify potential confounds and controls, and assist in developing and validating the resulting experimental analysis.

We did not use generative AI tools to generate or modify the experimental datasets or to fabricate, alter, or selectively choose experimental measurements. Tasks involving translation, qualitative or thematic data analysis, and human-subject research were not applicable to this work.

Additionally, we used generative AI tools for brainstorming, literature search and synthesis, critical assessment of novelty and positioning, software development and debugging, organization of the manuscript, and drafting and editing scientific text for clarity.

All AI-assisted research contributions were reviewed and validated by the authors. Mathematical statements and derivations used in the work were checked by the authors against their definitions and where applicable verified numerically. AI-assisted experimental and analysis code was tested using analytic or synthetic sanity checks, endpoint identities, independent replay of saved checkpoints, and numerical comparison against previously saved experimental outputs. Experimental conclusions were based on results produced by the implemented experiments rather than on generative-AI predictions Literature and novelty claims arising during AI-assisted discussion were checked against the cited literature. The authors made the final decisions regarding the mathematical formulation, hypotheses, experimental design, interpretation, claims, and manuscript content.

We take responsibility for the final content of this work, including text, claims, code, analyses, or other artifacts produced with the aid of generative AI.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJ4-rAVtl.

Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198/.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1556–1568. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0c0bf917c7942b5a08df71f9da626f97-Paper.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Alexander Immer, Lucas Torroba Hennigen, Vincent Fortuin, and Ryan Cotterell. Probing as quantifying inductive bias. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1839–1851, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.129. URL https://aclanthology.org/2022.acl-long.129/.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A. Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, Volume 2 - 2008, 2008. ISSN 1662-5137. doi: 10.3389/neuro.06.004.2008. URL https://www.frontiersin.org/journals/systems-neuroscience/articles/10.3389/neuro.06.004.2008.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647/.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pp. 746–760. Springer, 2012.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14/.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4738–4750. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf.

A PROOFS

A.1 EQUIVALENCE OF THE PREDICTIONS AND BACKPROPAGATED GRADIENTS

Write

$$\mathcal{H} = \frac{\tau}{2} I - \boldsymbol{r} \cdot \boldsymbol{\sigma}, \qquad \boldsymbol{r} = \left(-\nu_2, \nu_3, \frac{\nu_4 - \nu_1}{2} \right), \tag{23}$$

where $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$ denotes the Pauli matrices,

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{24}$$

Since $(\boldsymbol{r} \cdot \boldsymbol{\sigma})^2 = \|\boldsymbol{r}\|^2 I$, for $\boldsymbol{r} \neq 0$ the projector onto the smallest-eigenvalue eigenspace of $\mathcal{H}$ is

$$P_- = \frac{1}{2} \left(I + \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \cdot \boldsymbol{\sigma} \right). \tag{25}$$

Under the standard identification $\mathbb{CP}^1 \simeq \mathcal{S}^2$, this projector corresponds to $\boldsymbol{p} = \boldsymbol{r}/\|\boldsymbol{r}\|$. Thus the eigenvector construction and direct normalization define the same prediction map.

Consequently, for any differentiable task loss $\mathcal{L} = \ell(\boldsymbol{p})$, both constructions give

$$\boldsymbol{g} := \nabla_{\boldsymbol{r}} \mathcal{L} = \frac{I - \boldsymbol{p}\boldsymbol{p}^\top}{\|\boldsymbol{r}\|} \nabla_{\boldsymbol{p}} \ell, \qquad \nabla_{\boldsymbol{\nu}} \mathcal{L} = \left(-\frac{g_3}{2}, -g_1, g_2, \frac{g_3}{2} \right)^\top. \tag{26}$$

Their gradients with respect to any shared upstream parameters therefore agree by the chain rule. Their degeneracies also coincide: $\boldsymbol{r} = 0$ precisely when $\lambda_1 = \lambda_2$, at which point neither construction determines a unique sphere prediction.

Parameterization and optimization The equivalence is under the stated change of variables. Define

$$A = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}, \qquad \boldsymbol{r} = A\boldsymbol{\nu}. \tag{27}$$

Then ordinary Euclidean gradient descent directly on $\boldsymbol{\nu}$ induces

$$\Delta \boldsymbol{r} = -\eta A A^\top \boldsymbol{g} = -\eta \, \mathrm{diag}(1, 1, \tfrac{1}{2}) \boldsymbol{g}, \tag{28}$$

whereas gradient descent directly on $\boldsymbol{r}$ gives $-\eta \boldsymbol{g}$.

A.2 THE TRACE RECEIVES NO TASK-LOSS GRADIENT

The gradient in Appendix A.1 satisfies

$$(1, 0, 0, 1) \cdot \nabla_{\boldsymbol{\nu}} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \nu_1} + \frac{\partial \mathcal{L}}{\partial \nu_4} = 0, \tag{29}$$

so it has no component in the trace direction. This does not require the two diagonal-entry gradients individually to vanish. Nor does it require the trace output to remain constant when shared upstream features change during training. This result is intuitive as eigenvalue shifts by some scalar γ (i.e. $\mathcal{H} + \gamma I$) do not change the eigenvectors but do change the trace by 2γ.

B EXPERIMENTAL DETAILS

All experiments are implemented in PyTorch (Paszke et al., 2019) and run on NVIDIA A100 GPUs. Each experiment is repeated over five independent training seeds.

B.1 SURFACE-NORMAL ESTIMATION

Model and task We use the baseline-normal architecture from DSINE (Bae & Davison, 2024) on NYUv2 (Silberman et al., 2012). The representation is the 128-dimensional feature vector immediately before the final 1×1 convolution. The final convolution has four output channels. The first three determine the predicted surface normal after normalization; the fourth channel is excluded from the prediction and loss and is held fixed throughout optimization. Training uses the objective in eq. (20). As in the main analysis, eventual difficulty is the final model's angular error.

Training and checkpoints The model is trained for 20 epochs using AdamW and the optimization settings of the baseline-normal experiment configuration. Automatic mixed precision is used on CUDA, and gradients are clipped according to the configured gradient clipping threshold. The fourth output channel of the final convolution is restored after every optimizer step so that it remains fixed despite the optimizer update. We evaluate the frozen training trajectory at epochs $\{0, 1, 2, 3, 4, 5, 7, 10, 15, 20\}$. The checkpoint at epoch 0 is the initialized model before training.

Data and evaluation The NYUv2 model-training split contains 795 images. Evaluation uses 327 CAL images and 327 image-disjoint TEST images. We deterministically sample 83,712 valid pixels from each split. The same sampled pixel locations are used across training seeds and checkpoints. The eventual-difficulty target is computed from the final model of the corresponding training seed and held fixed across the complete checkpoint trajectory.

B.2 CIFAR-100 CLASSIFICATION

Model and task We use the `VisionTransformer` implementation provided by `torchvision`, adapted to CIFAR-100 inputs. The model uses 32×32 RGB inputs and 4×4 patches, with hidden dimension 192, 12 transformer layers, 3 attention heads, and MLP dimension 768. Dropout and attention dropout are set to zero. The representation is the 192-dimensional CLS token immediately before the final $192 \to 100$ linear classifier. Training uses the objective in eq. (21). As in the main analysis, eventual difficulty is the final model's NLL.

Training and checkpoints The model is trained for 300 epochs with AdamW, learning rate 10^{-3}, weight decay 0.05, and batch size 256. The learning-rate schedule uses a 10-epoch linear warmup followed by cosine decay. Training augmentation consists of random 32×32 crops with padding 4, random horizontal flips, and RandAugment with two operations and magnitude 9. Each training batch then uses either Mixup or CutMix with equal probability, with $\alpha_{\text{mixup}} = 0.2$ and $\alpha_{\text{cutmix}} = 1.0$. Label smoothing is 0.1. CAL and TEST use deterministic preprocessing without training augmentation. We evaluate checkpoints at epochs $\{0, 1, 2, 5, 10, 20, 40, 80, 120, 160, 200, 250, 300\}$, including the initialized model at epoch 0.

Data and evaluation The official CIFAR-100 training set is split deterministically into 45,000 TRAIN examples and 5,000 CAL examples using split seed 424242. The official 10,000-example test set is used as TEST. The same split is used across training seeds. The eventual-difficulty target is computed from the final model of the corresponding training seed and held fixed across checkpoints.

B.3 CIFAR-100 INPAINTING

Model and task The inpainting model uses the same `torchvision` `VisionTransformer` implementation and transformer architecture: 32×32 RGB inputs, 4×4 patches, hidden dimension 192, 12 transformer layers, 3 attention heads, and MLP dimension 768. Each image produces 64 spatial tokens. The CLS token passes through the transformer but is not used by the reconstruction head.

A central 16×16 region is masked. Before model normalization, the masked pixels are replaced by the CIFAR-100 channel mean. A shared $\mathrm{Linear}(192, 48)$ head is applied independently to each spatial token, where the 48 outputs represent the RGB values of one 4×4 patch. The analyzed representation is the 192-dimensional spatial token immediately before this reconstruction head. Training uses the objective in eq. (22). As in the main analysis, eventual difficulty is the final model's per-patch reconstruction MSE.

Training and checkpoints The model is trained for 100 epochs with AdamW, learning rate 10^{-3}, weight decay 0.05, and batch size 256. A 5-epoch linear warmup is followed by cosine decay. Training augmentation consists of random 32×32 crops with padding 4 and random horizontal flips; the central mask is applied after the geometric augmentation. CAL and TEST evaluation is deterministic. We evaluate checkpoints at epochs $\{0, 1, 2, 5, 10, 20, 40, 60, 80, 100\}$, including the initialized model at epoch 0.

Data and evaluation We use the same deterministic 45,000 TRAIN / 5,000 CAL split of the CIFAR-100 training set as in classification, together with the official 10,000-image TEST set. Each image contributes the 16 spatial tokens corresponding to the masked region, yielding 80,000 CAL observations and 160,000 TEST observations. The eventual-difficulty target is computed from the final model of the corresponding training seed and held fixed across checkpoints.

B.4 Checkpoint and evaluation details

Passive directions For every task and training seed, 256 passive directions are generated from independent zero-mean isotropic Gaussian draws, sampled once and held fixed across all checkpoints. In the two CIFAR-100 experiments, $\boldsymbol{a}_j \sim \mathcal{N}(0, I_d/d)$ with $d = 192$. For surface-normal estimation, the Gaussian probe directions are additionally rescaled to a common norm matching the weight norm of the fourth output channel in the baseline model's final 1×1 convolution.

The probe directions are independent of the eventual-difficulty property. No centering is applied as part of the probe construction. Centering is instead applied where required by the individual statistics, as defined in the main text.

Fixed task-readout subspace For the orthogonal and strict analyses, the task-readout projector defined in the main text is constructed from the final readout W and held fixed across all checkpoints. For surface-normal estimation, W contains the three rows of the final convolution that determine the predicted normal; the fixed fourth output channel is not part of the task readout. Thus the task-readout subspace removed by these controls does not change across the training trajectory.

Calibration and TEST evaluation All fitted analysis quantities are estimated using CAL and then frozen before TEST evaluation. For A_G, the bandwidth of each scalar coordinate is its population standard deviation on CAL at the corresponding checkpoint. The primary bandwidth is one CAL standard deviation. For calibrated R^2, the primary analysis uses 20 CAL quantile bins. Bin boundaries and the mean eventual difficulty within each bin are estimated on CAL and then applied unchanged to TEST.

For surface-normal estimation and inpainting, observations from the same image are kept together whenever an analysis procedure requires grouping. TEST is not used to select bandwidths, calibration mappings, learned-probe regularization, or other fitted analysis quantities.

C Passive probe controls

We evaluate four constructions to separate the effects of feature normalization and removal of the final task-readout row space. Let $h_\perp(x) = (I - P_W)h(x)$ and $\boldsymbol{a}_{j,\perp} = (I - P_W)\boldsymbol{a}_j$, where P_W is the projector defined in the main text. The four scalar coordinates are

$$\begin{aligned} q_j^{\mathrm{raw}}(x) &= \boldsymbol{a}_j^\top h(x), & q_j^{\mathrm{unit}}(x) &= \boldsymbol{a}_j^\top \frac{h(x)}{\|h(x)\|}, \\ q_j^{\mathrm{orth}}(x) &= \boldsymbol{a}_{j,\perp}^\top h_\perp(x), & q_j^{\mathrm{strict}}(x) &= \frac{\boldsymbol{a}_{j,\perp}^\top h_\perp(x)}{\|\boldsymbol{a}_{j,\perp}\|_2 \|h_\perp(x)\|}. \end{aligned} \tag{30}$$

Raw applies neither control, unit normalizes the representation only, orth removes the final task-readout row space, and strict combines readout removal with normalization.

As mentioned in section B.4, the experiment implementations differ slightly in the fixed radial scaling assigned to probe directions. Such fixed positive per-probe rescaling does not affect Pearson squared, absolute Spearman, calibrated R^2, or the one-standard-deviation A_G statistic. The same final task-readout subspace is used at every checkpoint.

Table 3 reports the mean within-seed endpoint change and its sample standard deviation across the five training seeds. All entries are multiplied by 10^3.

The qualitative task-level patterns are unchanged across the control constructions. Surface-normal estimation shows positive nonlinear and calibrated changes when normalization or readout removal is applied separately or jointly, classification remains close to its initialization level, and inpainting

Table 3: Passive-probe control analysis. Entries are mean within-seed endpoint change Δ with sample standard deviation across five training seeds, in units of 10^{-3}.

Task	Probe family	ΔA_K	ΔA_G	$\Delta R^2_{\mathrm{cal}}$
Surface normals	Raw	$+7.27 \pm 5.28$	$+7.01 \pm 1.19$	$+36.37 \pm 3.98$
	Unit	-0.92 ± 1.60	$+5.79 \pm 1.42$	$+38.12 \pm 4.92$
	Orth	$+9.26 \pm 4.82$	$+6.39 \pm 1.32$	$+31.90 \pm 3.77$
	Strict	-0.79 ± 2.04	$+5.45 \pm 1.33$	$+34.63 \pm 4.61$
Classification	Raw	-0.11 ± 0.37	-0.03 ± 0.22	$+0.84 \pm 0.47$
	Unit	-0.19 ± 0.36	-0.08 ± 0.21	$+0.60 \pm 0.37$
	Orth	$+0.05 \pm 0.55$	$+0.02 \pm 0.43$	$+1.07 \pm 0.93$
	Strict	-0.25 ± 0.53	-0.13 ± 0.37	$+0.32 \pm 0.62$
Inpainting	Raw	$+3.88 \pm 0.87$	$+2.36 \pm 0.55$	$+4.54 \pm 1.11$
	Unit	$+3.93 \pm 0.88$	$+2.35 \pm 0.56$	$+4.55 \pm 1.12$
	Orth	$+8.00 \pm 1.52$	$+3.98 \pm 0.70$	$+8.35 \pm 1.43$
	Strict	$+8.73 \pm 1.61$	$+4.30 \pm 0.79$	$+9.42 \pm 1.61$

Table 4: Sensitivity of calibrated R^2 to the number of CAL quantile bins. Entries are mean within-seed endpoint change $\Delta R^2_{\mathrm{cal}}$ with sample standard deviation across five training seeds, in units of 10^{-3}.

Task	Probe family	$B = 5$	$B = 10$	$B = 20$	$B = 40$
Surface normals	Raw	$+30.52 \pm 3.92$	$+34.60 \pm 3.85$	$+36.37 \pm 3.98$	$+37.11 \pm 4.01$
	Strict	$+23.27 \pm 3.34$	$+30.59 \pm 4.19$	$+34.63 \pm 4.61$	$+36.39 \pm 4.79$
Classification	Raw	$+0.34 \pm 0.40$	$+0.57 \pm 0.46$	$+0.84 \pm 0.47$	$+0.98 \pm 0.45$
	Strict	$+0.09 \pm 0.57$	$+0.23 \pm 0.60$	$+0.32 \pm 0.62$	$+0.29 \pm 0.63$
Inpainting	Raw	$+4.25 \pm 1.04$	$+4.41 \pm 1.09$	$+4.54 \pm 1.11$	$+4.62 \pm 1.12$
	Strict	$+8.29 \pm 1.48$	$+9.02 \pm 1.57$	$+9.42 \pm 1.61$	$+9.60 \pm 1.63$

shows positive changes throughout. For surface normals, the near-zero strict A_K change arises when readout removal and normalization are combined; the corresponding A_G and calibrated-R^2 changes remain positive.

D Calibration sensitivity

The primary calibrated-R^2 analysis uses 20 CAL quantile bins. To assess sensitivity to this choice, we repeat the endpoint analysis with $B \in \{5, 10, 20, 40\}$ bins while leaving the probe coordinates and eventual-difficulty targets unchanged. Table 4 reports raw and strict probes, the two families used in the main analysis.

The endpoint conclusions are stable across the tested bin counts. Surface-normal estimation and inpainting retain positive calibrated-R^2 changes under both raw and strict probes, whereas classification remains close to zero. The magnitude increases somewhat with finer calibration for surface normals, but the cross-task pattern does not depend on the primary choice $B = 20$.

E Learned-probe protocol

For each task, training seed, and checkpoint, we fit a separate ridge regression from the raw representation to the same eventual-difficulty target used in the main analysis. This provides the learned-probe quantities reported in the main-text comparison.

Let $\boldsymbol{z}_i$ denote the raw representation after feature-wise standardization estimated from the relevant CAL training fold. The learned scalar coordinate is

$$\hat{e}_i = \boldsymbol{b}^\top \boldsymbol{z}_i + c \,, \tag{31}$$

where $\boldsymbol{b}$ and c are fitted by minimizing squared error with an ℓ_2 penalty on $\boldsymbol{b}$. The intercept is not regularized.

Regularization selection The ridge parameter is selected independently for each training seed and checkpoint by five-fold cross-validation on CAL using $\lambda \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000\}$. Feature means and standard deviations are estimated within each cross-validation training fold and applied to its held-out fold, so held-out CAL observations do not contribute to their own standardization.

For surface-normal estimation, folds are grouped by image so that pixels from the same image cannot appear in both the fitting and validation portions of a fold. Inpainting is treated analogously: all 16 masked-patch observations from one image are assigned to the same fold. Classification uses ordinary five-fold splitting over images.

After selecting λ, feature standardization and the ridge model are refit using the complete CAL set and evaluated once on TEST. The regularization parameter is allowed to differ across checkpoints because the learned observer is permitted to adapt independently to each representation.

Matched evaluation Pearson squared and absolute Spearman are computed directly between the learned scalar prediction $\hat{e}$ and eventual difficulty on TEST. For the matched calibrated-R^2 comparison, the learned scalar is passed through the same 20-bin CAL-to-TEST calibration procedure used for passive probes: quantile-bin boundaries and within-bin target means are estimated from CAL, frozen, and then evaluated on TEST.

TEST observations are not used for feature standardization, ridge fitting, regularization selection, or scalar calibration. Direct ridge TEST R^2 is treated as a diagnostic quantity and is not included in the main passive-versus-learned comparison.