READING TRAINING DYNAMICS THROUGH REPRESENTATION DEPENDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural network training continually reshapes how inputs relate to one another in representation space. Tracking these relationships offers a view of learning that prediction error and gradient magnitude alone do not fully describe. We introduce representation dependence (RD), a training signal that aggregates dependence between hidden representations of different inputs. We analyse what RD captures beyond loss and gradient norm, and how changes in training data affect its evolution. In NanoGPT, adding RD features to a baseline combining loss, gradient norms and representation statistics reduces mean absolute error in predicting validation deterioration 250 updates ahead by 13.6%. RD also tracks representation changes under loss-matched label corruption in ViT and separates instruction-tuning fault windows in Pythia. In grokking, changes in RD accompany delayed generalisation after training examples have already been fitted. These findings show that the evolution of cross-example relationships provides useful information for predicting and interpreting training behaviour.

1 INTRODUCTION

Training changes both a model's prediction error and the relationships between its hidden representations of different inputs. Examples can become more aligned or develop different patterns of dependence as the model learns. Training loss records prediction error, while gradient norm measures the magnitude of the gradient driving optimisation. We study how to turn this relational evolution into a signal that can be tracked throughout training, and whether it provides useful information beyond conventional training metrics.

For example, during Pythia instruction tuning, a temporary mixture of chat templates is accompanied by a shift in dependence among the representations of fixed clean examples (Figure 1). Measuring the same probe throughout this interval lets us follow the model's response while the training data change.

Existing diagnostics inspect gradient statistics and attention entropy to understand optimisation and training instability (Schneider et al., 2021; Gray et al., 2024; Zhai et al., 2023). Representation analysis offers complementary tools: anisotropy describes common directions in word and contextual representations (Mu & Viswanath, 2018; Ethayarajh, 2019), spectral rank supports representation assessment (Garrido et al., 2023), and centred kernel alignment (CKA) and singular vector canonical correlation analysis (SVCCA) compare representations across layers, checkpoints, or models (Kornblith et al., 2019; Raghu et al., 2017). These studies establish the value of examining hidden states. Our focus is on relationships among inputs within the current model, measured repeatedly to follow how these relationships evolve and what they indicate about training.

To use these relationships as a training signal, we need to understand what they measure beyond the objective and how reliably a finite set of examples captures their evolution. The analysis must distinguish changes in the model from variation in the monitoring inputs, and separate uncertainty due to probe selection from the approximation introduced by sampling pairs.

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.

We formulate *representation dependence* (RD) as a family of training-time signals. At each checkpoint, we extract representations for a set of examples and average a dependence score across example pairs (Figure 1). A fixed probe follows the same inputs through time; a streamed-batch monitor reuses activations from the training forward pass. Pearson correlation measures angular alignment after centring. A pretrained mutual-information estimator, InfoNet (Hu et al., 2024), supplies a nonlinear instance without fitting a critic at each checkpoint. Both use the same averaging procedure, with different scores capturing different forms of dependence.

Our analysis explains why RD carries information beyond conventional scalar summaries and how training data change the relationships it measures. A construction based on classical deep-linear dynamics (Saxe et al., 2014) gives identical loss and full gradient-norm histories with different RD. For a finite-width, two-layer nonlinear multilayer perceptron (MLP), we then derive how changing the training examples alters probe representations through input overlap, shared active units, and backpropagated error. A finite-step bound relates this response to the direction and size of the RD change. Finally, we quantify how sampling pairs and probes affects estimation accuracy, applying classical U-statistic analysis (Clémençon et al., 2016) to the monitoring problem.

We evaluate these signals in NanoGPT, ViT, Pythia and grokking, testing prediction of future validation deterioration and representation changes during perturbations or delayed generalisation. Loss and gradient norm are the core scalar baselines in the prediction and perturbation studies.

Our contributions are:

- **A relational training signal.** We formulate RD as a common framework for tracking cross-example relationships throughout training, with linear and nonlinear measurements that summarise how a model organises inputs within its hidden representations.
- **Theoretical analysis.** We show that identical loss and gradient-norm histories can coexist with different RD, derive its response to data perturbations in a two-layer nonlinear network, and characterise convergence under pair and probe sampling.
- **Predictive and interpretive value.** RD features improve forecasts of validation deterioration beyond conventional monitors. Perturbation and grokking studies show how RD changes under altered supervision and during generalisation after training fit.

2 Representation Dependence

Representation dependence (RD) tracks how relationships between inputs change throughout training. At each checkpoint, we extract a hidden representation for each monitored input and average a dependence score over pairs of inputs. We use Pearson correlation for linear alignment and a pretrained mutual-information estimator for nonlinear dependence.

2.1 Monitoring formulation

Let θ_t be the model parameters at checkpoint t, and let $\mathcal{X} = \{x_1, \dots, x_N\}$ contain $N \geq 2$ monitored inputs. A fixed representation map converts the activation at layer ℓ into a D-dimensional vector $h_{t,i} = h_\ell(x_i; \theta_t) \in \mathbb{R}^D$. For a pair score $g : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ and a set $\mathcal{P}$ of K distinct pairs (i, j) with $i < j$ and $1 \leq K \leq \binom{N}{2}$, define

$$\mathrm{RD}_t^{(g)} = \frac{1}{K} \sum_{(i,j)\in\mathcal{P}} g(h_{t,i}, h_{t,j}). \tag{1}$$

The layer, map, and score remain fixed throughout a trajectory, so changes in RD can be compared across checkpoints.

With *fixed probes*, we encode the same inputs at every checkpoint to isolate changes in the model. With *streamed batches*, we reuse the current training batch's activations; the score then reflects both the model and the sampled inputs. Fixed probes require an additional forward pass, whereas streamed batches reuse the training forward pass. The monitoring interval sets the temporal resolution. Appendix A specifies these choices for each experiment.

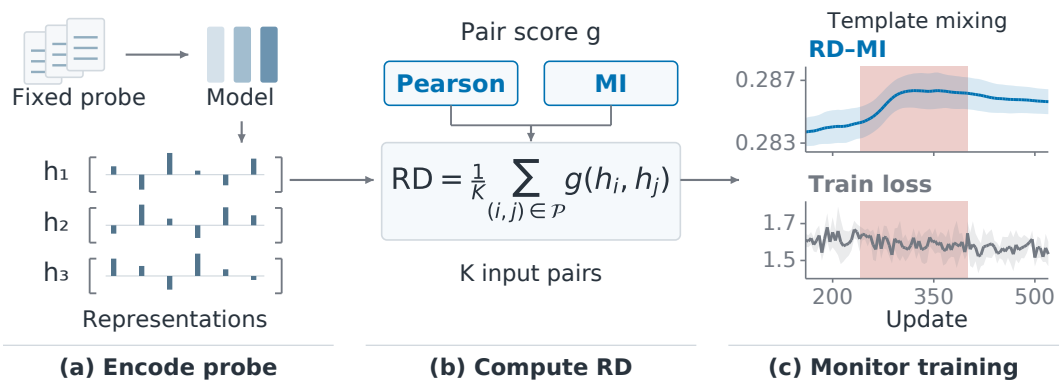

Figure 1: **Monitoring training through representation dependence (RD).** (a) Encode a fixed probe at a chosen layer. (b) Average a pair score g over K input pairs to obtain RD: g is Pearson correlation or an MI estimate, and h_i denotes the representation of input i at the current checkpoint. (c) Track RD during training: Pythia template-mixing trajectories show mean $\pm$ SD over five fine-tuning seeds, with fault updates 240–399 shaded. RD–MI uses InfoNet at the first block; training loss is shown below. Measurements are from Section 4.3.

2.2 Pearson and Mutual-Information Scores

Pearson correlation. Pearson correlation compares the directions of two representations after centring each vector across its coordinates. Let I_D be the D-dimensional identity matrix, $\mathbf{1} \in \mathbb{R}^D$ the all-ones vector, and $C = I_D - \mathbf{1}\mathbf{1}^\top/D$ the centring matrix. Define $z_{t,i} = Ch_{t,i}/\|Ch_{t,i}\|_2$, with $z_{t,i} = 0$ when $Ch_{t,i} = 0$. The pair score is $g_{\mathrm{P}}(h_{t,i}, h_{t,j}) = z_{t,i}^\top z_{t,j}$. Its complete-pair mean, denoted by RD-P$_t$, is

$$\text{RD-P}_t = \frac{\|\sum_i z_{t,i}\|_2^2 - \sum_i \|z_{t,i}\|_2^2}{N(N-1)}. \tag{2}$$

Thus all pairs can be averaged exactly in $O(ND)$ time without constructing a pairwise matrix. Centring removes constant coordinate offsets, and normalisation removes positive per-input scaling. Omitting centring gives the cosine score with the same computational reduction.

Mutual information. To measure nonlinear dependence, treat the corresponding coordinates of a pair as a sample $\mathcal{S}_{ij} = \{(h_i[d], h_j[d])\}_{d=1}^D$, where d indexes coordinates and the checkpoint index is omitted. We transform each marginal to normalised ranks and pass the resulting sample to a fixed pretrained estimator $\mathcal{E}$. This gives the pair score $g_{\mathrm{MI}}(h_i, h_j) = \mathcal{E}(\mathrm{rank}(\mathcal{S}_{ij}))$. We implement $\mathcal{E}$ with InfoNet (Hu et al., 2024), which requires no critic fitting at the monitored checkpoint, and call the resulting mean RD–MI. The score concerns dependence across corresponding representation coordinates; the representation map and coordinate correspondence are fixed. Appendix B.1 clarifies this score interpretation and gives an example of nonlinear dependence that linear and spectral summaries leave unchanged.

Scoring a subset of pairs reduces the cost of RD–MI. Reusing the same pairs across checkpoints permits paired comparisons over time. Section 3.3 gives the convergence rate as the pair budget increases; Appendix A.5 evaluates this trade-off empirically.

3 Theoretical Analysis

We first show that loss and gradient-norm histories leave some representation geometry undetermined. We then analyse how altered training examples change Pearson RD, and how pair sampling affects RD–MI precision. The Pearson analysis uses complete-pair means on fixed probes with nonzero centred representations. Proofs are in Appendix B.2 and Appendix B.4.

3.1 Relation to loss and gradient norm

Proposition 1 (Identical scalar histories, different RD)**.** There exist two gradient-flow trajectories of a two-layer linear network, trained on the same squared-loss task and input distribution, whose loss and full parameter-gradient norm agree at every time, but whose Pearson RD differs on the same fixed probe.

The construction uses the balanced mode dynamics of Saxe et al. (2014). The models share learned mode strengths but differ in orientation relative to the inputs, changing cross-example angles while preserving loss and gradient magnitude.

3.2 Response to training updates

Geometry of the response. Let $\mu = N^{-1}\sum_i z_i$ be the mean normalised direction at a checkpoint, where z_i is defined in Section 2. Because $\|z_i\|_2 = 1$, Eq. 2 reduces to RD-P $= (N\|\mu\|_2^2 - 1)/(N-1)$. Larger RD therefore means a stronger common direction among the probes. Equivalently, the mean squared distance between their unit directions satisfies

$$\frac{1}{\binom{N}{2}} \sum_{i<j} \|z_i - z_j\|_2^2 = 2(1 - \text{RD-P}). \tag{3}$$

This connects RD to representation anisotropy and angular dispersion (Ethayarajh, 2019; Li et al., 2026). For an update that changes the mean direction from μ to $\mu + \Delta\mu$,

$$\Delta\text{RD-P} = \frac{N}{N-1}\left(2\mu^\top \Delta\mu + \|\Delta\mu\|_2^2\right). \tag{4}$$

This identity holds for any architecture and optimiser with nonzero centred representations at both endpoints. Both useful learning and altered training data can strengthen a shared direction, so the sign of RD change alone does not distinguish them.

A two-layer network. Consider an input $x \in \mathbb{R}^p$, a hidden layer $h_W(x) = \text{ReLU}(Wx) \in \mathbb{R}^D$ and logits $f(x) = Vh_W(x) \in \mathbb{R}^m$, with $W \in \mathbb{R}^{D\times p}$ and $V \in \mathbb{R}^{m\times D}$. Both weight matrices are updated by stochastic gradient descent (SGD) with step size $\eta > 0$. For a labelled input (x, y), let $e(x, y) = \nabla_f \ell(f(x), y) \in \mathbb{R}^m$ be the derivative of the per-example loss ℓ. Write $D_x = \text{diag}(\mathbf{1}\{Wx > 0\})$ for the initial ReLU activation mask. Let P and Q be the empirical distributions of ordinary and alternative update examples, allowing weighted averages. The ordinary first-layer gradient is

$$A_P = \mathbb{E}_{(x,y)\sim P}[D_x V^\top e(x, y)x^\top]. \tag{5}$$

Replace P by $P_\varepsilon = (1-\varepsilon)P + \varepsilon Q$, where $\varepsilon \in [0, 1]$ is the alternative-data mixture weight. Defining A_Q analogously and $\Delta A = A_Q - A_P$, the two updates from the same checkpoint produce $W_c = W - \eta A_P$ and $W_a = W - \eta(A_P + \varepsilon\Delta A)$.

Proposition 2 (RD response to a data update)**.** Suppose both updates preserve the activation mask $D_i = D_{x_i}$ of each probe. At the ordinary endpoint W_c, define the centred vector $s_i = Ch_{W_c}(x_i)$, its norm $r_i = \|s_i\|_2 > 0$, the unit direction $z_i = s_i/r_i$, and their mean $\mu = N^{-1}\sum_i z_i$. Let

$$v_i = -\eta\varepsilon C D_i \Delta A x_i, \qquad \gamma_i = \frac{2(I_D - z_i z_i^\top)\mu}{(N-1)r_i}. \tag{6}$$

Here v_i is the centred representation difference between the altered and ordinary endpoints. If $\kappa_i = \min_{0\le u\le 1}\|s_i + uv_i\|_2 > 0$, then

$$\text{RD-P}(W_a) - \text{RD-P}(W_c) = \sum_i \gamma_i^\top v_i + \mathcal{R}, \qquad |\mathcal{R}| \le \frac{3}{N}\sum_i \frac{\|v_i\|_2^2}{\kappa_i^2}. \tag{7}$$

For a training example (x, y), the contribution to the probe update is proportional to $(x_i^\top x)D_i D_x V^\top e(x, y)$. Input overlap and shared active units determine how its error changes the probe representation. The factor $I_D - z_i z_i^\top$ in γ_i removes changes in magnitude and retains changes in direction. Their alignment with μ determines the first-order RD response. For fixed ΔA and path norms bounded away from zero, the first-order term is $O(\eta\varepsilon)$ and the remainder is $O(\eta^2\varepsilon^2)$; the first-order term may vanish. Appendix B.2 also treats changing activation masks and smooth activations.

General networks. At a differentiable checkpoint with model parameters θ, let $J_i = \partial h_\ell(x_i;\theta)/\partial\theta$ be the Jacobian of a probe representation. An infinitesimal update $d\theta$ gives $dh_i = J_i\, d\theta$ and $d\text{RD-P} = \sum_i \gamma_i^\top C J_i\, d\theta$, with γ_i evaluated at the current representations. This applies the gradient-based sensitivity principle used in training-data influence analysis (Pruthi et al., 2020; Silva et al., 2022) to RD's angular geometry. Appendix B.3 expands the attention, feed-forward, residual and normalisation contributions in Transformers (Vaswani et al., 2017).

3.3 Convergence Rate of RD–MI

Fix a checkpoint and its $M = \binom{N}{2}$ pair scores $q_{ij} = g_{\text{MI}}(h_i, h_j)$, including any estimator randomness. Denote their complete mean by $U_N = M^{-1}\sum_{i<j} q_{ij}$ and, for $M > 1$, their variance by $S_g^2 = (M-1)^{-1}\sum_{i<j}(q_{ij} - U_N)^2$.

Proposition 3 (Pair-sampling convergence)**.** Sample $1 \le K \le M$ of the $M > 1$ pairs uniformly without replacement, and let $\widehat{U}_K$ be their mean. Conditional on the fixed scores,

$$\mathbb{E}[\widehat{U}_K] = U_N, \qquad \sqrt{\mathbb{E}[(\widehat{U}_K - U_N)^2]} = S_g\sqrt{\frac{1}{K} - \frac{1}{M}}. \tag{8}$$

The root-mean-square error decreases at rate $O(K^{-1/2})$ and vanishes when all pairs are used. The case $M = K = 1$ is also exact. Now suppose the probes are independent draws from a common distribution, independent of the trained model and a fixed symmetric scorer. The population estimation error is $O_P(N^{-1/2} + K^{-1/2})$ under finite score variance. Here O_P denotes order in probability. Appendix B.4 derives this rate using U-statistic analysis (Clémençon et al., 2016). These rates describe convergence to the chosen MI estimator's mean score; they do not remove estimator bias.

4 Experiments

We evaluate RD in four settings: forecasting validation deterioration in NanoGPT, tracking label corruption in ViT, separating instruction-tuning fault windows in Pythia, and following delayed generalisation in grokking. Training loss and global gradient norm are the core baselines in the prediction and perturbation studies, alongside spectral and task-specific measurements. We use the output of the first Transformer block for RD in all main experiments; layer ablations appear in Appendix C.

Evaluation. We report uncertainty over complete training runs or pretrained models; figure captions specify the corresponding aggregation and error bands. Reported confidence intervals are 95% bootstrap intervals. Measurements that require probe labels are identified separately in the tables. Model configurations, predictor inputs, and sampling details appear in Appendix A.

4.1 NanoGPT: representation dynamics and future deterioration

Training loss can continue to fall while validation loss rises. Figure 2 illustrates this divergence in NanoGPT (Karpathy, 2022) on WikiText-2 (Merity et al., 2017): RD–MI and RD–Pearson track the accompanying changes in hidden-space relationships.

Predicting future overfitting. We test whether these measurements help predict subsequent validation performance. In a separate study, we trained a 124M-parameter NanoGPT on a 15,000-token WikiText-2 subset for 30,000 updates. Ridge regression uses current monitoring signals to predict validation deterioration 250 updates ahead, measured as validation negative log-likelihood (NLL) above its running minimum. We fitted on four training seeds and tested on the fifth, rotating across all five seeds. Each input set used the same evaluation procedure; regularisation was selected using only the four fitting seeds. Full settings appear in Appendix A.

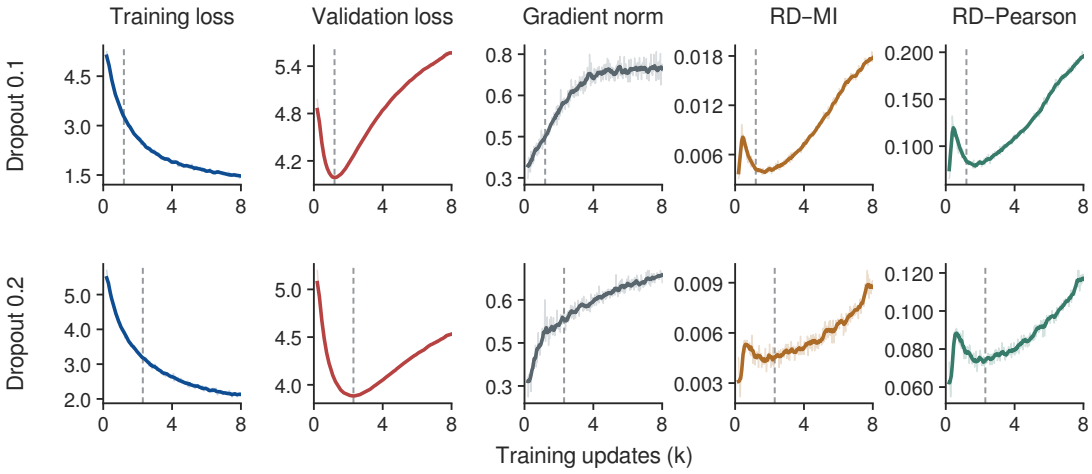

Figure 2: **Representation dynamics during overfitting.** Rows show dropout 0.1 and 0.2, with learning rates 0.001 and 0.0005, respectively. Columns show training loss, validation loss, global gradient norm, RD–MI and RD–Pearson. Faint lines are raw measurements; solid lines are 11-point moving averages. Dashed lines mark the smoothed validation minima.

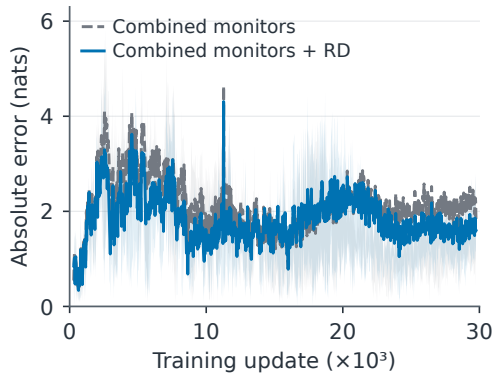

Figure 3: **Forecasting overfitting.** Ridge regression predicts NanoGPT's validation NLL above its running minimum 250 updates ahead. Lines show mean absolute error over five held-out seeds; bands show the interquartile range. Grey: combined monitors; blue: adding the RD features in Table 1.

Inputs	MAE ↓	R^2 ↑
Training monitors		
Training loss	3.209 ± 1.780	0.006
Gradient norm	3.759 ± 2.212	-0.459
Loss + gradient norms	2.858 ± 1.894	0.152
Spectral statistics	2.053 ± 1.407	0.530
Combined monitors	2.130 ± 1.395	0.515
Combined monitors + RD	$\mathbf{1.840 \pm 0.960}$	**0.654**
Validation feedback		
Validation loss	0.475 ± 0.160	0.968

Table 1: **Forecast accuracy.** Combined monitors include loss, gradient norms, and input and representation statistics. RD adds first-block Pearson and related linear features (Appendix A). All predictors include update and learning rate. MAE: mean ± SD (nats) over five held-out seeds; R^2: pooled test predictions. Validation feedback uses current validation loss.

Adding these linear RD features reduced prediction mean absolute error (MAE) from 2.130 to 1.840 nats (Table 1), a 13.6% reduction with improvement in four of five seeds. The paired reduction was 0.290 [0.009, 0.683] nats. Figure 3 shows how prediction error evolves during training. RD features therefore improved forecasting beyond the combined monitoring signals in this setting.

4.2 ViT: RESPONSE TO LABEL CORRUPTION

We test how RD responds to incorrect supervision when training losses are closely matched. We branched eight independently pretrained ViT-B/16 models (Dosovitskiy et al., 2021) on ImageNet-1K (Deng et al., 2009) from their 50k checkpoints. For five updates, we replaced each batch with incorrectly labelled image–label pairs selected to match the original per-example losses under the current model. A paired branch continued normal training; both branches then resumed ordinary data. We measured RD on five fixed probe populations.

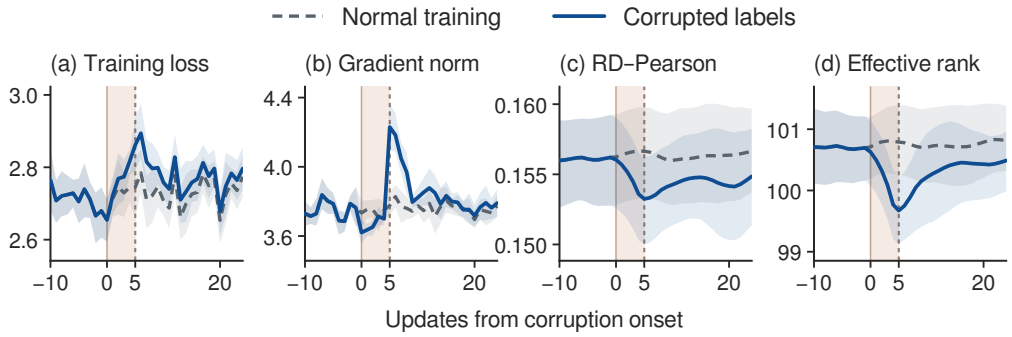

Figure 4: **Representation changes under loss-matched label corruption.** Eight ViT models continue normal training or receive loss-matched, incorrectly labelled image–label pairs at updates 0–4 (shaded). The dashed boundary at update 5 marks the return to ordinary training. Lines connect mean raw scores at recorded updates; bands give 95% bootstrap intervals across models, after averaging two continuation orders and, for representation metrics, five probe populations. RD and effective rank use the first Transformer block. Loss is in nats; no temporal smoothing is applied.

Figure 4 shows that RD decreases during corruption and remains separated from normal training into the recovery period, despite closely matched training losses. Gradient norm and effective rank also respond to the intervention. At the final corruption update, held-out top-1 accuracy fell by 1.77 percentage points relative to normal training. RD thus tracks changes in relationships among probe examples during a perturbation that affects model performance. Matching details and additional replacement fractions appear in Appendix A.2.

4.3 Pythia: INSTRUCTION-TUNING FAULT WINDOWS

We fine-tuned Pythia-2.8B (Biderman et al., 2023) with LoRA (Hu et al., 2022) on UltraChat (Ding et al., 2023), introducing mismatched instruction–response pairs, truncated responses, or mixed chat templates during updates 240–399 of a 600-update run. A fixed clean 64-example probe tracks representation dependence throughout training. Both RD variants use the first Transformer block for all three interventions; we evaluated five new seeds per intervention after selecting the monitoring configuration on a separate development set.

Table 2 reports rank separation between observations inside and outside the intervention window. RD–MI exceeds training loss and gradient norm in all three settings, with median separation of 0.550, 0.717 and 0.612. Activation SD is strongest for template mixing. The comparison includes ordinary training drift outside the known fault window. Layer ablations and implementation details are reported in Appendices C and A.

Table 2: **Pythia instruction-tuning fault-window separation.** Medians over five seeds per fault (updates 240–399). $D = |2\,\mathrm{AUROC} - 1|$ compares fault-window scores with all other observations, including recovery: 0 denotes AUROC 0.5; 1 denotes perfect separation in either direction. RD uses a fixed clean 64-example probe at the first block: Pearson uses masked-token representations; MI averages hidden coordinates into 32 groups per token. Bold marks column maxima; the additional labelled-probe comparison is reported in Appendix A.

	Fault-window separation $D \uparrow$		
Monitor	Mismatch	Truncation	Template mixing
Training monitors			
Training loss	0.129	0.114	0.064
Gradient norm	0.061	0.061	0.030
Activation SD (L16)	0.037	0.143	**0.750**
RD–Pearson (L1)	0.182	0.204	0.276
RD–MI (L1)	**0.550**	**0.717**	0.612

4.4 GROKKING

Figure 5 follows delayed generalisation after training fit (Power et al., 2022). Two runs with identical hyperparameters reach near-zero training loss, but only one generalises within 50k updates. In that run, mean Pearson falls while mean absolute Pearson and MI rise around the test-accuracy transition. Four public trajectories (Smith, 2022) show a similar early-to-late ordering of Pearson excursions, MI rises and generalisation within 15k updates.

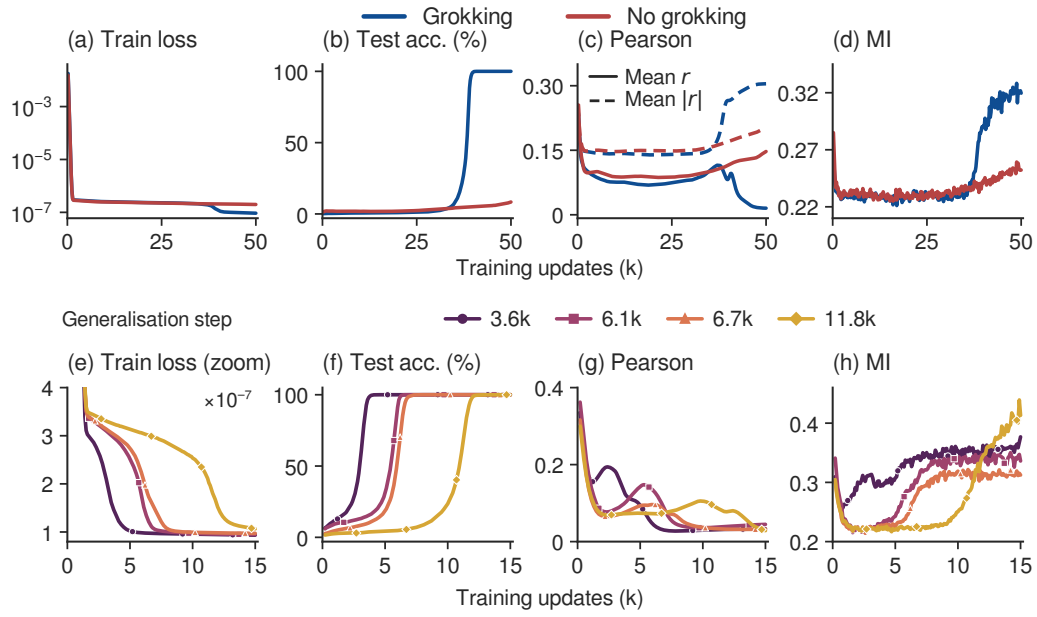

Figure 5: **RD changes accompany grokking at different training stages.** One-layer Transformers learn addition modulo 113 with weight decay 1. **Top:** two runs with identical hyperparameters and 20% training data, differing in initialisation and data split. **Bottom:** four public runs with 30% training data, labelled by the first sustained 95% test-accuracy crossing. Panel (e) zooms to $[0.8, 4.0] \times 10^{-7}$; earlier, larger losses lie outside the axes. Pearson is the mean correlation; dashed lines in (c) show mean absolute correlation. All curves start at update 200, without smoothing. Settings: Appendix A.4.

5 Related Work

Training monitoring. Gradient statistics, curvature and attention entropy provide established views of optimisation dynamics (Mahsereci et al., 2017; Schneider et al., 2021; Zhai et al., 2023). Gradient confusion and coherence measure agreement between example or mini-batch updates (Sankararaman et al., 2020; Chatterjee, 2020). Example forgetting and dataset cartography characterise learning through per-example training behaviour (Toneva et al., 2019; Swayamdipta et al., 2020). Representations also support monitoring decisions: Symbolic Early Stopping uses their evolution to inform stopping (Tomilov et al., 2026). RD measures relationships among the hidden representations of a fixed set of inputs. We study what this signal adds to conventional training monitors, both by analysing trajectories with matching scalar histories and predicting subsequent validation deterioration.

Representation geometry during learning. Anisotropy describes shared directions in word and contextual representations (Mu & Viswanath, 2018; Ethayarajh, 2019; Godey et al., 2024); its evolution across depth and training has also been studied (Machina & Mercer, 2024). A closely related measure, average pairwise cosine distance, tracks perplexity and supports label-free prioritisation of language models (Li et al., 2026). Pearson RD is directly related to centred angular dispersion (Eq. 3). We use this connection to study what loss and gradient-norm histories leave undetermined, and how training data alter relationships among probe representations.

Other geometric summaries capture different properties. CKA and SVCCA compare representations across layers, checkpoints or models (Kornblith et al., 2019; Raghu et al., 2017), while RankMe summarises their spectrum (Garrido et al., 2023). Studies of feature learning and generalisation connect representation geometry to model behaviour (Chou et al., 2025; 2026). Loss plateaus and delayed generalisation further motivate examining changes beyond training fit (Gopalani & Hu, 2025; Power et al., 2022). Our experiments follow these relationships under altered supervision and during generalisation.

Dependence estimation. The Hilbert–Schmidt independence criterion (HSIC), distance correlation and the randomised dependence coefficient (RDC) measure departures from independence (Gretton et al., 2005; Székely et al., 2007; Lopez-Paz et al., 2013). Mutual Information Neural Estimation (MINE) fits a critic, whereas InfoNet amortises estimation through pretraining (Belghazi et al., 2018; Hu et al., 2024). Finite-sample behaviour is central to interpreting neural MI estimates (Poole et al., 2019; McAllester & Stratos, 2020). We use InfoNet as a nonlinear estimator within RD. The dependence is measured between probe representations, a different quantity from the input–representation dependence studied in information-plane analyses (Goldfeld et al., 2019).

6 Conclusion

Cross-example representation dependence offers a view of training through the evolving relationships among inputs. Our analysis shows that identical loss and gradient-norm histories can coexist with different representation dependence, and characterises its response to changes in training data. Experiments show that RD features improve forecasts of validation deterioration and track representation changes during controlled perturbations and delayed generalisation. These results support using cross-example relationships alongside conventional training metrics to understand learning.

AI use statement. Generative AI tools assisted with literature discovery, manuscript development, experiment planning, code drafting, and preparation of figures and LaTeX. The authors reviewed the resulting text, code, measurements, and citations and take responsibility for the paper.

Reproducibility statement. Appendix A documents the models, data, monitoring choices and evaluation protocols. Additional analyses cover layer and cadence sensitivity, pair-sampling precision and computational cost. Appendices B.1 and B.2 give the nonlinear example and supporting derivations. Configurations, recorded trajectories and analysis code are retained for reproduction.

REFERENCES

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 531–540. PMLR, 2018.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023.

Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? fixing ImageNet out-of-distribution detection evaluation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2471–2506. PMLR, 2023. URL https://proceedings.mlr.press/v202/bitterwolf23a.html.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020.

Chi-Ning Chou, Hang Le, Yichen Wang, and Sueyeon Chung. Feature learning beyond the lazy–rich dichotomy: Insights from representational geometry. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10700–10740, 2025.

Chi-Ning Chou, Artem Kirsanov, Yao-Yuan Yang, and SueYeon Chung. Diagnosing generalization failures from representational geometry markers. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c2fQBcoKhU.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Stephan Clémençon, Igor Colin, and Aurélien Bellet. Scaling-up empirical risk minimization: Optimization of incomplete U-statistics. *Journal of Machine Learning Research*, 17(76):1–36, 2016. URL https://jmlr.org/papers/v17/15-012.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051. Association for Computational Linguistics, 2023.

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.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 55–65. Association for Computational Linguistics, 2019.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10929–10974. PMLR, 2023.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. Anisotropy is inherent to self-attention in transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 35–48. Association for Computational Linguistics, 2024.

Ziv Goldfeld, Ewout Van Den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2299–2308. PMLR, 2019.

Pulkit Gopalani and Wei Hu. What happens during the loss plateau? understanding abrupt learning in transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Gavia Gray, Aman Tiwari, Shane Bergsma, and Joel Hestness. Normalization layer per-example gradients are sufficient to predict gradient noise scale in transformers. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2411.00999.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert–Schmidt norms. In *Algorithmic Learning Theory*, pp. 63–77. Springer, 2005.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Zhengyang Hu, Song Kang, Qunsong Zeng, Kaibin Huang, and Yanchao Yang. InfoNet: Neural estimation of mutual information without test-time optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19283–19303. PMLR, 2024.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Andrej Karpathy. nanoGPT: The simplest, fastest repository for training and fine-tuning medium-sized GPTs. https://github.com/karpathy/nanoGPT, 2022.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

Yanhong Li, Ming Li, Karen Livescu, and Jiawei Zhou. On the predictive power of representation dispersion in language models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qvVrAMdK2F.

David Lopez-Paz, Philipp Hennig, and Bernhard Schölkopf. The randomized dependence coefficient. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Anemily Machina and Robert Mercer. Anisotropy is not inherent to transformers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4892–4907. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.274.

Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*, 2017.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 875–884. PMLR, 2020.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5171–5180. PMLR, 2019.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

A. Rényi. On measures of dependence. *Acta Mathematica Academiae Scientiarum Hungaricae*, 10: 441–451, 1959. doi: 10.1007/BF02024507.

Karthik Abinav Sankararaman, Soham De, Zheng Xu, W. Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8469–8479. PMLR, 2020.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6120.

Frank Schneider, Felix Dangel, and Philipp Hennig. Cockpit: A practical debugging tool for the training of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20825–20837, 2021.

Andrew Silva, Rohit Chopra, and Matthew Gombolay. Cross-loss influence functions to explain deep network representations. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 1–17. PMLR, 2022. URL https://proceedings.mlr.press/v151/silva22a.html.

Jess Smith. grokking-saved-runs. Hugging Face model repository, 2022. URL https://huggingface.co/smithjessk/grokking-saved-runs. Revision 3799fe7b4c6abd4d81df7e55972fbefc6a9f0c3f.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9275–9293. Association for Computational Linguistics, 2020.

Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

Ivan Tomilov, Rodion Zamotaev, Natalia Gusarova, and Aleksandra Vatian. Symbolic early stopping in neural sequence models via mapper-induced symbolic dynamics. *Technologies*, 14(6):339, 2026. doi: 10.3390/technologies14060339.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40770–40803. PMLR, 2023.

Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. *Journal of Data-centric Machine Learning Research*, 2(3):1–32, 2024. URL https://data.mlr.press/assets/pdf/v02-3.pdf.

A Experimental Settings and Measurement Studies

Shared measurement definitions. Training loss is the mini-batch objective. Probe task NLL is next-token negative log-likelihood on supervised response positions. Gradient norm is the global ℓ_2 norm over parameter gradients. The Pythia parameter-norm baseline is the mean of per-tensor ℓ_2 norms over all model parameters, including the frozen base.

RD–Cosine averages uncentred cosine similarity across the declared example pairs and is invariant to positive per-example rescaling. Activation standard deviation and root-mean-square (RMS) summarise activation magnitude. Let σ_j denote the singular values of the centred example-by-feature matrix, and $r_{\max}$ its maximum possible rank. Effective rank is $\exp[-\sum_j p_j \log p_j]$; stable rank is $\sum_j \sigma_j^2 / \max_j \sigma_j^2$; participation rank is $(\sum_j \sigma_j^2)^2 / \sum_j \sigma_j^4$; and normalised singular-value-decomposition (SVD) entropy is $-\sum_j p_j \log p_j / \log r_{\max}$, where $p_j = \sigma_j^2 / \sum_k \sigma_k^2$.

Activation CKA compares the current fixed-probe sample Gram matrix with its step-zero counterpart (Kornblith et al., 2019). SVCCA compares current and step-zero fixed-probe activations, retaining principal components explaining 99% of variance before canonical correlation (Raghu et al., 2017). Attention entropy averages causal attention entropy over valid heads and query positions.

A.1 NanoGPT forecasting protocol

The two natural-overfitting examples use 12 layers, width 256, context 256, batch size 128, and weight decay 0.05. Their learning rates and dropout probabilities are $(10^{-3}, 0.1)$ and $(5\times10^{-4}, 0.2)$. Figure 2 shows the MI and Pearson scores for each trajectory. Both learning rate and dropout vary between the two runs. The validation minimum is marked retrospectively.

The forecasting study uses 12 layers, 12 attention heads, width 768, context 256, batch 128, bfloat16, learning rate 0.003 with minimum 0.0003, weight decay 0.1, dropout 0, and gradient clipping at 1. It trains on the first 15,000 tokens of the WikiText-2 training stream for 30,000 updates. The five training seeds are 8171–8175. Monitoring occurs every 25 updates. Validation deterioration is

$$y_t = u_t - \min_{s \le t} u_s,$$

where u_t is validation NLL on the disjoint outcome sample. The prediction target is y_{t+250}. Each predictor includes update number and learning rate. The remaining features enter as both raw values and differences from their within-run median over updates 100–300. Prediction begins at update 300; final observations without a 250-update-ahead target are excluded. The probe and outcome sets contain different examples. Monitoring preserves the state of the training sampler's random-number generator.

The loss-and-gradient predictor uses training loss and global and layer-wise gradient norms. The spectral predictor uses participation/stable ranks, SVD entropy, and pooled spectral summaries. The combined monitors additionally include input composition, parameter/activation scale, and probe-activation CKA and SVCCA. The RD augmentation contains Pearson, cosine, covariance, and dot-product scores on the current training batch at the first Transformer block (code index 0). These four

scores and their early-reference differences add eight features to the 142 combined-monitor features. Appendix C compares this readout with the original three-layer augmentation. The covariance feature is the unnormalised centred product $(h_i - \bar{h}_i\mathbf{1})^\top(h_j - \bar{h}_j\mathbf{1})$; the dot-product score is $h_i^\top h_j$. Both retain magnitude information removed by correlation normalisation.

Each outer fold holds out one complete training seed. Inner folds hold out complete seeds from the remaining training set to select ridge $\alpha \in \{0.3, 3, 30, 300\}$. Median imputation and feature standardisation are fit within each training fold. Sample weights give each training trajectory equal total weight. MAE is computed for each held-out seed, then averaged across seeds. The paired gain interval resamples these five outer-seed errors.

The five seeds are new development runs of a setting selected from a six-setting, three-seed screen: token caps 15k, 60k and 240k, and no-weight-decay, weight-decay-1 and dropout-0.1 variants at a 30k token cap. The screen compared four RD augmentations (normalised or full linear scores, each on training batches or a fixed probe) against seven baseline panels. Selection used prediction gains at a 250-update horizon; the full training-batch RD feature set and combined monitors were then fixed for the five new seeds. The reported five-seed comparison concerns the selected 15k configuration; all six settings are retained in the development record. The same folds and alpha grid are used for the additional loss-only and gradient-only rows.

Validation-loss reference. The validation-loss predictor uses current next-token NLL on the fixed probe, its change from the early reference, update number and learning rate. The probe and the outcome sample are disjoint; this predictor receives no future losses. The combined monitors exclude this validation-loss input. Table 1 reports mean and SD of MAE across held-out seeds; R^2 pools their test predictions.

A.2 ViT LABEL-CORRUPTION PROTOCOL

Eight ViT-B/16 models (seeds 27301–27308) are pretrained on all 1,281,167 ImageNet-1K training images with batch 128, float32, and AdamW at learning rate 10^{-4} and weight decay 0.05. From each 50k checkpoint, we restore model, optimiser and scheduler states for two continuation orders, each with normal training and 25%, 50% and 100% replacement branches (64 trajectories). Continuations use batch 256. Injection occurs at updates 50,200–50,204; records cover ten preceding and twenty subsequent updates. Loss and gradient norm are measured before the optimiser step, and representations are measured after that update.

The five fixed, 128-example probes are CIFAR-100, OpenImage-O, iNaturalist, NINCO and SSB-Hard (Zhang et al., 2024; Bitterwolf et al., 2023). RD and effective rank use flattened first-block token representations. For effective rank, coordinates are centred across examples before computing singular-value energies. The 5,000-image evaluation set is disjoint from the matching candidates and probes. Figure 4 averages populations and continuation orders within each model, then reports means and pointwise 95% model-bootstrap intervals (50,000 draws, RNG seed 20260920).

Matching the training losses. At each affected update, we evaluate the original correctly labelled batch under the current intervention model. We then select distinct images and incorrect labels from a 40,960-image candidate pool to match the replaced positions' per-example losses. The pool size was chosen using separate development models. Both selection-time and actual training losses must meet relative tolerances of 1% for mean loss, 2.5% for Wasserstein distance normalised by mean target loss, and 5% for the 10th, 25th, 50th, 75th and 90th percentiles. All 240 affected updates pass, including all 80 at each replacement fraction. The loss objective and its gradient are computed normally on the selected pairs.

Replacement fraction. For each monitor, Table 3 reports the mean absolute intervention-minus-control difference over the five affected updates, divided by the control's temporal SD over the preceding ten updates (with a 10^{-6} floor). We average within each model before bootstrapping. RD and effective rank respond more strongly as the replacement fraction increases; effective rank has the largest response at full replacement. Appendix C compares first-block RD with the original block-6 analysis on the same trajectories.

Table 3: **ViT responses at different replacement fractions.** Mean absolute difference from normal training during the five corrupted-label updates, divided by each monitor's normal pre-window SD. Brackets give 95% bootstrap intervals over eight models. RD and effective rank use the same first-block representations.

Monitor	25%	50%	100%
Training loss	0.037 [0.028, 0.049]	0.087 [0.068, 0.109]	0.220 [0.178, 0.268]
Gradient norm	0.544 [0.450, 0.652]	0.905 [0.762, 1.043]	0.957 [0.756, 1.134]
RD–Pearson	0.632 [0.518, 0.766]	0.901 [0.801, 1.007]	1.688 [1.283, 2.058]
Effective rank	0.629 [0.479, 0.792]	1.489 [1.059, 1.905]	3.840 [2.962, 4.806]

Independent OOD comparator. A separate study of six pretrained models substitutes Describable Textures Dataset (DTD) images (Cimpoi et al., 2014) for ordinary ImageNet batches. For RD–Pearson, known-window separation is 0.718 with OOD intervention and 0.689 with ordinary training; the corresponding values are 0.903/0.313 for gradient norm and 0.202/0.217 for loss. Thus gradient norm is more discriminative in that OOD setting. The main experiment instead studies in-domain wrong-label replacement.

A.3 Pythia layer transfer and temporal sampling

Pythia-2.8B starts from the pretrained update-100,000 checkpoint. Fine-tuning uses float32, LoRA rank 8, alpha 16, dropout 0.05, learning rate 2×10^{-5}, weight decay 0, clipping at 1, sequence length 512, physical batch 30, and two gradient-accumulation steps. The shared clean probe has 64 examples sampled with seed 950001 from the first 1,024 training examples. It remains fixed across all 15 evaluation runs. These training examples may also appear in later optimisation batches.

Each run lasts 600 updates. Mismatch replaces 30% of responses; truncation retains the first 25% of tokens in 32% of responses; template mixing changes the role tags in 28% of examples. Five fine-tuning seeds per family are 9511–9515, 9521–9525, and 9531–9535, respectively. Measurements are recorded every four updates in the scheduled monitoring window and every 20 updates outside it. The separation score D uses the recorded fault flags and all observations outside the fault window, including recovery.

We selected L1 from L1, L16 and L31 on the 83-run development set by maximising the lowest median RD–MI separation across fault families. The MI representation map averages contiguous hidden coordinates into 32 groups per token and flattens the token-by-group array. This map and all 2,016 unordered example pairs are retained on the new seeds. RD–Pearson uses a flattened masked-token map. The two variants therefore differ in both their representation map and estimator. Activation SD at L16 was selected on the same development set as the strongest shared representation-summary comparator and then fixed.

Uncertainty for pooled paired medians resamples five complete runs within each of the three families, for 50,000 bootstrap draws. These intervals describe fine-tuning variation conditional on the common pretrained base. The pooled paired median RD–MI advantage over the held-out response NLL gap is 0.049 $[-0.022, 0.116]$. This labelled reference uses the first 32 examples of the separate test split. For each prompt, it subtracts correct-response NLL from the NLL of the next example's response under a fixed cyclic assignment. Only final-assistant-response tokens contribute; sequences are truncated at 1,024 tokens. The gap averages the paired finite per-example losses. Test examples and donor assignments remain fixed across checkpoints. Its median separation is 0.534, 0.443 and 0.691 for mismatch, truncation and template mixing, respectively (RD–MI: 0.550, 0.717 and 0.612).

A.4 Grokking settings and event measurements

Models and observations. Both studies use addition modulo 113, represented as three-token inputs $(a, b, =)$, with target $(a + b) \bmod 113$. Each Transformer has one layer, four attention heads, residual width 128, a width-512 ReLU MLP and no layer normalisation. We measure the final-token residual on 128 fixed training examples. Raw Pearson and InfoNet scores are averaged over fixed sets of distinct unordered example pairs: 4,096 for the public trajectories and 2,048 for the two

development runs. The probe and pair identities stay fixed within each run. No marginal-shuffle subtraction is applied.

The public trajectories come from Smith (2022), revision `3799fe7b4c6abd4d81df7e55972fbefc6a9f0c3f`. These models use 30% training data, full-batch updates and learning rate 10^{-3}. The four weight-decay-1 seeds are 140483915967, 166300152443, 169752037408 and 37989087660; the five weight-decay-0 seeds are 153030051121, 240366304413, 56436254028, 122035194185 and 212270911129. The groups therefore differ in initialisation and training split as well as weight decay. Every released checkpoint is evaluated, giving 500 observations per model at 100-update intervals through update 49,900. Test accuracy is evaluated on the held-out examples after excluding the separate 128-example held-out input probe.

The bottom row in Figure 5 shows the four weight-decay-1 grokking models over updates 200–15,000, beginning when all models have fitted the training set. The legend values 3.6k, 6.1k, 6.7k and 11.8k identify seeds 169752037408, 140483915967, 37989087660 and 166300152443 by their first sustained 95% test-accuracy crossing. Panel (e) displays the range $[0.8, 4.0] \times 10^{-7}$; earlier losses above this range are outside the displayed axes. The nine-model event and drift analysis uses the full released training history. One regularised model has a later transient accuracy dip to 75.8% at update 41,100, recovering to 100% at the next checkpoint.

The same-hyperparameter example uses development seeds 28107 and 28108, 20% training data (2,553 examples), full-batch AdamW with learning rate 10^{-3}, betas $(0.9, 0.98)$, weight decay 1 and ten-update linear warm-up. Both trajectories extend to 50k updates and are recorded every 200 updates. Both reach perfect training accuracy at update 200; the figure shows all 250 observations from that point onwards in its top row.

Panel (c) reports both the mean pairwise Pearson coefficient and the mean of its absolute value over the same 2,048 pairs; panel (g) reports the signed mean only. All MI curves use raw InfoNet scores. Each row shows its own time range and score scales. The seed determines both initialisation and the train/test split. Loss and test accuracy are measured after each update. Gradient norm is the global ℓ_2 norm of the full-training mean-loss gradient used in that update, before the optimiser step and excluding decoupled weight decay.

Event and drift comparison. Training fit is the first checkpoint with at least 99% training accuracy. Delayed generalisation requires at least 95% test accuracy for three consecutive checkpoints, first reached at least ten times later than training fit. All four regularised public models satisfy this definition, with first sustained crossings at updates 6,100, 11,800, 3,600 and 6,700. For the same model order, the first 10–90% accuracy intervals are $[1300, 6000]$, $[8400, 11700]$, $[700, 3500]$ and $[3700, 6500]$. Event change is the absolute difference between a metric's interval endpoints. Drift is the median absolute change over all equal-duration windows starting at a released checkpoint after training fit and lying wholly outside the event interval. For each weight-decay-0 model, we average absolute changes over the same four event windows. Aggregation treats each model as one statistical unit. Effective rank centres each coordinate across probe examples before applying the shared spectral definition above; CKA compares with the initial representation.

These are retrospective development comparisons. The nine-model analysis uses the complete selected public cohort; the two-run example was selected from the two completed trajectories in an eight-seed development programme.

Alternative nonlinear readout. We also average a pretrained neural estimate of HGR maximal correlation (Rényi, 1959) over the same representations and pairs. The corratlas1D estimator uses checkpoint `g0_L4_indvlow_absoff/step50000.pt` with its stored soft-rank preprocessing. Table 4 shows the corresponding raw-score changes. In the public cohort, all four event changes also exceed their own median fitted-training drift, whose group mean is 0.0049.

A.5 Pair-budget precision and computational cost

In a separate six-model ViT measurement study, we use recorded raw InfoNet scores for every unordered pair of each fixed 128-example probe: $M = 8,128$ pairs, five populations, two paired branches, and all 35 observations before, during and after the intervention. Scores are held fixed while pair indices are subsampled. For each model and population, 64 deterministic random permu-

Table 4: **HGR readout on the same grokking comparisons.** The public row gives mean absolute event-window changes across four grokking models and five weight-decay-0 models. The two-run row gives signed changes between updates 35.8k and 39.2k for the model that generalises and the model that does not generalise by 50k.

Comparison	Grokking	Comparison models
Public trajectories (absolute change)	0.0763	0.0045
Same-hyperparameter example (signed change)	+0.0721	+0.0081

tations define nested budgets $K \in \{16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8128\}$. A sampled set is reused across all updates and both arms, preserving pairing in temporal and intervention contrasts. Every full-cache mean is checked against the archived all-pair summary before sampling.

Figure 6 compares finite-budget precision and the resulting paired trajectories.

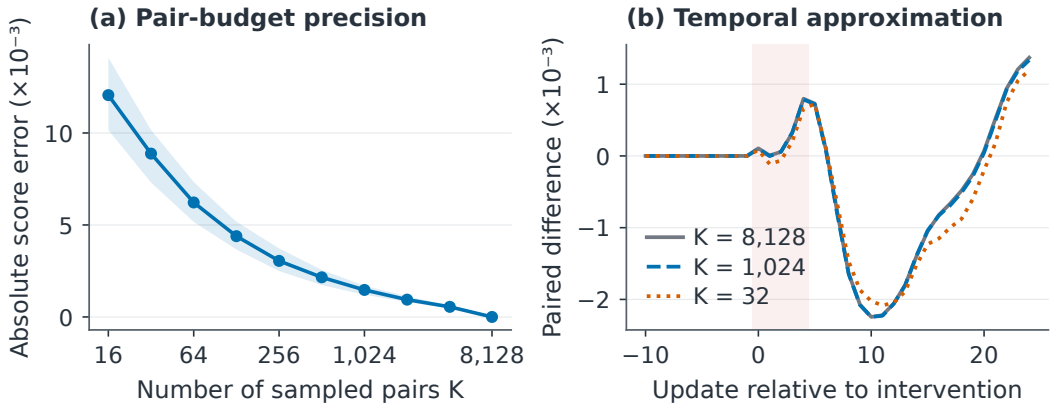

Figure 6: **Precision of a finite-pair monitoring approximation.** Left: absolute error against the complete cached InfoNet-score mean, averaged across models, populations, arms, and updates. The line averages 64 pair draws; the band gives the 10th–90th percentile across pair draws. Right: mean intervention-minus-control trajectories across all six models and five populations, using the first preassigned pair draw at each budget. The reference averages all recorded pairwise InfoNet scores. Red shading marks the intervention observations.

Mean absolute score error decreases from 0.00888 at $K = 32$ to 0.00147 at $K = 1024$, and is zero at the complete budget by construction. This estimates the error introduced by pair subsampling, conditional on the cached InfoNet outputs and their estimator randomness. Population MI error and estimator-training convergence are separate quantities.

Table 5: **Monitoring cost at two model operating points.** H200 host-time medians over five timing blocks; NanoGPT uses 64 sampled pairs and Pythia uses 256. The complete-operation row includes training and is timed directly.

Measurement (ms)	NanoGPT	Pythia
Training forward/backward	81.33	4035.31
Training with detached capture	92.53	7873.28
InfoNet raw score	527.18	5091.29
One marginal permutation	546.39	5212.09
Complete monitored operation	1199.01	18309.28

H200 timings include device synchronisation and report medians of five timing blocks. NanoGPT uses a width-768, 12-layer model, batch 128, sequence length 256, bfloat16, a 64-example probe, three layers, and 64 sampled pairs. Pythia uses 2.8B parameters, float32, a 64-example probe, length 512, three layers, and 256 sampled pairs. The Pythia training timer measures one physical micro-batch, not an accumulated optimiser step. Raw InfoNet, a fixed marginal permutation, and the complete monitored operation are timed separately with synchronisation. The complete-operation mea-

surement includes training and capture. Table 5 reports these sampled-pair configurations. The main Pythia experiment evaluates all 2,016 pairs and was not timed in this study. At the measured 256-pair budget, performing one complete monitored operation and nineteen train-only micro-batches takes about 17.7% longer than twenty train-only micro-batches. This overhead applies to the measured micro-batch configuration.

B Supporting Analysis

B.1 Exact nonlinear dependence example

Let a feature index Q be uniform over nine coordinates and define

$$x = (-1, -1, -1, 0, 0, 0, 1, 1, 1).$$

Pair x either with $y_{\mathrm{dep}} = (1, 1, 1, 0, 0, 0, 1, 1, 1) = x^2$ or with $y_{\mathrm{ind}} = (0, 1, 1, 0, 1, 1, 0, 1, 1)$. The two mini-batches $\{x, y_{\mathrm{dep}}\}$ and $\{x, y_{\mathrm{ind}}\}$ have identical coordinate-value marginals, raw Gram matrix $\mathrm{diag}(6, 6)$, centred Gram matrix $\mathrm{diag}(6, 2)$, singular values, ranks, zero Pearson and cosine pair score, and squared mean normalised direction $\|\mu\|_2^2 = 1/2$. The joint frequency table for $(x_Q, y_{\mathrm{ind},Q})$ factorises. The dependent pairing satisfies $y_{\mathrm{dep},Q} = x_Q^2$. Hence

$$I(x_Q; y_{\mathrm{ind},Q}) = 0, \qquad I(x_Q; y_{\mathrm{dep},Q}) = H(x_Q^2) = 0.636514 \text{ nats}. \tag{9}$$

Here H denotes discrete entropy with natural logarithms. Nonlinear dependence therefore distinguishes these two arrangements, although Pearson RD and all the listed second-order summaries are identical.

The construction uses discrete coordinates with ties. For generic finite vectors with distinct coordinates, treating the coordinate index as the sole randomness would instead give empirical discrete MI equal to $\log D$ for both dependent and permuted pairings. RD–MI applies a fixed pretrained estimator to the coordinate-pair sample; it is not this empirical discrete MI. A population-MI interpretation would additionally require a sampling model for coordinates. The example separates the dependence functionals and does not claim that InfoNet recovers these exact nine-coordinate values.

B.2 Information and update-response proofs

Throughout this analysis, RD-P denotes the complete-pair, signed Pearson mean on a fixed probe. All centred representations appearing in a normalised direction are nonzero.

Proof of Proposition 1. Consider $f(x) = W_2 W_1 x$, target sx with $s > 0$, and $L = \frac{1}{2}\mathbb{E}\|W_2 W_1 x - sx\|^2$, where $x \in \mathbb{R}^2$ and $\mathbb{E}[xx^\top] = I_2$. For example, a finite training distribution assigns equal probability to $\sqrt{2}e_1$ and $\sqrt{2}e_2$, where e_1, e_2 are the standard basis vectors. Choose any hidden width $D \geq 3$ and a matrix $B \in \mathbb{R}^{D\times 2}$ with $B^\top B = I_2$ and $\mathbf{1}^\top B = 0$; two orthonormal vectors in the $(D-1)$-dimensional subspace $\mathbf{1}^\perp$ suffice. For a fixed orthogonal $U \in \mathbb{R}^{2\times 2}$, initialise

$$W_1 = B\,\mathrm{diag}(\sqrt{q_1}, \sqrt{q_2})U^\top, \qquad W_2 = U\,\mathrm{diag}(\sqrt{q_1}, \sqrt{q_2})B^\top, \tag{10}$$

with $0 < q_2(0) < q_1(0) < s$. Both factors follow ordinary Euclidean gradient flow. Writing $E = W_2 W_1 - sI_2$, the gradients are $\nabla_{W_1} L = W_2^\top E$ and $\nabla_{W_2} L = E W_1^\top$. Substitution shows that this balanced, decoupled family is invariant, and its mode dynamics are the classical deep-linear solution (Saxe et al., 2014):

$$\dot{q}_k = 2q_k(s - q_k), \qquad q_k(t) = \frac{s}{1 + (s/q_k(0) - 1)e^{-2st}}. \tag{11}$$

Indeed, writing $a_k = \sqrt{q_k}$ in Eq. 10 gives $\dot{a}_k = a_k(s - a_k^2)$ for each factor. The two trajectories with $U = I_2$ and with U a rotation by $\pi/4$ therefore have the same $q_k(t)$ and

$$L(t) = \tfrac{1}{2}\sum_{k=1}^{2}(s - q_k(t))^2, \tag{12}$$

$$\|\nabla L(t)\|^2 = \|\nabla_{W_1} L(t)\|_F^2 + \|\nabla_{W_2} L(t)\|_F^2 = 2\sum_{k=1}^{2} q_k(t)(s - q_k(t))^2. \tag{13}$$

On the same fixed probe $\{e_1, e_2\}$, the hidden Gram matrix is $U\,\mathrm{diag}(q_1, q_2)U^\top$. The condition $\mathbf{1}^\top B = 0$ ensures that coordinate centring leaves each representation unchanged. Consequently,

$$\text{RD-P}_{I_2}(t) = 0, \qquad \text{RD-P}_{\pi/4}(t) = \frac{q_1(t) - q_2(t)}{q_1(t) + q_2(t)} > 0 \quad (t < \infty). \tag{14}$$

The mode solution preserves the strict ordering, proving the proposition. Under Euclidean gradient flow, $\dot{L} = -\|\nabla L\|^2$ links the two scalar histories.

The separation also holds on larger probes. Take n_1 distinct positive multiples of e_1 and n_2 distinct positive multiples of e_2, with $N = n_1 + n_2$. Within-group Pearson scores equal one in both trajectories, whereas their between-group scores differ. Thus

$$\text{RD-P}_{\pi/4} - \text{RD-P}_{I_2} = \frac{2n_1 n_2}{N(N-1)} \frac{q_1 - q_2}{q_1 + q_2}. \tag{15}$$

At any fixed finite time, continuity preserves a nonzero difference under sufficiently small non-collinear perturbations of these probes.

A finite-displacement identity and bound. Let $s_i = Ch_i$, $r_i = \|s_i\| > 0$, $z_i = s_i/r_i$ and $\mu = N^{-1}\sum_i z_i$. For centred increments v_i with $\kappa_i = \min_{u\in[0,1]} \|s_i + uv_i\| > 0$, differentiation gives

$$D\text{RD-P}[s](v) = \sum_i \gamma_i^\top v_i, \qquad \gamma_i = \frac{2}{(N-1)r_i}(I - z_i z_i^\top)\mu. \tag{16}$$

Here $s = (s_1, \ldots, s_N)$, $v = (v_1, \ldots, v_N)$, and $D\text{RD-P}[s](v)$ is the directional derivative at s along v. To bound the finite remainder, set $r_i(u) = \|s_i + uv_i\|$, $z_i(u) = (s_i + uv_i)/r_i(u)$, $\alpha_i = z_i(u)^\top v_i$, and $w_i = (I - z_i(u)z_i(u)^\top)v_i$. Direct differentiation yields

$$z_i'(u) = \frac{w_i}{r_i(u)}, \qquad z_i''(u) = -\frac{2\alpha_i w_i + z_i(u)\|w_i\|^2}{r_i(u)^2}. \tag{17}$$

Since $w_i \perp z_i(u)$ and $\alpha_i^2 + \|w_i\|^2 = \|v_i\|^2$, we have $\|z_i'\| \le b_i$ and $\|z_i''\| \le 2b_i^2$, where $b_i = \|v_i\|/\kappa_i$. Each pair consequently satisfies

$$\left|\frac{d^2}{du^2}(z_i(u)^\top z_j(u))\right| \le 2b_i^2 + 2b_i b_j + 2b_j^2 \le 3(b_i^2 + b_j^2). \tag{18}$$

Each index occurs in $N-1$ pairs, giving a mean second-derivative bound $6N^{-1}\sum_i b_i^2$. Integrating the Taylor remainder contributes a factor $\int_0^1 (1-u)\,du = 1/2$, hence

$$\left|\text{RD-P}(s+v) - \text{RD-P}(s) - \sum_i \gamma_i^\top v_i\right| \le \frac{3}{N}\sum_i \frac{\|v_i\|^2}{\kappa_i^2}. \tag{19}$$

A sufficient norm condition is $\|v_i\| < r_i$, giving $\kappa_i \ge r_i - \|v_i\|$. In particular, if $\|v_i\| \le r_i/2$, the bound is at most $12N^{-1}\sum_i \|v_i\|^2/r_i^2$.

Subtracting the endpoint mean-direction identities gives Eq. 4. If $\mu = 0$, the initial score is its minimum $-1/(N-1)$, and the change equals $N\|\mu^+\|^2/(N-1)$, where μ^+ is the endpoint mean direction. For fixed directions ξ_i with $v_i = \eta\xi_i$ and a nonzero mean tangent response $\tau = N^{-1}\sum_i (I - z_i z_i^\top)\xi_i/r_i$, this becomes $N\eta^2\|\tau\|^2/(N-1) + O(\eta^3)$, a second-order response.

Proof of Proposition 2. For an update batch $\{(x_b, y_b)\}_{b=1}^{B_0}$, write $D_b = D_{x_b}$ and $e_b = e(x_b, y_b)$. Simultaneous SGD gives

$$W^+ = W - \frac{\eta}{B_0}\sum_b D_b V^\top e_b x_b^\top, \qquad V^+ = V - \frac{\eta}{B_0}\sum_b e_b h_W(x_b)^\top. \tag{20}$$

All quantities on the right are evaluated before the update. We use the usual ReLU convention of a zero derivative at zero preactivation. For $\ell(f, y) = \frac{1}{2}\|f - y\|^2$, e_b is the output residual; for softmax cross-entropy it is the predicted probability vector minus the target vector.

When the activation mask of probe i stays fixed, its finite hidden displacement is exactly

$$h_{W^+}(x_i) - h_W(x_i) = -\frac{\eta}{B_0} \sum_b (x_i^\top x_b) D_i D_b V^\top e_b. \tag{21}$$

The hidden update uses the initial readout. Only probe gates must remain fixed; training gates are evaluated before the update. Away from ReLU kinks, the matrix multiplying e_b is explicitly

$$J_\theta h_W(x_i) J_\theta f(x_b)^\top = (x_i^\top x_b) D_i D_b V^\top, \tag{22}$$

for $\theta = (W, V)$. This hidden-to-output response differs from the output tangent kernel, which uses $J_\theta f$ on both sides (Jacot et al., 2018).

If both ordinary and altered-data updates preserve each initial probe gate, subtracting their versions of Eq. 21 gives the exact centred endpoint difference $v_i = -\eta\varepsilon C D_i \Delta A x_i$. Applying Eq. 19 at the ordinary endpoint, with γ_i and κ_i evaluated there, proves Eq. 7. The condition on gates can be checked coordinatewise: writing $p_i = W x_i$ and $d_i = (W^+ - W)x_i$, it is $\mathbf{1}\{p_{ik} > 0\} = \mathbf{1}\{p_{ik} + d_{ik} > 0\}$. For nonzero p_{ik}, $|d_{ik}| < |p_{ik}|$ is sufficient; zero-margin coordinates require the direct check.

Gate changes and smooth activations. For a ReLU coordinate whose gate may change, define

$$a_{ik} = \mathrm{ReLU}(p_{ik} + d_{ik}) - \mathrm{ReLU}(p_{ik}) - D_{i,kk} d_{ik}. \tag{23}$$

Scalar case analysis gives $|a_{ik}| \le |d_{ik}| \mathbf{1}\{|p_{ik}| \le |d_{ik}|\}$. Set $F_i = (\sum_k d_{ik}^2 \mathbf{1}\{|p_{ik}| \le |d_{ik}|\})^{1/2}$. The actual centred displacement is $v_i + C a_i$, where $v_i = C D_i d_i$ and $\|C a_i\| \le F_i$. If $\rho_i = \|v_i\| + F_i < r_i$, then

$$\left| \Delta\mathrm{RD\text{-}P} - \sum_i \gamma_i^\top v_i \right| \le \sum_i \|\gamma_i\| F_i + \frac{3}{N} \sum_i \frac{\rho_i^2}{(r_i - \rho_i)^2}. \tag{24}$$

For the altered-versus-ordinary contrast, compute both gate errors from the common initial checkpoint, set $v_i = C D_i (d_i^a - d_i^c) = -\eta\varepsilon C D_i \Delta A x_i$, and use their sum $F_i^a + F_i^c$ as F_i. Evaluate r_i and γ_i at the actual ordinary endpoint. Near-zero preactivations can produce first-order gate errors, which Eq. 24 retains.

For $h_W(x) = \sigma(Wx)$ with a smooth activation σ, replace every training and probe gate in A_P, A_Q and Eq. 20 by $D_x = \mathrm{diag}(\sigma'(Wx))$; in particular, $D_i = \mathrm{diag}(\sigma'(p_i))$. If $|\sigma''| \le \beta$ along the coordinate update intervals, Taylor's formula gives the alternative bound

$$\|\sigma(p_i + d_i) - \sigma(p_i) - D_i d_i\|_2 \le \frac{\beta}{2} \left(\sum_k d_{ik}^4 \right)^{1/2}. \tag{25}$$

Use this bound as F_i in Eq. 24, or the sum of both activation remainders for the two-endpoint contrast.

B.3 Local interpretation in a Transformer

For a differentiable representation $h_i(\theta)$, let $J_i = \partial h_i / \partial \theta$ and $J_f(x) = \partial f(x)/\partial\theta$ denote its Jacobian and the output Jacobian, respectively. Then $\delta h_i = J_i \delta\theta$; under an infinitesimal SGD update with empirical batch distribution P, $\delta h_i = -\eta \mathbb{E}_{(x,y)\sim P}[J_i J_f(x)^\top e(x,y)]$, where $e(x,y) = \nabla_f \ell(f(x), y)$. Finite updates additionally have a network-linearisation remainder.

For standard single-head attention (Vaswani et al., 2017), let the rows of $X \in \mathbb{R}^{T\times d}$ be T token vectors of width d. The query and key projection matrices $W_Q, W_K \in \mathbb{R}^{d\times d_k}$ give $Q = XW_Q$ and $K = XW_K$; the value projection $W_U \in \mathbb{R}^{d\times d_v}$ gives $U = XW_U$. Define the masked attention logits $S = QK^\top/\sqrt{d_k} + \mathrm{mask}$ and row-wise attention weights $A = \mathrm{softmax}_{\mathrm{row}}(S)$. With output projection $W_O \in \mathbb{R}^{d_v\times d}$, the output is $O = AUW_O$. Holding the mask fixed, its differential is

$$\delta O = (\delta A) U W_O + A(\delta U) W_O + AU(\delta W_O), \tag{26}$$

$$\delta S = (\delta Q K^\top + Q \delta K^\top)/\sqrt{d_k}, \tag{27}$$

$$\delta Q = \delta X W_Q + X \delta W_Q, \qquad \delta K = \delta X W_K + X \delta W_K, \tag{28}$$

$$\delta U = \delta X W_U + X \delta W_U. \tag{29}$$

For corresponding rows of A and S, represented as column vectors a and s, softmax satisfies $\delta a = (\mathrm{diag}(a) - aa^\top)\delta s$, restricted to the visible positions of a fixed mask. The first term in δO changes attention routing; the other two change value content and its projection.

For example, a sequential pre-normalisation residual block has $Y = X + F(X)$ and $Z = Y + G(Y)$, where F includes attention and its input normalisation, and G includes the feed-forward network and its input normalisation. Then

$$\delta Y = (I + J_F)\delta X + \delta_{\mathrm{par}} F, \qquad \delta Z = (I + J_G)\delta Y + \delta_{\mathrm{par}} G. \tag{30}$$

Here J_F and J_G are input Jacobians acting on vectorised token arrays, and δ_{par} denotes the change due to each block's parameters with its input held fixed. The parameter terms include the corresponding normalisation parameters. Applying the fixed representation map to these layer changes supplies δh_i in Eq. 16.

B.4 Convergence of the sampled MI mean

Proof of Proposition 3: finite pair budget. Index the unordered pairs by e. Condition on their M finite scores q_e, including the scorer, preprocessing and realised scorer randomness. Write $U = M^{-1}\sum_e q_e$ and $S_g^2 = (M-1)^{-1}\sum_e (q_e - U)^2$ for $M > 1$. If I_e indicates inclusion in a uniformly selected K-pair subset, $\mathbb{E}I_e = K/M$ and $\mathbb{E}I_e I_f = K(K-1)/[M(M-1)]$ for $e \neq f$. Expanding the variance of $K^{-1}\sum_e I_e q_e$ gives

$$\mathbb{E}[\widehat{U}_K \mid q] = U, \qquad \mathbb{E}[(\widehat{U}_K - U)^2 \mid q] = S_g^2 \left(\frac{1}{K} - \frac{1}{M}\right). \tag{31}$$

Taking square roots proves the RMSE for any fixed score array, without independence or symmetry assumptions. When $M = K = 1$ the mean is exact. For $S_g > 0$, achieving RMSE at most $a > 0$ requires $K \geq \lceil S_g^2/(a^2 + S_g^2/M)\rceil$, subject to $1 \leq K \leq M$.

Independent-probe rate. Assume the trained model, representation map and scoring function are fixed independently of $X_1, \ldots, X_N \overset{\mathrm{iid}}{\sim} P$. Let $q(x, y) = q(y, x)$ be a fixed score with $\theta = \mathbb{E}q(X, X')$ and $\sigma_q^2 = \mathrm{Var}\, q(X, X') < \infty$, where X, X' are independent draws from P. Conditional on the probes, select the K-pair subset uniformly without replacement. Define

$$a(x) = \mathbb{E}[q(x, X')] - \theta, \tag{32}$$

$$b(x, y) = q(x, y) - \theta - a(x) - a(y), \qquad \zeta_1 = \mathbb{E}a(X)^2, \quad \zeta_2 = \mathbb{E}b(X, X')^2. \tag{33}$$

For $U_N = M^{-1}\sum_{i<j} q(X_i, X_j)$,

$$U_N - \theta = \frac{2}{N}\sum_i a(X_i) + \frac{1}{M}\sum_{i<j} b(X_i, X_j).$$

Since $\mathbb{E}a(X) = 0$ and $\mathbb{E}[b(X, X') \mid X] = 0$, the terms are orthogonal. This degree-two U-statistic decomposition (Clémençon et al., 2016) gives

$$V_N = \mathrm{Var}\, U_N = \frac{4\zeta_1}{N} + \frac{2\zeta_2}{N(N-1)}, \qquad \sigma_q^2 = 2\zeta_1 + \zeta_2. \tag{34}$$

Total variance and $\mathbb{E}S_g^2 = M(\sigma_q^2 - V_N)/(M-1)$ imply, for $M > 1$,

$$\mathbb{E}(\widehat{U}_K - \theta)^2 = V_N + \frac{M-K}{K(M-1)}(\sigma_q^2 - V_N) \leq \sigma_q^2\left(\frac{2}{N} + \frac{1}{K}\right). \tag{35}$$

The inequality uses $V_N \leq 2\sigma_q^2/N$ and $(M-K)/[K(M-1)] \leq 1/K$. For $N = 2$ the variance is directly σ_q^2. Chebyshev's inequality gives $\widehat{U}_K - \theta = O_P(N^{-1/2} + K^{-1/2})$ as $N, K \to \infty$ with the score fixed. When $\zeta_1 > 0$, sampling K proportional to N retains the complete statistic's root-N order, while increasing K to M removes only the pair approximation.

For an ideal-MI target θ_*, the decomposition $\widehat{U}_K - \theta_* = (\widehat{U}_K - \theta) + (\theta - \theta_*)$ separates sampling error, controlled by N and K under the stated assumptions, from the fixed scorer's discrepancy.

Temporal differences. If the same pair indices are reused at checkpoints s and t, apply Eq. 31 to $d_e = q_{e,t} - q_{e,s}$. With finite-population variance S_d^2, the difference estimate is unbiased and has variance $S_d^2(1/K - 1/M)$. Appendix A.5 evaluates this approximation on complete cached trajectories.

C Layer Ablation

C.1 NanoGPT and ViT

The main experiments use first-block RD. We compare this choice with multi-layer NanoGPT features and the original block-6 ViT readout, using the same recorded trajectories and evaluation procedures.

For NanoGPT, the first-block predictor adds eight features: Pearson, cosine, covariance and dot-product scores, each as a current value and an early-reference difference. The three-layer predictor adds the same features from blocks 1, 7 and 12 (code indices 0, 6 and 11), giving 24 RD features. Both augment the same 142 combined-monitor features. Each predictor independently selects ridge regularisation within the same nested seed folds. First-block RD reduces MAE by 0.290 [0.009, 0.683] nats, retaining most of the 0.361 [0.051, 0.819] reduction from the three-layer predictor. Both improve on the combined monitors in four of five seeds (Table 6). The first-block comparison is an additional layer ablation on these five runs; the original development selection used the three-layer predictor.

Table 6: **NanoGPT RD layer ablation.** Validation deterioration predicted 250 updates ahead with identical nested seed folds. MAE is mean $\pm$ SD over five held-out seeds; R^2 pools all held-out predictions. The baseline features stay fixed while the RD layers vary.

Inputs	Added RD features	MAE $\downarrow$	R^2 $\uparrow$
Combined monitors	0	2.130 ± 1.395	0.515
+ first-block RD	8	1.840 ± 0.960	0.654
+ three-layer RD	24	1.769 ± 0.888	0.685

For ViT, we compare the first-block readout with the original block-6 analysis on the same eight models at 50k. The difference in response between 100% and 25% replacement is 1.056 [0.668, 1.433] control-SD units for the first block and 0.882 [0.558, 1.253] for block 6. Their respective paired differences from gradient norm are 0.643 [0.216, 1.017] and 0.469 [0.108, 0.859]. Both layers have a larger response increase than gradient norm under the same evaluation procedure.

C.2 Pythia

We compare RD–MI at Pythia layers L1, L16 and L31 on the same 15 fine-tuning trajectories, with five new seeds per fault family. All layers use the same fixed clean probe, 32-group representation map and InfoNet estimator. L1 was selected on the separate 83-trajectory development set by maximising the lowest family-median fault-window separation D.

Table 7 shows that L1 has the highest median separation in all three families. Its score is within 0.05 of the best layer in 11 of 15 runs, supporting the choice of layer across new fine-tuning seeds.

Table 7: **Pythia layer comparison on new fine-tuning seeds.** Median RD–MI separation D over five seeds per family. L1 was fixed using the separate development set. All layers use the same representation mapping, estimator and monitoring observations.

Fault	L1 (selected)	L16	L31
Mismatch	**.550**	.397	.228
Truncation	**.717**	.528	.212
Template mixing	**.612**	.164	.438