MODE DYNAMICS: A FRAMEWORK FOR NEURAL NETWORK LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Mode Dynamics, a framework that connects coordinated parameter motion during neural network learning to changes in model outputs. Starting from a recorded parameter trajectory, we decompose its updates into collective modes and map each mode to its state-dependent output effect. The framework organizes this link through three mechanisms. *Decomposition* identifies collective modes in parameter motion and their time-varying amplitudes. *Reconstruction* tests whether the effects induced by those modes recover local and accumulated output change. *Concentration* quantifies how behavioral contribution is distributed across modes. Controlled scalar regression and next-token experiments directly test these mechanisms. They show that small sets of leading modes reconstruct behavior, induced effects depend on the training path, and modal complexity varies with task and data. We then evaluate the framework on held-out image and language-model outputs using CIFAR-100 autoencoding and a public Pythia-70M pretraining trajectory. Across these settings, the results connect compact modal subspaces of recorded parameter motion to evolving behavior.

1 INTRODUCTION

Understanding how neural networks learn remains difficult because optimization unfolds in a high-dimensional parameter space, while the object of interest is the learned function. A central question is therefore not only whether gradient descent lowers loss, but which collective degrees of freedom drive changes in predictions over time.

Existing work provides two complementary perspectives. Kernel and linearization analyses describe regimes in which function-space dynamics remain close to initialization (Jacot et al., 2018; Lee et al., 2019; Chizat et al., 2019), whereas studies of feature learning emphasize that the parameter-to-function map can evolve substantially (Woodworth et al., 2020; Yang & Hu, 2021; Mei et al., 2019; Seleznova & Kutyniok, 2022; Loo et al., 2022). Separately, gradients, optimization trajectories, and adaptation updates often occupy low-dimensional subspaces (Gur-Ari et al., 2019; Li et al., 2018; 2021; Aghajanyan et al., 2021; Hu et al., 2021). These observations do not by themselves identify which trajectory directions carry behavioral change or whether their output effects remain fixed throughout training.

Mode Dynamics addresses this gap by treating parameter updates as discrete velocities. Collective directions extracted from the recorded update trajectory serve as modes, and their checkpoint-dependent coefficients are amplitudes. Each mode maps to an induced output effect that can change as the network moves through parameter space. Three linked mechanisms organize the description: *decomposition* into a shared mode basis with checkpoint-dependent amplitudes, *reconstruction* of one-step and accumulated output change from the realized modal contributions, and *concentration* of behavioral energy across modes. Reconstruction also tests *path dependence* by comparing induced effects evaluated along the recorded trajectory with an initialization-frozen baseline.

Contributions. Our main contributions are as follows.

- We introduce Mode Dynamics, a trajectory-level framework that maps collective parameter motion to state-dependent output effects and organizes this link through three mechanisms: decomposition, reconstruction, and concentration.

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 show that mode-induced output effects are path dependent rather than fixed at initialization, and that trajectory modes outperform dimension-matched random subspaces.
- We evaluate concentration separately from reconstruction fidelity, since energy concentration and output fidelity need not coincide.
- We isolate these mechanisms in controlled regression and next-token tasks, then test transfer to CIFAR-100 autoencoding and a public Pythia-70M pretraining trajectory (Biderman et al., 2023), ranking modes on the Pile validation split and scoring without reranking on the untouched Pile test split (Gao et al., 2020).

2 Related Work

Dynamical-systems perspectives. In physics, normal-mode analysis and dynamic mode decomposition motivate representing high-dimensional evolution through collective coordinates and time-dependent amplitudes (McGaughey & Larkin, 2013; Schmid, 2010). Koopman methods seek coordinates with linear evolution (Lusch et al., 2018) and have been applied to neural-network training (Dogra & Redman, 2020; Manojlović et al., 2021), while Correlation Mode Decomposition groups neural parameters with shared temporal profiles (Brokman et al., 2024). Mode Dynamics instead decomposes realized parameter updates without fitting an evolution operator or clustering parameters around representative profiles. It maps each direction to a state-dependent output effect and tests whether its realized contributions reconstruct behavior.

Learning dynamics and linearization. Infinite-width and lazy-training results describe regimes with approximately fixed linearized dynamics (Jacot et al., 2018; Lee et al., 2019; Chizat et al., 2019). Feature-learning, kernel-evolution, and mean-field work instead emphasizes time-dependent representations (Woodworth et al., 2020; Yang & Hu, 2021; Mei et al., 2019; Seleznova & Kutyniok, 2022; Loo et al., 2022). Deep linear networks also admit exact mode-wise learning dynamics under structured assumptions (Saxe et al., 2014). Mode Dynamics analyzes realized finite nonlinear trajectories and directly compares evolving induced effects with a frozen description.

Low-dimensional optimization and adaptation. Gradients and training trajectories often occupy low-dimensional subspaces or manifolds (Gur-Ari et al., 2019; Li et al., 2018; 2021; Mao et al., 2024), and low-rank update families are central to efficient adaptation (Aghajanyan et al., 2021; Hu et al., 2021; Sonthalia et al., 2025). These works motivate compact parameter-space descriptions. Mode Dynamics asks the complementary question of how trajectory directions contribute to behavior over time.

Behavioral attribution and mechanistic interpretability. Sparse-feature and circuit methods primarily analyze activations and computation within trained models (Elhage et al., 2022; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2024; Lieberum et al., 2024; Anthropic, 2025; Gao et al., 2025). Mode Dynamics complements this work by studying how behavior develops over training, linking full-parameter trajectory directions to time-varying output effects and testing whether their contributions reconstruct output change. Path-integrated attribution is also used to credit input features along an input interpolation (Sundararajan et al., 2017) and to credit training examples by accumulating per-checkpoint gradient terms (Pruthi et al., 2020). We instead integrate along the recorded parameter trajectory and attribute output change to collective update directions. Across these literatures, Mode Dynamics fills the specific gap between trajectory-level update modes and the evolving behavior they produce.

3 Mode Dynamics Framework

The framework starts from the ordered parameter states $\theta_0, \theta_1, \ldots, \theta_T$ and a chosen model output $f(x, \theta)$. It then follows one dependency chain. The parameter states define updates, whose decomposition produces modes and amplitudes. The parameter-to-output Jacobian maps each mode to an induced effect. Amplitude weighting gives realized contributions that reconstruct output change, and their distribution defines behavioral concentration.

3.1 PARAMETER-UPDATE DECOMPOSITION

Let $\theta_t \in \mathbb{R}^P$ be the parameters at checkpoint t and

$$v_t = \theta_{t+1} - \theta_t \tag{1}$$

the observed checkpoint-to-checkpoint parameter update. With checkpoint index treated as discrete time, v_t is the corresponding discrete analogue of velocity. Stacking the T updates gives $V = [v_0, \ldots, v_{T-1}] \in \mathbb{R}^{P \times T}$. Let $r = \operatorname{rank}(V)$. The thin singular value decomposition (SVD) $V = U\Sigma W^\top$ produces orthonormal trajectory modes $u_1, \ldots, u_r$ and their checkpoint-dependent amplitudes

$$a_{k,t} = u_k^\top v_t = (\Sigma W^\top)_{k,t}. \tag{2}$$

Consequently, every recorded update has the exact decomposition

$$v_t = \sum_{k=1}^{r} a_{k,t} u_k. \tag{3}$$

The trajectory directions u_k are fixed for the analyzed trajectory. Only their amplitudes vary across checkpoints. The basis is retrospective: it is computed from the complete recorded trajectory rather than used to predict future updates online. It depends on the parameter coordinates and checkpoint schedule. Individual modes are non-unique up to sign and rotations within degenerate singular subspaces, as detailed in Appendix A.

3.2 MODE-INDUCED EFFECTS AND LOCAL RECONSTRUCTION

The key step is to associate each parameter-space mode with the output effect it induces. For model output $f(x, \theta)$, let $J_\theta f(x, \theta) = \partial f(x, \theta)/\partial\theta$ denote the Jacobian of the outputs with respect to the parameters. The induced effect of a unit movement along mode u_k at state θ_t is the Jacobian–vector product (JVP)

$$\phi_k(x, \theta_t) = J_\theta f(x, \theta_t) u_k. \tag{4}$$

Weighting this unit effect by the observed amplitude defines the realized contribution of mode k as

$$g_{k,t}(x) = a_{k,t} \phi_k(x, \theta_t). \tag{5}$$

Let $\Delta f_t(x) = f(x, \theta_{t+1}) - f(x, \theta_t)$ be the observed one-step output change. Its local modal reconstruction is

$$\Delta f_t(x) \simeq \sum_{k=1}^{r} g_{k,t}(x). \tag{6}$$

Equation (3) is an exact parameter-space identity, whereas Eq. (6) is a first-order approximation in output space.

3.3 ACCUMULATED RECONSTRUCTION AND PATH DEPENDENCE

The observed output evolution obeys the exact telescoping identity

$$f(x, \theta_t) = f_0(x) + \sum_{\tau<t} \Delta f_\tau(x), \tag{7}$$

where $f_0(x) = f(x, \theta_0)$. Replacing each local change by its modal reconstruction gives

$$\widehat{f}_t(x) = f_0(x) + \sum_{\tau<t} \sum_{k=1}^{r} g_{k,\tau}(x). \tag{8}$$

We use path dependence in a specific operational sense. Although each mode u_k is fixed, its induced effect $\phi_k(x, \theta_\tau)$ can change with the parameter state visited along the recorded trajectory. The frozen baseline preserves the same modes and amplitudes but evaluates every induced effect at one reference state. In all reported frozen baselines, $\theta_{\mathrm{ref}} = \theta_0$, so we call this the *initialization-frozen reconstruction*.

$$\widehat{f}_t^{\,\mathrm{fixed}}(x) = f_0(x) + \sum_{\tau<t} \sum_{k=1}^{r} a_{k,\tau} \phi_k(x, \theta_{\mathrm{ref}}). \tag{9}$$

The evolving and frozen reconstructions therefore differ only in whether ϕ_k is evaluated at θ_τ or at θ_{ref}. Their contrast tests whether amplitude accumulation with one reference-state Jacobian adequately describes behavior along the observed trajectory. It does not compare alternative training paths or imply that a fixed endpoint has multiple outputs.

3.4 BEHAVIORAL CONCENTRATION

Having defined the realized contributions $g_{k,t}$, we ask how their behavioral energy is distributed. We separately test whether a small ranked subset reproduces output change.

Checkpoint-specific energy. The behavioral energy and normalized share of mode k at checkpoint t are

$$E_{k,t} = \|g_{k,t}\|_2^2, \qquad \rho_{k,t} = \frac{E_{k,t}}{\sum_j E_{j,t}}, \tag{10}$$

where the empirical norm is evaluated over the stated inputs and output coordinates. These shares describe the distribution of individual realized-contribution energies at a particular training stage. Because the contributions need not be orthogonal in function space, the shares are not explained-variance fractions of their sum.

Trajectory-level ordering. Let $\mathcal{T}$ denote the experiment-specific set of checkpoints used for behavioral ranking. To obtain one ordering across these checkpoints, define

$$\begin{aligned} G_k(x) &= \sum_{t\in\mathcal{T}} g_{k,t}(x), \\ \bar{E}_k &= \|G_k\|_2^2, \qquad \bar{\rho}_k = \frac{\bar{E}_k}{\sum_j \bar{E}_j}. \end{aligned} \tag{11}$$

Here $\bar{E}_k$ is the energy of the mode's net accumulated contribution, not the time-integrated energy $\sum_{t\in\mathcal{T}} E_{k,t}$. We call the ordering induced by $\bar{\rho}_k$ *behavioral-energy ordering*. We also use *parameter-energy ordering*, which ranks modes by parameter-motion energy $\sigma_k^2 = \sum_t a_{k,t}^2$, equivalently the squared norm of the mode's amplitude sequence, and requires no analysis inputs. Each experiment states its ordering explicitly.

Concentration summaries. Let $\rho_{(1),t} \geq \rho_{(2),t} \geq \cdots$ denote the checkpoint-specific shares in descending order. The cumulative energy share of the leading K modes is $C_K(t) = \sum_{k=1}^{K} \rho_{(k),t}$. Our primary threshold-free summary of the full distribution is the participation ratio (PR)

$$\mathrm{PR}(t) = \frac{1}{\sum_k \rho_{k,t}^2}. \tag{12}$$

For an intuitive thresholded count, we additionally report

$$K_{\mathrm{eff}}(\tau, t) := \min\{K : C_K(t) \geq \tau\}, \tag{13}$$

using $\tau = 0.9$ unless stated otherwise. The same cumulative, PR, and effective-count summaries apply to the trajectory-level shares $\bar{\rho}_k$.

Top-K reconstruction fidelity. For a stated ordering, Top-K reconstruction retains the K highest-ranked modes and measures how well they reproduce output change using metrics such as R^2, reconstruction error, KL divergence, or prediction agreement. Rapid fidelity improvement with K supports behavioral compactness, but fidelity is distinct from energy concentration because the realized contributions $g_{k,t}$ can align or cancel in function space. PR and K_{eff} require the relevant full energy distribution, whereas Top-K fidelity can also be evaluated for a truncated set of leading modes.

3.5 COMPUTING MODE-INDUCED EFFECTS

Equation (4) can be evaluated without constructing the full Jacobian, using either automatic differentiation (AD) to compute JVPs or directional finite differences. Regression and controlled next-token prediction use finite differences, whereas CIFAR-100 and Pythia use forward-mode AD and quadrature along the straight *checkpoint chord* joining consecutive checkpoints. Appendix A gives estimator boundaries. The dominant costs are $O(TP)$ trajectory storage and $O(KT)$ direction–state evaluations, multiplied by Q when each checkpoint chord uses Q quadrature nodes. Appendix B gives the full accounting.

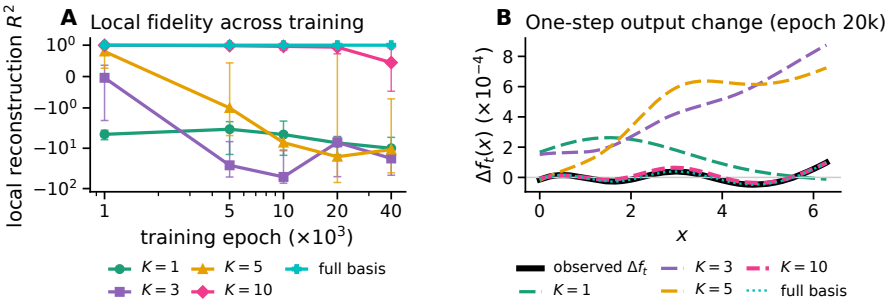

Figure 1: **Leading-mode reconstructions approximate local output changes. A:** Median local R^2 for $K \in \{1, 3, 5, 10\}$ and the full basis over 15 trajectories. Capped bars show the interquartile range and the vertical axis is symmetric-log. **B:** The observed and reconstructed one-step output change for a representative run at epoch 20k. Modes follow the parameter-energy ordering of the update SVD.

4 Experiments

We evaluate the framework in four settings: scalar regression, controlled next-token prediction, CIFAR-100 autoencoding, and a public Pythia-70M pretraining trajectory. All four test whether leading trajectory-derived modes reconstruct observed output change. The two controlled experiments provide detailed mechanism and control tests. CIFAR-100 and Pythia-70M provide compact transfer tests across image and language outputs. Appendix Table 2 summarizes the complementary tasks and metric coverage, and the appendix gives complete protocols.

For every reported reconstruction R^2, let s denote the reference model output or output change, stacked over the stated examples and output coordinates, and let $\widehat{s}$ denote its modal reconstruction. We report

$$R^2 = 1 - \frac{\|\widehat{s} - s\|_2^2}{\|s - \overline{s}\|_2^2}. \tag{14}$$

Here $\overline{s}$ is the mean of the entries of s. Values below zero indicate reconstruction error greater than that of the constant baseline $\overline{s}$. The reference is always the model output being reconstructed, not the task target. Each experiment states whether this reference is a one-step output change, an accumulated change from initialization, or a model output at a specified checkpoint. For Pythia, the same ratio is applied to vocabulary-centered logit change pooled over contexts, as specified in Appendix F.

4.1 Controlled Mechanistic Validation

4.1.1 Scalar Regression

Regression tests local fidelity and temporal accumulation in a setting where prediction curves and residuals can be inspected directly. We fit $y = \sin x$ on $[0, 2\pi]$ with a one-hidden-layer multilayer perceptron (MLP) using mean-squared error (MSE) and stochastic gradient descent (SGD). The main grid contains 15 trajectories, with five seeds at each width in $\{32, 64, 128\}$. Controls include randomly oriented, dimension-matched parameter subspaces, an initialization-frozen baseline, and an empirical finite-width neural tangent kernel (NTK). Appendix C gives optimization, checkpoint, robustness, and control details.

Local reconstruction. Figure 1 shows that a single mode is insufficient, whereas Top-10 approaches the full-basis reconstruction at early and intermediate checkpoints. For the reference width-64 architecture, Top-10 uses only $10/193 \approx 5.2\%$ of the available trajectory modes. Near convergence, one-step output changes have negligible variance, so tiny absolute errors can make local Top-K R^2 unstable. Full-basis local fidelity and accumulated Top-K reconstruction nevertheless remain high.

Path-dependent induced effects. Figure 2 shows that Top-10 reconstruction is nearly exact through 20k, then becomes less accurate and more variable at later checkpoints. The initialization-frozen baseline degrades progressively. Behavioral-energy and parameter-energy orderings retain

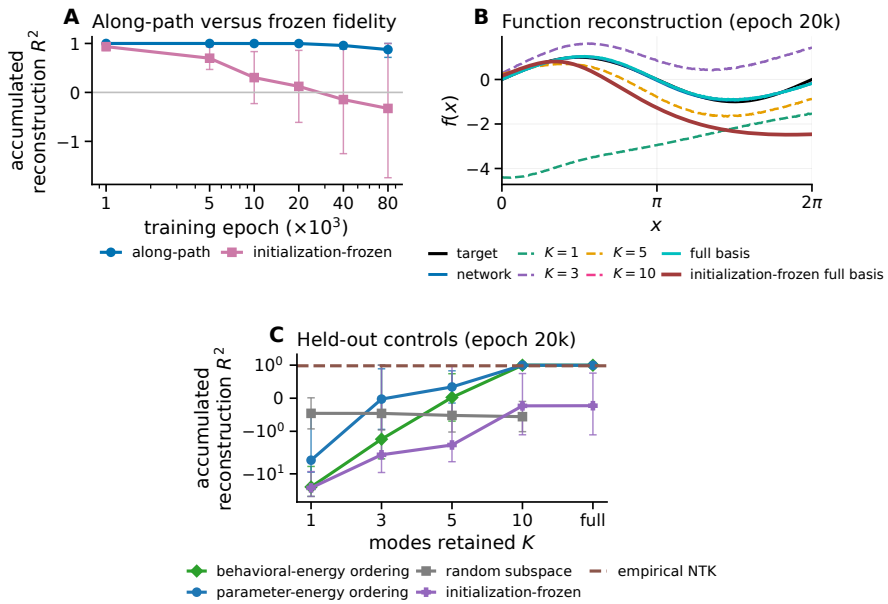

Figure 2: **Path-dependent induced effects improve accumulated reconstruction. A:** Mean Top-10 accumulated reconstruction R^2 over 15 runs for along-path and initialization-frozen induced effects. **B:** Representative along-path Top-K function reconstruction at epoch 20k with the initialization-frozen full-basis baseline. **C:** Controls at epoch 20k over five width-64 seeds, scored on a separate analysis set held out from behavioral-energy ranking. Capped bars show one standard deviation. Panel A uses logarithmic epoch spacing, Panel C uses symmetric-log vertical scaling, and Panels A–B use trajectory-level behavioral-energy ordering. Protocols and control definitions are in Appendix C.

high Top-10 accumulated R^2, whereas dimension-matched random subspaces perform poorly on average. The empirical NTK also remains predictive of the regression function, while Mode Dynamics attributes output change along the realized finite-network trajectory. Reconstruction remains strong across seeds, widths, and checkpoint schedules in the SGD grid (Appendix Figure 7).

Behavioral concentration. The checkpoint-by-checkpoint analysis in Appendix Figure 5 complements the trajectory-level reconstruction ordering by showing how checkpoint-specific behavioral energy is distributed during learning. PR remains small throughout training. Separately, trajectory-derived orderings reach high fidelity by $K = 10$, while random-subspace and initialization-frozen controls remain poor (Figure 2C). Across the 15 trajectories, Top-10 accumulated R^2 at epoch 20k is 0.998 ± 0.002. At epoch 10k, the checkpoint-specific mean ranked realized-contribution energy profile has $\mathrm{PR} = 3.7$ and $K_{\mathrm{eff}}(0.9) \approx 4$.

4.1.2 Controlled Next-Token Prediction

We use small decoder-only transformers on three controlled last-position token-prediction tasks. The tasks are *copying*, which predicts the visible final content token, *induction*, which retrieves the token that followed a repeated anchor, and *random-label memorization*, which assigns one fixed random content-token label to each sequence. Let L denote sequence length and V_{c} content-vocabulary size. The systematic grid crosses five seeds, depths $\{1, 2, 3\}$, and 15 valid (L, V_{c}) pairs, yielding 225 runs per task and 675 total. The pairs are drawn from lengths $\{4, 6, 8, 10, 12\}$ and vocabularies $\{6, 8, 12, 16\}$ subject to $V_{\mathrm{c}} \geq L - 1$. All models reach perfect training accuracy. Top-K logit reconstructions are evaluated across five seeds in a reference configuration (depth 1, sequence length 4, content vocabulary 6). No train–test split is used, so all metrics are in-sample on the finite training sequences, and seed variation includes the generated sequences or labels. For predicted-token decisions, we report *Top-1 prediction agreement*, the fraction of examples for which the trained model and Top-K reconstruction have the same highest-logit token. Architecture, optimization, sequence construction, and estimator details are in Appendix D.

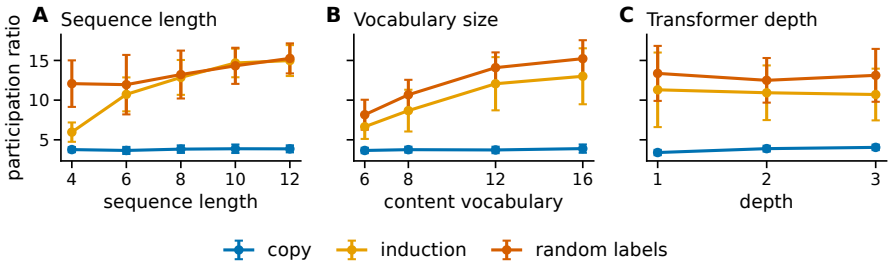

Figure 3: **Behavioral PR varies across next-token tasks and data conditions.** At epoch 50k, the full 675-run grid shows participation ratio versus sequence length in Panel A, content vocabulary in Panel B, and transformer depth in Panel C. Error bars denote one standard deviation over the corresponding conditions and seeds. All metrics are evaluated on training sequences. Matched-condition optimization, accuracy, and PR trajectories are in Appendix D.

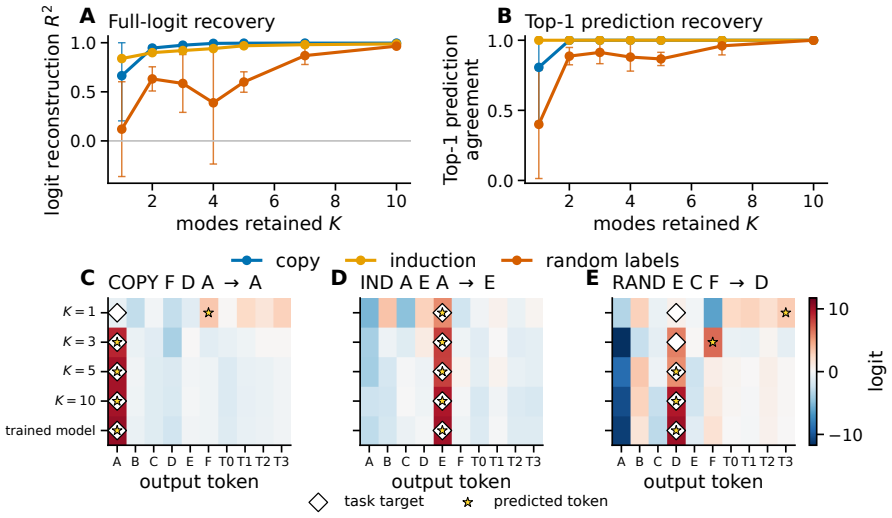

Figure 4: **Small sets of leading modes reconstruct next-token training-set behavior. A–B:** Training-set logit R^2 and Top-1 prediction agreement compare each Top-K reconstruction with the trained full-network output, averaged over five seeds. Capped bars show one population standard deviation, and modes use trajectory-level behavioral-energy ordering. **C–E:** Selected training examples illustrate low-to-high-K recovery at epoch 50k. Rows progress through cumulative Top-K modal reconstructions for $K \in \{1, 3, 5, 10\}$ to the same trained full-network output. Diamonds mark targets and stars mark row predictions, overlapping when correct. Selection details are in Appendix D.4.

Concentration across task and data conditions. Figure 3 summarizes the epoch-50k grid, while matched-condition trajectories in Appendix Figure 8 show that copying remains compact as induction and random-label memorization recruit additional modes during training. Across the full grid at epoch 50k, pooled PR is 3.8 ± 0.4 for copying, 11.0 ± 3.9 for induction, and 13.0 ± 3.2 for random labels. Behavioral PR is therefore associated with task identity even when all models ultimately fit their training sets. In the marginal summaries over valid pairs, larger sequence length or vocabulary is associated with higher PR for induction and random-label memorization, while copying remains nearly unchanged (Panels A–B). Because $V_c \geq L - 1$ couples the available sequence and vocabulary conditions, these panels are descriptive marginal associations rather than isolated causal effects. Restricted-support checks in Appendix D preserve the task ordering under this coupling. Within the tested grid, mean PR varies less across depths than across tasks despite increasing parameter count (Panel C). This does not establish scale invariance. Appendix D.3 compares trajectory dimension with static endpoint rank.

Behavioral reconstruction. Figure 4 links concentration to behavioral fidelity without equating energy and accuracy. Across the shared $K = 1, \ldots, 10$ sweep, copying reaches 100% Top-1 prediction agreement by $K = 2$ and induction by $K = 1$, while random labels approach consistent recovery only near $K = 10$. Additional modes continue to improve the numerical accuracy of reconstructed logits, especially for induction. Panels C and E visibly correct a low-K decision, whereas Panel D preserves induction's already-correct decision as its logits become more faithful. Appendix D.4 repeats the same cumulative row-wise view for three additional selected examples per task.

4.2 Validation on CIFAR-100 and Pythia-70M

The controlled settings isolate reconstruction, path dependence, and behavioral concentration. We next evaluate the framework on held-out CIFAR-100 images and a public Pythia-70M pretraining trajectory. Table 1 summarizes the main reconstruction, concentration, decision, and control results. We summarize the protocols here and provide full details in Appendices E and F. For these checkpointed systems, discrete parameter updates define the trajectory SVD, while each pointwise response term in Equation (8) is replaced by a response integral along each checkpoint chord, estimated by quadrature.

CIFAR-100 autoencoding. We train five convolutional autoencoders on CIFAR-100 (Krizhevsky, 2009) and record 1,961 checkpoints per trajectory. For each trajectory we estimate the leading 50 update-SVD modes in the parameter-energy ordering. The *locked* evaluation was fixed before any test score was inspected. It uses the path-integrated contributions of these modes to reconstruct the decoder's RGB output change from initialization on 100 test images per seed, held out from training, modal scoring, and mode selection (Appendix E). The 50-mode reconstruction attains pooled RGB-change $R^2 = 0.912$ while retaining under half of the parameter-motion energy (Table 1 and Appendix Figures 13–14). In a separate training-split analysis, fixed signed permutations preserve update magnitudes and pairwise angles but randomly reassign components and flip signs across parameters. Their strongly negative result shows that recovering output change requires preserving which parameters carry the coordinated motion, not update geometry alone. Because only a truncated 50-mode basis is estimated, this experiment supports compact reconstruction but does not identify a full-basis behavioral PR.

Pythia-70M next-token pretraining. We analyze 51 released Pythia-70M checkpoints spanning pretraining steps 0–143k, yielding 50 checkpoint updates and a corresponding full numerical mode basis. The reconstructed output is the final-position logit change from initialization, centered by subtracting each context's mean logit across the vocabulary. The analysis is likewise locked, with every setting fixed before any Pile-test access (Appendix F). Modes are ranked once on 128 Pile-validation contexts and scored without reranking on 128 untouched Pile-test contexts. Using two quadrature nodes per checkpoint chord ($Q = 2$), the leading 20 modes attain held-out pooled centered Δlogit $R^2 = 0.9668$, with validation behavioral energy PR 5.09 and $K_{\text{eff}}(0.9) = 13$ (Appendix Figure 15). The signed-permutation and initialization-frozen controls both fail on the same held-out contexts (Table 1), indicating that coordinate identity and along-path evaluation both matter for reconstruction in this setting. Despite this high geometric fidelity, the 20-mode reconstruction has zero Top-1 prediction agreement. Two *post-hoc* refinements probe this decision gap by changing analysis settings after the locked result and rescoring the same frozen test contexts (Appendix Table 3). Increasing quadrature order makes the full-update reference converge toward the endpoint change. Because this reference integrates complete checkpoint updates without modal projection and therefore realizes Equation (7) directly, its residual error is numerical rather than modal. The nested 101-checkpoint schedule supplies more modes to retain: its 80-mode reconstruction reaches 64.1% agreement at mean Kullback–Leibler (KL) divergence 0.531 nats (natural-log units), compared with 96.9% agreement and 0.00425 nats for the full-update reference. At 80% retention, the remaining decision gap therefore reflects truncation rather than quadrature error. Relaxing the rank cutoff, the trained model's predicted token lies within that reconstruction's ten highest-logit candidates for 88.3% of contexts, while the signed-permutation control never reaches the top fifty of a 50,304-token vocabulary (Appendix Figure 16). We interpret this single public trajectory as evidence for compact centered logit-change geometry, not decision-preserving model compression.

Table 1: **Summary of CIFAR-100 and Pythia-70M results.** CIFAR "±" values are population standard deviations over five trajectories, except the control, which is over three seed-0 signed permutations. Pythia rows above the *Post-hoc sensitivity* break use analysis settings fixed before any Pile-test access. Those below vary checkpoint density and quadrature afterwards on the same frozen test contexts. The bracketed interval is a 95% context-bootstrap confidence interval.

Analysis	Retained set / control	Metric	Result
CIFAR-100 autoencoding			
Test reconstruction	Top-50	pooled RGB-change R^2	0.912 ± 0.012
Image-level fidelity	Top-50	median per-image R^2	0.892
Parameter-motion energy	Top-50	fraction retained	$43.8\% \pm 0.4\%$
Coordinate control (training split)	signed permutation	pooled RGB-change R^2	-0.642 ± 0.344
Pythia-70M next-token pretraining			
Held-out reconstruction	20/50 modes (40%), $Q = 2$	pooled centered Δlogit R^2, Top-1 prediction agreement	$0.9668\ [0.9662, 0.9673]$, 0%
Validation concentration	51 checkpoints / 50 modes	PR, $K_{\text{eff}}(0.9)$	5.09, 13
Coordinate / state controls	signed-perm. 50-mode path, init.-frozen all-mode	pooled centered Δlogit R^2	$-0.890,\ -0.018$
Post-hoc sensitivity			
Full-update convergence	51 ckpt, no modal projection	Top-1 prediction agreement	77.3% at $Q = 2$ (locked), 97.7% at $Q = 8$
Distribution / decision recovery	101 ckpt, 80/100 modes (80%), $Q = 8$	mean KL (nats), Top-1 prediction agreement	0.531, 64.1%
Full-update reference	101 ckpt, no modal projection, $Q = 8$	mean KL (nats), Top-1 prediction agreement	0.00425, 96.9%

5 Discussion

Across all four settings, small subsets of trajectory modes reconstruct substantial output change. Along-path effects outperform initialization-frozen effects in the tested comparisons. Trajectory-derived directions outperform dimension-matched random subspaces, and signed-permutation controls fail to reconstruct output change. In the controlled experiments, behavioral concentration varies systematically with task and data, while in the transfer settings it remains only partially characterized.

Implications. Trajectory modes provide a parameter-space account of how outputs change during learning, complementing activation-level methods that study computation in a fixed trained network. Reconstruction attributes output change to coordinated parameter directions, providing a basis for monitoring behavior changes and auditing the associated parameter motion. Concentration distinguishes changes carried by a few directions from broadly distributed changes, which may inform hypotheses about behavioral fragility or robustness. Safety and alignment are natural settings for these questions, but demonstrating such uses requires causal evidence beyond this paper.

Scope and limitations. The method requires checkpoints containing the same complete set of parameter tensors in a consistent coordinate order. Computing mode-induced effects also requires JVPs through the model, making the method heavier than analyses based only on final weights. The framework quantifies concentration without explaining why it arises. It does not localize modes to model components or circuits or establish that they are causally necessary. The transfer analyses have different basis limitations: the image-autoencoder basis is truncated, while the language-model mode count is limited by recorded intervals and is therefore resolution-dependent. Evidence is limited to one 730k-parameter autoencoder architecture and one 70M-parameter language-model pretraining trajectory, so generalization across architectures and scales remains unresolved. The language-model results also show that near-perfect centered-logit reconstruction can still leave a decision gap.

Future work. Questions of resolution and scaling are computational rather than conceptual. Experiments with denser checkpoint schedules would test whether concentration stabilizes with temporal resolution, while broader architectures and larger models would test generality. Online variants that update the mode basis as training proceeds would be needed for real-time monitoring. Relating modes to layerwise activations and localized circuits would connect training dynamics to computation in the trained network. Moving from attribution to causality requires interventions that amplify, suppress, or remove modes during training and measure their behavioral consequences.

Ethics Statement

The work analyzes controlled regression and next-token tasks, CIFAR-100, and public Pile validation and test text using released Pythia checkpoints. It involves no human-subjects research or model deployment. The resulting trajectory attributions do not guarantee causal control or model safety.

Reproducibility statement

The Appendix documents the experimental and analysis protocols, numerical methods, and controls for each setting. Upon acceptance, we will release the code and supporting materials needed to reproduce the reported results. Bitwise reproducibility across hardware backends is not claimed.

AI use statement

Generative AI assisted with implementing experiment code, scripting figures, and drafting and editing prose. The authors developed the framework and designed the experiments, controls, and frozen protocols. They reviewed and revised all AI-assisted text and code. They ran every experiment, performed sanity and consistency checks, verified every reported number against saved artifacts, and take responsibility for the final content.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2021. URL https://arxiv.org/abs/2012.13255.

Anthropic. Circuit tracing: Revealing computational graphs in language models. https://transformer-circuits.pub/2025/attribution-graphs/methods.html, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, U. S. V. S. N. 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, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Callum McDougall, C. Daniel Freeman, Theodore Sumers, Edward Rees, et al. Towards monosemanticity: Decomposing language models with dictionary learning. https://transformer-circuits.pub/2023/monosemantic-features, 2023.

Jonathan Brokman, Roy Betser, Rotem Turjeman, Tom Berkov, Ido Cohen, and Guy Gilboa. Enhancing neural training via a correlated dynamics model. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.13247.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/ae614c557843b1df326cb29c57225459-Abstract.html.

Akshunna S. Dogra and William T. Redman. Optimizing neural networks via Koopman operator theory. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.02361.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022. URL https://arxiv.org/abs/2209.10652.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. URL https://arxiv.org/abs/2101.00027.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024. URL https://arxiv.org/abs/2406.04093.

Leo Gao, Achyuta Rajaram, Jacob Coxon, Soham V. Govande, Bowen Baker, and Dan Mossing. Weight-sparse transformers have interpretable circuits. *arXiv preprint arXiv:2511.13653*, 2025. URL https://arxiv.org/abs/2511.13653.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1812.04754.

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. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018. URL https://arxiv.org/abs/1806.07572.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019. URL https://arxiv.org/abs/1902.06720.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1804.08838.

Tao Li et al. Low dimensional landscape hypothesis is true: Dnns can be trained in tiny subspaces. *arXiv preprint arXiv:2103.11154*, 2021. URL https://arxiv.org/abs/2103.11154.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Krámár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024. URL https://arxiv.org/abs/2408.05147.

Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Evolution of neural tangent kernels under benign and adversarial training. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4bc4e9ecd5ae4a75048dc216a770cba1-Abstract-Conference.html.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 2018. URL https://www.nature.com/articles/s41467-018-07210-0.

Iva Manojlović, Maria Fonoberova, Ryan Mohr, Aleksandr Andrejčuk, Zlatko Drmač, Yannis Kevrekidis, and Igor Mezić. Applications of Koopman mode analysis to neural networks. In *Proceedings of the AAAI 2021 Spring Symposium on Combining Artificial Intelligence and Machine Learning with Physical Sciences*, volume 2964 of *CEUR Workshop Proceedings*, pp. 182, 2021. URL https://ceur-ws.org/Vol-2964/article_182.pdf.

Jialin Mao et al. The training process of many deep networks explores the same low-dimensional manifold. *Proceedings of the National Academy of Sciences*, 2024. URL https://www.pnas.org/doi/10.1073/pnas.2310002121.

Alan J. H. McGaughey and Jason M. Larkin. Predicting phonon properties from equilibrium molecular dynamics simulations. *Annual Review of Heat Transfer*, 17:49–87, 2013. doi: 10.1615/AnnualRevHeatTransfer.2013006915.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layer neural networks: Dimension-free bounds and kernel limit. In *Proceedings of the 32nd Conference on Learning Theory*, 2019. URL https://proceedings.mlr.press/v99/mei19a.html.

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://arxiv.org/abs/2002.08484.

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

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. URL https://doi.org/10.1017/S0022112010001217.

Mariia Seleznova and Gitta Kutyniok. Neural tangent kernel beyond the infinite-width limit: Effects of depth and initialization. In *Proceedings of the 39th International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/seleznova22a.html.

Rishi Sonthalia, Michael Murray, and Guido Montúfar. Low rank gradients and where to find them. *arXiv preprint arXiv:2510.01303*, 2025. URL https://arxiv.org/abs/2510.01303.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Christopher Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. https://transformer-circuits.pub/2024/scaling-monosemanticity/, 2024.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Nathan Srebro, and Daniel Soudry. Kernel and rich regimes in overparametrized models. *arXiv preprint arXiv:2002.09277*, 2020. URL https://arxiv.org/abs/2002.09277.

Greg Yang and Edward J. Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

SUPPLEMENTARY MATERIAL

Table 2 summarizes the experimental settings and their metric coverage.

Table 2: **Experimental overview and metric coverage.**

setting	task / model	reconstructed output	mode ordering	reported measures
Scalar regression	sine fitting one-layer MLP	scalar prediction curve	parameter-energy (local) behavioral-energy (accumulated)	local/accumulated Top-K R^2 PR and auxiliary K_{eff}
Next-token prediction	copy / induction / random labels decoder-only transformer	last-position logits and predicted-token decisions	behavioral-energy	Top-K logit R^2 and Top-1 prediction agreement PR and auxiliary K_{eff}
Pixel-space autoencoding	CIFAR-100 reconstruction convolutional autoencoder	RGB decoder-output change	parameter-energy	Top-50 RGB-change R^2 retained parameter-motion energy, no full PR
Public pretraining trajectory	Pythia-70M validation ranking / locked Pile-test scoring	vocabulary-centered final-position next-token logit change	behavioral-energy on dedicated ranking contexts	held-out pooled Δlogit R^2, PR, K_{eff} post-hoc KL and Top-r prediction agreement

A METHODOLOGICAL SCOPE

Trajectory scope and mode definition. Mode Dynamics retrospectively analyzes a finite recorded trajectory in a fixed parameter coordinate system. It assumes neither infinite width, Gaussian initialization, nor an NTK regime. The complete trajectory defines the regression, controlled next-token, and Pythia SVD bases, while behavioral-energy rankings also depend on the stated analysis inputs and intervals. These full SVDs reconstruct every recorded parameter update up to numerical precision. Individual directions remain coordinate-dependent, arbitrary in sign, and defined only up to rotations within degenerate singular subspaces. Near-degeneracy can destabilize individual directions while preserving their joint span. The CIFAR-100 analysis instead estimates only the leading 50 modes with randomized truncated SVD.

Exactness and numerical estimation. The main-text JVP maps modes to output space and requires the output map to be differentiable almost everywhere along the chosen interpolation. Unlike full-SVD update reconstruction, local output reconstruction is first-order. Regression and controlled next-token JVPs use forward finite differences without an ϵ sweep. For regression, the along-path estimator applies one finite difference to each aggregate projected update, whereas the initialization-frozen estimator sums modewise finite differences at θ_0. As $\epsilon \to 0$, the estimators converge to state-specific JVPs, with along-path evaluation at the current states and frozen evaluation at initialization. Their contrast therefore tests the stated frozen baseline, while finite ϵ adds estimator approximation. CIFAR-100 uses forward-mode AD along straight segments between consecutive checkpoints, avoiding finite-difference error but retaining quadrature, floating-point, nonsmooth-derivative, checkpoint-spacing, and interpolation dependencies. Pythia uses the same chord-wise forward-mode construction with two-point Gauss–Legendre quadrature and vocabulary-centered final-position logit change in the locked analysis. Checkpoint density and quadrature order are distinct. Additional saved checkpoints change the update matrix and mode basis, shorten the piecewise-linear chords, and can change truncated projections. Additional quadrature nodes leave the checkpoints and basis fixed and only improve integration along those chords. For Pythia, *full-update* means integrating complete checkpoint updates without any modal projection, while *all-mode* means summing the full numerical mode basis. The full-update Jacobian line integral is an exact differential and therefore recovers the endpoint output change for any connecting chord in the exact-integration limit. Residual full-update error diagnoses quadrature or floating-point error. An all-mode modal reconstruction can additionally inherit numerical SVD and projection error. Truncated modal attributions, unlike the full sum, can depend on the chosen chord path and checkpoint schedule.

Uncertainty reporting. Unless stated otherwise, error bars, bands, and "$\pm$" values are descriptive population standard deviations over the runs identified in each caption, not confidence intervals. Pythia confidence intervals instead use 2,000 context-bootstrap resamples of its locked confirmatory Pile-test contexts.

B COMPUTATIONAL COMPLEXITY

For T updates in P parameters, explicitly storing the trajectory requires $O(TP)$ memory. A dense SVD costs $O(\min\{T^2P, TP^2\})$ in the worst case. Evaluating K directions at T checkpoints by

finite differences requires $O(KT)$ additional forward evaluations on the selected examples. With Q quadrature nodes per checkpoint chord, path-integrated evaluation requires $O(KTQ)$ direction–node evaluations. AD JVPs have the analogous direction–state evaluation burden and can be batched. Leading-mode truncation, randomized or streaming SVD, checkpoint subsampling, minibatch energy estimates, and layerwise decompositions can reduce these costs.

C Scalar Regression

C.1 Protocol and Reproducibility

Data and training. The reference experiment draws 5,000 inputs independently and uniformly from $[0, 2\pi]$ with targets $y = \sin x$. A one-hidden-layer width-64 tanh MLP (193 parameters) is trained from fan-in-scaled uniform initialization for 80k epochs using full-batch SGD on MSE at learning rate 0.01, with checkpoints every 10 epochs. The robustness grid spans widths 32, 64, and 128 with five seeds each (15 trajectories). Each seed controls both the input draw and parameter initialization.

Evaluation and controls. A deterministic 400-point analysis grid on $[0, 2\pi]$ is alternated into 200 ranking points and 200 scoring points held out from behavioral-energy ranking. This is a ranking–scoring analysis split, not a task train–test split. For each width-64 trajectory, the random baseline uses 20 nested, dimension-matched random subspaces formed by orthonormalizing Gaussian parameter directions. This construction samples orientations without favoring any parameter coordinate or learned update direction. Because these subspaces stop at dimension 20, the random control has no full-basis entry and is plotted only up to that budget. A dimension-193 random subspace would span parameter space and make full-basis reconstruction trivial. The initialization-frozen baseline evaluates induced effects at initialization, while the empirical NTK baseline simulates full-batch gradient descent in the initialization-linearized finite-width model at the same learning rate. These controls test generic low-dimensional projection and the chosen initialization linearization. Without cross-trajectory mode swaps, they do not establish that individual modes are trajectory-specific. Parameter-energy ordering is not an independent random control because integrated squared amplitudes equal squared singular values.

Analysis schedule. Induced effects use forward finite differences with $\epsilon = 0.001$. Behavioral energy and local reconstruction are evaluated at 1k, 5k, 10k, 20k, and 40k. Accumulated reconstructions continue through the terminal 80k checkpoint.

C.2 Optimization, Concentration, and Robustness

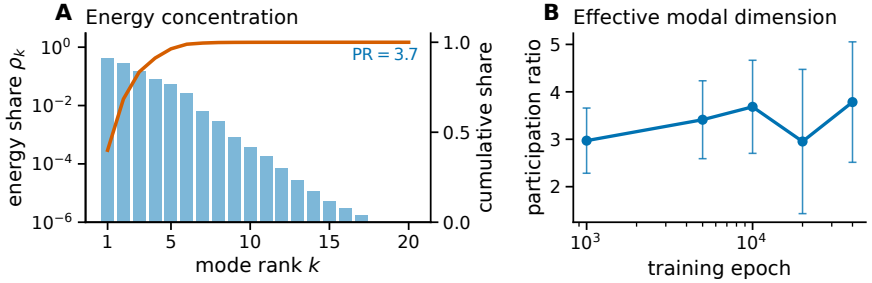

Figure 5: **Regression output-change energy is concentrated across training. A:** Mean ranked behavioral energy at epoch 10k, its cumulative share, and the corresponding participation ratio. **B:** Mean participation ratio across the five analyzed nonterminal checkpoints through 40k, with capped one-standard-deviation bars. Panel A averages the same five-seed, three-width sine grid.

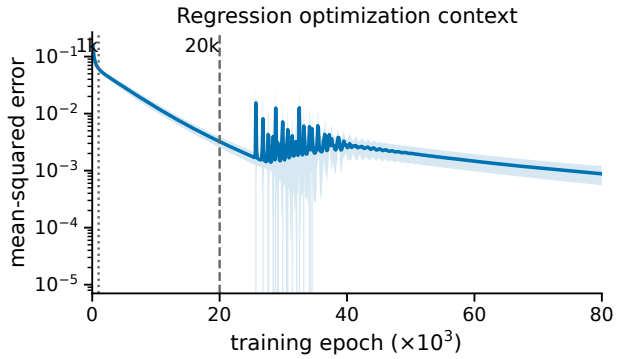

Figure 6: **Regression optimization context.** Mean width-64 sine-target MSE through 80k. Shading shows one standard deviation over five seeds. Vertical markers locate the representative 1k and 20k analysis checkpoints.

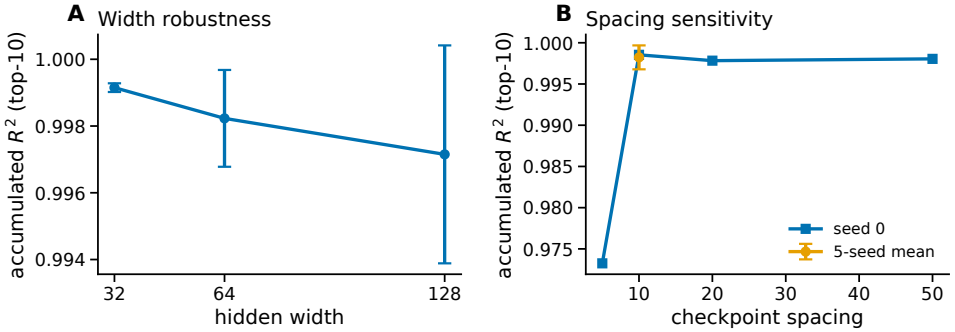

Figure 7: **Seed, width, and checkpoint-spacing robustness. A:** Sine-target Top-10 accumulated R^2 at 20k, aggregated over five seeds at each width. **B:** Sine Top-10 accumulated R^2 at 20k versus checkpoint spacing for width 64. Top-10 uses trajectory-level behavioral-energy ordering. The default spacing of 10 averages five seeds, while additional spacing runs use seed 0 only.

The loss decreases monotonically through 25k and then rises on 22% of logged steps between 25k and 40k, the same interval over which Top-10 accumulated reconstruction degrades and its across-seed spread widens. Figure 7 shows that high-fidelity Top-10 accumulated reconstruction is reproducible across seeds and widths. Panel B is a descriptive spacing check. Changing spacing also changes the update matrix, SVD basis, and temporal approximation, so this comparison does not isolate discretization error. As a compact target-function check, the same 15-run grid fitted $\sin x + 0.5 \sin 2x + 0.25 \sin 3x$ and achieved Top-10 accumulated $R^2 > 0.99$ at epoch 20k.

D Controlled Next-Token Prediction

D.1 Protocol and Reproducibility

Model and tasks. The decoder-only transformers use hidden dimension 32, two attention heads, learned token and position embeddings, and a linear output head. Each block contains pre-normalized causal self-attention followed by a pre-normalized fourfold-expansion GELU MLP, with no dropout. Depth ranges from one to three. For sequence length L and content-vocabulary size V_c, four task tokens are added and each condition contains $V_c(V_c - 1)$ training sequences. Copying predicts the visible final content token, induction predicts the token following a repeated anchor, and random-label memorization assigns each unique sequence one fixed, uniformly sampled content-token label.

Training grid. The systematic grid crosses three tasks, depths $\{1, 2, 3\}$, lengths $\{4, 6, 8, 10, 12\}$, and content vocabularies $\{6, 8, 12, 16\}$, subject to $V_c \geq L - 1$, with five seeds per valid condition (675 runs). The seed determines sampled sequences, fillers, anchor positions, or random labels as

applicable. Models train for 100k steps using full-batch SGD at learning rate 0.01 and last-position cross-entropy, with parameters recorded every 100 steps. No train–test split is used.

Behavioral concentration and reconstruction (in-sample). PR and reconstruction are evaluated in-sample using all $V_c + 4$ last-position logits on the finite training sequences. The complete recorded trajectory defines the SVD basis. Behavioral-energy statistics use the outgoing intervals at 1k, 5k, 10k, 20k, and 50k, with JVPs estimated by forward finite differences at $\epsilon = 0.001$. At each checkpoint, PR uses the full realized-contribution energy distribution. Trajectory-level reconstruction ordering aggregates contributions across the five intervals. Top-K reconstruction uses this trajectory-level behavioral-energy ordering and is evaluated at 50k in the reference configuration (depth 1, length 4, content vocabulary 6).

D.2 Training and Behavioral Concentration Diagnostics

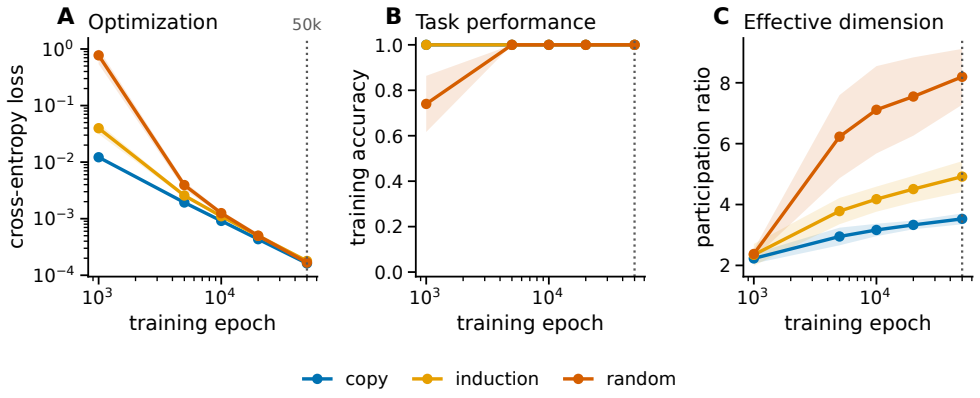

Figure 8: **Next-token optimization and participation-ratio trajectories. A–C:** Cross-entropy, training accuracy, and participation ratio over time for copying, induction, and random labels in the matched reference configuration (depth 1, sequence length 4, content vocabulary 6). Lines and shading are mean $\pm$ standard deviation over five seeds, and the dotted line marks the 50k checkpoint used for the main-text task/data comparison.

Figure 8 shows that copying reaches a compact modal distribution early, whereas induction and random-label memorization recruit additional modes as training proceeds. All tasks reach perfect training accuracy by the endpoint, so the PR differences describe how learning is distributed rather than differences in final fit.

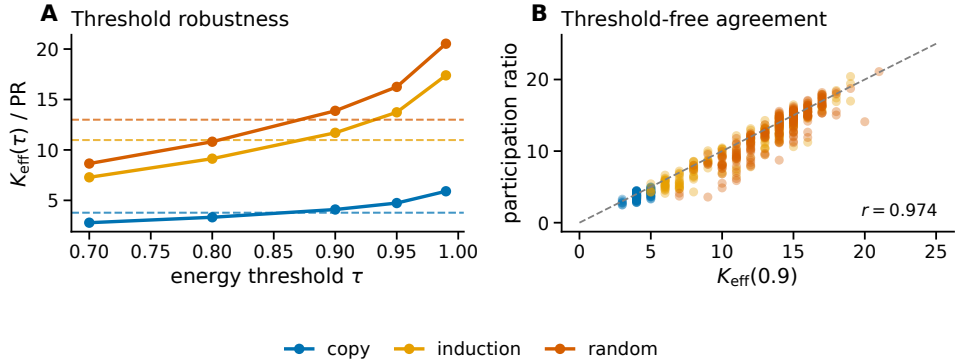

Figure 9: **Next-token behavioral-concentration diagnostics. A:** $K_{\text{eff}}(\tau)$ varies with the selected energy threshold from 0.70 to 0.99, with dashed horizontal lines showing the threshold-independent mean PR for each task. **B:** PR strongly agrees with the auxiliary $K_{\text{eff}}(0.9)$ count.

Figure 9 shows that task ordering is stable across cumulative-energy thresholds and that PR tracks $K_{\text{eff}}(0.9)$ while remaining threshold-free.

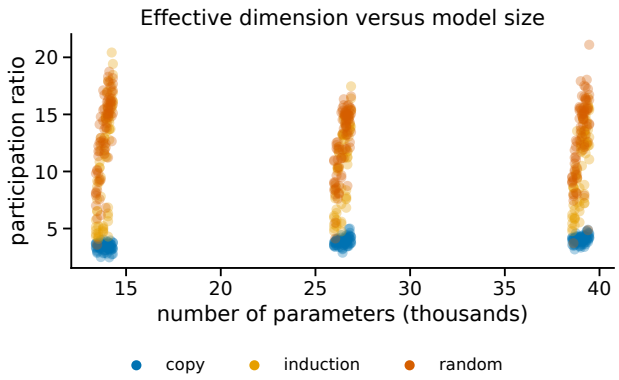

Figure 10: **Behavioral PR versus parameter count.** Across the controlled grid (approximately 13k–39k parameters), PR separates more strongly across task groups than across the displayed parameter-count range. This is a descriptive bounded-grid association rather than a variance decomposition or claim of scale invariance.

Figure 10 compares behavioral PR with parameter count over the bounded grid. Restricted-support checks preserve the task ordering despite the coupling between sequence length L and content vocabulary V_c. With $V_c \in \{12, 16\}$ ($n = 150$ per task), mean PR is 3.8 ± 0.5, 12.5 ± 3.5, and 14.7 ± 2.2 for copying, induction, and random labels, respectively. With $L \in \{4, 6\}$ ($n = 120$ per task), the corresponding values are 3.7 ± 0.4, 8.3 ± 2.9, and 12.0 ± 3.4. Within these marginal summaries, mean PR varies less across depth levels than across task constructions. The full grid contains 15 valid length–vocabulary pairs satisfying $V_c \geq L - 1$. Consequently, the main figure's marginal length and vocabulary curves are associated summaries rather than independent interventions.

D.3 Trajectory dimension and static endpoint rank

Trajectory concentration and endpoint low rank are related but distinct objects. Mode Dynamics measures how many trajectory-derived directions carry behavioral change during training, whereas low-rank adaptation (LoRA) imposes a static low-rank parameterization (Hu et al., 2021). We compare behavioral PR at epoch 50k with a separate endpoint head-spectrum PR computed from the singular-value energy of the output-head change from initialization.

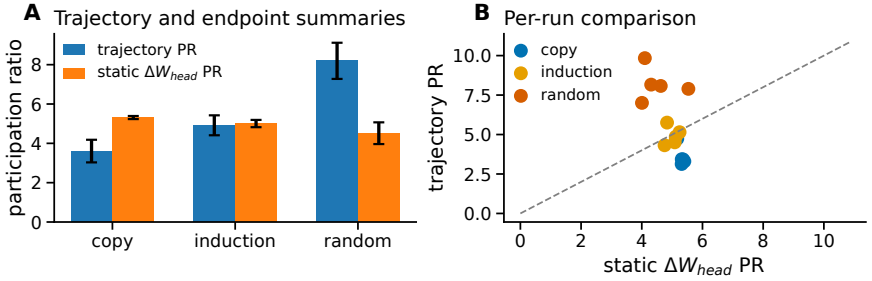

Figure 11: **Dynamic trajectory and static endpoint summaries. A:** Behavioral PR of realized-contribution energy at epoch 50k and endpoint head-spectrum PR across tasks. **B:** Per-run comparison. The quantities are not commensurate. The dynamic statistic uses full-parameter realized contributions at 50k, whereas the static statistic uses the 0–100k output-head change and is capped by head rank.

Figure 11 shows similar static output-head dimensionality across tasks despite descriptive differences in trajectory-level behavioral PR, so endpoint head rank does not substitute for the trajectory analysis and is not a quantitative rank-gap estimate. This diagnostic does not evaluate LoRA fine-tuning, mode rotation, or cross-seed alignment.

D.4 Per-Example Top-K Behavioral Reconstruction

Among examples with correct trained-model and Top-10 predictions and monotonically improving per-example logit R^2, we rank by the number of prediction changes, then the number of distinct predictions, and then the $K = 1$-to-Top-10 improvement. The main figure shows the highest-ranked case. Figure 12 shows three additional cases per task in a 3×3 layout, while preferring distinct target tokens. Induction has no prediction changes, consistent with its perfect aggregate $K = 1$ agreement, but its selected examples still show improving logits. These behavioral reconstructions do not test circuit localization or semantic monosemanticity.

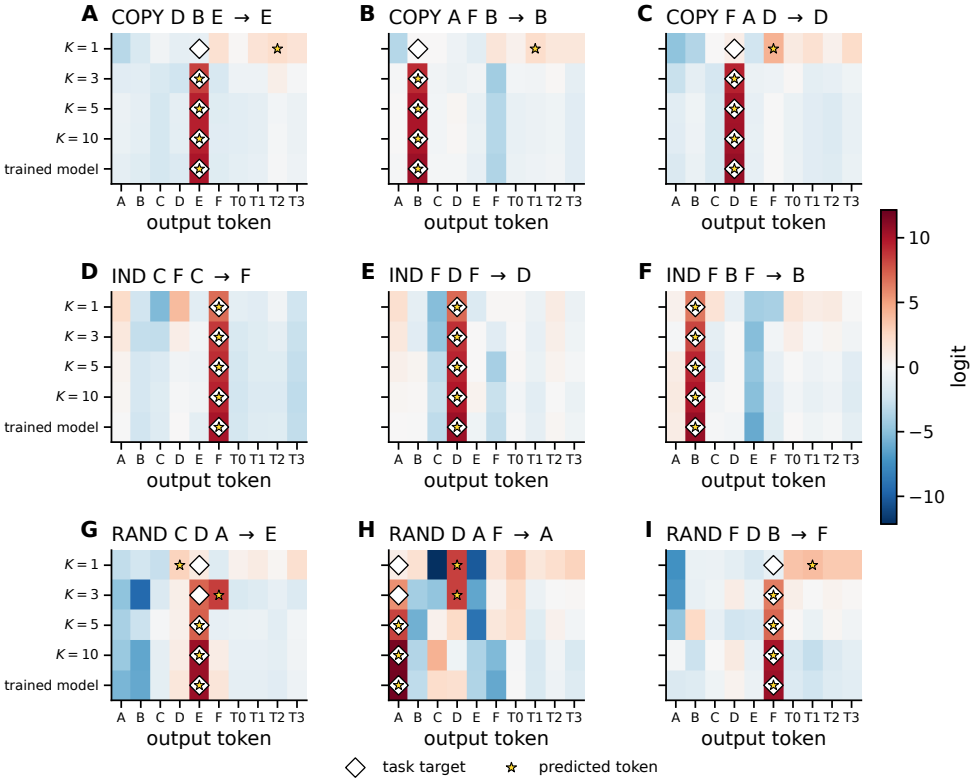

Figure 12: **Selected per-example cumulative logit reconstructions. A–C:** Copying. **D–F:** Induction. **G–I:** Random labels. Each subplot is one training example. Rows progress through cumulative Top-K modal reconstructions to the trained-model full-network reference. White diamonds and smaller gold stars are centered on the task target and row prediction, respectively, so a correct prediction appears as a star over a diamond.

Across the main and supplementary examples, copying and induction often recover the correct decision before every logit is numerically reconstructed, whereas random-label examples generally require broader sets of leading modes.

E Pixel-Space Reconstruction on CIFAR-100

E.1 Protocol and Reproducibility

Task and model. This experiment analyzes pixel-level reconstruction rather than classification. We train a convolutional autoencoder on all 50,000 CIFAR-100 training images using mean absolute RGB pixel loss. Class logits receive zero loss weight. Inputs use no augmentation and are normalized by channel means $(0.5071, 0.4867, 0.4408)$ and standard deviations $(0.2675, 0.2565, 0.2761)$. The encoder applies four convolution–group-normalization–ReLU stages with 48, 96, 192, and 192 channels and one max-pool downsampling to a 16×16 spatial bottleneck. The no-skip decoder

bilinearly upsamples once, applies a convolution–group-normalization–ReLU stage, and uses a sigmoid RGB head. Neither side has residual or refinement blocks. The model has 730,087 trainable parameters.

Training and design selection. All five runs use Adam with learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, numerical-stability constant 10^{-8}, no weight decay, batch size 256, and 50 epochs. Training uses shuffled minibatches on Apple MPS, with NumPy and PyTorch seeds 0–4 controlling initialization and sampling. The architecture, optimizer, integration settings, and $K = 50$ endpoint were selected using seed-0 decoder fidelity (including endpoint test-image L1), training-split integration calibration, and exploratory modal analyses before running seeds 1–4. Seed 0 remains in the five-run aggregate, so the uncertainty summary is descriptive and includes the design seed. No test-image modal reconstruction score was inspected before the protocol, source hashes, and exact test indices were frozen.

Evaluation. The CIFAR development analysis evaluates modal fidelity in-sample on a deterministic, seed-specific class-balanced subset of 100 training images, one per CIFAR-100 class. The locked evaluation instead evaluates the already selected Top-50 reconstruction on a fixed class-balanced subset of 100 test images per seed. These images are held out from training, modal scoring, mode selection, and reranking, but not from the earlier endpoint decoder-L1 check. A separate eight-image seed-0 training-split analysis supplies the full $K \in \{1, 3, 5, 10, 20, 30, 40, 50\}$ curve and qualitative examples. The class-balanced training and test subsets use the stated training seed plus 10,000 and 20,000, respectively. Randomized SVD uses the training seed.

E.2 Path-Integrated Reconstruction and Metrics

Trajectory basis and retained parameter motion. There are 196 optimizer steps per epoch. Storing parameters every five steps gives 1,961 checkpoints and $N = 1{,}960$ update intervals. For $i = 0, \ldots, N-1$, let $\Delta\theta_i = \theta_{i+1} - \theta_i$. The main framework stacks these updates as columns of V. The implementation forms $D = V^\top \in \mathbb{R}^{N \times P}$, where $P = 730{,}087$, and estimates its first 50 right singular vectors using scikit-learn randomized SVD with four power iterations and `random_state=seed`. These right singular vectors are the parameter modes u_k defined from V, and $\Pi_K = \sum_{k=1}^{K} u_k u_k^\top$ projects onto the first K modes. By Equations (2)–(3), the parameter-motion energy $\sigma_k^2 = \sum_i a_{k,i}^2$ is the total squared motion along mode k. Figure 13B reports the retained fraction $\sum_{k \le K} \sigma_k^2 / \|V\|_F^2$, whose denominator includes all recorded updates rather than only the 50 estimated modes. This parameter-space diagnostic is distinct from the behavioral cumulative share $C_K(t)$ defined in Section 3.4.

Path-integrated output reconstruction. The Top-K RGB-change reconstruction integrates the projected updates along the straight checkpoint chords:

$$\begin{aligned} \widehat{\Delta f}_K(x) &= \sum_{i=0}^{N-1} \int_0^1 J_\theta f(x, \theta_i + c\Delta\theta_i)\, \Pi_K \Delta\theta_i \, dc \\ &\approx \sum_{i=0}^{N-1} \sum_{q=1}^{Q} w_q J_\theta f(x, \theta_i + c_q \Delta\theta_i)\, \Pi_K \Delta\theta_i. \end{aligned} \tag{15}$$

Here $Q = 4$ and (c_q, w_q) are the Gauss–Legendre nodes and weights on $[0, 1]$. Because $\Pi_K \Delta\theta_i = \sum_{k \le K} a_{k,i} u_k$, the integrand sums the main-text realized contributions $a_{k,i}\phi_k(x, \theta_i + c\Delta\theta_i)$ along the chord. Thus Equation (15) is the path-integrated counterpart of the main modal reconstruction, not a new decomposition. JVPs are computed directly with forward-mode AD and batched over the selected images. On eight seed-0 images, full-update calibration over two 100-step windows gave chord-integration $R^2 = 0.983$ for steps 0–100 and 0.999 for steps 96–196 at the chosen five-step spacing and four-point rule. This checks short-window numerical integration, not a full-basis reconstruction ceiling over all 50 epochs.

Fidelity metrics. The reconstruction reference is the trained decoder's output change $\Delta f(x) = f(x, \theta_{\text{final}}) - f(x, \theta_0)$, and adding $\widehat{\Delta f}_K(x)$ to the initial output gives the reconstructed final decoder

output. We apply Equation (14) with $s = \Delta f$ and $\widehat{s} = \widehat{\Delta f}_K$, stacked over the selected images, pixels, and RGB channels. Thus the reference is the trained decoder's RGB change, not the ground-truth image or class label. Per-image R^2 applies the same equation separately to each image. Because its denominator can be small for nearly constant image-wise changes, we additionally record $\|\widehat{\Delta f}_K - \Delta f\|_2 / \|\Delta f\|_2$ and emphasize the pooled score and distributional quantiles rather than the minimum. The implementation adds 10^{-12} to variance and norm denominators for numerical stability. Decoder mean absolute RGB error instead compares $f(x, \theta_{\text{final}}) \in [0, 1]$ with the ground-truth image and answers a separate question.

E.3 Signed-Permutation (Norm-Matched) Coordinate Control

Each control draw samples one parameter permutation and one independent Rademacher sign vector, initialized with the training seed plus 30,000. The same fixed signed permutation is applied to the Top-50 projected update at every interval and quadrature node. This transformation exactly preserves each projected update's ℓ_2 norm and all temporal inner products while randomizing its global parameter-coordinate assignment. Three independent draws are evaluated on seed 0. The controls therefore test whether learned coordinate alignment matters at matched instantaneous update norm. They do not provide five-seed uncertainty for the random baseline. They were fixed and scored in the training-split development analysis and were not recomputed on the held-out test images. Because the transformation crosses parameter tensors and can move energy into the RGB-irrelevant classifier head, it is neither layer- nor support-matched and does not isolate fine-grained trajectory specificity.

E.4 Additional Results

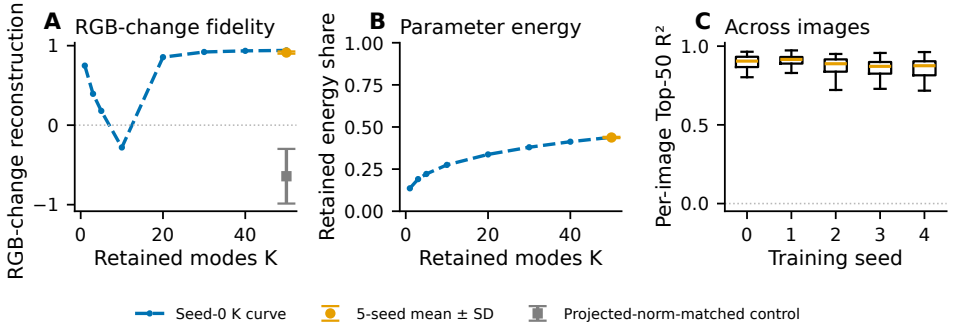

Figure 13: **A small parameter-energy-ranked subspace reconstructs the RGB change induced by CIFAR-100 training. A:** Training-split seed-0 eight-image K curve, five-seed Top-50 mean $\pm$ population standard deviation (100 training images per seed), and three seed-0 norm-matched controls. **B:** Fraction of parameter-motion energy retained. **C:** Training-split per-image Top-50 R^2 (100 images per box), with extreme fliers hidden only visually. Panel A reconstructs the trained decoder's RGB change from initialization, not the CIFAR target, and modal scoring uses training images.

Aggregate fidelity. The five training-split pooled Top-50 scores underlying Figure 13 are 0.922, 0.934, 0.911, 0.908, and 0.892. Excluding the design seed gives 0.911 ± 0.015 over seeds 1–4, leaving the descriptive conclusion unchanged. The locked held-out scores are 0.924, 0.927, 0.902, 0.896, and 0.908, giving 0.912 ± 0.012 across seeds while retaining $43.8\% \pm 0.4\%$ of parameter-motion energy. Pooling the 500 held-out seed–image evaluations gives median per-image $R^2 = 0.892$, interquartile range $[0.839, 0.919]$, and tenth percentile 0.763. Overall, 99.4% of scores are positive. The held-out median per-image relative ℓ_2 error is 0.291 with interquartile range $[0.250, 0.335]$. The Panel A control spread is over three seed-0 random draws, whereas the Top-50 error bar is over five training seeds.

Exploratory K-dependence and examples. The detailed seed-0 curve in Figure 13A is non-monotonic at small and intermediate K, including negative R^2 at $K = 10$. Parameter-energy ordering does not guarantee monotonic output-space reconstruction, so the curve documents the observed progression rather than implying that every additional mode improves fidelity. Figure 14 shows the corresponding cumulative reconstructions for fixed examples.

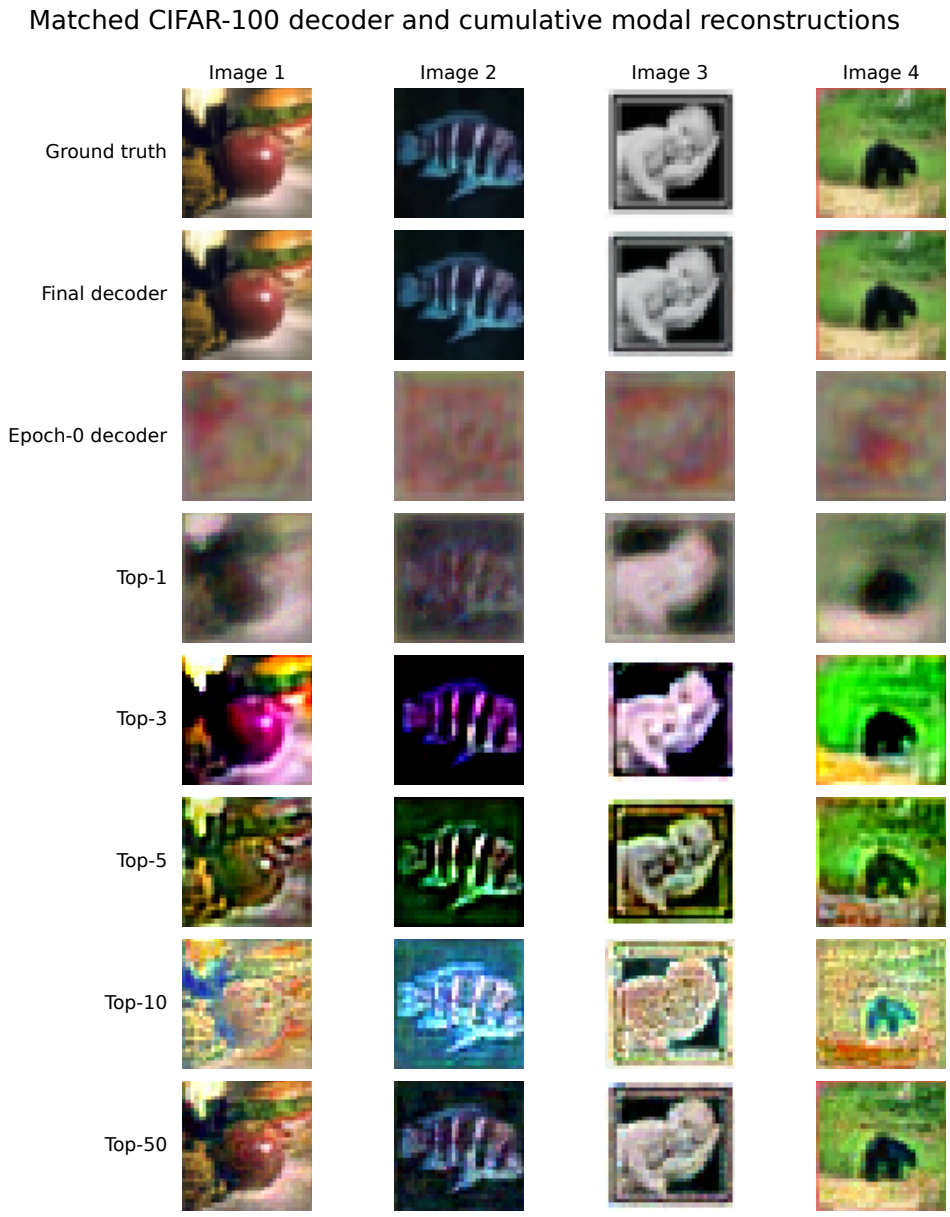

Figure 14: **Matched CIFAR-100 decoder outputs and cumulative modal reconstructions.** Columns are fixed evaluation images. Rows show the ground truth, final decoder, epoch-0 decoder, and cumulative Top-1, Top-3, Top-5, Top-10, and Top-50 path-integrated predictions from the detailed seed-0 run. These examples illustrate pixel-level recovery, while aggregate claims use the five-seed, 100-image results in Figure 13.

F PUBLIC PYTHIA-70M PRETRAINING TRAJECTORY

F.1 PROTOCOL AND DATA SEPARATION

Model and checkpoints. Pythia-70M is a decoder-only GPT-NeoX-style transformer released with intermediate pretraining checkpoints (Biderman et al., 2023). We analyze those released checkpoints without additional training or fine-tuning. Models and interpolated parameters are evaluated in float32 with eager attention on Apple MPS. The analysis uses 51 fixed checkpoints spanning step 0 through step 143000, with dense sampling through step 512 and broad coverage thereafter. The exact early schedule is $0, 4, 8, 16, 32, 64, 128, 256, 512$, followed by 1000, 2000, and $1000\,\mathrm{round}(1 + 142i/40)$ for $i = 1, \ldots, 40$. Thus the later intervals span either 3,000 or 4,000 optimizer steps. Flattening every trainable tensor in a fixed manifest gives 70,426,624 parameters per checkpoint and 50 checkpoint-to-checkpoint updates. Eigendecomposition of the 50×50 row Gram matrix yields a numerically full-rank 50-mode basis. The maximum observed orthonormality error is 1.08×10^{-5}.

Evaluation contexts. We tokenize without adding special tokens and deterministically select 256 nonoverlapping blocks of 129 tokens from the official Pile validation split (Gao et al., 2020). A SHA-256 hash of the fixed seed, source line number, and text determines candidate-document order and each token offset. The validation and test procedures scan 1,024 and 512 candidates, respectively, accepting the first nonoverlapping blocks. For each block, the first 128 tokens form the input prefix and the final token is retained only for task-level diagnostics. The first 128 blocks define the behavioral ranking set, and the remaining 128 form a disjoint validation scoring set used for development. Before any Pile-test access, we froze the validation-derived ordering, deterministic test-context selector, retained mode budgets, controls, two-node quadrature rule, and primary pooled R^2 metric. The confirmatory evaluation then deterministically selects 128 blocks from the previously untouched official Pile test split. The test procedure creates no ranking set and performs no reranking or tuning. Stored line numbers, token offsets, selection hashes, source-file digests, and frozen artifact hashes provide an audit trail.

F.2 BEHAVIORAL ORDERING AND PATH RECONSTRUCTION

Ordering. The trajectory modes are defined only by the parameter trajectory. To obtain the behavioral-energy ordering, we integrate each mode's realized contribution to the 50,304 final-position logits over all 50 checkpoint chords on the ranking contexts. Each chord uses two-point Gauss–Legendre quadrature and forward-mode JVPs. The two nodes are $c = (1 \pm 1/\sqrt{3})/2$ on the unit chord, each with weight $1/2$. At each node c, we evaluate the model at interpolated parameters $\theta_i + c\Delta\theta_i$. These are analysis points, not recovered intermediate training checkpoints. We subtract each effect's mean across vocabulary before applying Equation (11), removing the additive logit direction that leaves softmax probabilities unchanged. The ranking therefore uses full output-vector effects but no true next-token labels, losses, or decisions. Once estimated, the order is frozen for all scoring analyses.

Held-out reconstruction and controls. On the untouched Pile-test contexts, we path-integrate cumulative validation-ranked subsets with $K \in \{1, 3, 5, 10, 20, 50\}$ and compare them with the actual initialization-to-final centered-logit change. Let $C(z) = z - \frac{1}{V}\sum_j z_j$ center one context's vocabulary logits. The pooled score is

$$R^2_{\mathrm{pooled}} = 1 - \frac{\sum_c \|C(\widehat{\Delta z}_c) - C(\Delta z_c)\|_2^2}{\sum_c \|C(\Delta z_c)\|_2^2}. \tag{16}$$

Its 95% interval resamples the 128 contexts with replacement 2,000 times using seed 10,070 and recomputes this ratio while retaining all vocabulary coordinates within each sampled context. The full-update reconstruction integrates complete checkpoint updates along the path and supplies a numerical reference without modal projection. The initialization-frozen Top-K curve instead evaluates each cumulative K-mode displacement at the initial parameters. Its $K = 50$ endpoint is the initialization-frozen all-mode reconstruction up to numerical precision and remains distinct from full-update. A fixed signed parameter permutation of the 50-mode projected update preserves its norm at every interval but disrupts learned coordinate alignment.

F.3 RESULTS AND BOUNDARIES

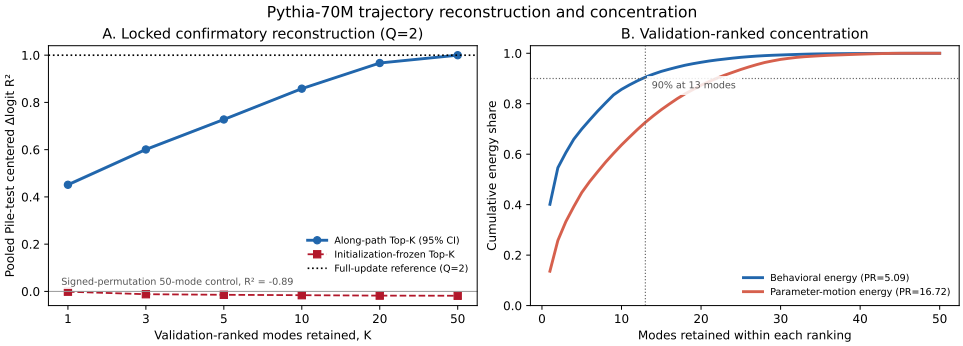

Figure 15: **Pythia-70M centered-logit change is concentrated in a behaviorally ranked trajectory subspace. A:** Locked confirmatory Pile-test pooled centered-Δlogit R^2 across validation-ranked K-mode subsets. Blue evaluates effects along the checkpoint path. Red is the initialization-frozen Top-K reconstruction, with $K = 50$ equal to the initialization-frozen all-mode reconstruction. The dotted line is full-update. Shading is a 95% context-bootstrap interval, and the annotation gives the signed-permutation 50-mode control. **B:** Validation-ranking cumulative behavioral energy in behavioral-energy ordering and cumulative parameter-motion energy in parameter-energy ordering. Legends report PR. Guides mark the 90% behavioral-energy threshold at 13 modes. The curves use separate rankings and are not a common-order comparison.

Figure 15A shows untouched-test pooled centered-Δlogit R^2 values of 0.451, 0.601, 0.728, 0.858, 0.9668, and 0.9998 for $K = 1, 3, 5, 10, 20, 50$, respectively. The $K = 20$ 95% context-bootstrap interval is $[0.9662, 0.9673]$. The full-update integration reference is 0.999809. The all-mode ($K = 50$) pooled score differs from the full-update reference by less than 10^{-7}. The signed-permutation 50-mode control gives -0.890, and the initialization-frozen all-mode reconstruction gives -0.018.

On the frozen validation ranking set, behavioral energy is more concentrated than parameter motion. Behavioral PR is 5.09 with $K_{\text{eff}}(0.9) = 13$, compared with PR 16.72 for the parameter-motion energy shares $\sigma_k^2 / \sum_j \sigma_j^2$. The behavioral-energy and parameter-energy rank positions have Spearman correlation 0.878, indicating substantial overlap without equivalence. We did not compute a parameter-ranked held-out Top-K reconstruction, so this correlation is not a reconstruction-sensitivity result.

Post-hoc checkpoint and quadrature sensitivity. These checks vary three analysis settings chosen after the locked result, namely checkpoint density, quadrature order, and retained mode budget, and leave the evaluation data untouched. We formed a nested 101-checkpoint schedule that retains all 51 original checkpoints and adds 50 released snapshots, reducing every later interval to at most 2,000 optimizer steps. This post-hoc schedule re-estimates the 100-mode basis and behavioral-energy ordering on the same validation ranking contexts, then scores the same frozen Pile-test contexts without test-set reranking. Its flattened float32 checkpoint stack occupies approximately 28.5 GB, while path analysis scales with checkpoint intervals, retained directions, quadrature nodes, and scoring contexts. An asymptotic checkpoint-density sweep was therefore computationally out of scope. Table 3 separates this checkpoint-resolution change from quadrature refinement on each fixed schedule, and Figure 16 shows the corresponding retained-fraction curves.

The absolute PR and K_{eff} increase when twice as many update intervals make twice as many modes available, while their fractions remain of similar order and decrease modestly. Table 4 compares the two schedules at matched retained fractions. Thus individual modes and absolute counts are schedule-dependent, but both resolutions show the same qualitative progression from rapid centered-logit recovery to later decision recovery.

For full-update, the large improvement from $Q = 2$ to $Q = 4$ and the much smaller change from $Q = 4$ to $Q = 8$ identify quadrature as the dominant numerical error and show convergence on both checkpoint schedules. Across the dense truncated set $K \in \{20, 30, 40, 50, 60, 80\}$, the $Q = 4$ and $Q = 8$ curves differ by at most 3.6×10^{-5} in pooled centered-Δlogit R^2 and one of 128 contexts in Top-1 prediction agreement. At $K = 80$, the dense $Q = 2/4/8$ results are pooled centered-Δlogit

Table 3: **Post-hoc Pythia checkpoint and quadrature sensitivity.** Concentration uses the fixed validation ranking contexts. The $Q = 2$ columns use the all-mode modal reconstruction, while $Q = 4, 8$ use the exact checkpoint displacement. The all-mode endpoint differs from full-update only at numerical precision. All output metrics use the same 128 frozen Pile-test contexts. The reported R^2 is the pooled centered-Δlogit score in Equation (16). KL is mean $D_{\mathrm{KL}}(p_{\mathrm{trained}} \| p_{\mathrm{recon}})$ in nats. Top-1 prediction agreement compares the trained model's and reconstruction's highest-logit tokens. PR and K_{eff} fractions divide each statistic by the available mode count.

Checkpoints	Modes	PR / modes	$K_{\mathrm{eff}}(0.9)$ / modes	Full-update pooled centered Δlogit R^2 $Q = 2/4/8$	Full-update mean KL (nats) $Q = 2/4/8$	Full-update Top-1 prediction agreement $Q = 2/4/8$
51 (locked)	50	5.09 (10.2%)	13 (26.0%)	0.999809/0.999966/0.999997	0.1463/0.0366/0.00246	77.3%/95.3%/97.7%
101 (post-hoc nested)	100	7.82 (7.8%)	21 (21.0%)	0.999705/0.999947/0.999994	0.2954/0.0312/0.00425	77.3%/96.1%/96.9%

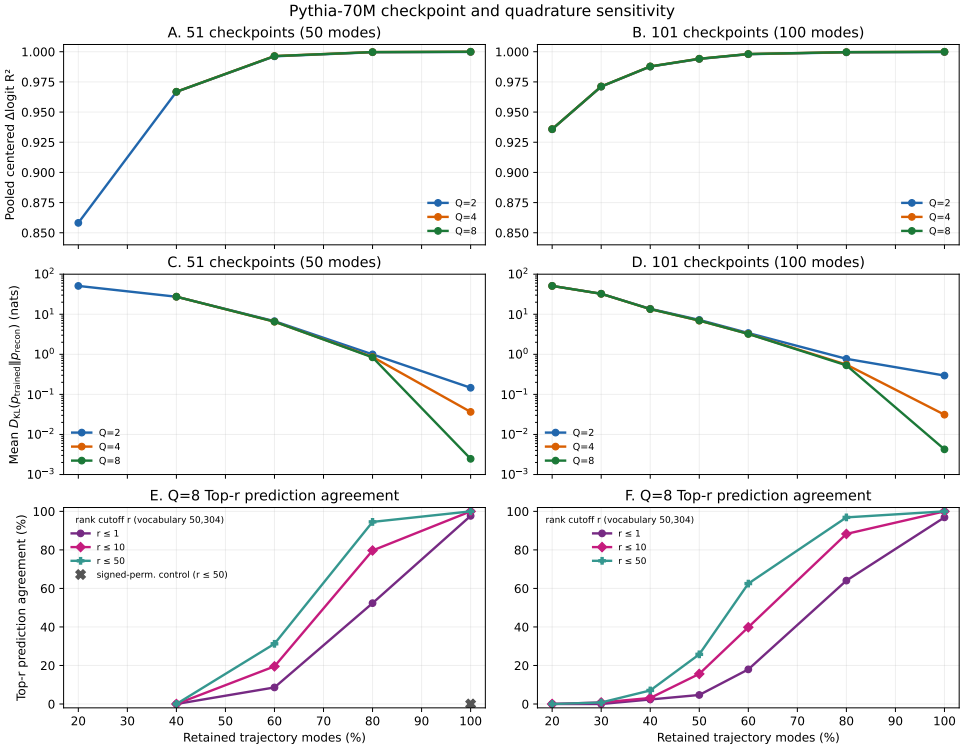

Figure 16: **Post-hoc Pythia checkpoint-resolution and quadrature sensitivity.** Columns hold the trajectory discretization fixed at 51 checkpoints (50 modes) or 101 nested checkpoints (100 modes). Rows report pooled centered-Δlogit R^2, mean full-vocabulary $D_{\mathrm{KL}}(p_{\mathrm{trained}} \| p_K)$ in nats on a log scale, and $Q = 8$ Top-r prediction agreement. Lower KL is better. Here K is the retained-mode count, while Top-r prediction agreement is top-r accuracy with the trained model's highest-logit token in place of the ground-truth label: the fraction of contexts where that token lies among the reconstruction's $r \in \{1, 10, 50\}$ highest-logit candidates. It is therefore neither true-token accuracy nor candidate-set intersection, and $r = 1$ is exactly the Top-1 prediction agreement reported elsewhere. Even $r = 50$ covers only 0.10% of the vocabulary. Panel E also marks the signed-permutation control at full retention, scored at the same $Q = 8$. It is defined only for the 51-checkpoint 50-mode path, so panel F has no counterpart. The 51-checkpoint $Q = 8$ sweep starts at $K = 20$ modes (40% retention), so panel E has no 20% or 30% points. Within a schedule, quadrature leaves the mode order fixed. The 101-checkpoint $Q = 4, 8$ endpoints use the exact checkpoint displacement, numerically equivalent to retaining all modes.

$R^2 = 0.999472/0.999645/0.999681$ and Top-1 prediction agreement $59.4\%/64.1\%/64.1\%$. The corresponding mean KL values are $0.773/0.554/0.531$ nats. At $Q = 8$, perplexity is 56.00 versus 35.04 for the trained model, while full-update KL and perplexity are 0.00425 nats and 34.94. Thus

Table 4: **Matched retained fractions across checkpoint schedules.** Two-node ($Q = 2$) held-out scores on the same 128 frozen Pile-test contexts, with modes ranked on the validation contexts. Retained fraction is the kept mode count divided by the available mode count, so 80% is 40 of 50 modes for the 51-checkpoint basis and 80 of 100 for the 101-checkpoint basis. The reported R^2 is the pooled centered-Δlogit score in Equation (16).

	pooled centered Δlogit R^2		Top-1 prediction agreement	
Retained	**51 ckpt**	**101 ckpt**	**51 ckpt**	**101 ckpt**
20%	0.8582	0.9357	0%	0%
40%	0.9668	0.9877	0%	1.6%
60%	0.9962	0.99795	10.9%	17.2%
80%	0.99958	0.99947	46.1%	59.4%
100%	0.99981	0.99970	77.3%	77.3%

the remaining fixed-schedule decision gap at this retained fraction is truncation-sensitive rather than a failure of quadrature convergence.

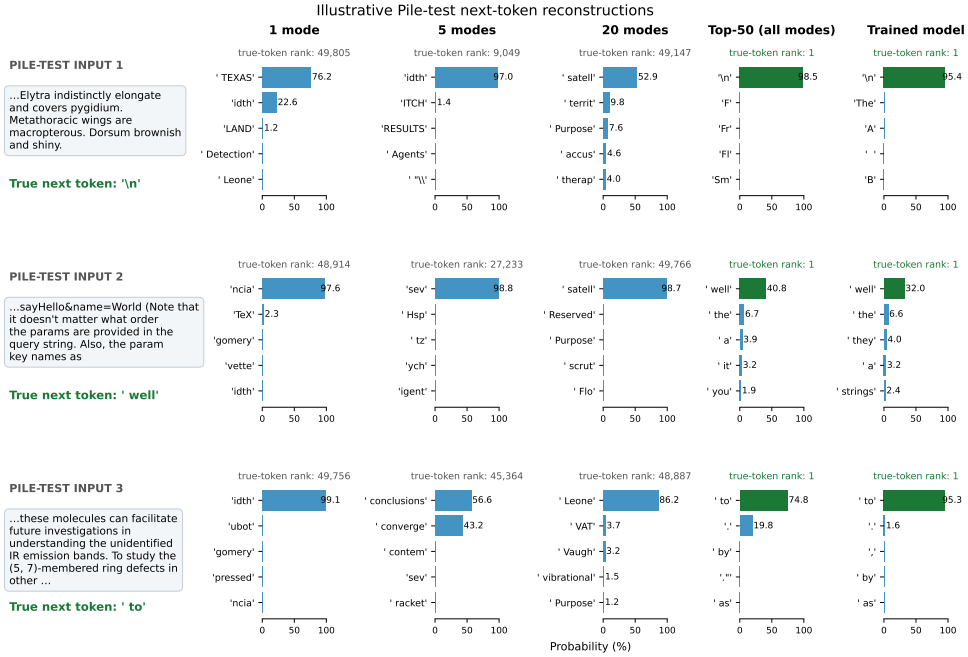

Figure 17: **Illustrative untouched-test next-token distributions across cumulative modal reconstructions.** Each row uses one Pile-test scoring context. Columns show cumulative 1-, 5-, 20-, and 50-mode reconstructions followed by the trained-model reference. Bars show the five highest-probability tokens, green marks the true next token, and each panel reports its rank. Examples were selected deterministically among contexts the trained model predicts correctly. The reported true-token rank is therefore also the rank of the trained model's predicted token and measures the same quantity that Figure 16E–F aggregates as Top-r prediction agreement, here for the locked two-node analysis rather than the eight-node sweep. These examples are explanatory rather than aggregate evidence.

Figure 17 illustrates why high pooled centered-Δlogit R^2 can coexist with decision errors. Under the locked two-point rule, the 20-mode reconstruction has zero Top-1 prediction agreement across the 128 scoring contexts, while the 50-mode along-path reconstruction reaches 77.3% and has perplexity 38.66. A near-perfect pooled centered-Δlogit R^2 therefore cannot certify decision recovery, because small errors near competing-token boundaries can change the highest-logit prediction. Figure 16E–F relaxes the rank cutoff, and Table 5 gives the corresponding values at 80% retention. Both full-update endpoints reach 100% by $r = 3$.

Table 5: **Top-r prediction agreement at 80% retention.** Eight-node ($Q = 8$) scores on the same 128 frozen Pile-test contexts, corresponding to Figure 16E–F. A context counts as agreeing when the trained model's highest-logit token lies among the reconstruction's r highest-logit candidates.

Rank cutoff r	1	3	5	10	20	50
51 checkpoints, 40/50 modes	52.3%	67.2%	71.9%	79.7%	89.1%	94.5%
101 checkpoints, 80/100 modes	64.1%	77.3%	84.4%	88.3%	91.4%	96.9%

Neither the signed-permutation control nor the 20-mode reconstruction ever places the trained model's token within the top 50 candidates on any of the 128 contexts, at either $Q = 2$ or $Q = 8$, whereas the locked 50-mode reconstruction reaches $77.3/94.5/100\%$ at $r = 1/3/10$. Eight-node quadrature moves the control only from -0.890 to -0.891 in pooled centered Δlogit R^2, confirming that scrambling parameter coordinates fails structurally rather than through integration error. This experiment supports compact reconstruction of centered logit-change variance for one public 70M-parameter trajectory. It does not establish decision-preserving model compression, mode monosemanticity, causal necessity, or scaling behavior.