PREDICTIVE RANK LANGUAGE MODELING: MEASURING REPRESENTATION CAPACITY IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models learn to predict future tokens by encoding information about their context. However, how the dimensionality and strength of predictive information interact with limited representation capacity remains unclear. To address this, we introduce *Predictive Rank Language Modeling* (PRLM), a controlled next-token learning environment with known predictive variables and a tunable distribution over future tokens. PRLM varies predictive rank, spectrum, and signal strength alongside residual width, while exposing only token sequences to the model. Experiments across 20 rank–width configurations reveal a gradual capacity frontier: increasing predictive rank reduces both predictive gain capture and global state decodability, whereas increasing residual width improves both. Matched controls further show that concentrating predictive energy raises gain capture from 27.2% to 82.6% at fixed rank and approximately matched Bayes KL. Under overload, weak global decoding nevertheless coexists with stable local effects and causally used predictive components. Sparse autoencoders likewise reconstruct activations accurately without reliably recovering the known concepts. Together, these findings show how predictive demand shapes learned representations and, using known ground truth, clarify what global decoding, local structure, sparse reconstruction, and causal use can establish about them.

1 INTRODUCTION

Modern autoregressive language models are trained by minimizing next-token cross-entropy (Brown et al., 2020). To reduce this loss, a transformer learns to encode effective information from the context in its residual stream (Elhage et al., 2021). The predictive contribution of such information varies across contextual factors: some strongly change the next-token distribution, some are redundant with others, and some matter only in particular contexts. This raises a basic question for language-model training and interpretability: how is predictive information allocated in model representations?

Prior work has addressed parts of this question through controlled learning settings and analyses of trained representations. Controlled studies relate feature statistics and width to loss (Liu et al., 2025) and data-generating dynamics to the geometry of learned belief states (Shai et al., 2024). Probes test which information is linearly accessible (Alain & Bengio, 2017), while sparse autoencoders seek interpretable features (Huben et al., 2024). SAEBench further shows that reconstruction quality alone does not reliably indicate downstream utility (Karvonen et al., 2025). However, it remains unclear how the data's predictive structure and the model's capacity jointly shape this allocation under next-token training.

To investigate this question, we introduce *Predictive Rank Language Modeling* (PRLM). A hidden state controls future-token probabilities (Figure 1a), but the model trains only on token sequences with ordinary causal language modeling; neither the state nor the Bayes-optimal distribution is provided as supervision. The known generator lets us vary *predictive rank*—the number of independent latent directions that change next-token log odds—as well as the spectrum and scale of those effects, while keeping the training interface fixed.

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

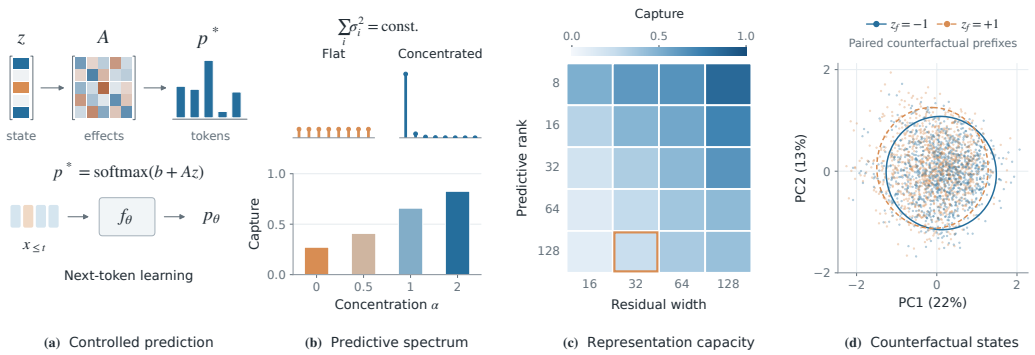

Figure 1: Predictive demand, representation capacity, and readout structure in PRLM. Full numerical results are reported in Table 1.

Our main result is a capacity frontier (Figure 1c). Across a 20-model grid, increasing predictive rank at fixed residual width reduces both the fraction of available predictive gain captured by the model and the amount of true predictive state recoverable by a single linear readout; increasing width at fixed rank improves both. Rank alone is insufficient: after matching overall Bayes KL, higher rank remains harder, while concentrating predictive strength into fewer directions at fixed rank greatly improves behavior (Figure 1b) and linear accessibility. Thus predictive dimensionality, spectral concentration, and loss-weighted signal strength jointly shape what next-token training learns and exposes.

PRLM also uses known ground truth to examine counterfactual representations (Figure 1d) and distinguish what common representation readouts can establish. Under predictive overload, global linear decoding can be weak while local effects remain consistent and predictive components remain causally used (Table 1C). Sparse autoencoders can reconstruct the same activations accurately while recovering few known predictive variables. These results make the interpretation of each readout concrete: global decodability measures joint-state access, local counterfactuals measure stable residual effects, ablations measure output use, and sparse reconstruction need not coincide with recovery of the ground-truth predictive variables.

In summary, this work makes three contributions. First, we introduce PRLM, a standard next-token learning environment in which the latent predictive state and Bayes-optimal distribution are known to the experimenter but never provided to the model (Section 3). Second, we establish how predictive rank, spectrum, and energy interact with residual width to determine representation capacity (Section 4.2). Third, using known predictive ground truth, we identify the evidential limits of common representation readouts: global probes can miss predictive information that is locally stable and causally used, while accurate sparse reconstruction need not recover the ground-truth predictive variables (Section 4.4).

2 Preliminaries

2.1 Autoregressive language models

An autoregressive language model maps a token prefix $x_{\leq t}$ to a distribution over the next token. In a decoder-only Transformer (Vaswani et al., 2017), the prefix is converted into a residual activation

$$h_t = f_\theta(x_{\leq t}) \in \mathbb{R}^{d_{\text{model}}}, \qquad p_\theta(x_{t+1} \mid x_{\leq t}) = \text{softmax}(W_U h_t)_{x_{t+1}}, \tag{1}$$

where d_{model} is the residual-stream width and W_U is the output map. Training minimizes next-token cross-entropy over observed sequences. Thus any contextual information used for prediction must be encoded in, transformed through, and read out from residual activations. This makes the residual stream the natural place to study representation capacity in a standard language-modeling objective.

2.2 PREDICTIVE INFORMATION

For a hidden variable to matter to next-token learning, it must change the conditional distribution of future tokens. Suppose a data-generating process has a hidden state z_t and a Bayes-optimal next-token distribution $p^*(\cdot \mid z_t)$. The loss-relevant content of z_t is therefore determined by the distribution it induces:

$$z_t \longmapsto p^*(\cdot \mid z_t). \tag{2}$$

Two states that induce the same distribution are predictively equivalent, even if their latent descriptions differ. Two states that induce different distributions create information that can reduce the autoregressive loss.

For a fixed prefix whose true next-token distribution is p^*, the next-token cross-entropy decomposes as

$$\mathbb{E}_{x\sim p^*}[-\log p_\theta(x \mid x_{\le t})] = H(p^*) + D_{\mathrm{KL}}(p^* \| p_\theta). \tag{3}$$

The entropy term is irreducible. The KL term is the reducible part of the next-token loss: it is small only when the model uses the prefix to approximate the true conditional distribution.

In a decoder-only model, reducing this KL term requires the prefix information to pass through the residual stream. Equation 1 computes logits from h_t, so any contextual factor that changes the model's next-token distribution must affect h_t or its readout through W_U. Representation analysis therefore tries to inspect this interface: probes ask what can be decoded from activations, interventions ask whether an activation component changes model outputs, and sparse decompositions ask whether activations can be described by reusable features.

The missing link is predictive ground truth. In natural corpora, we usually observe $x_{\le t}$ and h_t, but not the hidden state z_t that generated the prefix or the Bayes distribution $p^*(\cdot \mid z_t)$ that defines the KL term in Equation 3. As a result, an activation readout can reveal structure without telling us whether that structure is a loss-relevant predictive variable, a surface correlate, or a property of the readout method. PRLM keeps the training interface in Equation 1 unchanged while making the predictive map in Equation 2 known to the experimenter.

3 PREDICTIVE RANK LANGUAGE MODELING

3.1 TASK CONSTRUCTION

Predictive Rank Language Modeling (PRLM) is a family of synthetic token distributions with a hidden state that controls future-token probabilities. Each sequence contains ordinary state statements and payload tokens. State statements initialize or update k active variables among F available variables, producing a state

$$z_t \in \{0, \pm a_k\}^F, \qquad \|z_t\|_0 = k. \tag{4}$$

By default $a_k = 1$. In occupancy controls, we set $a_k = \sqrt{k_0/k}$ for a reference occupancy k_0, so that the state energy $\|z_t\|_2^2 = k a_k^2$ is held fixed while the number of active variables changes. After a statement, the generator samples payload tokens from

$$p^*(y \mid z_t) = \mathrm{softmax}(b + Az_t)_y, \tag{5}$$

where $A \in \mathbb{R}^{M\times F}$ and $b \in \mathbb{R}^M$ define a distribution over M payload types. The effect matrix is dense: each state variable changes many payload probabilities, and each payload probability depends on multiple variables. Thus the model cannot solve the task by mapping one latent variable to one output token.

The training interface is intentionally ordinary. The model receives only token IDs for statements, separators, and sampled payloads, and is trained with the causal language-modeling loss

$$\mathcal{L}(\theta) = -\sum_t \log p_\theta(x_{t+1} \mid x_{\le t}). \tag{6}$$

The latent state z_t, the matrix A, and the Bayes distribution $p^*(\cdot \mid z_t)$ are not provided to the model and are not used as auxiliary supervision. They are retained by the experimenter for data auditing, checkpoint selection, and post-hoc analysis. Appendix B gives the matrix construction and paired symbolic/language-like examples (Table 3).

3.2 PREDICTIVE RANK, SPECTRUM, AND STATE

PRLM separates the number of latent variables from the number of independent ways they affect prediction. Since softmax probabilities are invariant to a shared logit shift, we express the state's predictive effect as log odds relative to a reference payload:

$$\log \frac{p^*(y \mid z)}{p^*(0 \mid z)} = (b_y - b_0) + (A_{y,:} - A_{0,:})z. \tag{7}$$

We therefore define

$$B_{y,:} = A_{y,:} - A_{0,:}, \quad y = 1, \ldots, M-1, \qquad d_{\text{pred}} = \operatorname{rank}(B). \tag{8}$$

The *predictive rank* d_{pred} counts independent latent directions that can change the true next-token distribution. It can be varied while holding the surface vocabulary, the number of available variables, and the training objective fixed.

The spectrum specifies how predictive value is distributed across the rank-defined directions. We construct log-odds effect matrices by choosing orthonormal factors and setting

$$B = U \operatorname{diag}(\sigma_1, \ldots, \sigma_{d_{\text{pred}}}) V^\top, \qquad \sigma_i = c_\alpha i^{-\alpha}, \qquad \sum_i \sigma_i^2 = \text{constant}. \tag{9}$$

The exponent α controls spectral concentration, while the normalization keeps Frobenius energy fixed as rank or spectrum changes. For the thin SVD $B = U\Sigma V^\top$, we use the effect-weighted predictive coordinates

$$\widetilde{q} = \Sigma V^\top z \tag{10}$$

as the main target for global representation probes, because these coordinates weight state directions by their effect on log odds. We summarize spectral concentration by the participation ratio of the squared singular values (Litwin-Kumar et al., 2017), which we call the energy effective rank:

$$r_{\text{eff}} = \frac{(\sum_i \sigma_i^2)^2}{\sum_i \sigma_i^4}. \tag{11}$$

Together, d_{pred}, the predictive spectrum, the active-state distribution, and residual width define the capacity pressure studied in the experiments. The linear readout $\ell_t = W_U h_t$ limits the model's log-odds variation to at most d_{model} dimensions, connecting this capacity constraint to the softmax bottleneck (Yang et al., 2018). Appendix B.3 derives the predictive-target relationship.

3.3 EVALUATION

PRLM makes Equation 3 computable after training. Averaging over held-out payload positions gives the model's residual predictive error,

$$D_{\text{model}} = \mathbb{E}_t D_{\text{KL}}(p^*(\cdot \mid z_t) \| p_\theta(\cdot \mid x_{\le t})) \,. \tag{12}$$

where smaller values mean that p_θ is closer to the true conditional distribution. Because payload tokens are sampled from p^*, exact-token accuracy has an irreducible Bayes error. We therefore normalize the residual KL by the Bayes gain over a uniform payload predictor U. For held-out payload positions $\mathcal{P}$,

$$\text{Capture} = 1 - \frac{\mathbb{E}_{t \in \mathcal{P}}[D_{\text{KL}}(p_t^* \| p_{\theta,t})]}{\mathbb{E}_{t \in \mathcal{P}}[D_{\text{KL}}(p_t^* \| U)]}, \tag{13}$$

where $p_t^* = p^*(\cdot \mid z_t)$ and $p_{\theta,t} = p_\theta(\cdot \mid x_{\le t})$. Capture is one for Bayes-optimal prediction and zero for the uniform baseline. Since later payloads in the same turn can reveal information about the current distribution, we also compute the same quantity on only the first payload after each state statement:

$$\text{Capture}_{\text{first}} = 1 - \frac{\mathbb{E}_{t \in \mathcal{P}_{\text{first}}}[D_{\text{KL}}(p_t^* \| p_{\theta,t})]}{\mathbb{E}_{t \in \mathcal{P}_{\text{first}}}[D_{\text{KL}}(p_t^* \| U)]}, \tag{14}$$

where $\mathcal{P}_{\text{first}} \subset \mathcal{P}$ contains those first-payload positions.

Global probes measure predictive-state accessibility in the residual stream. We use $\widetilde{q} = \Sigma V^\top z_t$ as the probe target, weighting state directions by their effect on log odds. We fit ridge probes (Hoerl & Kennard, 1970),

$$(W^*, a^*) = \arg\min_{W,a} \sum_{t \in \mathcal{T}_{\text{fit}}} \|W h_t + a - \widetilde{q}_t\|_2^2 + \lambda \|W\|_F^2, \tag{15}$$

with λ selected on validation data, and report held-out $R^2_{\widetilde{q}}$ as a global access metric. The intercept a is unpenalized.

Finally, the same ground truth lets us assign different readouts to different claims about the representation. Local structure is measured with paired counterfactual prefixes that differ only in one latent sign,

$$\Delta h_f(c) = \frac{h(z_f = +1, c) - h(z_f = -1, c)}{2}, \tag{16}$$

where c denotes the other active variables and surface tokens. Per-example predictive gain is

$$g_i = D_{\mathrm{KL}}(p^*_i \| U) - D_{\mathrm{KL}}(p^*_i \| p_{\theta,i}), \tag{17}$$

and, following activation-ablation analyses (Wang et al., 2023), test causal use by removing fitted predictive components,

$$h_t^{(-j)} = h_t - (\widetilde{q}_{t,j} - \overline{q}_j) v_j, \tag{18}$$

where v_j is the ridge-fitted residual direction of predictive coordinate j. We compare payload-conditional KL damage against norm-matched random directions on the analyzed activation sample; Appendix G gives the fitting procedure. Following ground-truth dictionary evaluation (Karvonen et al., 2024), we evaluate sparse autoencoders by reconstruction fidelity, concept matching to signed latent variables, and ablation damage. These measurements distinguish joint-state access, stable local effects, output use, and recovery of the ground-truth predictive variables.

Table 1: **Numerical results across capacity, predictive load, overload, and readout diagnostics.** Panel A reports the complete rank–width grid. Panels B–D collect the controls and representation measurements discussed in the experiments. Capture (Cap.) and gain coverage are reported in percent; $D_U = D_{\mathrm{uniform}}$. In panel B, Fixed and Mix denote fixed-energy and mixture Capture. Both probes in panel C target $\widetilde{q}$; its gain coverage uses full-vocabulary probabilities (Table 16). In panel D, Matched/Wrong denotes fitted latent-direction removals for scale rows and SAE-unit removals for rank rows.

A. Capacity frontier: rank–width grid

d_{pred}	Capture (%) d_{model}				Linear $R^2_{\widetilde{q}}$ d_{model}			
	16	32	64	128	16	32	64	128
8	50.6	61.6	64.1	85.2	0.369	0.523	0.550	0.841
16	30.5	47.3	54.4	71.8	0.139	0.293	0.397	0.664
32	19.6	34.1	45.4	63.6	0.042	0.102	0.209	0.505
64	12.7	26.5	38.1	44.8	0.008	0.028	0.063	0.130
128	10.9	24.1	35.4	41.3	0.002	0.006	0.015	0.038

B. Effective predictive load controls at $d_{\mathrm{model}} = 32$

Rank

Rank	r_{eff}	D_U	Cap.	R^2
8	8.00	2.523	62.9	0.527
16	16.00	2.515	47.6	0.292
32	32.00	2.497	32.3	0.098
64	64.00	2.501	24.8	0.029
128	128.00	2.513	21.7	0.005

Spectrum

α	r_{eff}	D_U	Cap.	R^2
0	64.00	2.495	27.2	0.028
0.5	13.81	2.524	40.9	0.253
1	2.45	2.546	65.9	0.584
2	1.17	2.563	82.6	0.708

Occupancy: Capture (%)

k	Raw	Fixed	Mix
2	19.16	38.79	30.26
4	30.71	30.71	27.17
8	45.07	30.48	27.25
16	59.20	32.57	30.74

C. Overload representation diagnostics

Config.	Capture	Linear R^2	MLP R^2	Local resultant	Gain > 0	Fitted ΔKL	Random ΔKL	Fitted/random
r32/d16	19.57	0.042	0.067	0.991	77.24	0.00875	0.00077	11.4
r64/d16	12.71	0.008	0.018	0.982	71.22	0.00210	0.00018	11.7
r64/d32	26.50	0.028	0.048	0.992	79.53	0.00456	0.00053	8.6
r128/d16	10.92	0.002	0.005	0.984	70.07	0.00065	0.00011	5.9
r128/d32	24.13	0.006	0.013	0.956	77.34	0.00143	0.00021	6.8
r128/d64	35.39	0.015	0.023	0.955	85.31	0.00163	0.00032	5.1

D. Readout calibration under known predictive variables

Signal scale

Scale	Cap.	Latent R^2	Logit align.	CF align.	SAE corr.	Direction ΔKL matched/wrong
5	2.6	0.043	0.281	0.251	0.322	0.0046/0.0018
7	8.4	0.056	0.444	0.382	0.346	0.0174/0.0052
10	23.8	0.125	0.724	0.435	0.463	0.0822/0.0180
14	44.1	0.260	0.893	0.553	0.590	0.2145/0.0563
18	59.3	0.391	0.902	0.686	0.698	0.3257/0.1275

Predictive rank

Rank	SAE FVU	Concept corr.	SAE-unit ΔKL matched/wrong
8	0.0057	0.084	0.0174/0.0020
16	0.0082	0.083	0.0163/0.0016
32	0.0089	0.074	0.0126/0.0010
64	0.0097	0.074	0.0088/0.0011
128	0.0091	0.076	0.0072/0.0009

4 EXPERIMENTS

In this section, we conduct experiments to address the following research questions:

- **RQ1:** How does predictive rank affect learning at different residual widths? Does the same capacity trend hold for unseen combinations of predictive variables?
- **RQ2:** Is predictive rank sufficient to explain task difficulty? How do the spectrum, occupancy, and predictive effect scale alter the learned prediction?
- **RQ3:** What representation structure remains under predictive overload? When joint-state decoding is weak, do paired counterfactuals reveal stable and causally used components?
- **RQ4:** What do representation readouts establish under known predictive ground truth? Does accurate sparse reconstruction imply concept recovery or selective causal use?

4.1 EXPERIMENTAL SETUP

We briefly describe the models, datasets, and evaluation metrics below. Full training and split settings appear in Appendix A; generator construction and metric conventions appear in Appendix B.

Models and comparison conditions. We train four-block decoder-only Transformers from scratch with ordinary causal language-modeling loss. The main grid crosses predictive ranks $\{8, 16, 32, 64, 128\}$ with residual widths $\{16, 32, 64, 128\}$, holding the latent pool, occupancy, effect-matrix energy, dataset size, and training budget fixed. Additional experiments match Bayes KL, vary spectral concentration and occupancy, or change the predictive effect scale. The main grid uses one training seed; targeted replications evaluate seeds, training duration, architecture, and surface templates.

Datasets and evaluation metrics. Each model receives 12,000 training sequences, with 1,500 sequences each for validation, IID testing, and compositional testing. The compositional split holds out complete active-variable sets. Validation KL selects checkpoints. Following Section 3, we evaluate Capture, first-payload Capture, global predictive-state $R^2_{\tilde{q}}$, paired counterfactuals, controlled ablations, and SAE concept recovery.

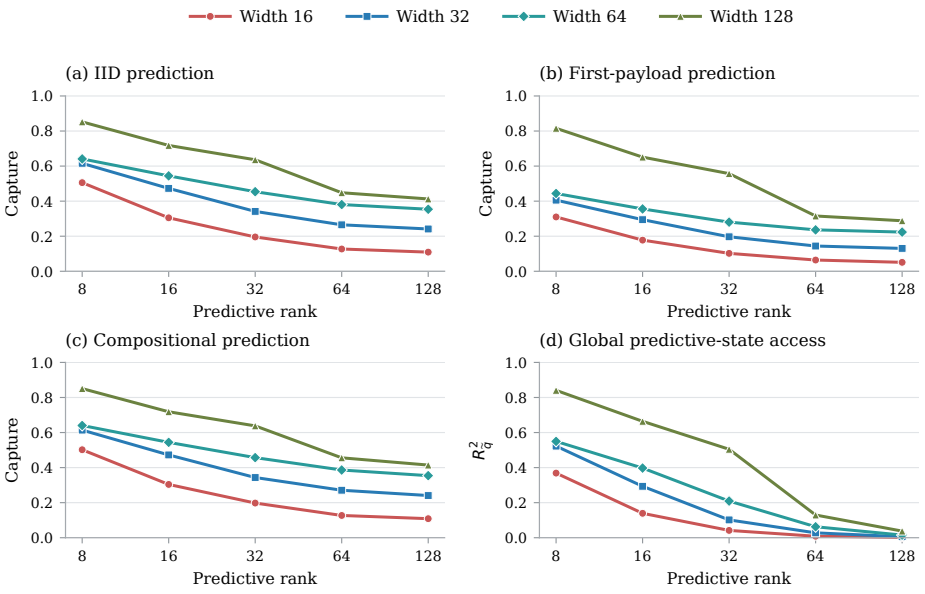

Figure 2: Prediction and global access across predictive ranks and residual widths. Panels show IID Capture, first-payload Capture, compositional Capture, and linear predictive-state $R^2_{\tilde{q}}$. Each curve fixes residual width; each point is one trained model.

4.2 Prediction and Generalization Across Capacity (RQ1)

To evaluate the effect of predictive demand and residual capacity, we train models on the 20 rank–width configurations. Table 1A reports IID Capture and global predictive-state access, while Figure 2 compares these measurements with first-payload and compositional prediction. Appendix C provides the complete grid and individual replications; Appendix A.2 specifies the held-out combinations. Based on these results, we make the following observations:

- **Obs 1: Prediction follows a gradual rank–width capacity trend.** At width 32, increasing rank from 8 to 128 lowers Capture from 61.6% to 24.1%; at rank 32, increasing width from 16 to 128 raises it from 19.6% to 63.6%. Performance changes gradually around equal rank and width.
- **Obs 2: The capacity trend generalizes beyond observed active-variable sets.** Compositional and IID Capture differ by at most 0.78 percentage points across all 20 models. However, useful prediction need not imply strong joint-state access: the rank-128, width-64 model retains 35.4% Capture with only 0.015 global $R^2_{\widetilde{q}}$.

The rank ordering also persists under seed, training-budget, architecture, and language-like surface controls (Appendix C.2; Tables 6 and 8).

4.3 Predictive Load Controls (RQ2)

To distinguish predictive dimensionality from signal strength, we perform four controlled comparisons: (1) match $D_{\text{uniform}} \approx 2.5$ while varying rank; (2) fix rank at 64 and match Bayes KL while changing the singular-value spectrum; (3) compare occupancy before and after energy normalization; and (4) vary the predictive effect scale under validation-based stopping. Figure 3 reports these comparisons. Panels (a)–(c) use width 32; panel (d) uses width 128 with 64 latent variables. Table 1B summarizes the results; Appendix D gives calibration and occupancy cross-evaluations, and Appendix C.2 gives robustness checks. We observe that:

- **Obs 3: Rank and spectral concentration affect learning even at matched Bayes gain.** Capture falls from 62.9% to 21.7% between ranks 8 and 128 after KL matching. At fixed rank 64, reducing energy effective rank from 64.0 to 1.17 instead raises Capture from 27.2% to 82.6%. These controls identify spectral concentration as an additional determinant of predictive demand.
- **Obs 4: Occupancy and predictive value alter the available learning signal.** Raw occupancy increases Capture from 19.2% to 59.2%, whereas an energy-normalized mixture-trained model remains within 27.2%–30.7% across occupancies. Separately, increasing effect scale raises Capture from 2.6% to 59.3% and latent-state R^2 from 0.043 to 0.391.

4.4 Hidden Representations Under Overload (RQ3)

We next examine six models whose predictive rank exceeds their residual width. To visualize the effect of changing a predictive variable, we (1) construct 4,096 paired counterfactual prefix pairs per model, differing in one latent sign; (2) extract the final residual activation from each prefix; and (3) fit a common two-dimensional PCA projection to both signs within each model and display 1,000 paired examples. Figure 4 compares three models with predictive rank twice their residual width; Appendix F includes all six models. Table 1C quantifies global decoding, direction consistency, and ablation damage in the original activation space. Appendices E, F, and G give the probe controls, gain distributions, and intervention protocol. These comparisons yield two observations:

- **Obs 5: Pooled distributions and local effects reveal different structure.** Sign separation is more visible in the rank-64/width-32 and rank-128/width-64 projections than in rank-32/width-16, despite all three sharing a rank-to-width ratio of two (Figure 4). Nevertheless, within-variable counterfactual directions remain consistent across all six models, with mean resultants of 0.955–0.992. The full-dimensional resultants reveal consistent effects within individual variables.
- **Obs 6: Weak global access coexists with broadly useful, causally relevant components.** The tested MLP probes reach at most 0.067 $R^2_{\widetilde{q}}$, improving over linear probes by at most 0.025. Yet 70.1%–85.3% of held-out positions have positive gain, and fitted-component removals cause 5.1–11.7 times the KL damage of norm-matched random removals. These components retain measurable output relevance under weak joint-state recovery.

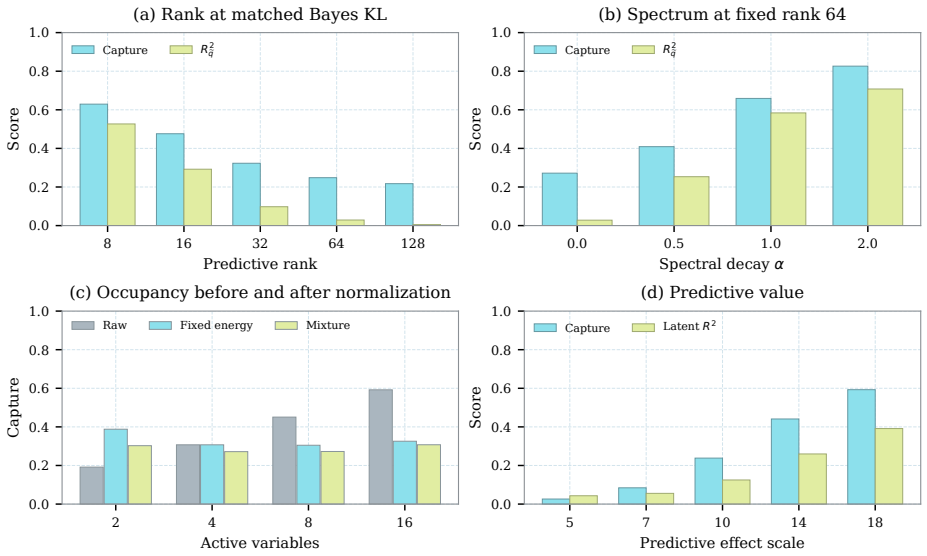

Figure 3: Controlled comparisons of rank, spectrum, occupancy, and predictive value. Bars show the original metric values. In (c), *Mixture* is one energy-normalized model evaluated at each occupancy. In (d), latent R^2 targets z; (a)–(b) target the effect-weighted coordinate $\widetilde{q}$.

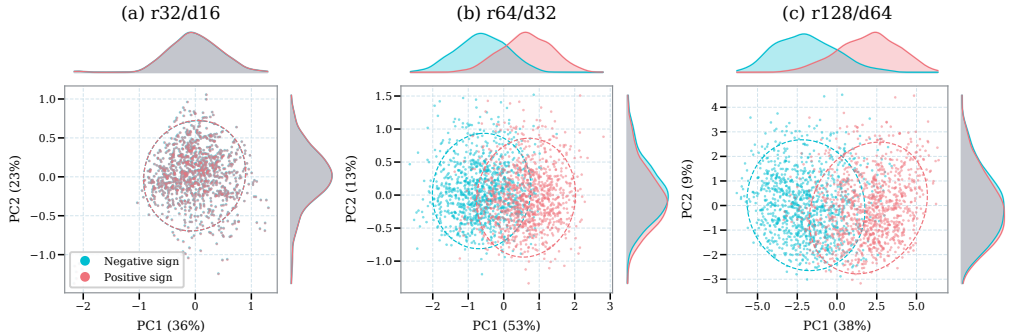

Figure 4: Counterfactual representations at rank/width 32/16, 64/32, and 128/64, all with rank-to-width ratio two. Panels show 1,000 paired contexts, marginal densities, and two-standard-deviation covariance ellipses. Each model has its own jointly fitted PCA coordinates; axes report explained variance. All six models appear in Figure 8.

Appendix F.3 compares these models with fixed SVD truncation on the same evaluation sample.

4.5 Calibrating Representation Readouts (RQ4)

Finally, we evaluate sparse readouts against the known predictive variables. Figure 5 compares matched- and wrong-unit SAE ablations; Table 1D reports reconstruction and concept recovery across predictive ranks and signal scales. Appendix H gives SAE selection, full-grid results, and exact-amplitude controls; Appendix I tests pairwise context modulation.

Obs 7: Accurate reconstruction and selective causal effects do not guarantee concept recovery. Stronger predictive effects improve several readouts together, but the width-32 capacity slice retains below 1% reconstruction FVU while mean concept correlation remains 0.074–0.084. None of its 1,280 train-selected concept matches exceeds 0.25 test correlation. Matched-unit ablations nevertheless cause more KL damage than wrong-unit and exact-amplitude controls. Thus sparse features can reconstruct activations and affect predictions while remaining weakly aligned with the known predictive variables.

5 Related Work

Superposition and Representation Capacity. This line studies how models encode more features than their representation dimensions would conventionally allow. Toy models of superposition (Elhage et al., 2022) show how sparsity and feature importance shape shared representations. Liu et al. (2025) connect superposition to loss scaling with model width, while temporal superposition (Sharma et al., 2026) examines feature organization under recurrent memory demands. The softmax bottleneck (Yang et al., 2018) instead constrains the rank of representable output distributions. PRLM connects these capacity perspectives by controlling the rank and spectrum of the data-generating log-odds effects.

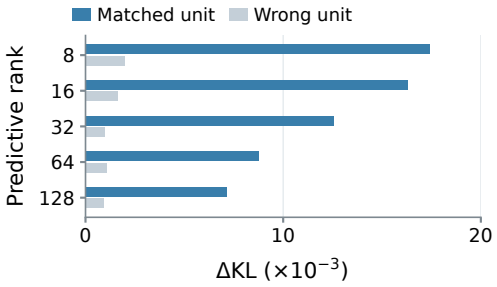

Figure 5: SAE ablation damage for matched and wrong units at residual width 32. Full results appear in Table 1D.

Controlled Language Modeling. Controlled sequence tasks expose underlying states or computations for studying learned representations. Othello models (Li et al., 2023) learn board-state representations from move sequences, while hidden Markov processes (Shai et al., 2024) connect transformer activations to known belief-state geometry. Grid-walker models (Brenner et al., 2026) further examine how predictive statistics shape world representations. Algorithmic tasks (Nanda et al., 2023) and compiled transformers (Lindner et al., 2023) provide complementary settings with accessible computational structure. PRLM directly controls the rank, spectrum, and strength of predictive effects under ordinary next-token training.

Analyzing Language-Model Representations. Representation analyses examine accessibility, structure, and use through different tools. Structural probes (Hewitt & Manning, 2019) recover syntactic geometry, tuned lenses (Belrose et al., 2025) recover intermediate predictions, and circuit tracing (Ameisen et al., 2025) constructs attribution graphs in replacement models. For sparse autoencoders, board-game evaluations (Karvonen et al., 2024) provide concept ground truth, while SAEBench (Karvonen et al., 2025) distinguishes reconstruction from downstream utility. Recent studies examine dictionary identifiability (Nelson et al., 2026) and benchmark reliability (Chanin, 2026). PRLM complements these efforts by evaluating access, structure, reconstruction, and causal use against the same known predictive variables and distributions.

6 Limitations & Future Discussion

PRLM studies small Transformers on synthetic sequences with known linear logit effects. The main grid uses one training seed, supplemented by targeted replications; the language-like condition tests alternative surface templates. Future work should test broader model scales, additional seeds, and nonlinear predictive mechanisms. Extending PRLM to natural corpora also requires estimating predictive variables or target distributions without generator ground truth, and testing whether the observed distinctions between global access, local structure, and causal use persist in pretrained models.

7 Conclusion

We introduced PRLM to control predictive rank, spectrum, and signal strength under ordinary next-token training, with known ground truth for representation analysis. Across 20 rank–width configurations and targeted controls, we find a gradual capacity frontier; concentrating predictive energy raises gain capture from 27.2% to 82.6% at fixed rank and approximately matched Bayes KL. Weak global decoding can coexist with predictive use, while accurate sparse reconstruction can coexist with weak recovery of the ground-truth predictive variables.

AI USE STATEMENT

Generative AI tools were used to assist with code review, analysis, and language editing. All reported quantities were checked against saved experimental artifacts, and the authors reviewed the resulting arguments and text. The authors take responsibility for the final content, claims, and artifacts.

ETHICS STATEMENT

The experiments use procedurally generated sequences and synthetic latent states, with no human-subject data or personal records. PRLM is intended to support controlled research on representation capacity and the interpretation of model behavior. Its results do not establish the safety, fairness, or reliability of deployed language models. In particular, successful reconstruction or selective causal effects should not be treated as evidence of complete understanding or as a safety guarantee. Applying these findings to natural-language systems requires additional validation beyond the controlled setting studied here.

REPRODUCIBILITY STATEMENT

The appendix documents the experimental setup and analyses in detail. Appendix A specifies model, data, optimization, split, and analysis settings; the other sections define the generator and evaluation procedures and report the complete capacity grid and targeted controls.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations, Workshop Track*, 2017. URL https://openreview.net/forum?id=HJ4-rAVtl.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit Tracing: Revealing Computational Graphs in Language Models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens, 2025. URL https://arxiv.org/abs/2303.08112. arXiv:2303.08112.

Sasha Brenner, Thomas R. Knösche, and Nico Scherf. Predictive Statistics Shape Emergent World Representations of Grid Walkers, 2026. URL https://arxiv.org/abs/2603.16689. arXiv:2603.16689.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

David Chanin. Are Sparse Autoencoder Benchmarks Reliable?, 2026. URL https://arxiv.org/abs/2605.18229. arXiv:2605.18229.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal

Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy Models of Superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

John Hewitt and Christopher D. Manning. A Structural Probe for Finding Syntax in Word Representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring Progress in Dictionary Learning for Language Model Interpretability with Board Game Models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 83091–83118. Curran Associates, Inc., 2024. doi: 10.52202/079017-2644. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9736acf007760cc2b47948ae3cf06274-Paper-Conference.pdf.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart Mcdougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders in Language Model Interpretability. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29223–29264. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

David Lindner, Janos Kramar, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. Tracr: Compiled Transformers as a Laboratory for Interpretability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 37876–37899. Curran Associates, Inc., 2023. doi: 10.52202/075280-1649. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/771155abaae744e08576f1f3b4b7ac0d-Paper-Conference.pdf.

Ashok Litwin-Kumar, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and L.F. Abbott. Optimal Degrees of Synaptic Connectivity. *Neuron*, 93(5):1153–1164.e7, 2017. doi: 10.1016/j.neuron.2017.01.030.

Yizhou Liu, Ziming Liu, and Jeff Gore. Superposition Yields Robust Neural Scaling. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 159269–159305. Curran Associates, Inc., 2025. doi: 10.52202/085713-5320. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e97ac22927560eb2de6b658498cbc575-Paper-Conference.pdf.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Walter Nelson, Theofanis Karaletsos, and Francesco Locatello. Toward Identifiable Sparse Autoencoders. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=miLK9YcxtA.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers Represent Belief State Geometry in their Residual Stream. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 75012–75034. Curran Associates, Inc., 2024. doi: 10.52202/079017-2387. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8936fa1691764912d9519e1b5673ea66-Paper-Conference.pdf.

Pratyaksh Sharma, Alexandra M Proca, Lucas Prieto, and Pedro Mediano. Temporal superposition and feature geometry of RNNs under memory demands. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 118125–118166, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/bfcb583d225b1db8d3ca2331f6785774-Paper-Conference.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkwZSG-CZ.

APPENDIX

Table of Contents

A Experimental Details . . . 14
- A.1 Model and training protocol . . . 14
- A.2 Data separation and analysis sampling . . . 14

B Generator Construction and Metric Definitions . . . 15
- B.1 Serialized episodes and state updates . . . 15
- B.2 Constructing rank and spectrum . . . 15
- B.3 Why the predictive target differs from the latent state . . . 16
- B.4 Full-vocabulary KL and conditional payload KL . . . 16

C Complete Capacity Grid and Robustness Controls . . . 16
- C.1 Rank–width grid and compositional prediction . . . 16
- C.2 Optimization, seeds, architecture, and surface controls . . . 16
- C.3 Language-like surface comparison . . . 18
- C.4 Training dynamics . . . 19

D Predictive Load, Spectrum, and Occupancy . . . 19
- D.1 Matching Bayes gain . . . 19
- D.2 Fixed-occupancy and mixture-trained models . . . 21

E Global Representation Readouts . . . 21
- E.1 Linear and nonlinear probes . . . 21
- E.2 Score definition and negative controls . . . 22
- E.3 Signal-scale probe comparison . . . 22

F Local Counterfactuals and Gain Distributions . . . 24
- F.1 Paired prefixes and within-variable consistency . . . 24
- F.2 Distribution of predictive gains . . . 24
- F.3 Fixed SVD truncation reference . . . 26

G Causal Interventions and Geometry Checks . . . 26
- G.1 Fitted predictive-component removal . . . 26
- G.2 Validity of global geometry estimates . . . 27

H Sparse Autoencoder Analysis . . . 28
- H.1 SAE fitting . . . 28
- H.2 Reconstruction and signed-concept matching . . . 28
- H.3 Ablation controls and their interpretation . . . 29

I Pairwise Context Modulation . . . 30

A Experimental Details

The supplementary material follows the experimental questions in the main text. Appendix B specifies the generator and predictive targets; Appendix C reports the capacity grid and replications; Appendix D details the matched-load controls. Appendices E–G describe global readouts, local counterfactuals, and causal interventions. Appendices H and I give the sparse and pairwise analyses. Throughout, a configuration denotes a trained model, whereas an analysis row denotes a payload position or a counterfactual example. These are not interchangeable units of replication.

A.1 Model and training protocol

Models are trained from scratch with a next-token cross-entropy objective over all non-padding tokens. They are not trained to predict the latent states, match generator probabilities, or optimize a representation metric. Each decoder block applies pre-normalized causal self-attention followed by a pre-normalized GELU MLP, with residual connections around both sublayers. Token embeddings and learned absolute position embeddings are added at the input. A final LayerNorm precedes the bias-free language-model head. The token embedding and output head are untied except in the explicit tying control.

Table 2: Shared settings for the final rank–width experiment. Family-specific overrides are described in the corresponding appendix sections.

Setting	Value
Residual widths / attention heads	16/2, 32/4, 64/8, 128/8
Decoder blocks / MLP width	4 / $4d_{\text{model}}$
Dropout / maximum serialized length	0 / 384 tokens
Training / validation / IID / compositional sequences	12,000 / 1,500 / 1,500 / 1,500
Optimizer / learning rate / weight decay	AdamW / 2×10^{-4} / 0.01
Training batch / evaluation batch	64 / 128 sequences
Epoch budget / warmup / gradient clipping	72 / 500 steps / 1.0
Main training seed / language seed	101 / 9101
Checkpoint criterion	Minimum validation payload KL

The learning rate uses linear warmup followed by cosine decay to one tenth of its peak over the configured budget. The longer-training control extends the budget from 72 to 144 epochs and adjusts the schedule accordingly. In the signal-scale series, scale 5 stops at epoch 30 under the predictive-gain stopping rule; scales 7, 10, 14, and 18 complete 72 epochs.

A.2 Data separation and analysis sampling

Training, validation, and IID test sequences are generated from the same permitted pool of active-variable sets using separate samples. Compositional evaluation uses a disjoint pool of complete active sets. A deterministic partition assigns each complete active set to one of ten buckets. Buckets 0–7 are permitted for training, validation, and IID testing; buckets 8–9 are reserved for compositional testing. Active-variable replacement is disabled, so an episode cannot cross this partition through a later update.

For example, the distinction is between complete sets such as $\{i, j, k, \ell\}$, not between new token types or previously unseen individual variables. Each individual variable and sign remains part of the same generator vocabulary on both sides of the partition. This tests recombination under a shared predictive mechanism. It does not test a new effect matrix, unfamiliar variable semantics, or generalization to natural text.

The ordinary activation analysis permits up to 2,048 sequences per split; the extended predictive-target analysis permits up to 1,024 sequences and 16,384 payload rows per split. Its linear, quadratic, and MLP comparisons share the collected rows. SAE analysis uses first-payload activations, whereas the extended global analysis includes payload positions across ordinals.

B Generator Construction and Metric Definitions

B.1 Serialized episodes and state updates

The main experiment has 128 available variables, four active variables, 160 payload token types, 16 turns per episode, and four payload samples per turn. An episode starts with reset and initialization statements, followed by turn markers, a speaker token, a state update or a no-update token, a link token, and the payload samples. Variable addresses contain a group and a slot, with 16 slots per group. Four alternative statement templates express the same underlying operation; four speaker identities are sampled independently of the predictive state.

Initialization chooses an active set permitted by the split and assigns independent signs. On each turn, an active variable is selected and its sign flipped with probability 0.7; otherwise a no-update marker is emitted. There are no swaps in the final compositional experiments. State changes become effective at the last token of the statement, before the subsequent payloads are sampled. Conditional on that state, the four payloads are independent draws from Equation 5; the model nevertheless observes earlier sampled payloads when predicting later ones.

The language-like surface condition changes the statement serialization to templates such as "entity_0 attribute_1 has value positive". These are discrete generator tokens, not text passed through a pretrained tokenizer. The latent update process and payload mechanism are unchanged. This condition isolates sensitivity to surface serialization, not the ability to interpret unrestricted natural language.

Table 3: Equivalent statement templates for variable 1 (group 0, slot 1). The first four rows assign the positive initial value using alternative templates; the last row reports a subsequent sign flip. Each space separates generator tokens. These are statement fragments within a complete episode.

Symbolic surface	Language-like surface
`<INIT> <GROUP_0> <SLOT_1> <POS>`	`initially entity_0 attribute_1 is positive`
`<POS> <AT> <GROUP_0> <SLOT_1>`	`entity_0 attribute_1 has value positive`
`<SPK_0> <DECLARES> <GROUP_0> <SLOT_1> <POS>`	`speaker_0 says entity_0 attribute_1 is positive`
`<GROUP_0> <SLOT_1> <BECOMES> <POS>`	`set entity_0 attribute_1 to positive`
`<REPORT> <GROUP_0> <SLOT_1> <NEG>`	`reports entity_0 attribute_1 is negative`

B.2 Constructing rank and spectrum

The implementation fixes the first payload row of A to zero and uses zero bias, making the remaining rows the reference-token log-odds matrix B. Gaussian matrices are orthogonalized by QR decomposition to construct left and right factors. For F_p predictive variables and requested rank r, the nonzero singular values before the global scale are

$$s_j = \frac{\sqrt{F_p}\, j^{-\alpha}}{\left(\sum_{\ell=1}^{r} \ell^{-2\alpha}\right)^{1/2}}, \qquad j = 1, \ldots, r, \qquad B = \gamma U \operatorname{diag}(s_j) V^\top. \tag{19}$$

Consequently $\|B\|_F^2 = \gamma^2 F_p$ is fixed when rank or spectral concentration changes at fixed F_p and γ. The main grid uses $F_p = 128$, $\alpha = 0$, and $\gamma = 18$. The full-rank unrotated construction can use the identity right factor; lower-rank matrices use an orthogonal feature rotation. Spectrum comparisons explicitly enable feature rotation at every spectral setting.

This normalization holds matrix energy fixed, not payload entropy. Softmax is nonlinear, so equal Frobenius norm does not ensure equal D_U. Appendix D separates these controls. The energy effective rank,

$$r_{\text{eff}} = \frac{(\sum_j \sigma_j^2)^2}{\sum_j \sigma_j^4}, \qquad \sigma_j = \gamma s_j, \tag{20}$$

is unchanged by a common scale multiplier. It equals r for a flat spectrum and approaches one when a single singular direction dominates. It summarizes energy concentration, not the number of nonzero singular values or the number of active variables.

B.3 Why the predictive target differs from the latent state

Writing $B = U\Sigma V^\top$, define $q = V^\top z$ and $\widetilde{q} = \Sigma q$. Then $Bz = U\widetilde{q}$, and for any state difference δz,

$$\|B\delta z\|_2^2 = \|\Sigma V^\top \delta z\|_2^2. \tag{21}$$

This equality follows from $U^\top U = I$. It explains the effect weighting: an error in a large-singular-value direction produces a larger log-odds error. At deficient rank, latent differences in the null space of B do not change the predictive distribution. Reconstructing the entire z therefore imposes requirements that prediction itself need not satisfy.

We retain latent-state probes as a separate diagnostic. The four extended targets are z, q, $\widetilde{q}$, and the reference-token log odds Bz. The last two are related by an isometry and have the same squared-error geometry for correspondingly transformed predictions. Independently optimized finite-budget MLPs need not return identical scores. Nor is squared log-odds error identical to distributional KL: the latter depends on the probability distribution at the evaluated state.

B.4 Full-vocabulary KL and conditional payload KL

Let $\mathcal{V}_P$ be the payload vocabulary and $m_i = \sum_{y\in\mathcal{V}_P} p_{\theta,i}(y)$ its predicted probability mass. Extend p_i^* by zero outside $\mathcal{V}_P$, and define $\bar{p}_{\theta,i}(y) = p_{\theta,i}(y)/m_i$ inside it. Then

$$D_{\mathrm{KL}}(p_i^* \| p_{\theta,i}) = D_{\mathrm{KL}}(p_i^* \| \bar{p}_{\theta,i}) - \log m_i. \tag{22}$$

The main behavioral metric uses the left-hand side, accounting for both relative payload probabilities and probability assigned to non-payload tokens. Fitted-component diagnostics use payload-conditional KL. Table 4 reports both quantities; their difference equals $\mathbb{E}[-\log m_i]$.

For a set of evaluated positions $\mathcal{I}$, Capture is the ratio of aggregated gain to aggregated Bayes gain,

$$\mathrm{Capture}(\mathcal{I}) = \frac{\sum_{i\in\mathcal{I}} [D_{\mathrm{KL}}(p_i^* \| U) - D_{\mathrm{KL}}(p_i^* \| p_{\theta,i})]}{\sum_{i\in\mathcal{I}} D_{\mathrm{KL}}(p_i^* \| U)}. \tag{23}$$

First-payload Capture restricts $\mathcal{I}$ to the first payload of every turn, including no-update turns. It excludes earlier payload samples from the same turn while retaining previous-turn history. Negative gain and Capture values are retained. KL and gain use natural logarithms and are measured in nats.

C Complete Capacity Grid and Robustness Controls

C.1 Rank–width grid and compositional prediction

Table 5 reports all 20 rank–width combinations. The equal-rank/width condition provides a reference point for the empirical capacity trend.

The compositional column measures prediction on held-out complete active sets under the same effect matrix. Capture uses each evaluation split's own uniform-predictor KL. Table 4 reports the corresponding full-vocabulary probability accounting.

C.2 Optimization, seeds, architecture, and surface controls

Table 6 summarizes the targeted controls, and Table 7 lists the individual replications. Training seeds 101, 202, 303, and 404 are evaluated at ranks 16, 32, and 64 with width 32 and language seed 9101. These seeds determine sequence sampling and model initialization. A separate series changes the language seed to 9202.

For these four-run summaries, the reported standard error is $s/\sqrt{4}$, where s is the sample standard deviation across trained models. This uncertainty estimate does not apply to the one-seed cells

Table 4: Full-vocabulary and payload-conditional evaluation on the main grid. M is mean probability mass assigned to payload tokens. Full KL includes probability assigned outside the payload vocabulary; conditional KL renormalizes that mass away. Capture elsewhere uses full KL.

Rank	Width	D_U	Full KL	Conditional KL	M
8	16	2.3645	1.1691	1.1582	0.9893
8	32	2.3645	0.9086	0.9060	0.9974
8	64	2.3645	0.8491	0.8482	0.9992
8	128	2.3645	0.3494	0.3486	0.9994
16	16	2.4887	1.7298	1.7208	0.9912
16	32	2.4887	1.3120	1.3095	0.9975
16	64	2.4887	1.1343	1.1334	0.9991
16	128	2.4887	0.7023	0.7005	0.9987
32	16	2.5967	2.0885	2.0795	0.9912
32	32	2.5967	1.7100	1.7056	0.9957
32	64	2.5967	1.4181	1.4171	0.9990
32	128	2.5967	0.9449	0.9434	0.9987
64	16	2.6354	2.3005	2.2931	0.9927
64	32	2.6354	1.9370	1.9334	0.9965
64	64	2.6354	1.6324	1.6313	0.9990
64	128	2.6354	1.4547	1.4528	0.9985
128	16	2.6655	2.3744	2.3659	0.9916
128	32	2.6655	2.0224	2.0187	0.9964
128	64	2.6655	1.7223	1.7212	0.9989
128	128	2.6655	1.5658	1.5637	0.9981

Table 5: Complete rank–width grid on held-out data. Compositional Capture uses the compositional payload KL and its own uniform baseline and differs from IID Capture by at most 0.78 percentage points.

d_{pred}	d_{model}	Capture	First	Comp.	$R^2_{\tilde{q}}$
8	16	50.6	31.0	50.2	0.369
8	32	61.6	40.6	61.4	0.523
8	64	64.1	44.4	64.0	0.550
8	128	85.2	81.6	85.1	0.841
16	16	30.5	17.8	30.4	0.139
16	32	47.3	29.4	47.2	0.293
16	64	54.4	35.6	54.4	0.397
16	128	71.8	65.1	71.9	0.664
32	16	19.6	10.2	19.8	0.042
32	32	34.1	19.7	34.3	0.102
32	64	45.4	28.0	45.7	0.209
32	128	63.6	55.7	63.8	0.505
64	16	12.7	6.4	12.7	0.008
64	32	26.5	14.4	27.1	0.028
64	64	38.1	23.6	38.6	0.063
64	128	44.8	31.5	45.6	0.130
128	16	10.9	5.1	10.9	0.002
128	32	24.1	13.0	24.1	0.006
128	64	35.4	22.4	35.4	0.015
128	128	41.3	28.8	41.4	0.038

elsewhere in the grid. Payload samples from the same model, and especially positions in the same episode, are not independent model replications. We therefore do not attach across-seed error bars to the entire capacity surface.

Doubling training from 72 to 144 epochs improves Capture at each tested rank while preserving the rank ordering. Two- and six-block models, tied embeddings, and alternative statement templates provide additional controls at width 32.

Table 6: **Optimization, replication, architecture, and surface-form controls.** Capture is in percent. The natural-language-like condition changes only statement templates; the latent generator and payload process remain unchanged.

A. Longer training at $d_{\text{model}} = 32$				
Rank	72-epoch Capture	144-epoch Capture	72-epoch R^2	144-epoch R^2
16	47.28	51.47	0.293	0.358
32	34.15	38.05	0.102	0.132
64	26.50	30.63	0.028	0.037
128	24.13	28.02	0.006	0.009
B. Four-seed replication at $d_{\text{model}} = 32$				
Rank	Capture mean $\pm$ SE	First mean $\pm$ SE	R^2 mean $\pm$ SE	ΔKL mean $\pm$ SE
16	47.32 ± 0.06	29.46 ± 0.04	0.2950 ± 0.0014	0.05602 ± 0.00013
32	34.64 ± 0.37	20.13 ± 0.21	0.1030 ± 0.0019	0.01601 ± 0.00018
64	27.91 ± 0.50	15.31 ± 0.31	0.0288 ± 0.0005	0.00447 ± 0.00012
C. Architecture and language-like surface controls: Capture / R^2 at ranks 16, 32, 64				
Condition	Rank 16	Rank 32	Rank 64	Changed factor
2 layers	47.71 / .300	35.30 / .106	28.75 / .030	depth
6 layers	47.27 / .293	34.46 / .100	27.51 / .029	depth
Tied embeddings	49.71 / .337	37.63 / .133	30.75 / .039	tying
Language-like templates	49.05 / .314	35.14 / .107	28.30 / .029	surface form

Table 7: Individual replication runs at width 32. Seed changes affect model initialization and sampled sequences; the language seed determines the effect matrix. Capture values are percentages. ΔKL is fitted predictive-component removal damage using payload-conditional probabilities (Appendix G). Each row summarizes one trained model.

Rank	Seed	Language seed	Capture	First	$R^2_{\tilde{q}}$	ΔKL
16	101	9101	47.28	29.44	0.2933	0.05635
16	202	9101	47.50	29.55	0.2932	0.05611
16	303	9101	47.21	29.35	0.2990	0.05585
16	404	9101	47.30	29.50	0.2944	0.05577
16	101	9202	50.63	32.01	0.3061	0.05744
32	101	9101	34.15	19.68	0.1016	0.01569
32	202	9101	34.44	19.95	0.1013	0.01606
32	303	9101	35.74	20.69	0.1087	0.01649
32	404	9101	34.21	20.19	0.1004	0.01580
32	101	9202	35.85	21.00	0.1009	0.01622
64	101	9101	26.50	14.40	0.0278	0.00456
64	202	9101	28.13	15.75	0.0284	0.00477
64	303	9101	28.85	15.66	0.0291	0.00425
64	404	9101	28.14	15.43	0.0300	0.00431
64	101	9202	27.42	15.15	0.0272	0.00437

C.3 Language-like surface comparison

The surface comparison uses ranks 16, 32, and 64 at width 32, with 128 available variables, four active variables, and the same predictive mechanism as the symbolic grid. Table 3 pairs the four statement templates and an update example. The model vocabulary treats each displayed word or marker as one discrete token. The longer wording changes the serialized context while retaining the state and payload rules.

Table 8 reports both surfaces side by side. Language-like Capture is 49.05%, 35.14%, and 28.30% across the three ranks, compared with 47.28%, 34.15%, and 26.50% for symbolic statements. First-payload and compositional measurements preserve this ordering. Weighted predictive-state linear and MLP scores also decrease with rank on both surfaces; shuffled-target MLP scores remain close to zero. These comparisons support robustness of the rank trend to the tested serialization changes.

Table 8: Symbolic and language-like surfaces at width 32. Capture columns are percentages; probes target $\widetilde{q}$. Compositional Capture uses its own split-specific uniform baseline. Shuffled denotes the shuffled-target MLP.

Rank	Surface	Capture	First	Comp.	Linear	MLP	Shuffled
16	Symbolic	47.28	29.44	47.23	0.2933	0.3179	0.0033
16	Language-like	49.05	30.69	48.99	0.3140	0.3346	0.0027
32	Symbolic	34.15	19.68	34.35	0.1016	0.1369	-0.0002
32	Language-like	35.14	20.14	35.61	0.1070	0.1423	-0.0005
64	Symbolic	26.50	14.40	27.09	0.0278	0.0477	-0.0002
64	Language-like	28.30	15.90	28.86	0.0291	0.0505	-0.0010

C.4 Training dynamics

Figure 6 plots the recorded validation Capture at every completed epoch. The five width-32 models share a 72-epoch budget; the extended runs use 144 epochs. The signal-scale series uses width 128, with scale 5 ending at epoch 30 and the remaining scales completing 72 epochs. Markers identify the selected checkpoints. Table 9 provides completed and selected epochs, together with best and final validation scores.

The extended runs continue to improve beyond the original budget, preserving the ordering across ranks. In the signal-scale series, scale 7 reaches its selected score at epoch 29 and ends below the uniform baseline; scale 10 also finishes below its selected score. Scales 14 and 18 retain most of their selected performance. These trajectories explain why checkpoint-selected performance and the final-epoch score differ.

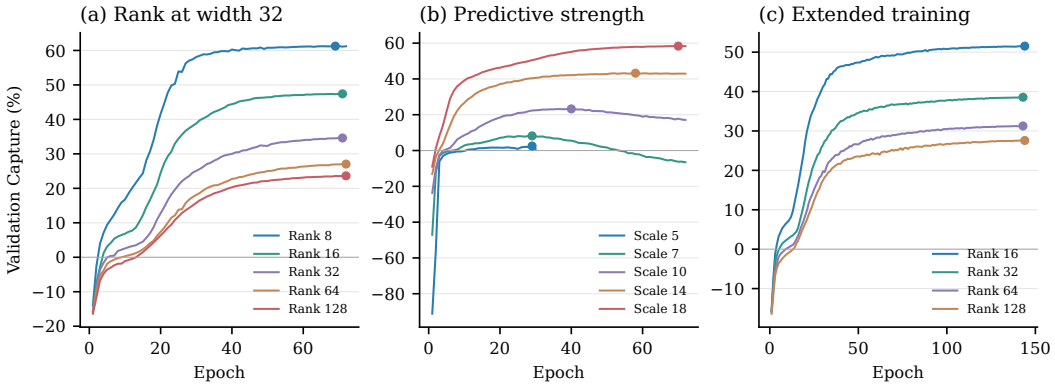

Figure 6: Epochwise validation Capture. (a) Rank sweep at width 32. (b) Signal-scale sweep at width 128. (c) Extended-budget runs at width 32. Lines show recorded scores without smoothing; dots mark selected epochs. Negative values are retained, and each panel uses its own vertical limits.

D Predictive Load, Spectrum, and Occupancy

D.1 Matching Bayes gain

The KL-matched rank experiment keeps 128 predictive variables and width 32 while varying rank. The spectrum experiment instead uses 64 predictive variables, rank 64, and width 32, varying $\alpha \in$

Table 9: Training duration and checkpoint selection. Best and final columns report validation Capture in percent. Grid and long runs use width 32; signal-scale runs use width 128. Scale 5 terminates under the predictive-gain stopping rule.

Configuration	Epochs	Selected epoch	Best	Final
grid_r8_d32	72	69	61.26	61.22
grid_r16_d32	72	71	47.41	47.33
grid_r32_d32	72	71	34.61	34.61
grid_r64_d32	72	72	27.04	27.04
grid_r128_d32	72	72	23.63	23.63
long_r16_d32	144	144	51.53	51.53
long_r32_d32	144	143	38.56	38.52
long_r64_d32	144	143	31.26	31.24
long_r128_d32	144	144	27.58	27.58
scale5_f64_d128	30	29	2.52	0.87
scale7_f64_d128	72	29	8.19	-6.57
scale10_f64_d128	72	40	23.25	17.08
scale14_f64_d128	72	58	43.20	42.93
scale18_f64_d128	72	70	58.33	58.30

$\{0, 0.5, 1, 2\}$. These are within-family controlled comparisons, not the same latent pool reused across every panel. Each family calibrates the common effect scale separately to a target mean $D_U = 2.5$.

Calibration uses 4,096 sparse signed states from an independent fixed seed and evaluates $\log M - H(p^*)$ as a function of effect scale. After checking that scale 64 reaches the target, bisection searches $[0, 64]$ for 48 iterations. Finite-sample calibration approximately matches test D_U to 2.5. Table 10 reports realized values and calibrated scales; Table 11 reports the fixed-energy sweep at widths 32 and 64.

Table 10: Bayes-KL-matched comparisons at width 32. Rank matching uses 128 predictive variables; spectrum matching uses 64. The calibration target is 2.5 nats. Realized test D_U is retained rather than replaced by the target. Capture and first-payload Capture are percentages.

Family	Rank	α	γ	D_U	Capture	First	$R^2_{\bar{q}}$
Rank	8	0.0	19.2149	2.5232	62.93	41.27	0.5267
Rank	16	0.0	18.1872	2.5150	47.60	29.55	0.2922
Rank	32	0.0	17.3323	2.4968	32.30	18.63	0.0978
Rank	64	0.0	17.1142	2.5011	24.83	13.67	0.0289
Rank	128	0.0	17.0036	2.5127	21.73	11.96	0.0051
Spectrum	64	0.0	16.9841	2.4948	27.17	16.95	0.0281
Spectrum	64	0.5	17.7843	2.5235	40.90	25.90	0.2534
Spectrum	64	1.0	21.3020	2.5456	65.89	49.04	0.5839
Spectrum	64	2.0	28.0168	2.5630	82.62	67.90	0.7078

Table 11: **Energy-matched predictive-spectrum sweep.** Algebraic rank is fixed at 64 and the singular values follow $\sigma_i \propto i^{-\alpha}$. Capture is reported in percent.

α	r_{eff}	d_{model}	Capture	First Capture	$R^2_{\bar{q}}$
0.0	64.00	32	29.63	17.93	0.027
0.0	64.00	64	42.97	30.51	0.086
0.5	13.81	32	41.05	26.03	0.252
0.5	13.81	64	51.54	37.17	0.316
1.0	2.45	32	64.01	47.91	0.584
1.0	2.45	64	69.60	57.61	0.642
2.0	1.17	32	80.08	65.51	0.726
2.0	1.17	64	84.80	76.89	0.794

At rank 64, increasing α concentrates singular-value energy while retaining algebraic rank. The matched comparisons quantify its effect on Capture at approximately equal mean Bayes gain.

D.2 Fixed-occupancy and mixture-trained models

The raw-occupancy series changes $k \in \{2, 4, 8, 16\}$ while retaining unit signed amplitudes. Its state energy is $\|z\|^2 = k$, so more active variables also change the signal delivered to softmax. The normalized series instead uses $a_k = \sqrt{4/k}$, equivalently rescaling the effect amplitude for fixed-k datasets, so that $\|z\|^2 = 4$. Matching state energy is not mathematically identical to matching Bayes KL; both the construction and measured D_U should be considered.

Fixed-condition models are trained separately at each occupancy. Each mixture model is trained on all four occupancies and evaluated on the fixed-occupancy datasets. Table 12 reports these cross-evaluations for both normalization conditions.

The raw series improves strongly as occupancy increases, while the energy-normalized mixture model is substantially flatter. This comparison shows why occupancy by itself is not a sufficient description of predictive load. It does not imply that occupancy is irrelevant under every generator or after every normalization. The effect-scale series provides a complementary intervention: it changes predictive strength while retaining the variable pool and architecture.

Table 12: **Complete occupancy controls.** Fixed models are trained and evaluated at one occupancy. Mixture models are trained jointly on $k \in \{2, 4, 8, 16\}$ and evaluated on each fixed condition. Capture is reported in percent.

Fixed condition	D_{uniform}	Capture	First Capture
$k = 2$, raw	1.717	19.16	15.95
$k = 4$, raw	2.672	30.71	17.82
$k = 8$, raw	3.468	45.07	27.43
$k = 16$, raw	4.044	59.20	40.83
$k = 2$, energy matched	2.650	38.79	28.20
$k = 8$, energy matched	2.653	30.48	20.46
$k = 16$, energy matched	2.660	32.57	25.77
Mixture model / test condition	**Raw-mixture Capture**		**Energy-mixture Capture**
$k = 2$, raw	15.59		14.72
$k = 4$, raw	28.11		27.17
$k = 8$, raw	41.56		37.09
$k = 16$, raw	55.11		46.50
$k = 2$, energy normalized	27.09		30.26
$k = 4$, energy normalized	28.11		27.17
$k = 8$, energy normalized	27.55		27.25
$k = 16$, energy normalized	27.88		30.74

E Global Representation Readouts

E.1 Linear and nonlinear probes

For each activation coordinate, training-set mean and standard deviation define the standardization applied to every split, with a minimum standard deviation of 10^{-6}. Ridge fitting centers the inputs and targets, leaving the intercept unpenalized. The implementation solves the summed-error objective

$$W_\lambda = (X_c^\top X_c + \lambda I)^{-1} X_c^\top Y_c, \qquad \widehat{Y} = (X - \bar{X}_{\text{train}}) W_\lambda + \bar{Y}_{\text{train}}. \tag{24}$$

The candidate penalties are $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ and are selected by validation R^2. These numerical penalties refer to the summed-error implementation; an averaged-error objective uses the corresponding penalty divided by the training-row count.

The quadratic probe augments standardized activations with upper-triangular products $h_i h_j$, including squares. When there are more than 2,048 possible products, a fixed-seed subset is sampled

without replacement and reused across splits. The MLP has one hidden layer of width 128, a GELU nonlinearity, and a linear output. In the extended analysis it is trained with AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch size 512, and at most 80 epochs, with validation patience ten. The shuffled-label control permutes training targets and uses at most 40 epochs and patience eight. Validation and test targets remain unshuffled.

E.2 Score definition and negative controls

For all global targets, the reported score is a joint, variance-weighted reconstruction score,

$$R^2_Y = 1 - \frac{\sum_{i \in \text{test}} \|Y_i - \widehat{Y}_i\|_2^2}{\sum_{i \in \text{test}} \|Y_i - \bar{Y}_{\text{train}}\|_2^2}. \tag{25}$$

Negative values indicate worse squared error than the training-mean baseline and are retained. For latent-state probes only, active-variable AUROC treats $|z_f| > 0$ as the positive label and uses the absolute predicted coordinate as its score; sign accuracy is evaluated only on active entries.

Table 13 supplies all probe classes and shuffled controls across the 20 models. Figure 7 makes the overload subset and the behavioral/access distinction explicit. A larger probe class does not guarantee improved held-out performance: finite data, regularization, and optimization can make the quadratic or MLP score worse. Low scores rule out strong recovery by the tested procedures at the stated budgets, not every possible nonlinear decoder.

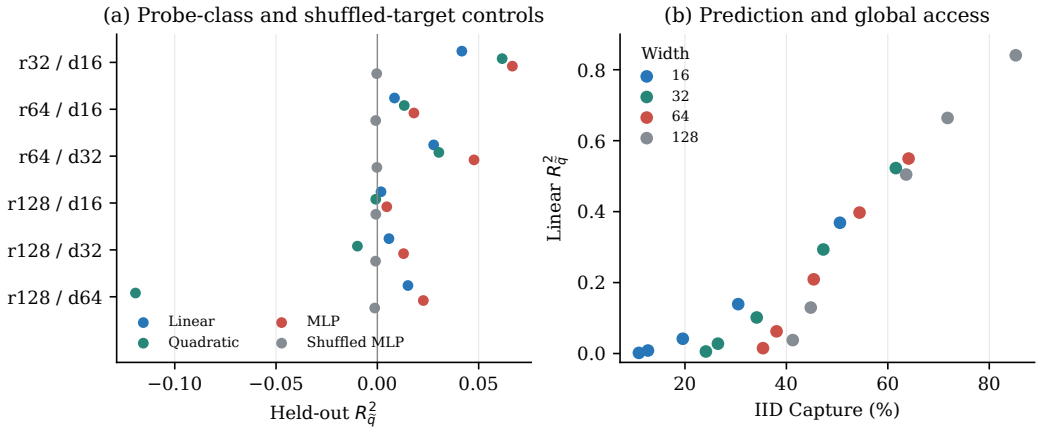

Figure 7: Supplementary global-access checks. (a) Linear, quadratic, MLP, and shuffled-target MLP scores on the six overloaded models, retaining negative R^2. Points show probe results for individual models. (b) All 20 grid models plotted by behavioral Capture and linear predictive-coordinate recovery, colored by residual width.

Table 14 compares latent and predictive targets and reports layerwise recovery at signal scales 10, 14, and 18. Table 13 provides shuffled-target controls for the complete grid; Table 15 reports latent-state probe results for all five signal scales.

E.3 Signal-scale probe comparison

Table 15 compares the original linear, quadratic, and MLP readouts of z across the five signal scales. This series uses a 64-unit hidden layer in its MLP; the extended predictive-coordinate probes above use 128 units. The scores retain each analysis's target and architecture. Active-variable AUROC increases from 0.744 to 0.923, and active-sign accuracy from 0.688 to 0.951. For each scale, the original linear latent-state probe scores above the corresponding MLP.

Table 13: Predictive-coordinate probes across the complete grid. All entries are held-out $R^2_{\widetilde{q}}$; negative values are retained. The shuffled-label MLP permutes training targets, while validation and test targets remain unshuffled. Each column has its own validation-based selection.

Rank	Width	Linear	Quadratic	MLP	Shuffled MLP
8	16	0.3687	0.3997	0.4087	0.0089
8	32	0.5227	0.5177	0.5242	0.0060
8	64	0.5497	0.4986	0.5547	0.0076
8	128	0.8407	0.8111	0.8406	0.0017
16	16	0.1392	0.1712	0.1755	0.0022
16	32	0.2933	0.3075	0.3179	0.0033
16	64	0.3973	0.3302	0.3974	0.0041
16	128	0.6641	0.6031	0.6622	0.0045
32	16	0.0417	0.0616	0.0666	-0.0003
32	32	0.1016	0.1259	0.1369	-0.0002
32	64	0.2092	0.1189	0.2122	-0.0008
32	128	0.5046	0.4184	0.4899	0.0002
64	16	0.0085	0.0133	0.0180	-0.0008
64	32	0.0278	0.0304	0.0477	-0.0002
64	64	0.0625	-0.0321	0.0766	-0.0007
64	128	0.1296	0.0088	0.1258	-0.0006
128	16	0.0018	-0.0008	0.0047	-0.0007
128	32	0.0057	-0.0098	0.0129	-0.0009
128	64	0.0151	-0.1194	0.0227	-0.0013
128	128	0.0378	-0.1055	0.0370	-0.0018

Table 14: **Additional predictive-target and signal-scale diagnostics.** Panel A compares global decoders for latent state z and weighted predictive state $\widetilde{q}$ under overload. Panel B shows active/sign recovery and where predictive-state decoding forms across the final blocks.

A. Held-out global decoding under overload

Config.	z linear	z quadratic	z MLP	$\widetilde{q}$ linear	$\widetilde{q}$ quadratic	$\widetilde{q}$ MLP	Centered-logit MLP
r32/d16	.009	.007	.009	.042	.062	.067	.068
r64/d16	.003	.001	.004	.008	.013	.018	.018
r64/d32	.012	-.005	.015	.028	.030	.048	.047
r128/d16	.002	-.001	.001	.002	-.001	.005	.005
r128/d32	.006	-.010	.008	.006	-.010	.013	.013
r128/d64	.015	-.119	.016	.015	-.119	.023	.022

B. Signal-scale recovery and layerwise formation at rank 64 / width 128

Scale	Active AUROC	Sign accuracy	Layer-2 R^2	Layer-3 R^2	Final-norm R^2
10	.803	.792	.049	.126	.125
14	.868	.870	.071	.262	.260
18	.923	.951	.096	.392	.391

Table 15: Global latent-state readouts in the signal-scale series ($F = 64$, width 128). Linear, quadratic, and MLP columns are R^2_z; AUROC and sign accuracy use the linear probe.

Scale	Linear	Quadratic	MLP	Active AUROC	Sign accuracy
5	0.0432	-0.0172	0.0423	0.7440	0.6880
7	0.0556	-0.0124	0.0496	0.7731	0.7016
10	0.1249	0.0894	0.1163	0.8035	0.7922
14	0.2598	0.2430	0.2335	0.8684	0.8702
18	0.3914	0.3753	0.3601	0.9228	0.9511

F Local Counterfactuals and Gain Distributions

F.1 Paired prefixes and within-variable consistency

Each single-variable counterfactual pair is serialized by the generator, with the targeted variable assigned opposite signs and the remaining context held fixed. Both members assign an explicit value to the target variable. The analysis extracts final residual activations immediately before a payload prediction. The 4,096 pairs per model vary both the selected variable and its surrounding context; each variable is evaluated in a subset of these pairs.

For a given variable f and nonzero half-difference $\Delta h_f(c)$ from Equation 16, define its unit direction $u_f(c) = \Delta h_f(c)/\|\Delta h_f(c)\|_2$. The resultant is

$$R_f = \left\| \frac{1}{n_f} \sum_{c=1}^{n_f} u_f(c) \right\|_2 . \tag{26}$$

A value near one indicates similarly oriented residual differences for a variable across sampled contexts. The resultant measures within-variable directional consistency; global probes measure joint-state recovery.

For the RQ3 point-cloud figures (Figures 4 and 8), PCA is fit jointly to both signs within each model, and 1,000 pairs are displayed. Marginal densities and covariance ellipses summarize the pooled projection. Each model has its own PCA basis. Full-dimensional resultants complement the projected distributions.

Figure 1d shows the rank-128, width-32 configuration highlighted in the capacity map, using the same joint PCA basis and 1,000 paired contexts as Figure 8. Blue and orange points are measured residuals for negative- and positive-sign counterfactual prefixes, pooled across targeted variables. Point drawing order is deterministically shuffled across signs to avoid systematic occlusion. Solid blue and dashed orange ellipses show two-standard-deviation covariance contours. This panel visualizes projected representations; it does not establish weak global decoding, local directional stability, or causal use. Those claims are supported by the corresponding quantitative analyses in Table 1.

F.2 Distribution of predictive gains

For each held-out payload position, the full-vocabulary gain is $g_i = D_{\mathrm{KL}}(p_i^* \| U) - D_{\mathrm{KL}}(p_i^* \| p_{\theta,i})$. A positive mean could in principle arise from a small set of large improvements. Reporting its distribution and the fraction $g_i > 0$ tests that possibility more directly. The exported sample contains 16,384 held-out positions for each overloaded model; the full-vocabulary and conditional results are compared in Table 16.

The allocation analysis adjusts sample gain for true entropy, the top-two probability margin, occupancy, and payload ordinal before summarizing feature-related associations (Table 17).

Table 16: **Predictive-gain distributions and normalization audit.** Each model is evaluated on 16,384 positions selected from the first 1,024 IID test sequences using seed 20260831, matching the original allocation analysis. Full-vocabulary gain is used in Figure 9B. Conditional gain renormalizes model probabilities over payload types, reproducing the earlier allocation metrics. Positions within a sequence are not independent replicates.

	Full vocabulary		Payload conditional	
Config.	Mean gain	Gain > 0 (%)	Mean gain	Gain > 0 (%)
r32/d16	0.508	77.24	0.517	77.61
r64/d16	0.345	71.22	0.352	71.72
r64/d32	0.709	79.53	0.713	79.61
r128/d16	0.288	70.07	0.297	70.72
r128/d32	0.637	77.34	0.640	77.41
r128/d64	0.933	85.31	0.934	85.33

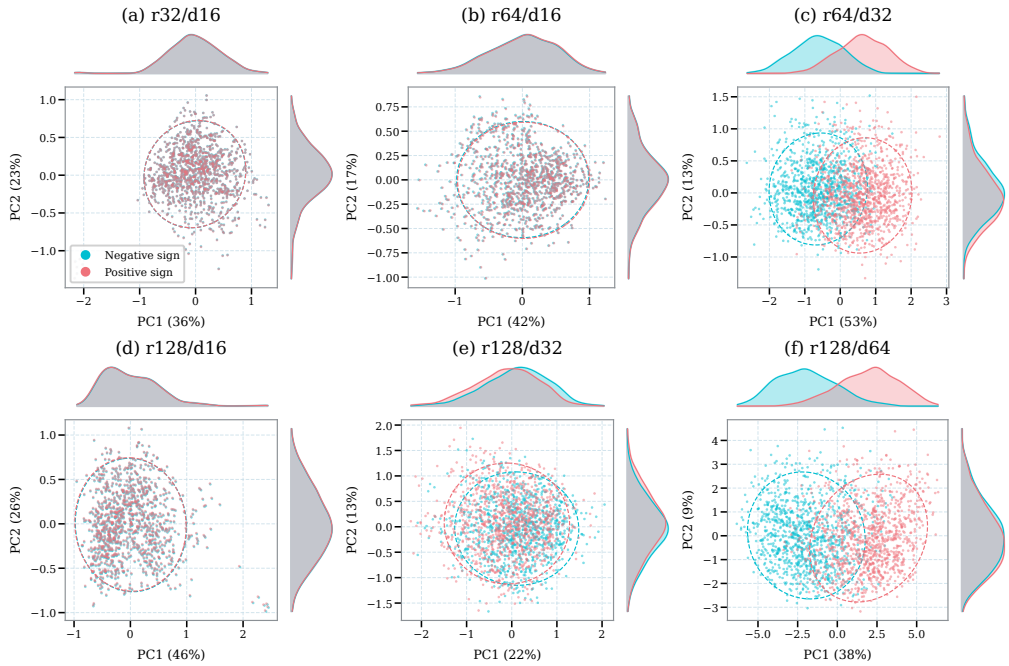

Figure 8: Counterfactual residual representations for all six overloaded models. Each panel displays 1,000 paired contexts pooled across targeted variables. Top and right curves are marginal kernel-density estimates; dashed ellipses show two-standard-deviation covariance contours. PCA is fit jointly to both signs within each model, with explained variance on the axes. Each model uses its own PCA coordinates.

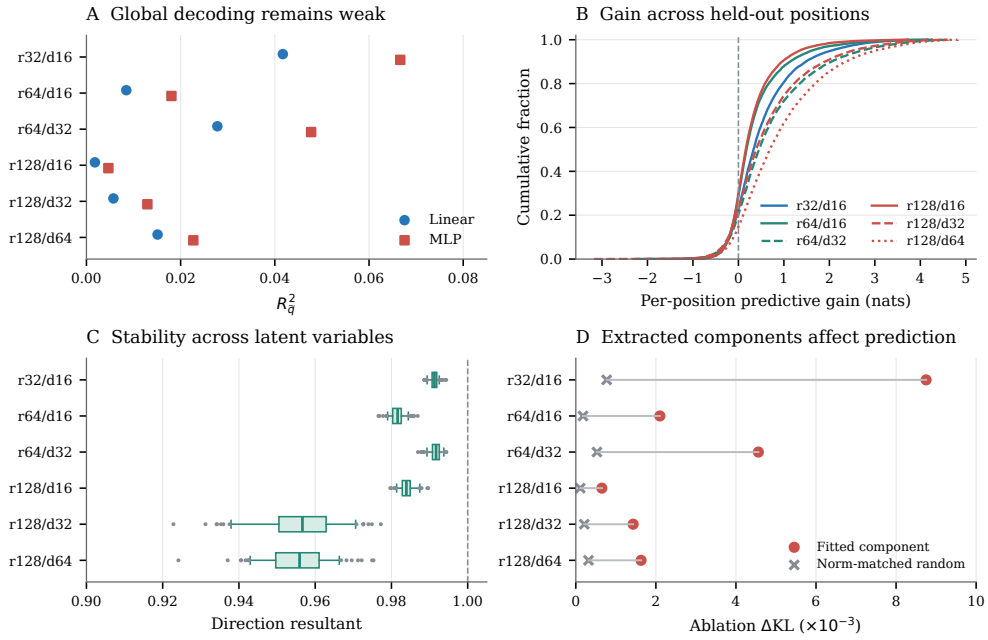

Figure 9: Full-dimensional overload diagnostics. (A) Global decoding. (B) Empirical gains at 16,384 held-out positions per model, with zero gain marked. (C) Resultants across 128 variables: boxes are quartiles and whiskers are the 5th–95th percentiles. (D) Fitted-component and norm-matched random ablations. Distributional summaries describe examples or variables within one model.

Table 17: **Sample- and feature-level allocation under overload.** These original diagnostics use payload-conditional model probabilities; Table 16 compares the full-vocabulary definition used in the main text. Feature effects are adjusted for true entropy, top-two probability margin, occupancy, and payload ordinal. The final column is the largest correlation between a single true singular-component magnitude and sample gain.

Config.	Mean gain	Gain > 0	Adjusted feature-effect range	Max component correlation
r32/d16	0.517	77.61%	$[-0.169,\ 0.156]$	0.070
r64/d16	0.352	71.72%	$[-0.100,\ 0.151]$	0.045
r64/d32	0.713	79.61%	$[-0.121,\ 0.125]$	0.052
r128/d16	0.297	70.72%	$[-0.112,\ 0.144]$	0.034
r128/d32	0.640	77.41%	$[-0.148,\ 0.147]$	0.037
r128/d64	0.934	85.33%	$[-0.157,\ 0.106]$	0.036

F.3 Fixed SVD truncation reference

A fixed spectral truncation provides a generator-based compression reference. For the saved thin SVD $B = U\Sigma V^\top$, retain the first m components:

$$B_m = U_{:,1:m}\Sigma_{1:m,1:m}V^\top_{:,1:m}, \qquad p_m(y \mid z) = \mathrm{softmax}([0; B_m z])_y. \tag{27}$$

The reference payload has zero logit and the generator bias is zero. We compute

$$C_m = 1 - \frac{\mathbb{E}_i D_{\mathrm{KL}}(p_i^* \| p_m(\cdot \mid z_i))}{\mathbb{E}_i D_{\mathrm{KL}}(p_i^* \| U)} \tag{28}$$

on the same 16,384 positions used for the gain distributions. The primary comparison sets $m = d_{\mathrm{model}}$. Table 18 also reports model Capture on this sample, with both payload-conditional and full-vocabulary normalization. These sample-level scores differ slightly from the full-test-set values in Table 1.

At r128/d32, full-vocabulary model Capture is 23.94%, exceeding the fixed truncation's 21.87%; at r128/d16, the corresponding scores are 10.85% and 10.93%. The remaining configurations score below their fixed truncations. This reference uses the true latent state and a fixed linear projection, whereas the trained model predicts from token prefixes.

The flat-spectrum grid has tied singular values, so the saved SVD selects one basis within each degenerate subspace. Accordingly, C_m characterizes that fixed choice rather than an optimized rank-m predictor or a universal capacity bound. Figure 10 and Table 19 show the complete retained-dimension sweeps. Keeping all predictive directions recovers the true distribution and gives Capture 100%.

Table 18: Fixed SVD truncation at $m = d_{\mathrm{model}}$ and model Capture on the same 16,384-position sample. All Capture values are percentages. Conditional normalizes model probabilities over payload types; full retains the full-vocabulary probabilities.

Configuration	SVD	Model conditional	Model full
r32/d16	45.22	19.90	19.55
r64/d16	22.60	13.34	13.06
r64/d32	46.94	27.02	26.89
r128/d16	10.93	11.17	10.85
r128/d32	21.87	24.08	23.94
r128/d64	44.74	35.12	35.09

G Causal Interventions and Geometry Checks

G.1 Fitted predictive-component removal

The component analysis fits an encoding map from $\widetilde{q}$ to final residual activations on the analyzed test sample using ridge penalty 0.1. For component j, let v_j be its fitted residual direction and $\bar{q}_j$

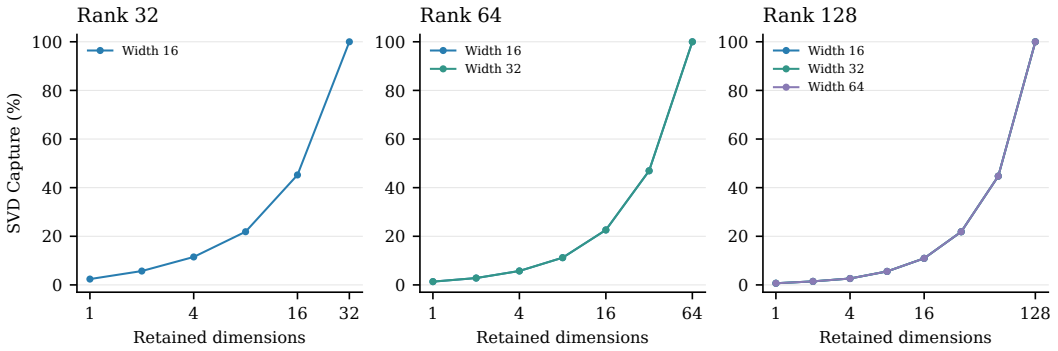

Figure 10: Generator-based Capture as retained SVD dimension increases. Each curve uses the saved basis and sample for one overloaded model. Width identifies the model used for comparison; the reference itself is computed from the generator.

Table 19: Retained-dimension sweep for fixed SVD truncation. m is the retained dimension; Capture is in percent and KL in nats. Widths listed together have identical results at the reported precision. Each configuration uses its saved SVD basis and evaluation sample.

Rank 32; width 16

m	Capture	KL
1	2.40	2.5366
2	5.70	2.4508
4	11.49	2.3003
8	21.88	2.0304
16	45.22	1.4237
32	100.00	0.0000

Rank 64; width 16, 32

m	Capture	KL
1	1.35	2.6031
2	2.82	2.5643
4	5.73	2.4874
8	11.20	2.3432
16	22.60	2.0422
32	46.94	1.4001
64	100.00	0.0000

Rank 128; width 16, 32, 64

m	Capture	KL
1	0.68	2.6414
2	1.45	2.6210
4	2.63	2.5895
8	5.55	2.5118
16	10.93	2.3688
32	21.87	2.0778
64	44.74	1.4696
128	100.00	0.0000

its mean over that sample. At each analyzed position, the intervention subtracts $(\tilde{q}_j - \bar{q}_j)v_j$ from the final residual activation and applies the frozen output head. The random control replaces v_j by a Gaussian direction scaled to the same norm and uses the same per-example amplitude. At most the first 32 predictive components are evaluated. The reported damage is averaged over these selected components.

The original component audit normalizes over payload logits. Its damage score is

$$\Delta_j = \mathbb{E}_{\text{test}}\left[D_{\text{KL}}(p^* \| \bar{p}_\theta^{(-j)}) - D_{\text{KL}}(p^* \| \bar{p}_\theta)\right]. \tag{29}$$

Both fitted and random removals use payload-conditional KL. The intervention acts directly on the final normalized activation through the frozen output head, measuring output sensitivity to the extracted component.

G.2 Validity of global geometry estimates

Global latent-to-residual directions are evaluated using encoding $R^2 > 0.01$, decoding $R^2 > 0.01$, active-variable AUROC > 0.6, sign accuracy > 0.55, cross-split direction stability > 0.5, and nonzero direction norms.

Configurations r16/d16, r32/d16, r64/d16, r64/d32, r128/d16, and r128/d32 fall below at least one regression criterion. Table 20 reports their overlaps alongside the causal diagnostics. The signal-scale series further characterizes how access and fitted geometry vary with predictive strength.

Table 21 reports direction rank, overlap, the covariance null, predictive variance, and activation rank across effect scales. These quantities describe different aspects of the fitted map and should be read together with the validity criteria above; active/sign recovery is reported separately in Table 14.

Table 20: **Global direction estimates and causal effects across the capacity grid.** Panel A reports mean absolute overlap among fitted latent directions. Panel B reports predictive-component ablation on the analyzed test sample ΔKL. Overlap is interpreted only when the estimator validity criteria in the text are satisfied.

	d_{model}			
Predictive rank	16	32	64	128
A. Mean direction overlap				
8	0.335	0.286	0.277	0.279
16	0.333	0.228	0.191	0.185
32	0.310	0.212	0.146	0.122
64	0.293	0.207	0.180	0.107
128	0.293	0.201	0.275	0.190
B. Predictive-component ablation ΔKL				
8	0.1281	0.1642	0.1697	0.2712
16	0.0345	0.0563	0.0642	0.1121
32	0.0087	0.0157	0.0201	0.0438
64	0.0021	0.0046	0.0057	0.0081
128	0.0007	0.0014	0.0016	0.0023

Table 21: **Direction geometry across the predictive-value sweep.** Direction rank is the effective rank of the fitted latent-direction Gram matrix. Predictive variance is the fraction of activation variance in the span of those directions.

Scale	Mean overlap	Direction rank	Covariance null	Predictive variance	Activation rank
5	0.668	2.17	0.132	0.880	34.78
7	0.366	6.12	0.110	0.843	52.49
10	0.121	25.58	0.088	0.739	81.85
14	0.050	46.77	0.085	0.740	89.06
18	0.041	45.15	0.088	0.790	83.10

H Sparse Autoencoder Analysis

H.1 SAE fitting

SAEs are fit to final residual activations before the first payload in each turn. The training mean is subtracted and a single training RMS scale is used for all coordinates and all splits. For normalized activation x, the encoder and decoder are

$$f(x) = \mathrm{ReLU}(W_e x + b_e), \qquad \widehat{x} = W_d f(x) + b_d, \tag{30}$$

with $4d_{\text{model}}$ dictionary units. Decoder columns are normalized to unit length after optimizer steps. The objective averages squared reconstruction error over examples and coordinates and adds λ times the mean sum of unit activations. This distinction fixes the scale of the sparsity coefficient.

The final-matrix sweep tests $\lambda \in \{0.003, 0.006, 0.01, 0.02, 0.04\}$. Candidates use Adam with learning rate 3×10^{-4}, betas $(0.9, 0.999)$, batch size 1,024, up to 4,000 steps, gradient clipping at 1.0, validation every 100 steps, and patience eight. The selected model balances reconstruction error, activation sparsity, and dead-unit fraction.

H.2 Reconstruction and signed-concept matching

For normalized test inputs, FVU is $\sum_i \|x_i - \widehat{x}_i\|^2 / \sum_i \|x_i\|^2$; L_0 counts units with positive activation. For every variable f, two binary concepts indicate its positive and negative active state. This gives 256 signed concepts for the 128-variable capacity grid. Each concept is matched to the unit with highest training correlation, and the matched correlation is reported on test data.

Multiple concepts may select the same unit. The number of distinct selected units measures dictionary reuse. Thresholded split/merge counts use training correlation 0.3; this differs from the reported 0.25 held-out-match check and the 0.4 coverage statistic. Each statistic uses its specified threshold.

Table 22 and Figure 11 extend the main width-32 slice to all 20 grid models, including sparsity, dead units, and selected regularization.

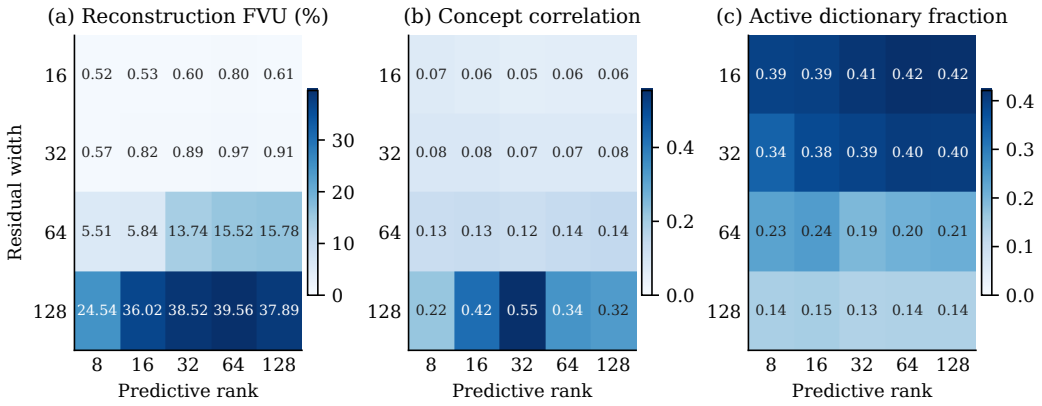

Figure 11: Sparse readouts across the full rank–width grid. Heatmaps report reconstruction FVU, mean held-out signed-concept correlation, and the active-unit fraction $L_0/(4d_{\text{model}})$. Each panel has its own color scale; colors are not comparable between metrics. Rows and columns identify the same models across panels.

Table 22: Sparse autoencoders across all 20 rank–width models. FVU and dead-unit fraction are percentages; L_0 is the mean number of active units. Correlation is averaged over matched signed-concept readouts. The dictionary has $4d_{\text{model}}$ units.

Rank	Width	λ	FVU (%)	L_0	Dead (%)	Correlation	Matched units
8	16	0.003	0.518	24.96	0.00	0.0722	56
8	32	0.003	0.574	43.42	0.00	0.0838	92
8	64	0.010	5.512	59.34	0.00	0.1256	141
8	128	0.020	24.536	73.11	0.00	0.2165	203
16	16	0.003	0.535	25.22	0.00	0.0649	58
16	32	0.003	0.818	48.31	0.00	0.0828	97
16	64	0.010	5.836	62.14	0.00	0.1300	152
16	128	0.020	36.023	74.72	0.00	0.4231	223
32	16	0.003	0.602	26.21	0.00	0.0548	59
32	32	0.003	0.890	49.86	0.00	0.0743	101
32	64	0.020	13.737	48.92	0.00	0.1240	160
32	128	0.020	38.521	67.98	0.00	0.5542	243
64	16	0.003	0.803	26.94	0.00	0.0579	57
64	32	0.003	0.971	51.23	0.00	0.0744	108
64	64	0.020	15.520	52.24	0.00	0.1379	159
64	128	0.020	39.564	69.79	0.00	0.3393	211
128	16	0.003	0.611	26.81	0.00	0.0580	58
128	32	0.003	0.908	51.11	0.00	0.0756	101
128	64	0.020	15.782	53.47	0.00	0.1432	152
128	128	0.020	37.895	69.84	0.00	0.3220	210

H.3 Ablation controls and their interpretation

The initial SAE ablation removes a train-matched unit and compares its KL effect with a wrong-unit removal. The separate paired audit chooses an active signed concept for each example, excludes control units with absolute training correlation at least 0.3 to that concept, and selects the remaining unit with the nearest activation amplitude. Because nearest amplitudes can still differ, an exact-amplitude control subtracts the target unit's activation times the unrelated unit's decoder direction. Both interventions start from the SAE reconstruction, which is converted back to raw activation units before the frozen language-model head is applied.

SAE ablation damage is measured relative to the reconstructed baseline. Reconstruction-induced KL change is reported separately. Exact-amplitude controls match the removed activation magnitude and decoder-direction norm; their KL difference measures selective output sensitivity.

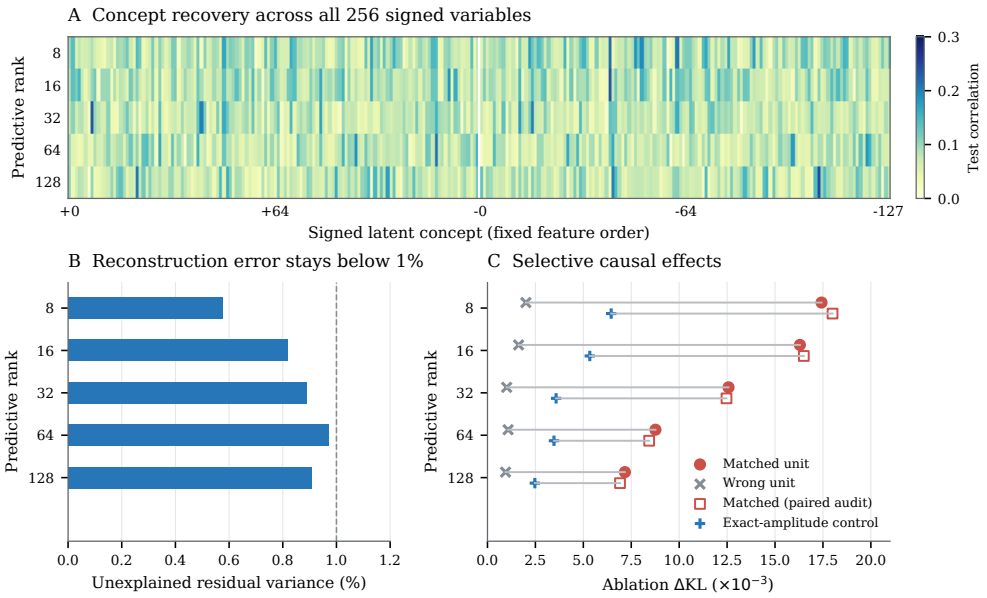

Figure 12: Detailed SAE diagnostics at width 32. (A) Matched-unit correlations for all 256 signed concepts, in fixed feature order. (B) Reconstruction FVU. (C) Matched versus wrong-unit removals and exact-amplitude controls.

Table 23 gives numerical reconstruction and ablation values for the width-32 slice. Table 24 reports the complementary signal-scale and matching diagnostics. Together with Figure 12, these separate fidelity of the activation reconstruction from correspondence to signed ground-truth concepts.

Table 23: SAE evaluation on the width-32 capacity slice.

Rank	FVU	L_0	Concept corr.	Matched ΔKL	Wrong ΔKL
8	0.0057	43.4	0.084	0.0174	0.0020
16	0.0082	48.3	0.083	0.0163	0.0016
32	0.0089	49.9	0.074	0.0126	0.0010
64	0.0097	51.2	0.074	0.0088	0.0011
128	0.0091	51.1	0.076	0.0072	0.0009

I Pairwise Context Modulation

For two active variables (i, j), four legal prefixes instantiate all sign combinations while retaining the other variables and surface context. Denote their activations by $h_{--}, h_{-+}, h_{+-}, h_{++}$. The exact four-corner decomposition is

$$a_0 = (h_{--} + h_{-+} + h_{+-} + h_{++})/4, \tag{31}$$
$$a_i = (-h_{--} - h_{-+} + h_{+-} + h_{++})/4, \tag{32}$$
$$a_j = (-h_{--} + h_{-+} - h_{+-} + h_{++})/4, \tag{33}$$
$$a_{ij} = (h_{--} - h_{-+} - h_{+-} + h_{++})/4. \tag{34}$$

For $s_i, s_j \in \{-1, +1\}$, these coefficients reconstruct each corner as $h_{s_i s_j} = a_0 + s_i a_i + s_j a_j + s_i s_j a_{ij}$. The pairwise term vanishes for any feature-wise additive representation on that quartet. The same decomposition is applied separately to pre-final-normalization activations, final normalized activations, payload logits, and SAE-unit activations.

Table 24: **Signal-scale SAE diagnostics.** Coverage is the fraction of signed concepts with held-out correlation above 0.4. Equal-amplitude controls remove the same activation magnitude from a unit not matched to the target concept.

Scale	FVU	L_0	Concept corr.	Coverage	Decoder alignment	Recon. ΔKL
5	.218	59.42	.322	3.13%	.266	-.0013
7	.251	56.95	.346	15.63%	.371	.0006
10	.383	63.18	.463	80.47%	.590	.0103
14	.384	56.54	.590	98.44%	.781	.0494
18	.356	46.64	.698	100%	.831	.1213

Scale	Matched ΔKL	Equal-amplitude	Excess	SE	Splitting	Merging
5	.000982	.000332	.000651	.000017	19.53%	48.72%
7	.003435	.001357	.002078	.000057	57.81%	29.61%
10	.017857	.006855	.011002	.000205	86.72%	11.34%
14	.063287	.032146	.031141	.000671	96.88%	4.72%
18	.116120	.076564	.039556	.001237	96.09%	3.82%

The implementation records both an interaction amplitude and an energy fraction. They are different summaries:

$$A_{ij} = \frac{\|a_{ij}\|_2}{\sqrt{\|a_i\|_2^2 + \|a_j\|_2^2}}, \tag{35}$$

$$E_{\text{int}} = \frac{\sum_c \|a_{ij}(c)\|_2^2}{\sum_c \left(\|a_i(c)\|_2^2 + \|a_j(c)\|_2^2 + \|a_{ij}(c)\|_2^2\right)}. \tag{36}$$

Table 25 reports interaction energy fractions in percent. SAE values average unitwise fractions; residual values use pooled residual energy.

The final-matrix configuration requests eight contexts per variable pair. With 128 variables, complete coverage gives $\binom{128}{2} = 8,128$ pairs and 65,024 quartets, evaluated on a fixed model. These four-corner comparisons measure pairwise context dependence, including nonlinear effects of normalization and the SAE encoder.

Table 25: **Pairwise context modulation.** Values are second-order interaction energy fractions from legal four-corner interventions (Equation 36); the SAE column averages unitwise fractions. The signal-scale series uses rank 64 / width 128; the capacity slice fixes width 32.

A. Signal-scale series

Scale	Pre-norm	Final norm	Payload logits	SAE units
5	12.22%	8.38%	4.92%	22.32%
7	7.55%	7.37%	4.49%	22.09%
10	7.58%	8.22%	4.82%	19.69%
14	7.74%	8.92%	6.08%	18.79%
18	4.73%	5.93%	4.83%	13.01%

B. Width-32 capacity slice

Rank	Final norm	Payload logits	SAE units
8	0.180%	0.111%	7.84%
16	0.167%	0.112%	9.53%
32	0.188%	0.095%	6.67%
64	0.277%	0.112%	8.12%
128	0.184%	0.140%	3.26%