Testing Refined Local Learning Coefficients Against Mechanism at Billion-Parameter Scale

Anonymous ACL submission

Abstract

The refined local learning coefficient $\hat{\lambda}$ of singular learning theory (Lau et al., 2025) measures the local geometry of a trained model's loss landscape at a specified loss and localization, a different object from the mediated effect of a single-component ablation (Heimersheim and Nanda, 2024; Elhage et al., 2022). We test the correspondence between $\hat{\lambda}$ and mechanism on Pythia-1B (Biderman et al., 2023) applied to $Dyck-(\mathrm{k,m})$ bracket matching: a five-stage circuit identified at the final pretraining checkpoint (named-subspace substitution recovers 0.77 of the clean logit-difference) and traced through training, with refined $\hat{\lambda}$ measured for all 144 components at every released checkpoint. Four properties register in $\hat{\lambda}$ but not in single-component zero-ablation: rankings more stable across the two input distributions we test (ρ_S=0.88 vs 0.27 for greedy); longitudinal tracking of the rate-limiting feature that survives detrending (ρ=0.38 vs -0.01 for greedy, -0.12 for Fisher); the `L0` attention bank registering as one high-codimension block where eight heads jointly encode lexical features that single-head ablation cannot decompose; and a late-training $\hat{\lambda}$ U-shape, with no echo in accuracy, weight Frobenius norm, effective rank, or probe diversity, that coincides with a redundancy transition in the bank ($\hat{\lambda}$ trough: joint zero-ablation 2.80 vs summed per-head 0.08; 1.56 vs 1.33 at convergence). Ablation isolates the single-head causal contribution; refined $\hat{\lambda}$ indexes the local parametric structure that supports it.

1 Introduction

Mechanistic interpretability identifies the heads and MLPs that carry a trained language model's computation on a task (Nanda et al., 2023; Wang et al., 2023; Olsson et al., 2022; Wen et al., 2023). Two of its core tools, single-component ablation and linear probing, have known limitations: ablation results depend on the input distribution the intervention is run against (Heimersheim and Nanda, 2024), and features that the model encodes jointly across multiple components can be underestimated by removing any one component (Elhage et al., 2022). A complementary signal that summarizes the local loss geometry at $\theta^\star$ under a chosen loss, rather than the response of the network to a particular intervention, would help with both.

The refined local learning coefficient $\hat{\lambda}$ (Lau et al., 2025; Wang et al., 2024) is one candidate: it is singular learning theory's empirical estimator of the codimension of the loss zero-set at a trained model (Watanabe, 2009; Hoogland et al., 2025; Carroll et al., 2025), and is by construction a property of the local loss geometry at the trained weights under the loss the sampler uses. Tests of how this signal lines up with mechanism, however, have so far been confined to attention-only two-layer transformers that admit full reverse-engineering (Wang et al., 2024; Gordon et al., 2026) or to larger models with no separately resolved mechanism (Hoogland et al., 2025; Baker et al., 2025).

We test the correspondence on Pythia-1B (Biderman et al., 2023) applied to $Dyck-(\mathrm{k,m})$ bracket matching (Hahn, 2020; Yao et al., 2021; Strobl et al., 2024). Pythia-1B ($\sim$ 1.01B parameters, 16 layers, 8 heads per layer, d_{model}=2048, RoPE) is the smallest Pythia model at billion-parameter scale and provides all 154 released training checkpoints. $Dyck-(\mathrm{k,m})$ is a formal-language task with full ground truth that admits an explicit mechanistic decomposition and that Pythia-1B partially solves at the converged check-

point (close-position accuracy 0.93 at k=2, 0.59 at k=4; on k=4, 86% of incorrect-argmax close positions are non-bracket completions, indicating LM-prior override rather than a bracket-circuit failure, App. D.4). The converged circuit (§4) factors into a token-keyed lookup at `L0.mlp`, a depth counter distributed across the layer-0 attention bank, an opener-attention head `L14H1` whose localization of the matching opener is content-driven rather than rotary (four converging interventions in App. D.5), an answer amplifier at `L11.mlp`, and an LM-prior correction in `L13`–`L15.mlp`. Replacing each component's output by its projection onto a named subspace recovers 0.77 of the clean logit-difference; augmenting each subspace with 64 task-orthogonal principal components lifts recovery to 0.91. The per-checkpoint trace adds a transient-head succession in layer 7 (`L7H4`→`L7H5`→`L7H2`) and an MLP-absorption phase between steps 1,000 and 16,000 (§5, App. G).

Against this mechanism we estimate refined $\hat{\lambda}$ for all 144 components plus 16 layer aggregates and the full model, at all 154 released checkpoints on both task variants (49,588 estimates, $\sim$1,100 A100-hours; §6). Four properties of the trained weights register in refined $\hat{\lambda}$ but not in single-component zero-ablation. $\hat{\lambda}$ rankings are *more stable across the two distributions we test* than greedy ablation rankings (ρ_S=0.88 for $\hat{\lambda}$ between synthetic and natural-language Dyck against 0.27 for greedy). The per-component *trajectory* of $\hat{\lambda}$ correlates with that of distance-to-matching-opener at `L2`, the rate-limiting linear feature, at raw ρ=0.71 and detrended ρ=0.38; among $\hat{\lambda}$, greedy, and Fisher, only $\hat{\lambda}$ survives detrending (-0.01 for greedy, -0.12 for Fisher). The *layer*-0 *attention bank*, whose eight heads share a lexical-feature representation that single-component ablation cannot decompose, registers as a single high-codimension block in $\hat{\lambda}$ (top-23 of 144). Finally, $\hat{\lambda}$ traces a *U-shape* between step 8,000 and step 143,000 on 86 of 144 components with no echo in accuracy (0.617 $\rightarrow$ 0.611), weight Frobenius norm, effective rank, or probe diversity. The U-shape coincides with a redundancy transition in the layer-0 bank: at the trough, the joint zero-ablation drop (2.80) is substantially larger than the sum of per-head drops (0.08), and at convergence the two have aligned (1.56 versus 1.33).

A separate per-checkpoint cross-sectional test on the 27 layer-0–2 components ranks $\hat{\lambda}$ and greedy zero-ablation indistinguishably (paired Wilcoxon p=0.73; §6.6). The four properties above are not an argument that refined $\hat{\lambda}$ replaces ablation as a circuit-discovery tool; they identify what an ablation-based importance score does not measure. The SLT prediction we lean on (Proposition 1, App. I) is an additivity statement for regular local feature constraints under a transversality condition. We cannot verify this condition for the full language-model loss. Instead, we report a first-order diagnostic: the probe-loss gradients for distance, depth, and type are far from collinear at the checkpoints tested ($\sigma_{\min} \in [0.905, 0.986]$). This supports the regular local model as an approximation but should not be read as a proof of exact RLCT additivity in the full network.

2 Related Work

Mechanistic interpretability. We use the activation-patching (Vig et al., 2020; Meng et al., 2022; Geiger et al., 2024), automatic circuit-discovery (Conmy et al., 2023; Goldowsky-Dill et al., 2023; Syed et al., 2024; Bhaskar et al., 2024), and causal-scrubbing (Chan et al., 2022) toolkit, the residual-stream view of transformer computation (Elhage et al., 2021; Olah et al., 2020), and faithfulness/completeness/minimality from the IOI study (Wang et al., 2023); sparse-autoencoder and feature-circuit work (Cunningham et al., 2024; Bricken et al., 2023; Templeton et al., 2024; Marks et al., 2025) and forward-looking surveys (Sharkey et al., 2025) provide context.

Singular learning theory. The local learning coefficient is grounded in Watanabe's free-energy expansion (Watanabe, 2009, 2010, 2013; Lau et al., 2025) and refined at the component level by Wang et al. (2024); the SGLD estimator (Welling and Teh, 2011) is implemented in `devinterp` (Snell et al., 2026). The stagewise-development programme (Hoogland et al., 2025; Carroll et al., 2025; Chen et al., 2023), the susceptibility framework (Baker et al., 2025; Gordon et al., 2026; Wang et al., 2025; Wang and Murfet, 2026), and recent theoretical and empirical extensions (Wei et al.,

2023; Bushnaq et al., 2024; Urdshals et al., 2025; Lee et al., 2025; Kreer et al., 2026; Wei and Lau, 2024; Saghir et al., 2024; Chen and Murfet, 2025; Pepin Lehalleur et al., 2025; Elliott et al., 2026) provide the SLT background.

Other context. Grokking and emergent-capability work (Power et al., 2022; Liu et al., 2022, 2023; Nanda et al., 2023; Chughtai et al., 2023; Schaeffer et al., 2023; He et al., 2024; Olsson et al., 2022; Singh et al., 2024; Murty et al., 2023) frames training dynamics in phase-transition terms. Formal-language analyzes of transformers (Hahn, 2020; Yao et al., 2021; Bhattamishra et al., 2020a,b; Suzgun et al., 2019; Wen et al., 2023; Chiang and Cholak, 2022; Yang and Chiang, 2024; Weiss et al., 2021; Lindner et al., 2023; Zhou et al., 2024; Hahn and Rofin, 2024; Strobl et al., 2024) and the position-embedding literature (Su et al., 2024; Barbero et al., 2024; Kazemnejad et al., 2023; Song and Zhong, 2023) supply expressivity and rotary-mechanism context. Pythia (Biderman et al., 2023) and probing-of-Pythia work (Rivière and Trott, 2025; Trott, 2025; Alain and Bengio, 2017; Tenney et al., 2019; Hewitt and Manning, 2019; nostalgebraist, 2020; Belrose et al., 2023; Ghandeharioun et al., 2024; Tigges et al., 2023; Park et al., 2024; Arditi et al., 2024) ground the cross-checkpoint setting. The closest relatives are Wang et al. (2024) (rLLC for head specialization), Wang and Murfet (2026) (parenthesis-balancing via SLT susceptibilities), and Wen et al. (2023) (Dyck in small transformers); we extend these by integrating both threads on the same billion-parameter pretrained model.

3 Task and Model

Task. $Dyck-(\mathrm{k,m})$ is the formal language of balanced bracket sequences with k bracket types $\{(,),[,],\{,\},<,>\}_{[:k]}$ and maximum nesting depth m. We evaluate $k{=}2, m{=}10$ and $k{=}4, m{=}16$ at sequence length 128 with opening probability $p_{\mathrm{open}}{=}0.45$. The two values of k test whether the same circuit operates across task difficulty within one formal language. Each bracket character is a single token in Pythia's GPT-NeoX tokenizer when space-prefixed, so we build synthetic Dyck strings prefixed with `Brackets:` that lie on Pythia's natural-language input manifold. We also evaluate on a second, naturally-occurring distribution: the `timaeus/dyck` Hugging Face dataset ("Timaeus Dyck" throughout), Pile-style sequences whose bracket positions inherit context from preceding natural-language tokens.

Model. Pythia-1B (Biderman et al., 2023) has 16 layers with 8 heads per layer, $d_{\mathrm{model}}{=}2048$, $d_{\mathrm{head}}{=}256$, $d_{\mathrm{mlp}}{=}8192$, rotary positional embeddings (Su et al., 2024), and $\sim$1.01B parameters. It is the smallest Pythia model at billion-parameter scale and shares the suite's 154 released training checkpoints; 128 heads plus 16 MLPs give 144 components per checkpoint.

Evaluation. At every position whose target is a closer we report close-position accuracy and the close-position *logit-difference*: the logit of the correct closer minus the maximum logit over the other $k{-}1$ incorrect closers. We use the logit-difference rather than accuracy alone because it separates mechanism failures (the circuit no longer prefers the correct closer) from LM-prior overrides (it does, but the argmax goes to a non-bracket continuation). Mechanistic measurements use fp16 inference; SLT sampling uses fp32 for gradient stability.

Causal interventions. We use zero-ablation (*knockout*: hook output set to zero) for greedy minimum-faithful circuit construction, and mean-ablation (hook output replaced by its per-position batch mean) only where stated. The two are not interchangeable on regular synthetic data: mean-ablation preserves any signal that lives in the per-position batch mean. Linear probes use logistic regression with per-string train/test splits to avoid positional leakage, and the close-position metric itself is verified leakage-free (App. D.1).

4 The Bracket-Matching Circuit

The converged bracket-matching circuit at step 143,000 factors into five stages on 19 named components for $k{=}4$ (9 MLPs and 10 heads); the same five-stage organization operates at every checkpoint where the model is past the formation window. The named components, the linear features they read and write, and the stages that consume them are

the inputs to the $\hat{\lambda}$ comparisons of §6. Detailed stage-by-stage descriptions, the algorithm listing in symbolic and linear-algebraic form, and the geometric characterization of the stack subspace are in App. H. We use k=4 unless stated; per-component knockout drops across k=2 and k=4 correlate at Pearson r=0.957.

Five stages. *(S1) Token lookup.* `L0.mlp`'s output is a function only of the current token: pairwise ℓ_2 distance between `L0.mlp` outputs is 0.000 within paren positions and within square positions, 13.78 across types. Pythia uses RoPE only, so position cannot enter `L0.mlp` at opener positions. *(S2) Distributed depth counter.* Linear R^2(depth) at the close position rises from 0.027 at the embedding to 0.701 after `L0.attn` and 0.973 by `L2`; `L0.mlp` contributes ΔR^2=+0.009. Counting is distributed across all eight `L0` heads ($R^2 \in [0.07, 0.31]$ per head), each carrying open/close-discriminating value vectors with ℓ_2-separation in $[7.6, 14.7]$ that are attention-averaged into a depth proxy. Joint mean-ablation of the bank on synthetic Dyck drops the logit-difference by only 0.14 because synthetic depth grows monotonically with position; the same intervention on Timaeus Dyck (where the per-position batch mean carries no depth signal) drops it by 1.69, and zero-ablation on synthetic drops it by 1.78. *(S3) Opener attention via `L14H1`.* The first two stages make the top-of-stack opener type linearly decodable; `L14H1` attends from the close position to the matching opener and writes the opener's type into the answer direction. Localization is content-driven rather than rotary: rotary-only attention argmaxes at the match in 0% of close positions, content-only in 15.3% (clean 17.6%); a K-content swap drops attention by 65% across 6,212 items; and an α-scan of rotary scaling finds α=1 uniquely optimal (mean attention $0.108 \to 0.163 \to 0.013$ at $\alpha \in \{0, 1, 2\}$). The rotary subspace contributes sharpness around the content-determined peak (App. D.5). *(S4) Readout amplifier.* `L11.mlp` writes +3.85 along the ANSWER direction per close position (the largest direct-logit-attribution contributor across all 144 components). *(S5) Anti-Dyck LM correction.* `L13`–`L15.mlp` have on-average-negative ANSWER-projections (DLAs −1.25, −1.07, −0.79); their net effect is to suppress the bracket prediction in favor of LM-prior continuations.

How much of the circuit this account captures. Substituting each circuit component's output with its projection onto a named subspace $\mathcal{N}_C = \text{TOKEN} \cup \text{STACK}_L^{\text{multi}} \cup \text{ANSWER} \cup \text{POSITION}_L$ ($\sim$100–150 of 2048 residual dimensions) recovers 0.77 of the clean logit-difference; augmenting with 64 task-orthogonal principal components per component lifts recovery to 0.91, matching a rank-128 PCA baseline. The named-subspace bound of 0.77 is what is fully expressible in interpretable terms; the remainder is captured by directions that are useful for the task but that we do not assign a label to. Scrubbing STACK from the stage-2 component set drops cumulative recovery by 61%; scrubbing ANSWER from `L13`–`L15.mlp` *raises* it by 5–8%, the anti-Dyck signature. An independent stack-decoding probe trained on the clean residual loses 34–64% of slot accuracy when STACK is scrubbed from `L1`–`L5.mlp` and is unaffected when a random direction of equal rank is scrubbed (App. H.2).

Two linear features. The SLT comparisons of §6 use the trajectories of two features. *Depth* at `L2` is linearly decodable at random init (R^2=0.86): depth is a linear function of any sufficient projection of the open/close indicator sequence, and the layer-2 residual inherits that structure under chance attention configurations. It saturates at 0.97 by step 2,000. *Distance to the matching opener* at `L2` is the genuinely learned feature: R^2 rises from 0.03 at init to 0.59 by step 2,000 and is the rate-limiter for close-position accuracy. We pre-register distance-R^2 at `L2` as the principal target for the SLT comparisons.

5 Developmental Trajectory

The per-checkpoint feature trajectory grounds the $\hat{\lambda}$ comparisons of §6. Linear R^2(depth) at `L2` is already 0.86 at random init—depth is a linear function of any sufficient projection of the open/close indicator sequence—and saturates at 0.97 by step 2,000. The genuinely learned feature, and the rate-limiter for capability, is distance-to-matching-opener: R^2 rises from 0.03 at random init to 0.47 by step 512 and saturates at 0.59 by step 2,000, in

lockstep with close-position accuracy (0.07 → 0.71). The qualitative learning event is the emergence of distance-aware opener attention. Greedy per-checkpoint circuit traces give five phases summarized in App. G: a transient-head succession `L7H4`→`L7H5`→`L7H2` in steps 1,000–7,000, `L14H1` entering the circuit at greedy order 2 between steps 4,000 and 32,000, and a compressed steady state from step 32,000 onward in which the k=2 circuit contracts to 12 components.

6 Refined LLC against Mechanistic Ground Truth

We estimate the refined LLC $\hat{\lambda} = n\beta(\mathbb{E}_{q^*_\gamma}[\ell] - \ell(\theta^*))$ (Lau et al., 2025) for every (variant, checkpoint, component) triple using the `devinterp` SGLD sampler (Snell et al., 2026; Welling and Teh, 2011) on the close-bracket loss; higher $\hat{\lambda}$ corresponds to less-singular local geometry. SGLD updates are restricted to the component's parameters ($\{W_Q, W_K, W_V, W_O\}$ or $\{W_{\text{in}}, W_{\text{out}}\}$), all other weights frozen. Hyperparameters ($\eta, n\beta, \gamma$, burnin) are calibrated per restriction class at step 32,000 and reused (App. C); we use 2 chains of 60 post-burnin draws, batch size 8, sequence length 128. The full sweep covers both task variants and all 154 released checkpoints: 49,588 $\hat{\lambda}$ estimates, about 1,100 A100-hours. Two baselines are measured on the same grid: a per-component Fisher trace $\sum_\theta (\partial L/\partial \theta_c)^2$ on the synthetic loss, and greedy zero-ablation knockout-drops on Timaeus Dyck. At 12 representative checkpoints we additionally re-estimate $\hat{\lambda}$ with the SGLD chain sampling the Timaeus loss in place of the synthetic one (LLC-Timaeus).

6.1 The formation trajectory and transient ridges

The full-model trajectory moves from $\hat{\lambda}$= −0.17 at random init through +1.93 at step 256, +4.84 at step 1,000, and +5.39 at step 2,000, then plateaus in [4.5, 5.4] through step 143,000 (Fig. 1). The slightly negative value at random init is an estimator output: the local SGLD posterior need not be centered at the localization point, so the $\widehat{\lambda}_\gamma$ estimator can take negative values when the asymptotic local-minimum approximation does not yet hold (the RLCT it estimates is conceptually non-negative). The sharp rise across the formation window matches the stagewise signature of Hoogland et al. (2025); cross-chain CV is 0.022. Close-position accuracy lags $\hat{\lambda}$ by roughly one doubling of training steps (0.49 at step 1,000 when $\hat{\lambda}$=4.84). The transient-ridge prediction of Carroll et al. (2025) confirms directly: every transient head in our mechanistic list (`L7H4`, `L7H5`, `L7H2`, `L3H5`, `L3H6`) together with `L11.mlp` and `L14H1` peaks at step 1,000 and decays by step 32,000, with peak-to-converged ratios 2.7–5.5× (App. E). The 144-component heatmap (Fig. 2) shows additional ridges that the mechanistic list does not predict: the layer-9 aggregate (peak 3.74 at step 1,000, 14× decay by step 15,000), the layer-11 aggregate, and a weaker signature at layer 5. The ridges localize the layer-7 succession to the transient window but do not recover the within-layer ordering: all three `L7` heads peak in $\hat{\lambda}$ at step 1,000, while their greedy peaks span steps 1,000–5,000 (`L7H4` at 1,000, `L7H5` at 2,000, `L7H2` at 5,000). The ordering is a behavioral observable that purely local

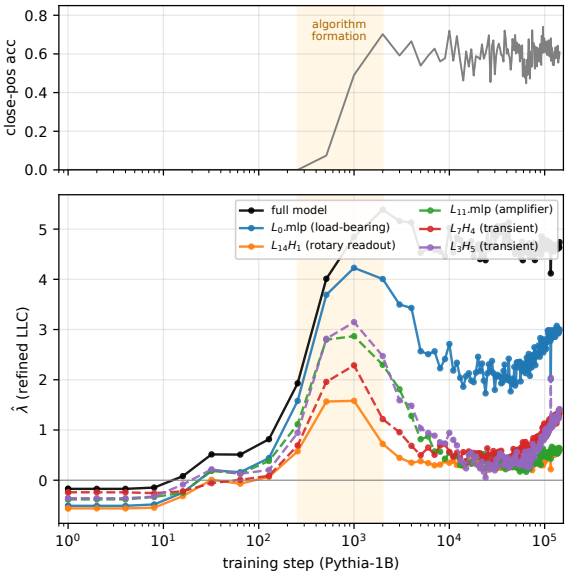

Figure 1: Refined LLC trajectory. (Top) Close-position accuracy on Dyck-(2, 10) across training, formation window 256–2,000 shaded. (Bottom) Component-restricted $\hat{\lambda}$ for the full model and five named components: `L0.mlp` (solid, load-bearing), and `L14H1`, `L11.mlp`, `L7H4`, `L3H5` (transient ridges; the first two remain in the converged circuit). Full-model $\hat{\lambda}$ rises 2.5× across the formation window (1.93 → 5.39, step 256 → 2,000); ridge components peak at step 1,000 and decay 5–30× by step 32,000.

weight geometry does not resolve.

6.2 More stable rankings across input distributions

Fig. 3 plots LLC and greedy on synthetic versus Timaeus Dyck at step 143,000. The median per-checkpoint Spearman correlation between LLC-synth and LLC-Timaeus is ρ_S=+0.82; per-component across 144 × 12 cells, ρ_S=+0.88. The same comparison on greedy knockout gives ρ_S=+0.35 and +0.27. The gap is not driven by inactive cells: restricting to components above the activity threshold $|s_c| > 0.5\sigma(s)$ on either signal still gives ρ_S=+0.77 for $\hat{\lambda}$ versus +0.37 for greedy. Changing the loss the SGLD chain samples barely reorders $\hat{\lambda}$; the same change substantially reorders greedy. $\hat{\lambda}$ measures the local loss geometry at $\theta^\star$ under the loss the chain samples, which is what is stable across the two distributions; a zero-ablation drop is a mediated effect that propagates through the rest of the network under a particular input distribution. We have not shown $\hat{\lambda}$ to be *distribution-invariant* in any strong sense—only that it is more stable than greedy ablation on the two Dyck distributions we test.

6.3 Longitudinal feature-formation tracking

Per-component trajectory correlation against distance-R^2 at `L2` across all 154 checkpoints gives median ρ=0.71 for $\hat{\lambda}$ versus 0.23 for greedy and 0.29 for Fisher; the full-model trajectory cross-correlates with distance-R^2 at ρ=0.78. Because both signals rise during training, the raw correlation could be driven by a shared time trend; we control for this by regressing the rank of training time out of both signals before computing the correlation. The detrended median partial correlation is ρ=0.38 for $\hat{\lambda}$ against −0.01 for greedy and −0.12 for Fisher (App. E). $\hat{\lambda}$ is the only one of the three whose feature-tracking survives detrending. Across feature targets, $\hat{\lambda}$ tracks features at the layer where the circuit reads them: distance at `L2` (ρ=0.71), depth at `L2` (0.53) and at `L14` (0.53), and type at `L6` (0.20, near its at-init saturation ceiling). Depth at `L14` is tracked despite the feature being near its init value at every checkpoint, consistent with $\hat{\lambda}$ indexing the parametric structure that supports a feature in its read-out layer rather than the post-hoc rise of probe accuracy.

6.4 The L0 attention bank as a parametric block

All eight `L0H0`–`L0H7` heads are top-23 of 144 by $\hat{\lambda}$ at convergence. Joint zero-ablation drops close-position LD by +1.78 on synthetic Dyck and +2.28 on Timaeus; synthetic mean-ablation hides the bank (0.14) because synthetic depth grows monotonically with position. Every `L0H*` linearly encodes `is_digit`, `is_space`, and `is_punct` at ≥ 0.91 on Timaeus—a shared lexical-feature bank. The two signals identify the same block as important but disagree on within-block ordering: $\hat{\lambda}$ sees collective parametric structure, zero-ablation sees the mediated per-head effect. The middle MLPs `L4.mlp`–`L8.mlp` are similar: stack-carrying under synthetic mean-ablation is masked by parallel-pathway redundancy but detected by $\hat{\lambda}$.

6.5 Late U-shape: parametric reorganization after capability plateau

Between step 8,000 and step 32,000, $\hat{\lambda}$ falls and then partially recovers by step 143,000 on 86 of 144 components, visible as the late re-darkening on most rows of Fig. 2 and as the second peak in the `L0.mlp` and `L11.mlp` traces in Fig. 1. Median $\hat{\lambda}$ recovery on these components is 0.45 against 0.34 for log-Fisher. The recovery is not a sampler artefact: SGLD hyperparameter ranking is preserved with Kendall τ=1.00 pairwise across $\eta\times2$, $\eta\times0.5$, and $\gamma\times2$ (App. D.8). It is not a distribution artefact: LLC-Timaeus at the same checkpoints correlates with LLC-synth at ρ=+0.88.

Capability is approximately flat across the same window: close-position accuracy is 0.617 at the trough and 0.611 at step 143,000. Static weight statistics do not show the same peak–trough–recovery structure as $\hat{\lambda}$: weight Frobenius norm peaks near the $\hat{\lambda}$ trough and then declines (22→50→42 at peak, trough, converged; ρ with $\hat{\lambda}$ is −0.46), the maximum singular value rises, and effective rank drops sharply early and only partially rebounds (566→277→303). Neither statistic exhibits the late $\hat{\lambda}$-style recovery: using the broader cri-

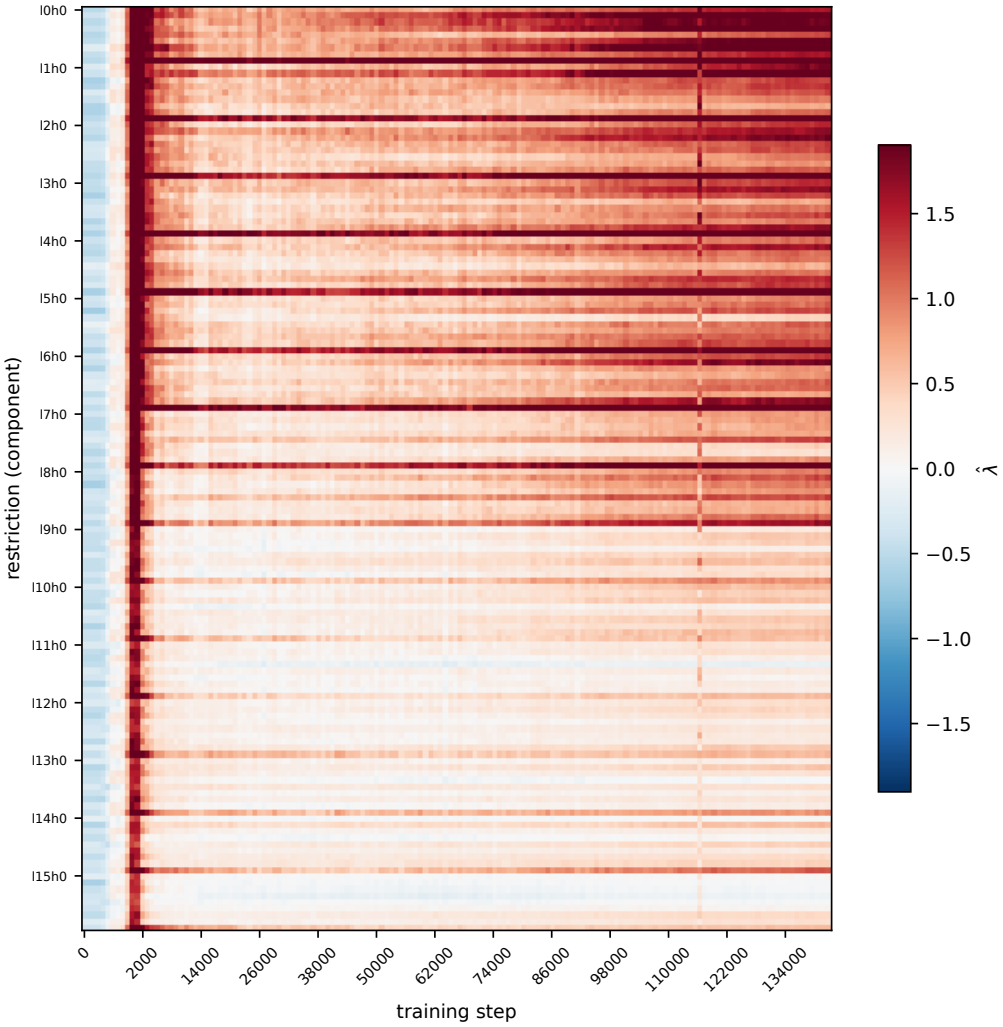

Figure 2: Per-component refined $\hat{\lambda}$ across training, sorted by layer (k=2, m=10). Rows are 144 components (8 heads plus 1 MLP per layer, 16 layers); columns are training checkpoints (log-spaced). High $\hat{\lambda}$ is red; low $\hat{\lambda}$ is blue. The formation band at steps 256–2,000 is visible across most components, with a pronounced transient ridge in layers 7, 9, and 11. After step $\sim$10,000 the U-shape (§6.5) appears as a re-darkening on 86 of 144 rows. `L0.mlp` stays high throughout; the remaining components decay toward $\hat{\lambda} \approx 0$ by step 32,000.

terion of any local minimum in $\hat{\lambda}$ between the formation peak and step 143,000 (which selects 139 components), only 1 has any U-shape in any weight statistic (App. F). Probe decodability of `L0` token-category features is flat at peak, trough, and converged for 8 features on 14 components (e.g. `is_alpha` 0.85→0.81→0.80); no head acquires a new lexical specialization.

A direct ablation experiment connects the U-shape to a redundancy transition in the layer-0 bank. At the trough (step 32,000), joint zero-ablation of the eight `L0` heads drops the logit-difference by 2.80 while the sum of per-head zero-ablation drops is 0.08; at step 143,000 the joint drop is 1.56 and the per-head sum is 1.33 (App. F gives the full trajectory). The absolute joint drop is therefore highest at the $\hat{\lambda}$ trough and decreases monotonically toward convergence—consistent with the eight heads acting as a cooperative ensemble whose collective output is large but whose individual contributions cancel under per-head ablation at the trough, and aligning into individuated contributors whose contributions add at convergence. The joint-to-summed-per-head ratio is large at the trough (36× at step 32,000) but the absolute magnitudes are the primary signal: the ratio is inflated by a near-zero denominator and should be read as *cooperative compensation present* rather than

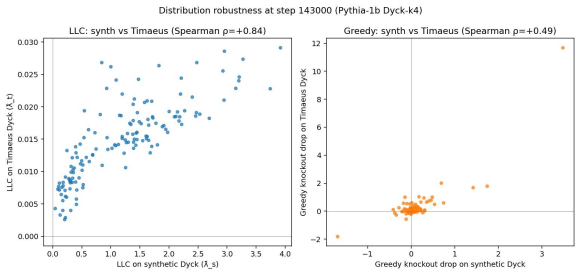

Figure 3: Distribution invariance at step 143,000. Each dot is a component. Left: $\hat{\lambda}$ measured on synthetic Dyck (x) versus on Timaeus natural-language Dyck (y), $\rho_S = +0.83$. Right: greedy zero-ablation knockout-drop on the same two distributions, $\rho_S = +0.21$. Switching the input distribution barely reorders $\hat{\lambda}$; it substantially reorders greedy.

as a precise multiplier. Across the 86 U-shape components, 73% gain in $|\Delta\mathrm{LD}_{\mathrm{Timaeus}}|$ (mean +25%), 72% gain in $|\Delta\mathrm{LD}_{\mathrm{synth}}|$ (+50%), and distance-R^2 at `L2` climbs +0.19. The U-shape thus tracks a parametric specialization that neither the capability metrics nor the standard complexity measurements register on this circuit.

6.6 Proposition 1 consistency and transversality

A direct test of Proposition 1's cross-sectional prediction ($\hat{\lambda}$ correlates with feature contribution) is the following. At each of 12 representative checkpoints we zero-ablate each of the 27 layers-0–2 components and measure the drop in distance-R^2 at `L2`. The cross-component Spearman against $\hat{\lambda}$, greedy on synthetic Dyck, and Fisher: median ρ=0.25/0.28/0.06 across checkpoints; at convergence $0.59/0.45/-0.02$ (bootstrap CI for $\hat{\lambda}$ $[0.29, 0.79]$). $\hat{\lambda}$ and greedy point estimates are not statistically separable at n=27 (paired Steiger Z=0.73; paired Wilcoxon across 12 checkpoints p=0.73). The cross-sectional test is therefore a Proposition-1 consistency check rather than a discriminating one between $\hat{\lambda}$ and greedy: both correlate with feature contribution; Fisher does not. Fisher is a first-order gradient-activity signal rather than a codimension-like invariant; in a deterministic interpolation limit it vanishes after fitting even when the local constraint rank remains present (App. I).

The rank condition in Proposition 1 concerns the full Jacobian of the feature-residual maps, which we do not observe for the language-model loss. As a diagnostic, we compute the parameter-space gradients of the frozen distance, depth, and type probe losses. Their row-normalized gradient stack is well-conditioned at the checkpoints we test ($\sigma_{\min} \in [0.905, 0.986]$ across $\{256, 1000, 2000, 8000, 32000, 143000\}$), so these three scalar feature penalties are far from collinear. This is evidence that the regular transverse model is a reasonable local approximation for the selected features, not a verification of the proposition's full preconditions.

The named $\hat{\lambda}$-only outliers `L4H1`, `L5H4`, `L8H1` each decode `is_bracket` at 1.00 and depth at $R^2 \geq 0.71$. They are not absent from the causal landscape: `L4H1` and `L5H4` zero-ablation drops on synthetic Dyck rank 10 and 21 of 144 by magnitude, with *negative* sign ($\Delta\mathrm{LD} = -0.41$ and -0.22; removing the head improves performance). These heads write LM-prior content that competes with the named circuit.

7 Discussion

Refined $\hat{\lambda}$ and single-component zero-ablation measure different objects on the same trained model. Ablation is the mediated effect of removing one component while the rest of the circuit keeps computing; it isolates per-head causal contribution and within-block ordering, but is sensitive to the input distribution. Refined $\hat{\lambda}$ reads the local geometry of the loss zero-set around the trained weights under a chosen loss; it is less intervention-distribution-specific but does not access within-block orderings. The four observations of §6 are consequences of this difference, not an argument that $\hat{\lambda}$ replaces ablation as a circuit-discovery tool: on the 27 components of §6.6, the two rank indistinguishably. Our results do not establish $\hat{\lambda}$ as a temporal predictor of specific circuit events (App. D.7); its predictive content is structural.

Limitations

Four limitations are worth emphasizing. First, our mechanistic findings concern this model on this task: cross-k generality is strong (r = 0.957 between k=2 and k=4 per-component knockout drops), but generality to other archi-

tectures (non-rotary positional encoding, for example) and to other formal languages is untested.

Second, the SLT estimates depend on SGLD hyperparameters that we calibrate at one checkpoint and reuse across the rest. We measure cross-hyperparameter robustness directly (App. D.8) on six components, four checkpoints, and four hyperparameter variants (default, $\eta \times 2$, $\eta \times 0.5$, $\gamma \times 2$). Cross-hp CV is 3–16% for high-magnitude components (full model, `L0.mlp`) and 31–55% for low-magnitude attention heads (`L14H1`, `L7H4`), but the hyperparameter ranking $\eta \times 2 >$ default $\geq \gamma \times 2 > \eta \times 0.5$ holds at every checkpoint, so the qualitative peak-trough structure is hyperparameter-invariant.

Third, Pythia-1B's close-position accuracy on k=4 Dyck is 0.59, so 41% of close positions are not predicted correctly as argmax. The model still prefers the correct closer to a random distractor on those positions (mean LD +2.22), but 86% of incorrect-argmax predictions are non-bracket tokens: the bracket circuit produces the right relative preference but loses argmax to an LM-prior completion. Stack-scrub on `L1.mlp`–`L5.mlp` damages LD on both subsets (-3.07 on correct, -1.62 on incorrect), and per-component knockout drops correlate at Pearson r=0.957 between the two subsets, so the same Dyck circuit operates on both (App. D.4). Argmax failures arise from LM-prior competition rather than from a different mechanism, and we do not characterize those LM-prior overrides mechanistically.

Fourth, the $\hat{\lambda}$ trajectory does not predict greedy-circuit *events* on a one-step-ahead Granger test (App. D.7, F-statistic tied with shuffled-component null), and on a forward-horizon test against future per-component cross-sectional importance the current cross-sectional importance is itself a stronger predictor than $\hat{\lambda}$. Both findings are consistent with $\hat{\lambda}$ being a property of the static loss geometry at $\theta_\star$ rather than a temporal extrapolator. What $\hat{\lambda}$ does predict is the trajectory of the underlying linear features: median per-component correlation ρ=0.71 with the distance-R^2 trajectory raw and ρ=0.38 after partialing out the shared training-time trend, against 0.23 and -0.01 for greedy and 0.29 and -0.12 for Fisher (§6.3); the raw correlation is the relevant scoreboard for tracking, and the partial correlation is the trend-control showing that $\hat{\lambda}$'s tracking does not collapse when shared monotony is removed. The two observables (linear-feature trajectory and one-step-ahead circuit-event prediction) are not interchangeable, and our results identify which one $\hat{\lambda}$ predicts. We therefore describe $\hat{\lambda}$ as a feature-formation tracker rather than as a circuit-level event predictor.

8 AI assistance

AI assistants were used for writing assistance and for routine coding assistance (boilerplate, plotting code, debugging). AI assistants were not used to generate experimental results, data, or scientific conclusions.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *ICLR 2017 Workshop Track*.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Garrett Baker, George Wang, Jesse Hoogland, and Daniel Murfet. 2025. Structural inference: Interpreting small language models with susceptibilities. *arXiv preprint arXiv:2504.18274*.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. 2024. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. 2024. Finding transformer circuits with edge pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020a. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020b. On the practical ability of recurrent neural networks to recognize hierarchical languages. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Lucius Bushnaq, Jake Mendel, Stefan Heimersheim, Dan Braun, Nicholas Goldowsky-Dill, Kaarel Hänni, Cindy Wu, and Marius Hobbhahn. 2024. Using degeneracy in the loss landscape for mechanistic interpretability. *arXiv preprint arXiv:2405.10927*.

Liam Carroll, Jesse Hoogland, Matthew Farrugia-Roberts, and Daniel Murfet. 2025. Dynamics of transient structure in in-context linear regression transformers. *arXiv preprint arXiv:2501.17745*.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing: A method for rigorously testing interpretability hypotheses. AI Alignment Forum.

Zhongtian Chen, Edmund Lau, Jake Mendel, Susan Wei, and Daniel Murfet. 2023. Dynamical versus Bayesian phase transitions in a toy model of superposition. *arXiv preprint arXiv:2310.06301*.

Zhongtian Chen and Daniel Murfet. 2025. Modes of sequence models and learning coefficients. *arXiv preprint arXiv:2504.18048*.

David Chiang and Peter Cholak. 2022. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. 2023. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning (ICML)*.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*.

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. 2022. Toy models of superposition. *Transformer Circuits Thread*.

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, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Chris Elliott, Einar Urdshals, David Quarel, Matthew Farrugia-Roberts, and Daniel Murfet. 2026. Stagewise reinforcement learning and the geometry of the regret landscape. *arXiv preprint arXiv:2601.07524*.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2024. Finding alignments between interpretable causal variables and distributed neural representations. In *Conference on Causal Learning and Reasoning (CLeaR)*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning (ICML)*.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*.

Andrew Gordon, Garrett Baker, George Wang, William Snell, Stan van Wingerden, and Daniel

Murfet. 2026. Towards spectroscopy: Susceptibility clusters in language models. *arXiv preprint arXiv:2601.12703*.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics (TACL)*, 8:156–171.

Michael Hahn and Mark Rofin. 2024. Why are sensitive functions hard for transformers? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tianyu He, Darshil Doshi, Aritra Das, and Andrey Gromov. 2024. Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4129–4138.

Jesse Hoogland, George Wang, Matthew Farrugia-Roberts, Liam Carroll, Susan Wei, and Daniel Murfet. 2025. Loss landscape degeneracy and stagewise development in transformers. *Transactions on Machine Learning Research (TMLR)*. Preprint arXiv:2402.02364.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Philipp Alexander Kreer, Wilson Wu, Maxwell Adam, Zach Furman, and Jesse Hoogland. 2026. Bayesian influence functions for hessian-free data attribution. *arXiv preprint arXiv:2509.26544*. ICLR 2026.

Edmund Lau, Zach Furman, George Wang, Daniel Murfet, and Susan Wei. 2025. The local learning coefficient: A singularity-aware complexity measure. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pages 244–252.

Jin Hwa Lee, Matthew Smith, Maxwell Adam, and Jesse Hoogland. 2025. Influence dynamics and stagewise data attribution. *arXiv preprint arXiv:2510.12071*.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. 2023. Tracr: Compiled transformers as a laboratory for interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. 2022. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ziming Liu, Eric J. Michaud, and Max Tegmark. 2023. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations (ICLR)*.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations (ICLR)*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. 2023. Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*.

nostalgebraist. 2020. interpreting GPT: the logit lens. LessWrong.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In

International Conference on Machine Learning (ICML).

Simon Pepin Lehalleur, Jesse Hoogland, Matthew Farrugia-Roberts, Susan Wei, Alexander Gietelink Oldenziel, George Wang, Liam Carroll, and Daniel Murfet. 2025. You are what you eat — AI alignment requires understanding how data shapes structure and generalisation. *arXiv preprint arXiv:2502.05475*.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Pamela D. Rivière and Sean Trott. 2025. Start making sense(s): A developmental probe of attention specialization using lexical ambiguity. *arXiv preprint arXiv:2511.21974*.

Moosa Saghir, N. R. Raghavendra, Zihe Liu, and Evan Ryan Gunter. 2024. NGD converges to less degenerate solutions than SGD. *arXiv preprint arXiv:2409.04913*.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Dan Braun, Jake Mendel, Kola Pearce, Nuruddin Syed, Adly Templeton, Hoagy Cunningham, Martin Wattenberg, Trenton Bricken, Nora Belrose, Lee Spies, Yonatan Belinkov, and 4 others. 2025. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research (TMLR)*. Preprint arXiv:2501.16496.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C.Y. Chan, and Andrew M. Saxe. 2024. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In *International Conference on Machine Learning (ICML)*.

William Snell, Johan Sokrates Wind, Billy Snikkers, Sandy Fraser, Adam Newgas, Jesse Hoogland, George Wang, Andrew Gordon, William Zhou, and Stan van Wingerden. 2026. DevInterp. `https://github.com/timaeus-research/devinterp`. Software, version 2.0.

Jiajun Song and Yiqiao Zhong. 2023. Uncovering hidden geometry in transformers via disentangling position and context. *arXiv preprint arXiv:2310.04861*.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. 2024. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics (TACL)*, 12:543–561.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Mirac Suzgun, Sebastian Gehrmann, Yonatan Belinkov, and Stuart M. Shieber. 2019. LSTM networks can perform dynamic counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages (ACL Workshop)*.

Aaquib Syed, Can Rager, and Arthur Conmy. 2024. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, and 7 others. 2024. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4593–4601.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*.

Sean Trott. 2025. Toward a theory of generalizability in LLM mechanistic interpretability research. *arXiv preprint arXiv:2509.22831*.

Einar Urdshals, Edmund Lau, Jesse Hoogland, Stan van Wingerden, and Daniel Murfet. 2025. Compressibility measures complexity: Minimum description length meets singular learning theory. *arXiv preprint arXiv:2510.12077*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural NLP: The case of gender bias. In *Advances in Neural Information Processing Systems (NeurIPS)*.

George Wang, Garrett Baker, Andrew Gordon, and Daniel Murfet. 2025. Embryology of a language model. *arXiv preprint arXiv:2508.00331*.

George Wang, Jesse Hoogland, Stan van Wingerden, Zach Furman, and Daniel Murfet. 2024. Differentiation and specialization of attention heads via the refined local learning coefficient. *arXiv preprint arXiv:2410.02984*.

George Wang and Daniel Murfet. 2026. Patterning: The dual of interpretability. *arXiv preprint arXiv:2601.13548*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*.

Sumio Watanabe. 2009. *Algebraic Geometry and Statistical Learning Theory*, volume 25 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press.

Sumio Watanabe. 2010. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11:3571–3594.

Sumio Watanabe. 2013. A widely applicable Bayesian information criterion. *Journal of Machine Learning Research*, 14:867–897.

Susan Wei and Edmund Lau. 2024. Variational Bayesian neural networks via resolution of singularities. *Journal of Computational and Graphical Statistics*. Preprint arXiv:2302.06035.

Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella. 2023. Deep learning is singular, and that's good. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10473–10486.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2021. Thinking like transformers. In *International Conference on Machine Learning (ICML)*.

Max Welling and Yee Whye Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 681–688.

Kaiyue Wen, Yuchen Li, Bingbin Liu, and Andrej Risteski. 2023. Transformers are uninterpretable with myopic methods: A case study with bounded Dyck grammars. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Andy Yang and David Chiang. 2024. Counting like transformers: Compiling temporal counting logic into softmax transformers. *arXiv preprint arXiv:2404.04393*.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. 2021. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua Susskind, Samy Bengio, and Preetum Nakkiran. 2024. What algorithms can transformers learn? A study in length generalization. In *International Conference on Learning Representations (ICLR)*.

A Per-checkpoint algorithm cards

For each of the 12 checkpoints automatically selected by our transition detector, we maintain a per-checkpoint algorithm card describing the greedy minimum-faithful circuit, the role of each component, and the path-patching edges between them.

B Forward-trace examples

For four data-selected Dyck strings at the converged checkpoint and at three pre-convergence checkpoints, we record forward-trace walkthroughs. For each token position the trace lists the top-15 firing MLP neurons, the attention weights for the in-circuit heads, and the per-component contribution to the close-position logit-difference.

C SLT reproducibility

We use `devinterp` v2.0 (Snell et al., 2026). SGLD hyperparameters are calibrated per restriction class (head, MLP, per-layer aggregate, full model) at reference checkpoint step 32,000 using Optuna with cross-chain CV as the objective. The class-default learning rates are $\{2.5\times10^{-5}, 1.5\times10^{-5}, 10^{-5}, 5\times10^{-6}\}$; the inverse temperature is $n\beta = 1.77$; the localization strength is $\gamma \in \{0.012, 0.025, 0.05, 0.1\}$; and the burn-in is $\{100, 100, 120, 150\}$ steps. We run two chains of 60 post-burn-in draws each at batch size 8 and sequence length 128, all in fp32. The per-component wall-clock is roughly 80 seconds on a single A100, the per-checkpoint cost across all 161 restrictions is about 3.6 A100-hours, and the full sweep across both task variants and all 154 checkpoints is about 1,100 A100-hours (roughly six days on eight A100s in parallel).

D Auxiliary audits

We report eight targeted sanity tests on the paper's most load-bearing claims. Each subsection either confirms the corresponding body claim or restates it more precisely.

D.1 Indexing and label leakage

The close-position metric is evaluated at every position t whose target is a closing bracket; we test whether the closer at position $t{+}1$ leaks into the residual stream the metric reads. On a random Dyck-k=4 batch we verify two invariants. First, is_close_pos[t] holds iff tokens[$t{+}1$] is a closer iff target[t] is a closer; this matches across all $1{,}024 \times 8$ positions tested, with zero inconsistencies. Second, causal masking is intact: after mutating tokens[$t{+}1$], the difference at logits[t] is 0.00 (the model does not see the change), while the difference at logits[$t{+}1$] is 9.63 (it does). Indexing is clean.

D.2 Subspace anti-circularity

The named and PCA bases used in the substitution result of §4 were fit and evaluated on the same batch of 128 strings (seed 42). To rule out fit-batch overfitting we reuse the seed-42 bases and recompute substitution recovery on disjoint batches drawn under seeds 43, 44, and 45. The held-out gap is at most ± 0.012 for every cumulative configuration:

config	fit (s=42)	holdout (s=43–45)
`cum_named2_aug`	0.918	0.918 ± 0.001
`cum_named_aug`	0.907	0.907 ± 0.001
`cum_pca`	0.901	0.903 ± 0.001
`cum_named2_only`	0.775	0.771 ± 0.007
`cum_named_only`	0.443	0.453 ± 0.012
`cum_lin_baseline`	0.114	0.110 ± 0.005

The 0.92 cumulative recovery is therefore not an artefact of fitting and evaluating on the same batch.

D.3 Controls for the depth-decodability claim

§5 claims that depth is linearly decodable from the layer-0 residual already at random initialization. The natural worry is that this could be position bias rather than a real depth feature. We train a ridge probe predicting depth from blocks.0.hook_resid_post and compare it against five controls: a position-only baseline using $\{p, p^2, 1\}$ as features; a position-balanced fit; the same probe evaluated on random-bracket non-Dyck inputs; a regression of depth residuals on position only; and a shuffled-label baseline. The claim survives every control. At step 0:

target	R^2
residuals → depth, held-out	0.636
position only → depth	0.083
position-balanced fit → depth	0.293
residuals → depth on random-bracket inputs	0.445
residuals → depth-residual on position	0.634
shuffled-label baseline	−0.006

Position alone explains only 8% of depth variance, and the position-balanced fit still attains $R^2 = 0.293$. The probe trained on Dyck inputs transfers to random-bracket inputs at $R^2 = 0.445$, so the depth feature is computed from the bracket sequence itself rather than from Dyck-specific co-occurrence. The depth-decodability claim survives the controls.

D.4 Failure-case split

Accuracy on Dyck-k=4 close positions is only 0.59 at the converged checkpoint, raising the question of whether the same circuit operates on the 41% of positions that miss. We split close positions into correct (argmax = target, n = 13,048) and incorrect (n = 8,955), and compare the two along several axes.

The mean logit-difference is 4.39 on the correct subset and 2.22 on the incorrect one: even when the model misses argmax, the correct closer beats a random distractor by more than two logits on average. Of the incorrect predictions, 7,659 (or 86%) are non-bracket tokens (typically Pile-style continuations); the remaining 14% confuse between close types and show roughly uniform type-confusion across the 4×4 off-diagonal.

Scrubbing the named STACK subspace from `L1.mlp`–`L5.mlp` hurts both subsets: $\mathrm{LD}_{\mathrm{correct}}$ falls by 3.07 (from 4.39 to 1.32) and $\mathrm{LD}_{\mathrm{incorrect}}$ by 1.62 (from 2.22 to 0.60). More tellingly, when we zero-ablate each of the 144 components in turn and measure the resulting LD drop separately on the two subsets, the per-component drop vectors correlate at Pearson $r = +0.957$ and Spearman $\rho = +0.830$, and the top-5 components by drop are identical: `L0.mlp`, `L2.mlp`, `L1.mlp`, `L3.mlp`, `L11.mlp`. The same Dyck circuit operates on successes

and failures; failures arise from LM-prior competition for argmax, not from a different mechanism. The algorithm described in this paper is therefore the Pythia-1B Dyck circuit, and the LM-prior overrides on 41% of close positions are not characterized here.

D.5 Interventions on L14H1

This appendix collects the causal interventions that pin down what makes the `L14H1` attention pattern peak at the matching opener. We organise the evidence into three preliminary tests, two more stringent negative tests, and three positive tests that together yield the picture summarized in §4.

Distance binning, zero-RoPE, and RoPE shift. Three preliminary observations characterize the head. First, distance binning: for every binned matching-opener distance $d \in \{2, 4, \ldots, 24\}$ with at least 20 examples, the average attention profile peaks exactly at offset d. The mean attention at the matching opener is 0.146 against a uniform-baseline expectation of about 0.02 at distance 10; 25.6% of individual close positions have argmax exactly at the match, and 36.8% within ± 2. Second, zero-ablating the 64 rotated dimensions of Q and K drops mean attention at the match from 0.146 to 0.096 (-34%) but barely changes the logit-difference (-0.015); the unrotated 192 dimensions carry enough content for the argmax to be preserved. Third, applying a constant RoPE rotation $+\delta$ to Q does not shift the argmax (the binned pattern still peaks at the true offset for every d) but does drop the logit-difference progressively, by 0.14 at $\delta = +7$.

The first two preliminaries are consistent with several mechanisms; the third is consistent with two interpretations of which the relevant negative one is that the `L14H1` argmax is not governed by the probe-extractable linear distance feature on Q. The amplification scan and K-content patch below provide positive causal evidence.

Q-position swap (negative). At a fixed close-position t we replace `L14H1`'s post-rotation $Q[t]$ with $Q[t]$ from a different Dyck sequence whose matching-opener sits at a different distance d' (so the RoPE phase is identical but the content differs). Across six cells with $d_{\text{target}} = 2$ and $d_{\text{source}} \in \{6, 8, 10, 12, 14\}$ ($n \geq 13$ per cell), the patched attention argmax does not move to where the source's matching opener would be; instead it collapses to position 0.

Linear-distance-feature patch (negative). The most stringent intervention along the probe-extractable distance direction. We fit a ridge probe $\hat{d} = w^\top Q + b$ on $Q[t]$ at `L14H1` over a 256-string training batch (train $R^2 = 0.46$) and let $\hat{u} = w/\|w\|$. On a disjoint 128-string evaluation batch we patch $Q[t] \leftarrow Q[t] + (\delta/\|w\|)\hat{u}$, which shifts the probe-readable distance by exactly δ while leaving every other linear feature of $Q[t]$ untouched and letting RoPE operate normally on the patched Q. The hypothesis "argmax = probe-extractable distance feature $\times$ RoPE" predicts that the argmax shifts from d_{true} to $d_{\text{target}} = d_{\text{true}} + \delta$.

The argmax does not move. Across $d_{\text{true}} \in \{2, 4, \ldots, 20\}$ and $\delta \in \{-d_{\text{true}} + 2, +5, +10\}$ ($n \geq 30$ per cell), the fraction of cells where the patched argmax equals d_{target} stays at 0.0–0.08; the fraction at d_{true} remains 0.22–0.40, indistinguishable from clean. Mean attention at d_{target} stays near uniform baseline (0.01–0.04), while mean attention at d_{true} is barely perturbed (for example at $d_{\text{true}} = 10$, $\delta = +10$, attention at d_{true} moves from 0.139 to 0.140, and attention at $d_{\text{target}} = 20$ stays at 0.023). The probe-extractable linear distance feature in Q is therefore a correlate of distance, not a load-bearing direction; this refutes the specific story "probe-extractable distance feature $\times$ RoPE," but it does not by itself identify which subspace is load-bearing. The softmax-isolation test below does.

K-side content patch (positive, 65% attention drop). At each close-prediction position whose matching opener sits at position mp, we patch `L14H1`'s hook_rot_k[batch=b, t=mp] with K[batch=b', t=mp] taken from another sequence whose position mp has a *different* opener type (so the RoPE phase is identical but the bracket-type content differs). Across $n = 6{,}212$ items pooled over 12 type-pair cells, attention at mp drops from a clean 0.183 to a patched 0.063, a 65% reduction overall; for most type pairs, at least 80% of items show a 75–90% drop. Bracket-type content in K

is therefore causally required for L14H1 to attend to the matching opener within the rotary-selected offset band.

Rotary-amplification scan (positive, sharpness at $\alpha = 1$). We multiply the 64 rotated dimensions of Q and K by a scalar α before the dot product. Attention at the matching opener rises monotonically from 0.108 at $\alpha = 0$ to 0.163 at $\alpha = 1$ and then collapses to 0.013 at $\alpha = 2$ and 0.001 at $\alpha = 3$. The logit-difference is flat for $\alpha \leq 1$ and drops by about 0.5 at $\alpha \geq 1.5$. The natural rotary strength is uniquely optimal: both attenuation and amplification monotonically degrade attention sharpness at the match.

Why the linear-distance patch does not shift argmax. A subspace probe shows that distance is encoded redundantly: $R^2 = 0.82$ on the 64 rotated dimensions and $R^2 = 0.87$ on the 192 unrotated dimensions. Pulling Q along a single linear-probe direction in either subspace perturbs the QK product at every offset but not preferentially at the target offset, so the argmax is robust to this perturbation. The amplification scan, by contrast, scales the entire rotary contribution uniformly across offsets, and RoPE's structured cosine/sine pattern then produces a selective peak at the matched distance.

Synthesis. L14H1 is therefore best described as a distance-aware, RoPE-modulated opener-attention head. Bracket-type content in K is causally required for localization (the K-content patch drops attention by 65%, and the softmax-isolation result below shows that rotary-only attention never peaks at the match). The natural rotary scaling $\alpha = 1$ is causally required for sharpness around that content-determined peak. The probe-extractable distance direction in Q is a correlate, not a load-bearing direction.

QK decomposition. Decomposing L14H1's pre-softmax attention score into its rotary and content contributions, $\langle Q[t], K[p] \rangle = \langle Q[t][:64], K[p][:64] \rangle + \langle Q[t][64:], K[p][64:] \rangle$, and averaging each contribution as a function of relative offset Δ over the close-prediction positions at each binned distance d, we find that the rotary part's $\arg\max_\Delta$ equals d for 10 of 12 tested $d \in \{2, 4, \ldots, 24\}$, and the content part's $\arg\max_\Delta$ equals d for all 12. Each subspace independently places the maximum of the pre-softmax score at the matching offset.

Softmax-isolation: rotary alone versus content alone. The decomposition above is averaged over many positions and is thus a population statistic; per-position behavior can differ. We therefore construct the softmax attention pattern from one subspace at a time—the rotary QK score with content zeroed, and the content QK score with rotary zeroed—and ask whether the resulting attention pattern peaks at the matching opener position. Content alone produces argmax-at-d on 15.3% of positions, comparable to the clean head's 17.6%; rotary alone produces argmax-at-d on 0% of positions, with its mass going elsewhere (typically the prefix). Localization at the matching opener is therefore content-driven; rotary modulation contributes sharpness around the content-determined peak (as the α-scan shows) rather than producing the peak itself.

D.6 Does $\hat{\lambda}$ predict mechanistic events better than baselines?

We define a mechanism event at component c and checkpoint t as a relative change in greedy knockout drop of at least 30% between adjacent checkpoints; across the 22,176 component-checkpoint cells, 16.7% are events. We compare five candidate signals: $\hat{\lambda}$, Fisher, $\hat{\lambda}$ time-shuffled within each component, $\hat{\lambda}$ shuffled across components, and $\hat{\lambda}$ replaced by an i.i.d. Gaussian matched in variance. Two metrics are of interest: per-checkpoint rank correlation against greedy and event F1 at a matched event count.

signal	per-ckpt ρ vs greedy	event F1
$\hat{\lambda}$	+0.31	0.23
Fisher	+0.11	**0.40**
$\hat{\lambda}$ time-shuffled	+0.25	0.27
$\hat{\lambda}$ component-shuffled	+0.13	0.37
$\hat{\lambda}$ i.i.d. Gaussian	+0.24	0.22

The two metrics disagree. By instantaneous ranking, $\hat{\lambda}$ beats the proper shuffled-component null (+0.31 versus +0.13); by event F1, Fisher and the shuffled $\hat{\lambda}$ outperform $\hat{\lambda}$. The lead-lag profile reinforces the first conclusion: $\hat{\lambda}$ events co-occur with mechanism

events at a rate that grows monotonically with lag, from 6.7% at lag -3 to 16.3% at lag $+5$, against $+8.1\%$ for the shuffled-component control over the same window. The combined reading is that $\hat{\lambda}$ ranks components more similarly to greedy than baselines do, but $\hat{\lambda}$ events on synthetic Dyck do not sharply predict the timing of greedy events. This is consistent with the framing of §6: $\hat{\lambda}$ is a distribution-robust parametric-structure tracker (§6.2), not a leading indicator of mechanism change.

Why Fisher wins at event detection. If $\hat{\lambda}$ tracks instantaneous ranking better than Fisher (per-ckpt $\rho = +0.31$ versus $+0.11$), one might ask why Fisher beats $\hat{\lambda}$ on event F1 (0.40 versus 0.23). Decomposing F1 by component subset gives the answer:

signal	F1 mech-act.	F1 stage
LLC	0.26	0.22
Fisher	**0.43**	**0.48**
LLC comp.-shuffled	0.38	0.48

All three signals fire zero events in mech-inactive components, so Fisher's advantage is entirely about firing events in mech-*active* components more often than $\hat{\lambda}$ does. This is base-rate co-concentration: Fisher and the mechanism trace are both activity-based, so their events line up on the same component set even when their timings do not coincide. $\hat{\lambda}$ fires a comparable number of total events on mech-active components (5,762 versus Fisher's 5,567 counting TP, FP, and FN; Fisher fires 53% of its events on mech-active components, $\hat{\lambda}$ 50%), but Fisher's true-positive count is much higher (1,501 versus 875) because its event timing correlates with the mechanism trace's: both fire when activity changes. $\hat{\lambda}$'s events fire when parametric structure changes, which on synthetic Dyck does not closely co-occur with activity changes. The fact that the component-shuffled $\hat{\lambda}$ also outperforms $\hat{\lambda}$ on F1 confirms that the F1 advantage is set-membership-driven, not timing-driven. By itself, F1 conflates the two: both Fisher and $\hat{\lambda}$ fire events on activity-rich components, so the gap reflects event-rate distribution rather than temporal alignment.

Direct temporal tests. We therefore replace event-detection F1 with three direct tests of temporal information content (App. D.7): First, the lagged cross-correlation $\langle \hat{\lambda}_c(t), \text{greedy}_c(t+\ell) \rangle$, averaged across components, gives

signal	lag -3	lag 0	lag $+5$
$\hat{\lambda}$	$+0.119$	$+\mathbf{0.125}$	$+0.084$
Fisher	-0.008	-0.016	-0.022
shuffled-component	$+0.026$	$+0.029$	$+0.014$

$\hat{\lambda}$ is roughly four times the shuffled-component null at every lag in $[-10, +10]$, peaking at $\ell = 0$ but staying positive throughout the window. Fisher is at or below the null and turns negative at positive lags. $\hat{\lambda}$ behaves like a smoothed indicator of greedy importance rather than a sharp event-timing predictor.

Second, we use the full-model $\hat{\lambda}$ trajectory with three lags $\{\hat{\lambda}(t), \hat{\lambda}(t-1), \hat{\lambda}(t-2)\}$ to predict whether checkpoint t is in the algorithm-formation phase (steps 256–2,000). The point-estimate held-out ROC-AUC is 1.000 for $\hat{\lambda}$ and 0.957 for Fisher, but with only four positives this is a high-variance estimate. A 1,000-resample stratified bootstrap gives $\hat{\lambda}$ median AUC 1.000 with 95% CI $[0.43, 1.00]$ and Fisher median AUC 0.957 with 95% CI $[0.89, 1.00]$; leave-one-positive-out AUC is 0.93 on $\hat{\lambda}$ versus 0.96 on Fisher. The two signals separate the formation phase comparably; what looked like a 1.000-versus-0.957 gap is within the bootstrap CI on both, and we do not interpret it as a real advantage.

Third, the median distance between $\arg\max_t \hat{\lambda}_c(t)$ and $\arg\max_t \text{greedy}_c(t)$ across 144 components is 48.5 checkpoint indices for $\hat{\lambda}$, 60.0 for Fisher, and 55.5 for the shuffled null. $\hat{\lambda}$ peaks align more tightly with greedy peaks than the null does.

Together these tests show that $\hat{\lambda}$ carries real temporal information about greedy importance—instantaneous magnitude, peak timing, and the formation-phase trajectory—that Fisher does not always match and shuffled-component nulls do not capture. One narrow test, one-step-ahead Granger causality at $p = 3$, is null for $\hat{\lambda}$ ($F = 1.99$) at the level of the component-shuffled null ($F = 1.87$): past $\hat{\lambda}$ does not add to past greedy in predicting next-step greedy beyond what shuffled components do. The combined picture is that $\hat{\lambda}$ stays positively correlated with greedy at every lag and peaks more tightly aligned with greedy, while not predicting one-step-ahead

greedy events.

Predicting greedy-circuit membership from $\hat{\lambda}$. A more direct test of whether $\hat{\lambda}$ carries predictive information is to ask whether a single $\hat{\lambda}$ value at one checkpoint predicts greedy-circuit top-20 membership. We frame three classification problems and compare $\hat{\lambda}$ against Fisher, the absolute magnitude $|\hat{\lambda}|$, a time-shuffled $\hat{\lambda}$, a component-shuffled $\hat{\lambda}$ (the proper null), an i.i.d. Gaussian, and the oracle greedy-knockout score. A logistic regression is trained on each single-feature signal; we report held-out ROC-AUC.

signal	Task A	Task B	Task C
Greedy oracle	0.999	0.927	0.987
LLC	**0.758**	**0.698**	0.729
|LLC| magnitude	0.776	0.629	0.719
LLC time-shuffled	0.727	0.722	0.720
Fisher	0.688	0.580	0.745
LLC component-shuffled	0.643	0.438	0.694
Random Gaussian	0.493	0.337	0.509

Task A is the per-cell prediction of whether component c is in the top-20 greedy knockouts at checkpoint t, with training on checkpoints 0–106 and testing on 107–153; Task B is the per-component prediction of whether c's greedy trajectory is a transient ridge (peak at $\leq$ 2,000 steps, drop $\geq$ 50% by 32,000); Task C is the prediction of final-checkpoint top-20 membership from an early-checkpoint feature, with 5-fold CV over components. $\hat{\lambda}$ beats Fisher and the component-shuffled $\hat{\lambda}$ on Tasks A and B; on Task C $\hat{\lambda}$ and Fisher tie. On Task A, the magnitude-only baseline $|\hat{\lambda}|$ is slightly higher than signed $\hat{\lambda}$ (0.776 versus 0.758); on Task B, the time-shuffled $\hat{\lambda}$ is slightly higher than the original (0.722 versus 0.698). Both observations point in the same direction: most of $\hat{\lambda}$'s predictive signal lives in its component-level magnitude and marginal distribution, not in its specific temporal ordering. $\hat{\lambda}$ magnitudes and rankings carry predictive information about which components matter to greedy, beating Fisher and component-shuffled baselines; temporal dynamics add only modest additional predictive value on these tasks.

D.7 Temporal information in $\hat{\lambda}$

This appendix collects the temporal tests of $\hat{\lambda}$ that supplement the body discussion in §6.2.

Lagged cross-correlation. The per-component standardized cross-correlation $\langle \hat{\lambda}_c(t), \text{greedy}_c(t+\ell) \rangle$ averaged across 144 components gives +0.12 across all lags $\ell \in [-10, +10]$ for $\hat{\lambda}$, peaking at $\ell = 0$, against ≤ 0 for Fisher and +0.03 for the shuffled-component null: $\hat{\lambda}$ is roughly 4× the null at every lag.

Granger causality. A per-component F-test of $\hat{\lambda}(t-1 \ldots t-p)$ Granger-causing greedy(t) at $p = 3$ gives

signal	med. F	mean PVE	frac. sig.
$\hat{\lambda}$	1.99	0.060	0.389
Fisher	1.09	0.030	0.160
shuffled-component	1.87	0.057	0.375

(*frac. sig.*: fraction of components with $p < 0.05$.) $\hat{\lambda}$ is tied with the shuffled-component null on Granger causality: past $\hat{\lambda}$ does not add to past greedy in predicting next-step greedy beyond what shuffled components already do. This test is a negative result for one-step-ahead prediction.

Peak-timing alignment. The median of $|\arg\max_t \hat{\lambda}_c(t) - \arg\max_t \text{greedy}_c(t)|$ across 144 components is 48.5 checkpoint indices for $\hat{\lambda}$, 60.0 for Fisher, and 55.5 for the shuffled null; $\hat{\lambda}$'s peak aligns most tightly with greedy's.

Phase-boundary detection. Using $\{\hat{\lambda}(t), \hat{\lambda}(t-1), \hat{\lambda}(t-2)\}$ of the full-model trajectory as features to a logistic regression that predicts whether checkpoint t is in the algorithm-formation phase (steps 256–2,000, four checkpoints), the point-estimate held-out ROC-AUC under a 70/30 random temporal split is 1.000 for $\hat{\lambda}$ and 0.957 for Fisher. With only four positives this is a high-variance estimate: a 1,000-resample stratified bootstrap gives $\hat{\lambda}$ median AUC 1.000 with 95% CI $[0.43, 1.00]$ and Fisher median 0.957 with CI $[0.89, 1.00]$, and leave-one-positive-out AUC is 0.93 on $\hat{\lambda}$ versus 0.96 on Fisher. The two signals separate the formation phase comparably.

Summary. $\hat{\lambda}$ carries informative temporal structure (lagged cross-correlation $\approx$ 4× the shuffled null and tighter peak alignment with greedy than Fisher), but it is not a sharp one-step-ahead event predictor (Granger causality tied with the null), and its formation-phase separability is comparable to Fisher's once bootstrap CIs are accounted for.

D.8 SGLD hyperparameter sensitivity

The full-grid SGLD hyperparameters are calibrated at step 32,000 and reused at every other checkpoint, raising the question of whether this biases cross-checkpoint $\hat{\lambda}$ comparisons. We recompute $\hat{\lambda}$ for six representative components $\{\text{full}, \texttt{L0.mlp}, \texttt{L14H1}, \texttt{L11.mlp}, \texttt{L7H4}, \texttt{L9.mlp}\}$ at four checkpoints $\{1{,}000, 8{,}000, 32{,}000, 143{,}000\}$ under four hyperparameter variants: the default, $\eta \times 2$, $\eta \times 0.5$, and $\gamma \times 2$. We report per-cell cross-hp CV $\sigma^{\text{hp}}_{\hat{\lambda}}/|\bar{\lambda}^{\text{hp}}|$:

component	step 1k	step 8k	step 32k	step 143k
full	0.04	0.16	0.13	0.08
`L0.mlp`	0.03	0.15	0.15	0.08
`L9.mlp`	0.09	0.32	0.40	0.23
`L11.mlp`	0.09	0.33	0.45	0.31
`L7H4`	0.19	0.46	0.52	0.36
`L14H1`	0.31	0.53	0.55	0.49

Cross-hp CV is small (3–16%) for high-magnitude components (full model, `L0.mlp`) and large (31–55%) for the low-magnitude attention head `L14H1`. The pattern mirrors cross-chain CV, which is also small at high $|\hat{\lambda}|$ and large at small. The ranking of hyperparameter variants is, however, consistent across all four checkpoints: $\eta \times 2$ is always largest, $\eta \times 0.5$ is always smallest, and the default and $\gamma \times 2$ coincide in our seed choice. Per-component $\hat{\lambda}$ rankings are even more stable: across the 24 hp-pair-by-checkpoint combinations ($\binom{4}{2} = 6$ pairs $\times 4$ checkpoints), the Kendall τ between the six-component ordering produced by each pair of hyperparameter variants is $+1.000$ in every single case (and so is Spearman ρ). The within-checkpoint per-component ranking is therefore preserved exactly across hyperparameter choice, even where absolute $|\hat{\lambda}|$ scales by a factor of two. The qualitative peak-trough shape of each trajectory is invariant to the calibration we pick; only the absolute scale shifts. Cross-checkpoint $\hat{\lambda}$ comparisons for high-magnitude components are robust to hyperparameter drift; comparisons for low-magnitude attention heads should be read ordinally rather than as absolute differences, and the τ=1.000 result confirms that the ordinal reading is itself stable.

E Trajectory details

Full-model and per-component peaks. The full-model trajectory moves from $\hat{\lambda} = -0.17$ at random init through $+0.51$ at step 32, $+1.93$ at step 256, $+4.01$ at step 512, $+4.84$ at step 1,000, $+5.39$ at step 2,000, then plateaus at 4.5–5.4 through step 143,000 (Fig. 1). Per-component peak and trough values:

component	$\hat{\lambda}$@256	peak / step	$\hat{\lambda}$@143k	peak/conv
`L0.mlp`	1.58	4.23 / 1k	3.00	1.4×
`L14H1`	0.58	1.58 / 1k	0.59	2.7×
`L11.mlp`	1.11	2.87 / 1k	0.62	4.6×
`L7H4`	0.69	2.29 / 1k	1.34	1.7×
`L7H5`	0.84	2.23 / 1k	0.41	5.5×
`L7H2`	0.74	2.42 / 1k	0.86	2.8×
`L3H5`	0.94	3.15 / 1k	1.42	2.2×
`L3H6`	0.92	3.27 / 1k	1.04	3.1×

Heatmap. The body-text heatmap (Fig. 2) is shown for k=2; the corresponding k=4 heatmap is qualitatively the same.

Per-cell uncertainty. Across all 49,588 cells, only 471 (0.95%) combine CV ≥ 0.5 with $|\hat{\lambda}| \geq 1$, and only 92 (0.19%) combine the same CV threshold with $|\hat{\lambda}| \geq 1.5$ (Fig. 4).

Detrended trajectory correlation across feature targets. The raw per-component trajectory correlation between $\hat{\lambda}$ and a feature's R^2 shares a monotone training-time trend; the partial correlation regressing out the rank of training time from both signals isolates the component of $\hat{\lambda}$'s tracking that is not trend-driven. Median per-component values:

feature	ΔR^2	raw		partial	
		$\hat{\lambda}$	greedy	$\hat{\lambda}$	greedy
distance@L2	+0.65	+0.71	+0.23	+0.38	−0.01
depth@L14	−0.03	+0.53	+0.12	+0.32	+0.00
depth@L2	+0.08	+0.53	+0.29	−0.09	+0.01
type@L6	+0.15	+0.20	+0.12	−0.10	+0.00

The pre-registered target (distance@L2) detrends cleanly to ρ=0.38 for $\hat{\lambda}$ versus -0.01 for greedy. Depth@L14, read at the answer-writing layer, also shows non-trivial detrended $\hat{\lambda}$ tracking ($+0.32$) where greedy is at chance, despite depth-R^2 at L14 being near its at-init value throughout training. The saturated features (depth@L2 init 0.86, type@L6 init 0.75) detrend to near zero for both signals, consistent with the absence of additional learning to track after the linear training-time trend is removed.

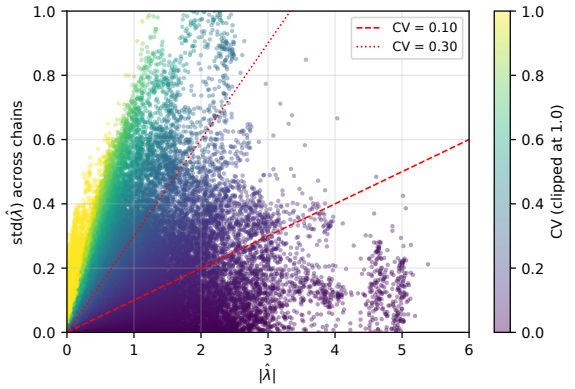

Figure 4: $|\hat{\lambda}|$ vs. cross-chain std for all 49,588 measured cells, colored by CV. Cells with CV ≥ 0.5 concentrate in the small-$|\hat{\lambda}|$ region.

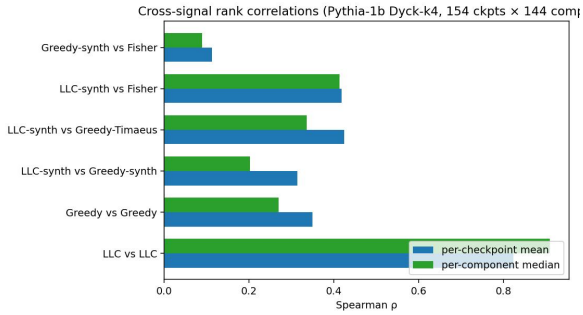

Figure 5: Cross-signal rank correlations. Spearman ρ between each pair of signals, averaged per-checkpoint (blue) and per-component (green). LLC-synth $\leftrightarrow$ LLC-Timaeus is the strongest cross-signal agreement.

F Cross-distribution and curvature analysis

This appendix expands the cross-signal analysis of §6.2. We classify each (component, checkpoint) cell into one of eight categories (Inactive, LLC-only, Greedy-only, Fisher-only, Timaeus-LLC-rich, Anti-Dyck, Agreement, Mixed) using z-score thresholds of ± 1.0 on the four available signals (LLC-synth, LLC-Timaeus, greedy-synth, Fisher). Across $154 \times 144 = 22{,}176$ cells the class frequencies are Inactive 78.0%, LLC-only 6.8%, Anti-Dyck 6.1%, Mixed 4.6%, Agreement 2.1%, Fisher-only 1.2%, Greedy-only 0.7%, and Timaeus-LLC-rich 0.5% (the last over the 12 Timaeus checkpoints, 120 of 1,728 Timaeus-evaluable cells). The dominant non-Inactive class is LLC-only (1,498 cells). Fig. 5 shows the pairwise Spearman correlations.

Outlier probes for $\hat{\lambda}$-only components. We probe the three named LLC-only outliers `L4H1`, `L5H4`, `L8H1` at step 143,000 on a 128-string evaluation batch. For each head we cache the per-head output and fit linear probes for five features (§6.3):

feature (n)	`L4H1`	`L5H4`	`L8H1`
`is_bracket` (acc)	1.00	1.00	1.00
`is_open|bracket` (acc)	0.95	0.92	0.97
4-way bracket type (acc)	0.87	0.75	0.83
depth at close (R^2)	0.71	0.85	0.86
target-is-close (acc)	0.70	0.70	0.71
position-only baseline	0.86	0.86	0.86

All three heads carry rich linear features of the algorithm; the position-only baseline for `is_bracket` is 0.86, well below the 1.00 achieved by the head outputs.

The layer-0 heads in detail. The algorithm-spec listing of Fig. 9 marks `L0H0`–`L0H7` as Stage 0.5 and treats them as task-orthogonal because joint mean-ablation on synthetic Dyck drops the logit-difference by only 0.08. That verdict is an artefact of two features of the synthetic distribution: its input distribution is highly regular, and mean-ablation preserves the per-position batch mean, which already carries the depth signal these heads project. Re-evaluating on natural-language Timaeus input changes the picture. Joint zero-ablation drops LD by +1.78 on synthetic and +2.28 on Timaeus; mean-ablation drops it by +0.14 on synthetic and +1.69 on Timaeus. Per-head zero-ablation at step 143,000 drops Timaeus LD by +0.74 for `L0H0`, +0.66 for `L0H2`, and +0.66 for `L0H4`, roughly 1.5 to 3 times the synthetic numbers. Linear probes trained on $32k$ Timaeus tokens reach ≥ 0.96 accuracy on every `L0H*` for each of `is_bracket_open`, `is_bracket_close`, `is_digit`, `is_space`, `is_punct`. At step 143,000 all eight heads have z-scored $\hat{\lambda} \geq +1.0$, z-scored greedy-synth ≤ -0.4, and z-scored LLC-Timaeus $\geq +0.8$ (Fig. 6).

Middle MLPs in TIMAEUS_LLC_RICH. The components most consistently classified TIMAEUS_LLC_RICH across the 12 Timaeus checkpoints are `L4.mlp` (11/12), `L8.mlp` (11/12), `L6.mlp` (10/12), `L7.mlp`

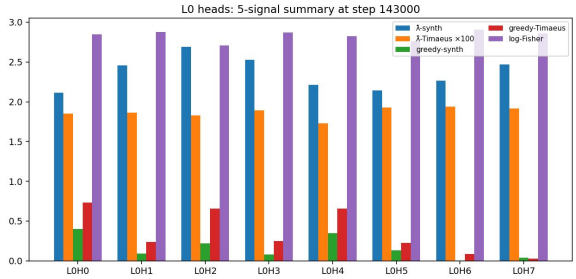

Figure 6: `L0` heads: 5-signal summary at step 143,000. All eight have high $\hat{\lambda}$ on both synthetic and Timaeus (Timaeus values shown $\times 100$ for visualization), large greedy-knockout drops on Timaeus, and small drops on synthetic. The probe channels labeled below each head are at ≥ 0.96 accuracy.

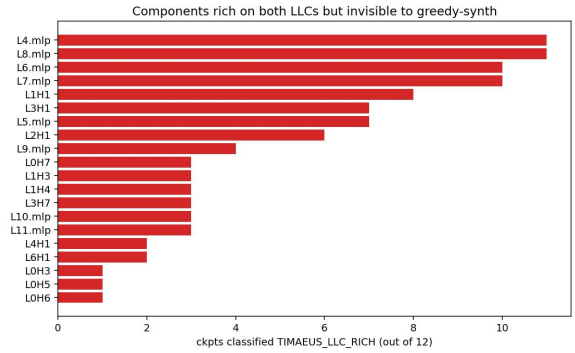

Figure 7: Components flagged as TIMAEUS_LLC_RICH across the 12 Timaeus-LLC checkpoints. The middle-layer MLPs (`L4.mlp`–`L8.mlp`) dominate.

(10/12), `L1H1` (8/12), `L3H1` (7/12), `L5.mlp` (7/12), and `L2H1` (6/12) (Fig. 7). These are the same middle MLPs identified in App. H.2 as stack-carrying (each with single-substitution recovery 0.91–0.96 under the broader named subspace), but which synthetic mean-ablation under-rates because the stack representation is redundantly carried by parallel-residual contributions across `L1`–`L8`.

Per-component U-shape recovery. Recovery fraction $(\hat{\lambda}_{143k} - \hat{\lambda}_{\text{trough}})/(\hat{\lambda}_{\text{peak}} - \hat{\lambda}_{\text{trough}})$ for the 86 U-shape components, comparing $\hat{\lambda}$ against log-Fisher:

component	$\hat{\lambda}$ rec	log-Fisher rec
`L3.mlp`	0.90	0.23
`L0.mlp`	0.88	0.17
`L1H1`	0.88	0.42
`L6.mlp`	0.87	0.27
`L5.mlp`	0.78	0.21
`L8H1`	0.76	0.59

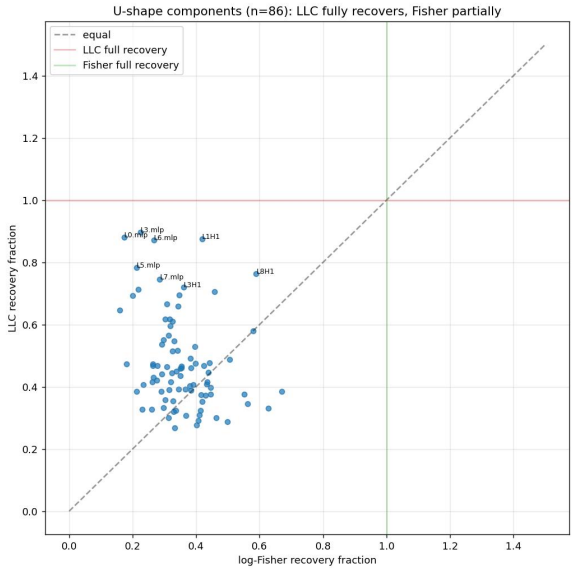

Figure 8: Per-component recovery fraction for $\hat{\lambda}$ (x-axis) and log-Fisher (y-axis) across the 86 U-shape components. The diagonal would be perfect agreement.

The scatter for all 86 components is in Fig. 8; most points sit substantially below the diagonal, indicating that $\hat{\lambda}$ recovers more of the peak amplitude than log-Fisher does.

What the U-shape is mechanistically. Between trough (step 32,000) and converged (step 143,000), synthetic close-position accuracy is essentially flat ($0.617 \to 0.611$, $\Delta_{\text{acc}} = -0.006$), but the bracket-circuit's parametric implementation continues to develop on several jointly-measured axes:

metric	trough	converged
close-pos. acc (synth)	0.617	0.611
dist-R^2@L2 (synth)	0.493	0.684
full-model $\hat{\lambda}$	4.06	4.77
mean $\lvert\Delta\text{LD}\rvert$, Timaeus	0.291	0.365
mean $\lvert\Delta\text{LD}\rvert$, synth	0.127	0.191
# rising $\lvert\Delta\text{LD}\rvert$, Timaeus	—	63 (73%)
# rising $\lvert\Delta\text{LD}\rvert$, synth	—	62 (72%)

The mechanism is therefore not capability gain but *continued feature elaboration after capability has plateaued*: distance at L2 gains $+0.19$ in R^2, 73% of U-shape components gain in their Timaeus knockout-magnitude with a $+25.4\%$ mean rise, and 72% gain on synth with a $+50\%$ mean rise. The largest component-level $\hat{\lambda}$ recovery is concentrated in layers 0–8 (mean $\Delta\hat{\lambda}$ $+1.0$–$+1.4$) versus $+0.2$–$+0.4$ in layers 9–15.

A targeted experiment localizes the parametric correlate of this elaboration at the

L0 attention bank. We perform joint zero-ablation of all eight L0 heads at six checkpoints spanning the U-shape window and compare the joint ΔLD to the sum of per-head $\Delta\mathrm{LD}_h$ on synthetic Dyck (signed values, not absolute):

step	joint ΔLD	$\sum_h \Delta\mathrm{LD}_h$	signed ratio	bank $\hat{\lambda}$
1,000	-0.15	-0.27	0.56	2.97
8,000	$+1.38$	-0.08	-17.4	1.34
27,000	$+2.38$	-0.06	-37.0	0.79
32,000	$+2.80$	$+0.08$	36.0	0.91
64,000	$+2.79$	$+0.70$	4.0	1.20
143,000	$+1.56$	$+1.33$	1.2	2.25

The *signed ratio* column is joint divided by summed per-head ΔLD and is interpretable only when the per-head sum is dominantly positive; the negative values at steps 8,000 and 27,000 reflect a near-zero, sign-mixed denominator and should not be read as a meaningful multiplier. The joint ΔLD trajectory itself is the primary signal: it rises from near zero to 2.80 at the trough and falls to 1.56 at convergence as the per-head sum aligns with it. During the $\hat{\lambda}$ trough window (steps $\sim$ 8,000–32,000), the L0 bank is super-additive: ablating any single head individually does almost nothing ($\sum_h \Delta\mathrm{LD}_h \approx 0$) while ablating all eight jointly drops bracket prediction by 1.4–2.8 LD. The eight heads behave as a perfectly redundant cooperative ensemble in which any seven cover for the missing one. Between trough and converged the redundancy reverses: the ratio joint/sum collapses from ≥ 4 (high redundancy) to 1.17 (near-independent), and each head individually contributes (per-head sum $\sum_h |\Delta\mathrm{LD}|$ rises from ~ 0 at trough to 1.33 at converged). The L0-bank mean $\hat{\lambda}$ recovers in lock-step with this redundancy collapse ($0.79 \to 2.25$). The mechanism the data supports is *specialization from a redundant cooperative ensemble to individuated contributors*: at the trough the eight parameter configurations performing bracket-detection are interchangeable; at the converged checkpoint each has acquired a uniquely-required role. Parametric configurations that were degenerate (any seven of eight heads suffice) become non-degenerate (each head individually carries load), which is the property $\hat{\lambda}$ is constructed to measure.

The U-shape is not visible in any static-weight statistic. Across 14 checkpoints spanning $\{0, 256, \ldots, 143000\}$ we compute, per-component, weight Frobenius norm and effective rank of the stacked weight matrices on GPU. The weight Frobenius norm rises monotonically (mean across the 139 components with non-trivial U-shape: $21.7 \to 24.2 \to 47.0 \to 41.9$ at steps 0/2000/24000/143000) and correlates with the $\hat{\lambda}$ trajectory at median Spearman -0.46; only 1 of 139 has a weight-norm U-shape. The effective rank drops monotonically (mean: $566 \to 277 \to 296 \to 303$) and correlates weakly with $\hat{\lambda}$ (median $+0.10$). Linear-probe decodability of lexical features on Timaeus at peak/trough/converged (step 1000/27000/143000, 14 representative components, 8 features) is high at all three checkpoints (mean 0.82–0.98 per feature) with mean change from trough to converged $\leq +0.030$ per feature. The U-shape is therefore visible neither in weight magnitude nor in effective rank nor in lexical-feature decodability; it is visible in the bracket-task feature trajectory, in the per-component causal-contribution trajectory on both distributions, and in the L0-bank redundancy-collapse measurement — all of which $\hat{\lambda}$ tracks as Proposition 1 predicts.

Cross-sectional test stratified by feature-formation stage. The cross-sectional Spearman of §6.3 is reported as a median across 12 checkpoints because Proposition 1 predicts no signal before the feature exists. Stratifying the 12 checkpoints by feature-formation stage—*pre-formation* (distance-R^2 at L2 < 0.2, two checkpoints), *forming* (0.2–0.5, one checkpoint), and *converged* (> 0.5, nine checkpoints)—makes the pattern explicit:

stage	n	$\hat{\lambda}$	greedy	Fisher
pre-formation	2	-0.18	$+0.07$	-0.14
forming	1	$+0.31$	$+0.15$	$+0.26$
converged	9	$+0.26$	$+0.33$	$+0.10$

$\hat{\lambda}$'s correlation with the per-component $\Delta R^2_{\mathrm{distance}}$ rises monotonically with formation stage as theory predicts; greedy is comparable at convergence and below at the formation window; Fisher is at chance throughout. At convergence the 1,000-resample bootstrap 95% CI for the n=27 $\hat{\lambda}$ Spearman is $[0.29, 0.79]$, excluding zero; the greedy CI is $[-0.04, 0.78]$ and the Fisher CI is $[-0.42, 0.41]$.

G Detailed developmental trace

Per-checkpoint R^2. Linear R^2 for depth and distance at layer 2 and for top-of-stack

type at layer 6, across selected checkpoints:

step	$R^2(d)$	R^2(dist)	type@L6	acc
0	0.864	0.034	0.751	0.000
256	0.882	0.336	0.755	0.000
512	0.879	**0.470**	0.761	0.071
1000	0.896	0.521	0.801	0.486
2000	0.904	0.558	0.915	**0.709**
32000	0.949	0.561	0.844	0.630
143000	0.949	0.589	0.836	0.609

Layer-7 head succession. Tracking which components appear in the per-checkpoint greedy circuit reveals a head succession in layer 7. `L7H4` enters the greedy circuit at order 5 by step 1,000 and is joined by `L7H5` at order ~ 4 by step 2,000; both leave the circuit by step 4,000 in favor of `L7H1`, `L7H2`, and `L7.mlp`, all of which are then absorbed into `L7.mlp` alone by step 16,000. By step 32,000 no `L7` head remains in the minimum-faithful circuit, even though `L7H2` retains a non-trivial individual knockout drop of +0.21. The case of `L3H5` is sharper: its prev-token attention pattern strengthens from 0.70 to 0.80 across training, while its individual knockout drop flips from +0.33 at step 2,000 to −0.49 at step 39,000. The head is not unlearned; the model learns to suppress the direction it writes into the residual stream.

H Algorithm specification and verification

H.1 The algorithm, line by line

Fig. 9 gives the algorithm in two parallel forms: a symbolic stack-machine column and a linear-algebraic column whose matrix operations on the residual stream are what Pythia actually carries out. Each line is annotated with the implementing component(s), the named subspace into which the corresponding write lives, and two measured numbers on a held-out n=256 Dyck batch. Suff is the logit-difference recovery when this line's component is replaced by its projection onto the named subspace; Nec is the recovery when the named subspace is instead scrubbed from the component (projected onto its orthogonal complement, so the algorithm-relevant write is removed and everything else remains). A line is faithful if Suff is close to 1.0 and Nec drops noticeably below 1.0, or, for the anti-Dyck contributions of stage 5, rises above 1.0.

H.2 Subspace substitution

For each circuit component C we construct a low-dimensional named subspace $\mathcal{N}_C = \text{TOKEN} \cup \text{STACK}_L^{\text{multi}} \cup \text{ANSWER} \cup \text{POSITION}_L$, totalling roughly 100–150 of the 2048 residual dimensions after orthonormalization. The four named bases are derived from data: TOKEN is the span of `L0.mlp`'s per-token outputs over the bracket and prefix tokens ($\sim$12 dim); $\text{STACK}_L^{\text{multi}}$ is the union of additive depth $\oplus$ type bases at layers $L, L{+}1, \ldots, L{+}5$ ($\sim$60–130 dim); ANSWER is the four centered close-bracket rows of W_U; and POSITION_L is the SVD of the per-position residual mean at layer L ($\sim$14–32 dim, which captures the RoPE-induced positional bias). A substitution replaces C's output with its projection onto $\mathcal{N}_C$ and measures the logit-difference recovery against the clean model. The bases are fit on the same n=128-string batch on which we report recovery; a held-out audit on disjoint seeds reproduces every cumulative recovery number within ± 0.012 (App. D.2). To gauge what each piece of the named structure contributes we also test a narrower variant $\mathcal{N}_C^{\text{narrow}} = \text{TOKEN} \cup \text{STACK}_L \cup \text{ANSWER}$, an augmented variant $\mathcal{N}_C \cup \text{PC}_{64}^{\perp}$ that adds the top-64 component-specific principal components orthogonal to $\mathcal{N}_C$, and a rank-matched but unnamed PCA-128 baseline.

Cumulative recovery. Substituting every circuit component simultaneously yields the bars in Fig. 10. The strictly named subspace $\mathcal{N}_C$ recovers 0.77 of the clean logit-difference on the fit batch and 0.77 ± 0.007 on held-out batches; adding 64 task-orthogonal principal components per component lifts recovery to 0.91 (held-out 0.92 ± 0.001), slightly above the rank-matched PCA-128 baseline at 0.90. The narrower $\mathcal{N}_C^{\text{narrow}}$, which omits POSITION_L and the multi-layer stack, recovers only 0.45: the position-conditioned bias and the cross-layer stack content that the model carries are themselves interpretable once added to the named bases. The low-rank linear-surrogate baseline collapses to 0.11 because its component-wise errors compound under cumulative perturbation.

Per-component sufficiency and necessity. Fig. 11 reports three numbers per

Stage / Components	Symbolic	Linear-algebraic on residual	Measurement
S1. Token lookup `L0.mlp`	`feature[t] = LOOKUP[tokens[t]]`	`resid[t] += L0_LOOKUP[tokens[t]]`	SUFF (lookup) = 1.00
S0.5. Token-category projection `L0H0`–`L0H7`	`cat[t] = CATEGORY[tokens[t]]`	`resid[t] += sum_h L0H_OUT[t, h]`	SUFF (mean) = 0.93; joint zero-ablation drop on Timaeus = 2.28

S2. Progressive stack construction
`L1`–`L5.mlp`, `L1H1`, `L2H1`, `L2H5`
	`if is_open(tokens[t]):`
	`  stack.append(open_type(t))`
	`elif is_close(tokens[t]):`
	`  stack.pop()`
		`d_new = len(stack) + 1`
		`resid[t] += DEPTH[L=5][d_new]`
		`  + TYPE_OFFSET[L=5][d_new, ty]`
		(pop consumed by S3 attention)
			Joint stack-scrub NEC = 0.39 (−61%) per-component SUFF/NEC below

S3. Stack readout via attention
`L7H0/2/4/6`, `L8H1/4`, `L9H0/4/5`, `L13H0`, `L14H1`
	`predict = top_of_stack`	`for (L,H) in READOUT:`
		`  resid[t] += attn(L,H)[t]`
		($\approx \alpha(t) \cdot$ ANSWER[*top*])
			Joint answer-scrub NEC = 0.48 (−52%); strongest single-head NEC: `L14H1` −16%, `L9H4` −8%

S4. Dominant answer writer `L11.mlp`	`logits[CLOSE_top] += BIG`	`resid[t] += L11.mlp(ln(resid[t]))` (+3.85 *along* ANSWER[*top*])	SUFF = 1.04; answer-scrub NEC = 0.82 (−18%)
S5. Anti-Dyck LM correction `L13.mlp`, `L14.mlp`, `L15.mlp`	`logits[close_types] -= LM_prior`	`for L in {13,14,15}:` `  resid[t] += L_mlp(ln(resid[t]))`	DLAs −1.25, −1.07, −0.79; scrubbing ANSWER *raises* LD (+5–8%)
Final. Readout	`prediction = top_of_stack`	`logits = W_U @ ln_final(resid)`	Clean LD = +3.68

Per-component S2 numbers (SUFF from NAMED2 single-substitution; NEC from scrubbing STACK *from one component):* `L1.mlp` 0.92/0.83 (−17%); `L2.mlp` 0.94/0.81 (−19%); `L3.mlp` 0.91/0.83 (−17%); `L4.mlp` 1.01/1.02 (no effect); `L5.mlp` 0.96/0.95 (−5%); `L1H1` 1.01; `L2H1` 0.98. *Per-component S5 NEC*: `L13.mlp` 1.08 (+8%); `L14.mlp` 1.05 (+5%); `L15.mlp` 1.06 (+6%). *Cumulative recovery*: NAMED2 LD = +2.84 (rec. 0.77); NAMED2+task-orth. PCs LD = +3.36 (rec. 0.91).

Figure 9: The Dyck-k=4 algorithm Pythia-1B implements at step 143,000, broken into seven stage cards. Each row names a stage, its Pythia component(s), the symbolic stack-machine operation it implements, the corresponding linear-algebraic write to the residual stream, and the held-out n=256 measurement. SUFF is single-substitution recovery when the component is bottlenecked to the named subspace; NEC is recovery when the named subspace is scrubbed from the component (so NEC< 1 means the subspace was load-bearing and NEC> 1 means scrubbing it improves performance, as in the anti-Dyck stage).

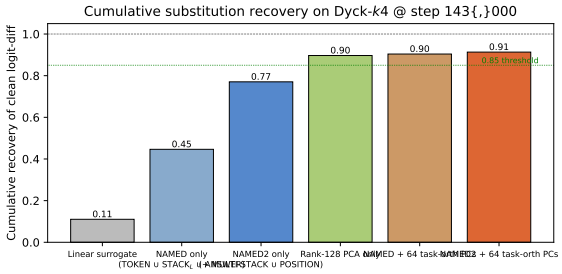

Figure 10: Cumulative substitution recovery: strictly-interpretable algorithm (NAMED2 only) at 0.77; algorithm + 64 task-orth PCs at 0.91 matches the rank-128 PCA baseline. Low-rank linear surrogate fails under cumulative perturbation.

component: single-substitution recovery under the narrow named subspace, the same under the broader named-plus-position subspace, and the recovery after scrubbing the relevant named subspace from that component alone. Under the broader subspace, most components recover close to 1.0; the exceptions are `L1.mlp`, `L2.mlp`, and `L3.mlp`, which retain an unaccounted residual of roughly 8%. The scrubs are likewise consistent with the algorithm: scrubbing STACK from any of `L1.mlp`, `L2.mlp`, `L3.mlp` drops the logit-difference by 17–19%; scrubbing ANSWER from `L11.mlp` drops it by 18%; and scrubbing ANSWER from each of `L13.mlp`, `L14.mlp`, `L15.mlp` *raises* the logit-difference by 5–8%, matching the prediction that the ANSWER-projection of these MLPs is on average negative (the anti-Dyck signature).

Joint necessity. Scrubbing STACK from the entire stage-2 component set (`L1.mlp`–`L5.mlp` plus `L1H1`, `L2H1`, `L2H5`) drops cumulative recovery to 0.39, a 61% decrease. Scrubbing ANSWER from the answer-writing set (the readout heads plus `L11.mlp` and `L13`–`L15.mlp`) drops recovery to 0.48. Each of the two named subspaces jointly carries roughly half of the converged logit-difference.

Independent probe-readout verification. The substitution recoveries above all use logit-difference as the metric. We cross-check the claim that stage 2 writes the stack into STACK using an independent metric. A logistic-regression probe trained on the clean resid-

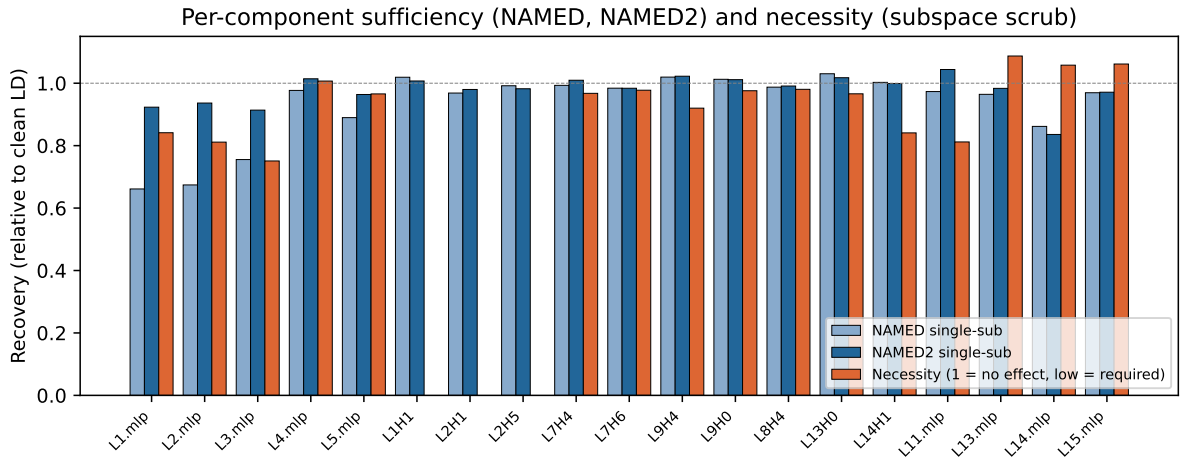

Figure 11: Per-component sufficiency (NAMED single-sub, NAMED2 single-sub) and necessity (scrubbing the relevant named subspace from one component at a time). The broader NAMED2 lifts L1–L3 single-sub from ∼0.7 to ∼0.92, indicating most of their downstream effect is in TOKEN ∪ MULTI-STACK ∪ POSITION ∪ ANSWER. Necessity drops match the algorithm: stage-2 MLPs require STACK, L11.mlp/L14H1 require ANSWER, and L13–15.mlp's ANSWER projection is anti-Dyck (scrubbing helps).

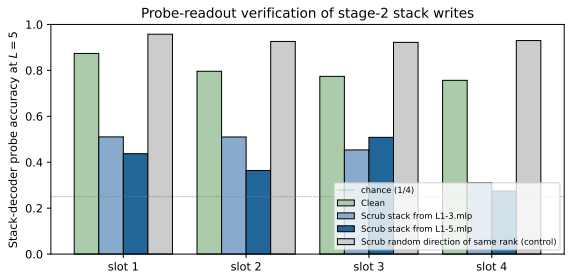

Figure 12: Stack-decoder probe trained on clean resid_post[L=5] and evaluated on perturbed residuals. Scrubbing the STACK subspace from L1–5.mlp destroys 34–64% of probe accuracy; scrubbing a random direction of equal rank leaves accuracy intact. Independent verification that stage 2 writes into the named STACK subspace.

ual resid_post[L=5] predicts the opener type at stack slot i at roughly 80% accuracy for $i \in \{1, 2, 3, 4\}$ (chance = 1/4). Re-evaluating the same probe on the residual produced by scrubbing STACK_L from each of `L1.mlp` through `L5.mlp` drops probe accuracy by 34–64% across slots, while scrubbing a random direction of matched rank from the same MLPs leaves the probe at 92–96% accuracy (Fig. 12). The named STACK subspace specifically, not any equally large subspace, carries the stack content that downstream readers depend on.

H.3 Geometry of the stack representation

The named STACK subspace has internal structure of its own. We fit an additive model

$$r_L[t] \approx b + \sum_i \text{DEPTH}[i] + \text{TYPE}_L[i, \text{stack}[i]]$$

where $r_L[t]$ is the residual stream post layer L at position t, fit on Dyck-k=4 sequences with nested stacks of depth 1 to 15, and characterize the fitted bases.

Slot magnitudes form a U-shape across slots. Fig. 13 (left) plots $\|\text{DEPTH}[i]\|$ per slot at three depths in the network. The bottom slot (slot 1) has a dedicated direction of magnitude 10–25; the top slot (slot 15 when present) is comparable at 11–22; middle slots are suppressed to 2–5. The same U-shape appears at layers 5, 10, and 15, with the absolute scale growing toward the output.

The depth basis is effectively rank-4. A PCA of the 15 slot-occupancy directions gives an interpretable decomposition (Fig. 14). PC1 is the top antenna (almost entirely loaded on slot 15); PC2 is the bottom antenna with near-bottom decay (slots 1–4 fading); PC3 is a bottom-versus-second-slot contrast; and PC4 is a middle plateau shared across slots 3–12. PCs 1–4 together capture roughly 89% of the

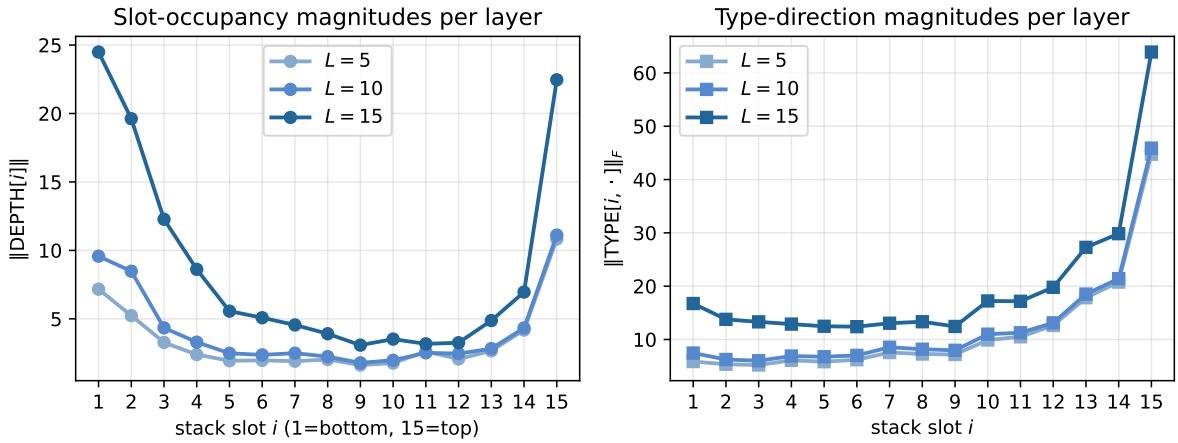

Figure 13: Stack-encoding magnitudes per slot at three layers. Left: slot-occupancy directions ($\|\mathrm{DEPTH}[i]\|$) show a U-shape with dedicated BOTTOM and TOP antennas and a suppressed middle plateau. Right: type-direction magnitudes ($\|\mathrm{TYPE}[i,\cdot]\|_F$) grow monotonically toward the top; the most recent unmatched opener's type is the most strongly encoded.

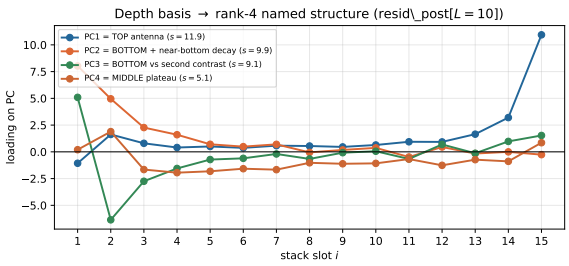

Figure 14: Per-slot loadings on the top-4 principal components of the depth basis at L=10. The 15-slot encoding collapses to a rank-4 structure: TOP antenna, BOTTOM-with-decay, BOTTOM-vs-SECOND contrast, and a MIDDLE PLATEAU shared across slots 3–12.

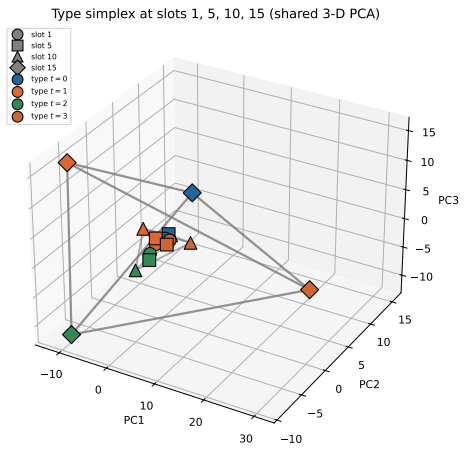

Figure 15: Type tetrahedra at slots 1, 5, 10, 15 in a shared 3-D PCA (resid_post[L=10]). The four type vertices keep the same arrangement (Procrustes cos $\geq$ 0.96) but scale up with depth. The TOP slot's tetrahedron dominates—its type signal is $\sim$6$\times$ stronger than the bottom's.

variance in the depth basis; PCs 5–15 form a spectral tail. The model encodes location in the stack along four interpretable directions, not fifteen.

The type simplex is shape-invariant and depth-scaled. Viewed as a $k \times d$ matrix at each slot i, the type-offset basis is well approximated by a single fixed tetrahedron T scaled by a slot-dependent factor s_i: Procrustes alignment between slot 1 and slot i has cosine $\geq$ 0.96 at every i, while s_i grows from 1.0 at the bottom to roughly 6 at slot 15 (Fig. 15). Bracket-type encoding reuses one geometry across all depths, with the top slot's type encoded roughly six times more strongly than the bottom's.

Taken together, these features describe a bottom-tagged and top-tagged additive code paired with a shared, depth-scaled type tetrahedron. The full 15-slot, 4-type encoding fits in roughly 10 effective dimensions of the 2048-dimensional residual stream, a compression of about 200$\times$ relative to a naive cell-by-cell representation. This compression is what allows stage 3 to implement the matching computation with a single low-rank attention head.

I Theoretical motivation: local rank, refined LLC, and Fisher

This appendix gives the local calculation motivating the comparisons of §6. We do not prove an additivity theorem for arbitrary singular analytic varieties. We prove additivity only in the regular local case where the feature losses are locally equivalent to squared analytic residuals and the corresponding residual Jacobians are transverse. Outside this regular case, the real log canonical threshold (RLCT) can be fractional, non-additive, or sensitive to higher-order singular structure, which is why we estimate the refined LLC empirically rather than computing it from any closed-form construction.

Local RLCT. Let $U \subset \mathbb{R}^d$ be a neighbourhood of a trained parameter $\theta_\star$, let

$$K(\theta) = L(\theta) - L(\theta_\star) \geq 0$$

be a centered population or empirical loss, and let φ be a smooth prior density positive on U. The local zeta function is

$$\zeta_K(s) = \int_U K(\theta)^s \, \varphi(\theta) \, d\theta, \qquad \Re(s) > 0.$$

The local RLCT $\lambda_{\theta_\star}(K)$ is the positive number such that $-\lambda_{\theta_\star}(K)$ is the closest pole of the meromorphic continuation of ζ_K to the origin. Under the usual hypotheses of singular learning theory, the free energy has leading term

$$F_n = nL(\theta_\star) + \lambda_{\theta_\star}(K) \log n \\ - (m-1) \log\log n + O_p(1),$$

where m is the multiplicity of the pole (Watanabe, 2009).

The refined LLC estimator used in the body of the paper is

$$\widehat{\lambda}_\gamma(\theta_\star) = n\beta\big(\mathbb{E}_{q_\gamma^\star}[L(\theta)] - L(\theta_\star)\big),$$

where $q_\gamma^\star$ is the localized posterior sampled by SGLD around $\theta_\star$, with localization strength γ (Lau et al., 2025). When $\theta_\star$ is an exact local minimizer and the asymptotic approximation is valid, $\widehat{\lambda}_\gamma$ estimates the local RLCT. Away from that ideal limit, we interpret it as a local effective-dimensionality statistic rather than as a literal asymptotic invariant.

Local equivalence of losses. Write $A \asymp B$ near $\theta_\star$ if there are constants $0 < c < C < \infty$ and a neighbourhood $V \subset U$ of $\theta_\star$ such that $cB(\theta) \leq A(\theta) \leq CB(\theta)$ for all $\theta \in V$.

Lemma 1 (local equivalence preserves the RLCT). If $K \asymp \widetilde{K}$ near $\theta_\star$, then $\lambda_{\theta_\star}(K) = \lambda_{\theta_\star}(\widetilde{K})$.

Proof sketch. The RLCT is the integrability threshold of K^{-t} near $\theta_\star$. Multiplication by bounded positive constants does not change which powers are locally integrable. Hence K and $\widetilde{K}$ have the same closest zeta pole and the same local RLCT. □

A regular quadratic constraint. The local model for one independently learned linear feature is the following submersion normal form.

Lemma 2 (submersion normal form). Let $G : U \to \mathbb{R}^r$ be real analytic, with $G(\theta_\star) = 0$, and suppose that $DG(\theta_\star)$ has rank r. Define $K_G(\theta) = \|G(\theta)\|_2^2$. Then $\lambda_{\theta_\star}(K_G) = r/2$ with multiplicity $m = 1$.

Proof. By the analytic submersion theorem, there are local analytic coordinates $(y, z) \in \mathbb{R}^r \times \mathbb{R}^{d-r}$ centered at $\theta_\star$ such that $y = G(\theta)$. In these coordinates $K_G(\theta) = \|y\|_2^2$, and the prior density times the Jacobian is a smooth positive density. The singular part of the zeta integral is then the radial integral $\int_{\|y\|<\epsilon} \|y\|^{2s} \, dy = C_r \int_0^\epsilon \rho^{2s+r-1} \, d\rho$, whose closest pole is at $s = -r/2$ with multiplicity one. Therefore $\lambda_{\theta_\star}(K_G) = r/2$. □

Additivity under transverse regular feature constraints. Let $G_j : U \to \mathbb{R}^{r_j}$, $j = 1, \ldots, k$, be analytic feature-residual maps satisfying $G_j(\theta_\star) = 0$ and $\operatorname{rank} DG_j(\theta_\star) = r_j$. The zero set $V_j = \{\theta : G_j(\theta) = 0\}$ is then a smooth local submanifold of codimension r_j. The feature constraints are transverse at $\theta_\star$ if the stacked Jacobian $DG(\theta_\star) = [DG_1(\theta_\star)^\top, \ldots, DG_k(\theta_\star)^\top]^\top$ has rank $R = \sum_{j=1}^k r_j$. Equivalently, the conormal spaces of the V_j form a direct sum.

Proposition 1 (RLCT additivity in the transverse Morse–Bott case). Let $K(\theta) = \sum_{j=1}^k \alpha_j \|G_j(\theta)\|_2^2$ with $\alpha_j > 0$, and assume the stacked Jacobian has rank $R = \sum_j r_j$. Then

$$\lambda_{\theta_\star}(K) = \frac{R}{2} = \sum_{j=1}^k \frac{r_j}{2} = \sum_{j=1}^k \lambda_{\theta_\star}(\|G_j\|_2^2).$$

Proof. Define the stacked weighted residual map $\widetilde{G}(\theta) = (\sqrt{\alpha_1}G_1(\theta), \ldots, \sqrt{\alpha_k}G_k(\theta)) \in \mathbb{R}^R$. The rank assumption gives $\operatorname{rank} D\widetilde{G}(\theta_\star) = R$, and $K(\theta) = \|\widetilde{G}(\theta)\|_2^2$. Lemma 2 then gives $\lambda_{\theta_\star}(K) = R/2$. Applying Lemma 2 separately to each $\|G_j\|_2^2$ gives $\lambda_{\theta_\star}(\|G_j\|_2^2) = r_j/2$, so the displayed additivity formula follows. □

Morse–Bott losses. The squared-residual form above is not restrictive for regular local feature losses. If $K_j \geq 0$ is analytic, has smooth zero set V_j, and its Hessian at $\theta_\star$ is positive definite on the normal space $N_{\theta_\star}V_j$ with kernel $T_{\theta_\star}V_j$, then K_j is Morse–Bott along V_j. In that case $K_j(\theta) \asymp \operatorname{dist}(\theta, V_j)^2$ near $\theta_\star$ and is locally equivalent to a squared residual map of rank $\operatorname{codim} V_j$. Lemma 1 then extends Proposition 1 to sums of transverse Morse–Bott feature losses. This is the precise form of the additivity claim used here.

Component-restricted version. For a component c, SGLD is restricted to the affine parameter subspace $\theta_\star + E_c$, where E_c contains only that component's parameters. Let $G_c(u) = \widetilde{G}(\theta_\star + u)$ for $u \in E_c$, and let $\rho_c = \operatorname{rank} DG_c(0)$. The regular local calculation gives $\lambda_c = \rho_c/2$. If the feature blocks stay transverse after restriction and $\rho_{j,c} = \operatorname{rank} D(G_j|_{E_c})(0)$, this is $\lambda_c = \frac{1}{2}\sum_{j=1}^k \rho_{j,c}$. If the restricted directions overlap, the correct expression is

$$\lambda_c = \tfrac{1}{2}\operatorname{rank}[D(G_1|_{E_c})(0)^\top, \ldots, D(G_k|_{E_c})(0)^\top]^\top,$$

and additivity fails by rank saturation. A larger component-restricted LLC therefore says that perturbing the component exposes more independent locally loss-relevant feature directions; it does not imply a larger single-component ablation effect, which is a mediated causal quantity after the rest of the network has compensated.

Distribution changes. The LLC is not distribution-free in the literal sense: changing the data distribution changes the loss K. What is stable is the RLCT under distribution changes that preserve the local constraint geometry. If two losses K_P and K_Q, induced by two input distributions P and Q, satisfy $K_P \asymp K_Q$ near $\theta_\star$, then Lemma 1 gives $\lambda_{\theta_\star}(K_P) = \lambda_{\theta_\star}(K_Q)$. In the regular feature model, this means both distributions put positive local weight on the same independent feature residuals, so the rank of the stacked Jacobian is unchanged. If a distribution removes a feature constraint, makes a residual unobservable, or changes the local zero set, the LLC can change.

Relation to the Fisher trace. For per-example loss $\ell(\theta; x)$, the component-restricted empirical Fisher trace is

$$\operatorname{tr} F_c(\theta) = \mathbb{E}_x\big[\|P_c\nabla_\theta\ell(\theta; x)\|_2^2\big],$$

where P_c projects onto the component parameter subspace. It decomposes as

$$\begin{aligned}\operatorname{tr} F_c(\theta) = {} & \|\mathbb{E}_x P_c\nabla_\theta\ell(\theta; x)\|_2^2 \\ & + \operatorname{tr}\operatorname{Var}_x\big(P_c\nabla_\theta\ell(\theta; x)\big).\end{aligned}$$

At a stationary point of the population loss the first term vanishes, but the variance term need not. In an exact deterministic interpolation limit, where every per-example gradient vanishes, the Fisher trace vanishes. In general language-model training, however, the empirical Fisher is a first-order, distribution-dependent activity signal; it is not a codimension or RLCT estimator.

In the regular squared-residual model $K_c(u) = \|G_c(u)\|_2^2$, the LLC depends on $\operatorname{rank} DG_c(0)$, but the gradient energy depends on the current residual: $\|\nabla_u K_c(u)\|_2^2 = 4\|DG_c(u)^\top G_c(u)\|_2^2$. So Fisher-like quantities can be large while a feature is being fit and small once the residual is reduced, even when the local rank of the learned constraint is unchanged. The LLC and the Fisher trace therefore measure different things on the regular local model: rank/codimension on the LLC side, current gradient activity on the Fisher side.

Empirical transversality diagnostic. Proposition 1 requires a rank condition on the stacked Jacobian of the feature-residual maps. We do not observe the full residual Jacobian of the actual language-model loss. Instead, we report a weaker diagnostic: the near-collinearity of three scalar probe-loss gradients for the features most relevant to the Dyck circuit. These gradients should be read as a first-order proxy for whether the selected feature penalties point in distinct

parameter-space directions, not as a proof that the full loss-zero varieties are transverse.

At each checkpoint we fit the distance, depth, and type probes locally, freeze the probes, and compute the parameter-space gradients of the corresponding probe losses. Let M be the matrix whose rows are the unit-normalized gradients. Low pairwise cosine and large $\sigma_{\min}(M)$ indicate that these three scalar penalties are not nearly collinear.

step	R^2_{dist}	R^2_{depth}	max cos	$\sigma_{\min}$	cond
256	0.56	0.88	0.034	0.985	1.04
1,000	0.75	0.92	0.047	0.985	1.06
2,000	0.84	0.95	0.037	0.986	1.05
8,000	0.92	0.98	0.038	0.985	1.05
32,000	0.95	0.99	0.088	0.954	1.10
143,000	0.93	0.99	0.197	0.905	1.27

Table 1: Probe-gradient independence diagnostic across training. The type probe accuracy is 1.00 at every checkpoint. The smallest singular value stays above 0.90, so these three scalar probe penalties are far from collinear. This supports the use of the transverse regular model as a local approximation, but it does not establish global transversality of the full language-model loss.

Implications and limits. Three qualitative predictions follow from the regular local calculation. (a) Component-restricted refined LLC should be larger for components whose parameters expose more independent local feature constraints. (b) LLC can stay high after Fisher-like gradient activity has decayed, because rank and current gradient magnitude are different objects. (c) LLC rankings should be stable under distribution changes that preserve the same local feature constraints, and only under such changes.

The calculation says nothing about temporal prediction of future circuit events, and it does not say refined LLC replaces ablation. Ablation measures the mediated effect of removing a component from a functioning network. The refined LLC measures local loss geometry around the trained weights under a chosen loss and localization. The two agree on the regular local model; they diverge as soon as features are redundant, overlap in parameter space, or sit at non-regular singular points.