# COMPUTATION PERSISTS, ROUTING DEGRADES: A MECHANISTIC ANALYSIS OF TRANSFORMER LENGTH EXTRAPOLATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Transformer models trained on algorithmic tasks often fail to extrapolate to sequence lengths beyond those observed during training. This behavioral degradation is frequently interpreted as a fundamental collapse in reasoning capacity, but we investigate whether this failure is mechanistically decomposable. Using dynamic programming (DP) tasks as case studies, we analyze the internal execution of RoPE-based Transformer model organisms. We find that length extrapolation failure is heterogeneous across learned mechanisms. Specifically, in the studied Longest Increasing Subsequence (LIS) models, internal representations associated with DP state construction remain substantially decodable beyond the training horizon, while downstream positional routing mechanisms degrade sharply. Crucially, through out-of-distribution (OOD) activation patching, we demonstrate that this unmodified OOD state representation remains functionally usable: correcting downstream routing alone recovers substantial behavior relative to the OOD baseline, and upstream restoration provides no additional benefit under this intervention. Furthermore, evaluating sequences at a fixed length reveals that dependency distance is a key correlate of representational degradation. In contrast, on the Longest Common Subsequence (LCS) task, where state construction itself requires global cross-sequence interactions, both intermediate representations degrade alongside behavioral performance under extrapolation. Layer-specific positional interventions and Randomized Positional Training further implicate positional geometry as a major constraint. Together, these results suggest that algorithmic extrapolation failure depends on the positional interaction requirements of individual learned mechanisms, rather than reflecting a uniform loss of algorithmic information.

## 1 INTRODUCTION

The capacity of Transformer models (Vaswani et al., 2017) to perform multi-step algorithmic reasoning remains a subject of ongoing investigation. Empirical studies consistently observe that while models can reliably internalize algorithmic rules on in-distribution data, their performance predictably degrades when tasked with out-of-distribution (OOD) length extrapolation (Anil et al., 2022; Dziri et al., 2023; Saparov & He, 2022).

Existing work predominantly measures *whether* algorithmic generalization succeeds, frequently treating OOD behavioral collapse as evidence that models rely on shallow, length-bounded heuristics rather than robust recursive logic (Delétang et al., 2022; Zhou et al., 2022). In this work, we instead ask: **Where does algorithmic execution break inside the network?**

Rather than treating behavioral failure as definitive evidence that internal reasoning has collapsed entirely, we draw on recent advances in mechanistic interpretability (Elhage et al., 2021; Wang et al., 2022) to investigate whether algorithmic execution decomposes into interacting components with heterogeneous failure modes. Specifically, we hypothesize that dynamic programming (DP) tasks partition into: a *DP-state representation* associated with constructing the hidden mathematical state, and a *downstream routing mechanism* associated with decoding that state via long-range attention pointers.

Our central finding is that length extrapolation failure is mechanism-dependent: in the studied Transformer model organisms, degradation increases when learned computation or routing requires positional dependencies beyond the training regime. We structure our findings around three pillars of mechanistic evidence:

1. **Representational Persistence:** In 1D DP tasks, intermediate representations associated with state construction remain substantially decodable beyond the training length, while downstream long-range routing diffuses catastrophically.



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

2. **Functional Usability:** Through OOD activation patching, we demonstrate that this persisting unmodified representation is causally functional. Correcting the downstream routing mechanism is sufficient to recover substantial behavior.

3. **Distance-Dependent Degradation:** By holding sequence length strictly constant, we reveal that dependency distance is a key correlate of representational degradation, explaining why circuits requiring unobserved long-range pointers fail while those relying on short-range interactions survive.

To evaluate this decomposition, we train autoregressive Transformer model organisms utilizing Rotary Positional Embeddings (RoPE) (Su et al., 2024) on the Longest Increasing Subsequence (LIS) and Longest Common Subsequence (LCS) tasks. While canonical LIS allows for global dependencies, the learned LIS state representation in our models is empirically associated with predominantly short-range interactions. Conversely, LCS state construction inherently necessitates long-range cross-attention, causing it to degrade symmetrically with routing.

Through a progression of behavioral scaling, linear probing, distance-controlled evaluations, out-of-distribution activation patching, and positional ablations, we systematically characterize the failure modes of algorithmic extrapolation. To strictly bound our scientific claims, Appendix A summarizes our specific methodologies and their respective evidentiary strength (e.g., distinguishing correlational probing from causal activation patching).

While we validate representational persistence in deeper LLaMA-style networks, we explicitly bound our claims to the studied model organisms and synthetic tasks. Extrapolating these specific mechanistic dynamics to billion-parameter production systems remains an area for future empirical investigation.

## 2 Related Work

**Limits of Algorithmic Length Generalization** Neural networks often struggle to generalize algorithmic behavior beyond the sequence lengths observed during training. Synthetic reasoning benchmarks such as SCAN (Lake & Baroni, 2018) have highlighted broader limitations in systematic and compositional generalization (Keysers et al., 2019), while standard Transformers have been shown to exhibit substantial length-extrapolation failures even on simple arithmetic tasks (Anil et al., 2022). These failures have motivated explanations based on pattern matching and shortcut learning (Dziri et al., 2023; Saparov & He, 2022), as well as methods that introduce randomized training distributions, explicit algorithmic supervision, or recursive latent-state computation (Ruoss et al., 2023; Altabaa et al., 2025). Rather than proposing another behavioral remedy, we investigate the internal mechanisms underlying length-extrapolation failure in standard Transformers, distinguishing degradation of intermediate algorithmic state from degradation of downstream routing.

**Mechanistic Interpretability and Causal Interventions** Mechanistic interpretability aims to identify the computational structures implemented by neural networks (Elhage et al., 2021; Conmy et al., 2023). Prior studies have recovered task-relevant circuits for phenomena including indirect-object identification and modular arithmetic (Wang et al., 2022; Nanda et al., 2023). Because correlational analyses alone cannot establish whether a representation is functionally required, recent work has increasingly emphasized causal interventions (Geiger et al., 2025). Activation patching tests causal dependence by replacing selected internal activations and measuring the resulting behavioral change (Meng et al., 2022; Heimersheim & Nanda, 2024), while causal scrubbing evaluates mechanistic hypotheses through targeted activation transformations (Chan et al., 2022). Our analysis combines these causal tools with representational measurements, including CKA (Kornblith et al., 2019) and RSA (Kriegeskorte et al., 2008), to distinguish persistence of internal state from functional usability under distribution shift.

**Positional Encodings and Length Extrapolation** Positional encodings provide Transformers with information about token order (Vaswani et al., 2017), and their behavior outside the training range can strongly affect length generalization. Absolute positional embeddings are tied to the positions encountered during training, while relative schemes such as ALiBi (Press et al., 2021) and Rotary Positional Embeddings (RoPE) (Su et al., 2024) modify how positional relationships enter the attention computation. Empirical studies nevertheless show that positional mechanisms can exhibit substantial extrapolation limits (Kazemnejad et al., 2023; Sun et al., 2023; Dziri et al., 2023). Several approaches, including YaRN (Peng et al., 2024), NTK-aware scaling (Liu et al., 2024), and Linear Position Interpolation (PI) (Chen et al., 2023), modify the positional geometry to improve long-context behavior. These methods provide useful interventions for testing whether OOD degradation is tied to positional compatibility rather than to a complete loss of internal algorithmic state. In this work, we use layer-specific positional interventions to examine whether restoring positional compatibility preferentially recovers the routing mechanism responsible for exact pointer retrieval.

# 3 EXPERIMENTAL METHODOLOGY AND SETUP

To isolate algorithmic execution from the confounding variables of natural language pre-training, we construct a controlled experimental framework utilizing canonical dynamic programming tasks.

## 3.1 TASK FORMULATION AND MODEL ARCHITECTURES

We primarily evaluate models on the Longest Increasing Subsequence (LIS) problem. Given an input sequence $S = [s_1, \ldots, s_N]$, the objective is to autoregressively output the longest strictly increasing subsequence. The classic DP solution constructs a memoization state $L[i]$ via a backward-looking recurrence relation: $L[i] = 1 + \max\{L[j] \mid j < i \text{ and } s_j < s_i\}$. To prevent the model from learning ambiguous tie-breaking heuristics, our dataset generator synthesizes sequences possessing exactly one mathematically valid LIS. We explicitly note that this unique-solution constraint is employed strictly to eliminate output generation ambiguity for clean mechanistic evaluation, and does not necessarily imply that the resulting sequence distribution represents naturally occurring algorithmic inputs. To validate our findings on 2D architectures, we extend our evaluation to the Longest Common Subsequence (LCS) task.

Our primary experiments employ a tractable 2-layer, 4-head decoder-only Transformer utilizing RoPE. To verify that our observations are not shallow artifacts, we explicitly validate our findings on deeper networks, including standard 8-layer and 12-layer GPT-2 style networks, and a 6-layer medium-scale model utilizing modern LLaMA-style primitives (RMSNorm, SwiGLU, and bias-free projections). We operationally define the *DP-state representation* as intermediate residual stream features that encode the algorithmic state, and the *downstream routing mechanism* as the final-layer attention heads that read this state to predict the target output. Full architectural dimensions and hyperparameters are detailed in Appendix A.

## 3.2 TRAINING, PROBING, AND EVALUATION PROTOCOLS

Models were trained on in-distribution sequences of length $N = 8$ and evaluated at OOD lengths $N \in \{12, 16, 24, 32\}$. For the primary evaluations, behavioral performance is reported as Sequence-Level Exact Match accuracy averaged across five distinct random seeds. Crucially, to ensure statistical rigor, all confidence intervals and $p$-values are calculated via bootstrapping at the *sequence or seed level*, preventing artificial inflation from correlated intra-sequence tokens.

Because our evaluations span global scaling and targeted single-checkpoint interventions, exact local OOD baselines vary slightly (e.g., $2.08\%$ vs. $1.18\%$ vs. $0.60\%$). A Master Experiment Table clarifying all baseline metrics, seeds, and intervention conditions is provided in Appendix A.

Linear probes were trained strictly on frozen residual stream activations from the $N = 8$ distribution. To ensure the network is actively computing state rather than exploiting spurious artifacts, we benchmark probe extraction against three stringent controls: a *Naive Positional Baseline*, a *Local Heuristic Baseline*, and a *Token-Shuffled Baseline* (which permutes the input to destroy algorithmic ordering). Optimization details and leakage controls for the linear probes are provided in Appendix D.

## 3.3 TARGETED INTERVENTION SETUPS

**Distance-Controlled Probing:** To evaluate whether degradation is driven by physical sequence length or internal geometric requirements, physical sequence length is held strictly constant at $N = 16$. Test tokens are bucketed by mathematical dependency distance (the required look-back jump, $d$). Balanced sampling reduces distributional differences across buckets, isolating dependency distance as the primary controlled factor.

**OOD Activation Patching:** We define a standard Target model (suffering OOD collapse at $N = 16$) and a Source model (supplied with corrected positional geometry via Linear Position Interpolation). During autoregressive generation, we systematically overwrite the entire Target residual stream activation vector with the Source vector at specific layers. Source and Target models receive identical ground-truth prefixes at each autoregressive step, ensuring patched activations correspond to the exact same input context (Appendix B).

# 4 WHERE DOES OOD FAILURE OCCUR?

A prevailing assumption in the literature is that when a Transformer fails at out-of-distribution (OOD) length extrapolation, its underlying capacity to maintain algorithmic state has degraded. We investigate whether algorithmic execution is instead mechanistically decomposable into interacting computational subsystems—specifically, DP-state

representation (e.g., tracking the Bellman state) and routing-associated representations (e.g., attending to and decoding target tokens)—and whether these subsystems exhibit heterogeneous degradation profiles under OOD conditions.

### 4.1 Behavioral Degradation vs. Representational Persistence

We first establish the baseline behavioral degradation of the network on the LIS task. Evaluated in-distribution ($N = 8$), the 2-layer models achieve a sequence-level exact match accuracy of $92.42\% \pm 0.82\%$ across 5 seeds. However, when evaluated on OOD sequences of length $N = 16$, behavioral performance declines to $2.08\% \pm 1.49\%$. To determine whether this behavioral decline corresponds to a loss of the underlying algorithmic information, we train linear probes to decode the ground-truth dynamic programming state (the $L[i]$ array) directly from the residual streams of Layer 0 and Layer 1. We evaluate the probes across a continuous scaling of sequence lengths ($N \in \{8, 12, 16, 24, 32\}$) to capture the derivative of failure. As illustrated in Figure 1, the analysis reveals a clear functional divergence:

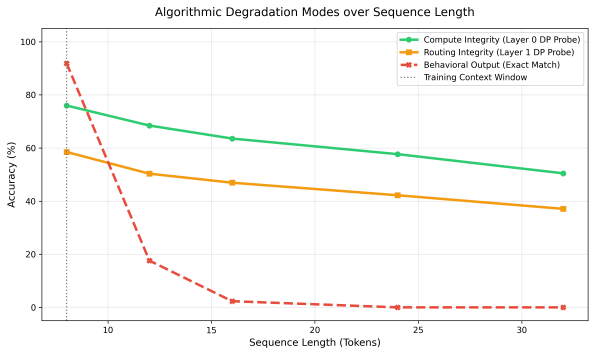


Figure 1: Degradation scaling curves illustrating divergence between behavioral exact match and representational decodability.

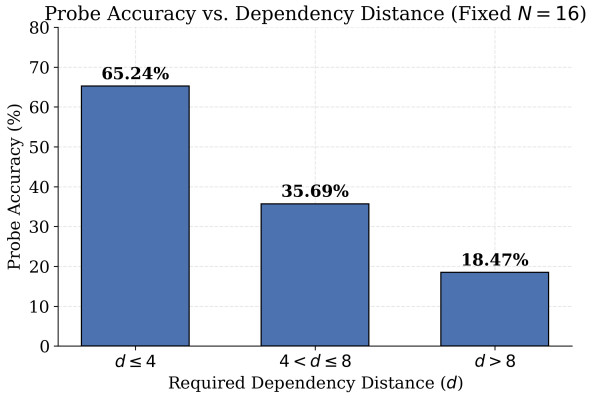


Figure 2: Probe Accuracy vs. Dependency Distance (fixed $N = 16$). Accuracy decreases monotonically as required distance exceeds the training horizon.

- **Behavioral Output** declines steeply, dropping from $> 90\%$ at $N = 8$, to $19.14\%$ at $N = 12$, and effectively reaching $0.00\%$ by $N = 24$.
- **Layer 1 Representational Decodability** degrades rapidly in tandem with behavioral output, dropping from $57.00\%$ at $N = 8$ to $36.22\%$ at $N = 32$.
- **Layer 0 Representational Decodability** remains substantially more robust. At $N = 16$, extraction accuracy is $62.06\%$. Even at $N = 32$—four times the training context window—the Layer 0 probe retains $50.69\%$ accuracy.

To rigorously contextualize the $61.58\%$ token-level OOD extraction accuracy and rule out spurious statistical artifacts (such as length-based or bag-of-words memorization), we benchmark the Layer 0 probe against three stringent controls on the OOD $N = 16$ dataset. First, a *Naive Positional Baseline* (predicting the most frequent DP state per index) achieves only $33.37\%$ token-level accuracy. Second, a *Local Heuristic Baseline* (greedily incrementing the state if the current token exceeds the previous) achieves $36.30\%$. Finally, under a *Token-Shuffled Baseline*—where the input sequence is randomly permuted prior to the forward pass to destroy positional order—probe accuracy collapses to $28.80\%$. The severe degradation under input shuffling suggests that probe decodability relies strictly on sequence-order-dependent representation. The significant delta between these heuristics ($\leq 36.30\%$) and our Layer 0 probe ($61.58\%$) provides evidence that the residual stream contains a representation that linearly encodes substantial information about the DP state, rather than defaulting to positional heuristics.

Bootstrap confidence intervals ($k = 10,000$ resamples) were computed at the sequence level. A paired $t$-test across matched test sequences showed that Layer 0 ($61.58\% \pm 0.53\%$, 95% CI: $[60.58, 62.62]$) significantly outperformed Layer 1 ($48.71\% \pm 0.56\%$, 95% CI: $[47.64, 49.77]$) under $2\times$ extrapolation ($p < 5.68 \times 10^{-124}$). These metrics suggest that substantial algorithmic-state information persists in the intermediate representations, degrading more slowly than behavioral performance.

### 4.2 Distance-Controlled Degradation

While the previous analysis evaluates sequence-level extrapolation, we hypothesize that the underlying representational degradation is governed by the specific geometric requirements of the learned algorithmic circuit. To test this, we evaluate probe accuracy under our strictly controlled $N = 16$ distance-bucketed setup.

When evaluated at this fixed sequence length, probe decodability is highly sensitive to the required dependency distance ($d$), as visualized in Figure 2:

- When the required look-back jump is well within the training horizon ($d \leq 4$), Layer 0 probe accuracy remains high at **65.24%**.
- When the distance approaches the edge of the training horizon ($4 < d \leq 8$), accuracy declines to **35.69%**.
- When the dynamic programming state inherently requires evaluating positional geometries outside the training horizon ($d > 8$), accuracy collapses to **18.47%**.

Because sequence length is held fixed at $N = 16$, these results isolate dependency distance as a key correlate of representational degradation.

### 4.3 Functional Bifurcation: State Decodability vs. Attention Mass

To mechanistically contextualize how routing and computation interact across network depth, we trained an expanded 4-layer LIS model. For each layer, we concurrently extracted the linear decodability of the DP state and the maximum attention probability mass placed on the mathematically correct routing pointer under OOD conditions ($N = 16$).

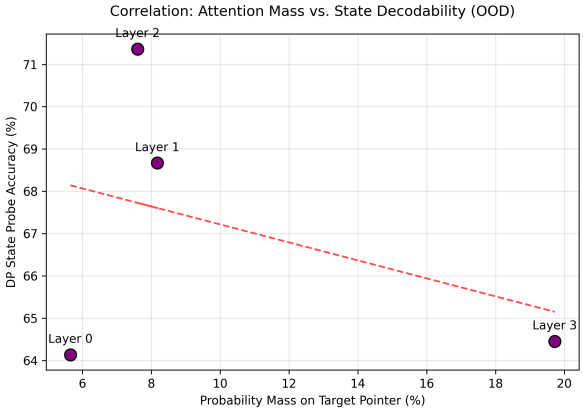


Figure 3: Correlation between DP State Decodability and Target Attention Mass (OOD $N = 16$). Intermediate layers compute without routing; the final layer routes but degrades.

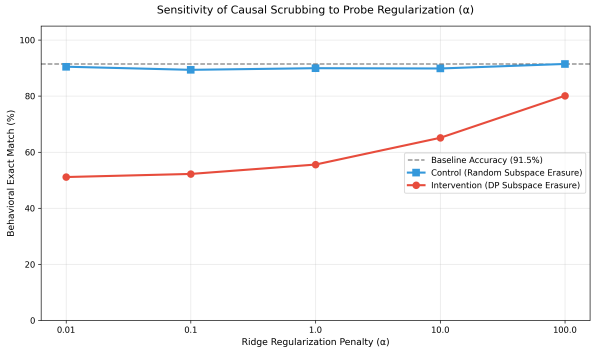


Figure 4: Sensitivity of Causal Scrubbing to Probe Regularization ($\alpha$). Erasing the specific DP Subspace substantially degrades exact match accuracy.

As illustrated in Figure 3, mapping these metrics reveals a clear separation between state decodability and routing-related attention across depth. State decodability peaks in intermediate layers (e.g., 71.36% in Layer 2), whereas target-pointer attention is concentrated more strongly in the final layer (19.72%). This pattern is consistent with a division in which intermediate layers contribute to state construction while the final layer contributes more strongly to pointer extraction, rendering its behavioral expression disproportionately vulnerable to out-of-distribution spatial geometries.

### 4.4 Scaling Validation of Representational Persistence

A critical limitation of evaluating miniature architectures is the risk that the observed decomposition might be an artifact of shallow optimization or legacy components. As model scale increases, representations often become increasingly distributed and attention heads highly polysemantic. To investigate whether this representational persistence scales as a stable architectural property, we extend our analysis to three deeper networks trained under identical conditions: an 8-layer GPT-2 style model ($d_{\text{model}} = 256$), a 12-layer GPT-2 style model ($d_{\text{model}} = 768$), and a 6-layer model configured with modern LLaMA-style primitives ($d_{\text{model}} = 512$, incorporating RMSNorm, SwiGLU activations, and bias-free projections).

Table 1: Representational Persistence across Scaled Architectures under $2\times$ Extrapolation ($N = 16$).

| Architecture | Peak Middle-Layer Probe | Final-Layer Probe | Behavioral Match |
|---|---|---|---|
| 8-Layer GPT-2 ($d = 256$) | $71.44\%$ (Layer 4) | $60.62\%$ (Layer 7) | $1.80\%$ |
| 12-Layer GPT-2 ($d = 768$) | $79.47\%$ (Layer 8) | $76.97\%$ (Layer 11) | $4.05\%$ |
| 6-Layer LLaMA ($d = 512$) | $77.53\%$ (Layer 4) | $77.00\%$ (Layer 5) | $3.09\%$ |

As detailed in Table 1, behavioral exact match severely declines across all deeper architectures at $2\times$ extrapolation. However, representational decodability is robustly maintained deep into the networks. Notably, in the LLaMA-style architecture, probe accuracy climbs steadily to peak at $77.53\%$ in Layer 4 before behavioral collapse occurs at the output ($3.09\%$). This structural discrepancy between internal representational preservation ($\sim 77\%$) and behavioral output ($\sim 3\%$) suggests that modern architectural primitives can retain substantial algorithmic information within the residual stream, while downstream attention mechanisms struggle to effectively route this state to the correct vocabulary predictions under OOD positional shifts.

Furthermore, unlike our 2-layer architecture which localized routing to a single dominant head (recovering $81\%$ via oracle intervention), applying targeted oracle interventions to individual heads in the final layer of these scaled models yielded maximum behavioral recoveries of only $1.90\%$ (8-layer), $7.50\%$ (12-layer), and $4.60\%$ (6-layer LLaMA). This indicates that as architectural capacity increases, the routing mechanism becomes highly distributed across multiple interacting heads, rendering single-head interventions insufficient. Nonetheless, the core phenomenon of upstream representational persistence decoupling from downstream behavioral collapse remains clearly observable across the studied model scales and primitive types.

## 5 Is the Surviving Representation Functionally Usable?

While the previous section demonstrates that substantial algorithmic-state information persists in intermediate layers out-of-distribution, linear decodability does not inherently guarantee functional execution. A residual stream might contain decodable information due to input statistics or positional correlations without the downstream network actually utilizing that specific representation. To rigorously answer whether this surviving representation remains functionally usable under length extrapolation, we perform targeted causal interventions.

### 5.1 Causal Subspace Erasure

First, we must establish that the downstream network actively and causally relies on the specific DP-state subspace identified by our Layer 0 probe. We achieve this via Regularized DP-Subspace Erasure (Causal Scrubbing) on in-distribution sequences.

We estimate the geometric subspace that Layer 0 utilizes to encode the DP state by extracting the weight matrix $W$ from our trained linear probe. To rigorously test erasure without introducing artifacts from pseudo-inverse hyperparameters, we implement Ridge-regularized projection matrices into the null space: $P_{\text{null}} = I - W^T(WW^T + \alpha I)^{-1}W$. During the forward pass of an in-distribution sequence, we project the Layer 0 activations through $P_{\text{null}}$, effectively erasing the specific DP representation. We sweep the regularization penalty $\alpha$ across five orders of magnitude (Figure 4).

As demonstrated in Figure 4, accurately erasing the specific DP subspace ($\alpha \leq 0.1$) degrades behavioral exact match from a baseline of $91.50\%$ down to $51.12\%$. Crucially, projecting out a Random Control subspace (an orthogonalized random Gaussian projection of identical dimensionality, as defined in Appendix D) preserves $> 89\%$ accuracy, irrespective of $\alpha$. The divergence between the intervention and control strictly isolates the DP subspace, providing evidence that downstream prediction relies causally on this specific computational representation and that this dependency is robust to inversion hyperparameters.

### 5.2 OOD Activation Patching (Causal Sufficiency)

Having established that the model causally relies on the Layer 0 representation in-distribution, we directly test whether the unmodified, degraded OOD Layer 0 representation is functionally usable. We employ an OOD Activation Patching protocol on $N = 16$ sequences.

We utilize a Target model (the standard, unmodified network suffering from OOD collapse) and a Source model (the network operating under Linear PI, which possesses partially recovered routing capabilities). During the autoregressive

generation of the standard Target model, we systematically patch its residual stream activations with the corresponding activations from the Source model. The results are summarized in Table 2.

Table 2: OOD Activation Patching Matrix ($N = 16$)

| **Intervention (Source → Target)** | **Behavioral Exact Match** |
|---|---|
| Standard Baseline (Broken) | $1.18\%$ |
| Patch Layer 0 Only | $1.12\%$ |
| Patch Layer 1 Only | $\mathbf{14.50\%}$ |
| Patch Both (Layer 0 + Layer 1) | $\mathbf{14.50\%}$ |

This intervention yields a critical insight: patching the PI-corrected Layer 1 onto the unmodified OOD Layer 0 recovers $14.50\%$ exact-match accuracy, which is identical to the ceiling achieved when patching both layers. Thus, within the tested intervention, Layer-1 restoration is sufficient for the recovered behavior, and Layer-0 restoration provides no additional benefit.

We note that the patched Layer 1 activations from the PI-rescued model contain all downstream processing changes induced by the altered positional geometry, not solely corrected attention probabilities. Therefore, while this does not strictly prove Layer 1 acts *exclusively* as a routing circuit, it provides powerful causal evidence that downstream processing at Layer 1 constitutes a dominant causal bottleneck. The unmodified OOD Layer 0 representation remains functionally usable, already containing sufficient information to support the recovered downstream behavior.

### 5.3 The Routing Bottleneck and Oracle Interventions

If correcting the downstream positional routing recovers a fraction of the performance ($14.50\%$), what is the theoretical maximum recoverability of the downstream module? To test this, we execute an inference-time Oracle routing intervention on the standard OOD dataset ($N = 16$, local baseline $0.60\%$).

Instead of patching from a rescued model, we freeze the network and overwrite the pre-softmax attention matrix of the primary Layer 1 routing head (identified strictly via maximum target attention mass on the in-distribution $N = 8$ validation set to prevent selection bias) with a causally masked, one-hot distribution pointing to the mathematically correct target tokens.

This Oracle intervention yields a massive behavioral recovery, elevating exact match accuracy from $0.60\%$ to $81.00\%$. To rigorously verify that this recovery is task-specific, we introduce a *Random Oracle* baseline, which forces the attention head to point to a randomly selected input token. Evaluated under identical conditions, the Random Oracle yields $0.00\%$ accuracy, supporting that the observed behavioral recovery depends on mathematically correct routing pointers rather than attention sharpening alone.

The divergence between the learned routing rescue via activation patching ($14.50\%$) and the ground-truth Oracle intervention ($81.00\%$) indicates that while restoring positional compatibility (via PI) rescues downstream processing enough to read the surviving state representation, it does not fully reconstruct the ideal routing computation. Furthermore, the massive $81.00\%$ recovery demonstrates that the downstream network retains the *capacity* to exploit the correct pointer even when native OOD routing fails. However, the inability to achieve full $100\%$ recovery indicates that routing is not the singular cause of OOD failure, and additional sources of OOD degradation remain beyond the tested routing intervention.

## 6 Why Does Routing Fail?

Having established that the surviving upstream representation is functionally usable and that downstream routing constitutes a major causal bottleneck, we now investigate the geometric mechanisms driving this routing failure.

### 6.1 Attention-Distance Analysis: Probability Diffusion

A natural hypothesis is that routing fails because the attention probability distribution physically diffuses (softmax entropy increases) at longer sequence lengths due to unobserved positional geometries. RoPE embeds position via rotational angles calibrated during training. In 1D tasks like LIS, our empirical models learn an implementation that relies predominantly on short-range comparisons ($d \leq 8$) to compute the state, safely falling within the training horizon. Conversely, routing the final answer frequently requires pointers from the end of the sequence to the beginning ($d > 8$).

To mechanistically quantify this, we extract the post-softmax probability mass from the primary LIS routing head on $N = 16$ sequences as a function of relative pointer distance (Figure 5).

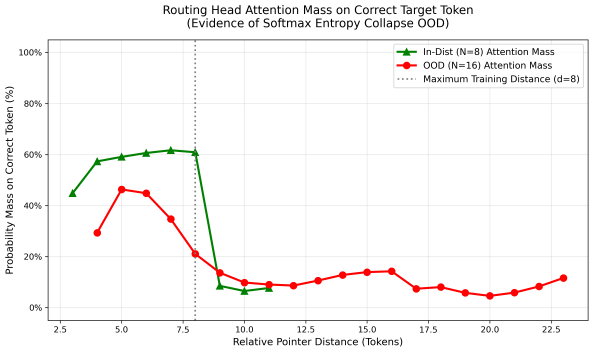


Figure 5: Routing Head Attention Mass on Correct Target Token (OOD $N = 16$). Probability mass diffuses catastrophically beyond $d > 8$.

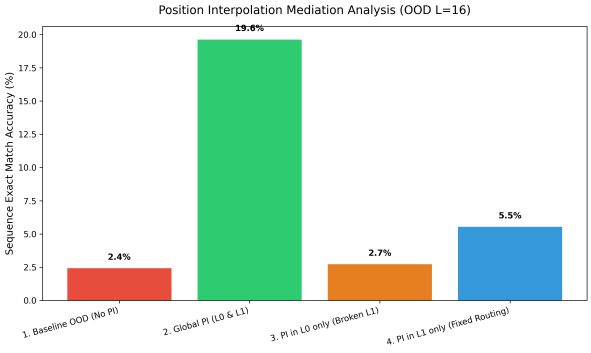


Figure 6: Layer-Specific PI Mediation. Applying PI solely to L1 yields partial recovery, while global PI yields the strongest positional recovery (19.00%)

Within the training horizon ($d \leq 8$), the LIS routing head places an average of $57.41\%$ probability mass on the correct token. Crossing the maximum training distance ($d > 8$) collapses this mass to $9.64\%$. This provides empirical evidence that positional extrapolation causes the attention distribution to diffusively collapse, hindering routing while local compute persists.

### 6.2 Positional Interventions Localize the Routing Bottleneck

To directly test whether this routing diffusion is intrinsically driven by out-of-distribution positional geometries, we evaluate layer-specific inference-time positional interventions using Linear Position Interpolation (PI) (Chen et al., 2023). By decoupling the positional embeddings across layers, we can pass interpolated rotational frequencies strictly to Layer 0 (Compute) or Layer 1 (Routing) during inference on OOD lengths ($N = 16$).

As illustrated in Figure 6, for the frozen checkpoint used in the PI intervention (local OOD baseline $1.30\%$), applying PI strictly to the compute layer (L0) yields effectively zero substantive recovery ($2.70\%$). In contrast, applying PI *solely* to the routing layer (L1) yields a statistically significant behavioral recovery ($5.50\%$, 95% CI: $[4.10, 6.90]$). Global PI applied to both layers yields the maximum positional recovery ($19.00\%$).

Coupled with our OOD Activation Patching results (Section 5.2), these results support a causal chain in which OOD evaluation introduces previously unseen relative positional distances, positional geometry degrades the routing mechanism's attention distribution, and restoring the positional geometry produces downstream behavioral recovery. Additional positional baselines (ALiBi, NTK, random jitter) and representational-similarity analyses (CKA/RSA) provide complementary evidence indicating the localized positional nature of this failure; full results are reported in Appendices E.1 and E.2.

## 7 Beyond the Core LIS Circuit

### 7.1 Cross-Task Validation: The 2D LCS Symmetry

A common limitation of mechanistic interpretability studies is the risk of overfitting explanations to a single algorithmic structure. To test whether the observed degradation profile is uniquely tied to the routing module or instead follows the positional interaction requirements of the task, we replicate our pipeline on the Longest Common Subsequence (LCS) task.

Unlike the 1D LIS task, the canonical LCS recurrence permits dependencies spanning the two sequences, and the learned implementation studied here exhibits substantial cross-sequence positional requirements merely to construct the DP Bellman state matrix. Because computing this state relies on long-range pointers, we hypothesize that the DP-state representation will degrade in tandem with routing. Models were trained strictly on input sequences of length $N = 6$ and evaluated up to $4\times$ the training length ($N = 24$).

Table 3: OOD Extrapolation Results for the 2D LCS Task.

| Sequence Length | Behavioral Exact Match | Layer 0 State Probe | Layer 1 State Probe |
|---|---|---|---|
| Length 6 ($1\times$) In-Dist | $79.09\%$ | $68.08\%$ | $72.17\%$ |
| Length 12 ($2\times$) OOD | $4.23\%$ | $48.88\%$ | $47.29\%$ |
| Length 24 ($4\times$) OOD | $0.00\%$ | $30.90\%$ | $31.10\%$ |

As shown in Table 3, at severe extrapolation lengths ($N = 24$), behavioral execution drops to $0.00\%$. Crucially, unlike the 1D LIS task, the internal representational decodability for both Layer 0 and Layer 1 in the LCS task degrades symmetrically to $\sim 31\%$. This provides complementary cross-task evidence: positional generalization limits act as a major constraint for *any* mechanism requiring exact long-range attention, whether utilized for downstream routing (as in LIS) or for global state computation (as in LCS).

### 7.2 Algorithmic Preservation via Randomized Positional Training

Finally, to further isolate the causal role of rigid training geometries, we tested whether removing strict positional horizons during training mitigates OOD collapse. We trained an identical 2-layer model where RoPE frequencies were randomly scaled by a uniformly sampled factor $s \in [1.0, 2.0]$ during the forward pass of each training batch.

Evaluated natively (without inference-time intervention) across 5 distinct random seeds, the standard model achieves $91.75\% \pm 0.82\%$ exact match in-distribution ($N = 8$) but collapses to $2.20\% \pm 1.45\%$ OOD ($N = 16$). In contrast, the models trained with Randomized Positional Training achieve $28.06\% \pm 1.12\%$ OOD accuracy, trading off a modest $7.5\%$ in in-distribution precision ($84.23\% \pm 0.95\%$).

This $> 12\times$ relative improvement in out-of-distribution generalization is highly informative. RPT exposes the model to a broader distribution of positional geometries and substantially improves OOD extrapolation. This result is consistent with the hypothesis that positional diversity during training reduces the network's sensitivity to unseen positional regimes. It provides empirical evidence suggesting that substantial algorithmic capacity remains available, while behavioral failure is predominantly a structural artifact of out-of-distribution positional embeddings hindering the learned mechanisms.

## 8 Discussion and Conclusion

Our findings demonstrate that OOD length extrapolation failure is not necessarily a uniform collapse of algorithmic capability. Instead, it is highly mechanism-dependent: in the studied Transformer model organisms, degradation is governed by the positional dependency distance required by the specific computational circuits the network has learned.

Crucially, we do not claim that computation is perfectly preserved out-of-distribution. Rather, our experiments demonstrate that substantial DP-state information persists and remains functionally usable. The OOD activation patching matrix isolates the routing mechanism as a major causal bottleneck, demonstrating that Layer-1 restoration is sufficient for the recovered behavior, and Layer-0 restoration provides no additional benefit under this intervention. However, the performance gap between this partial rescue ($14.50\%$) and the ground-truth Oracle intervention ($81.00\%$) indicates that restoring positional compatibility alone does not perfectly reconstruct the ideal routing computation. This suggests a nuanced reality: routing limits are a primary and isolable cause of OOD breakdown, but minor upstream computational degradation also contributes to the remaining failure.

Furthermore, our distance-controlled evaluations confirm that dependency distance is a key correlate of representational degradation. Mechanisms that rely on short-range interactions (like the learned empirical implementation of 1D LIS state construction) survive out-of-distribution, while mechanisms that inherently require unobserved, long-range relative distances (like LIS final routing or 2D LCS global state construction) suffer from catastrophic attention diffusion and positional geometric breakdown.

In summary, this work provides a rigorous mechanistic framework for understanding length extrapolation failure. Our results rest on three pillars: (1) **Representational Persistence** of local state construction, (2) **Functional Usability** validated via causal patching, and (3) **Distance-Dependent Degradation**. Because this persistence scales to deeper LLaMA-style architectures, and behavioral recovery can be achieved via positional interventions (PI) or training (RPT), we identify positional geometry as a major constraint on length extrapolation. We explicitly bound our claims to the studied model organisms and synthetic tasks. Determining how these computational topologies generalize to massive language models on natural language reasoning tasks remains a critical area for future research.

## REFERENCES


Awni Altabaa, Siyu Chen, John Lafferty, and Zhuoran Yang. Unlocking out-of-distribution generalization in transformers via recursive latent space reasoning. *arXiv preprint arXiv:2510.14095*, 2025.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. volume 35, pp. 38546–38556, 2022.

Lawrence Chan, Adria Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing: A method for rigorously testing interpretability hypotheses. 2:19, 2022.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. volume 36, pp. 16318–16352, 2023.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, et al. Neural networks and the chomsky hierarchy. 2022.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in neural information processing systems*, 36:70293–70332, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Nova Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Thomas Icard, and Christopher Potts. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

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

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. volume 36, pp. 24892–24928, 2023.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*, 2019.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pp. 2873–2882. PMLR, 2018.

Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of rope-based extrapolation. In *International Conference on Learning Representations*, volume 2024, pp. 48160–48185, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. volume 35, pp. 17359–17372, 2022.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023. 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. 2024:31932–31951, 2024.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. 2021.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. Randomized positional encodings boost length generalization of transformers. pp. 1889–1903, 2023.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. 2022.

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

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14590–14604, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. 2022.

Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, and Hanie Sedghi. Teaching algorithmic reasoning via in-context learning. *arXiv preprint arXiv:2211.09066*, 2022.

## A Experimental Setup and Hyperparameters

To ensure full reproducibility of all results, we detail the specific architectural configurations and training hyperparameters utilized across our experiments.

**1. Miniature Model Organisms (2-Layer):** The primary model used for LIS and LCS evaluations is a 2-layer, decoder-only causal Transformer.

- **Dimensions:** $d_{\text{model}} = 128$, $n_{\text{heads}} = 4$, head dimension $d_{\text{head}} = 32$, and $d_{\text{mlp}} = 512$.
- **Activation & Normalization:** GELU activations and standard LayerNorm.
- **Positional Encoding:** Rotary Positional Embeddings (RoPE) applied to queries and keys per head.
- **Optimization:** Trained using AdamW ($\beta_1 = 0.9, \beta_2 = 0.98$) with a learning rate of $1 \times 10^{-3}$, weight decay of $\lambda = 1.0$, and a batch size of 256. The high weight decay prevents rote memorization and encourages algorithmic circuit formation.

**2. Deeper GPT-2 Style Architectures:** For the scaling validations, we utilized standard GPT-2 dimensional configurations.

- **8-Layer Model:** $d_{\text{model}} = 256$, $n_{\text{heads}} = 4$, $d_{\text{mlp}} = 1024$.
- **12-Layer Model:** $d_{\text{model}} = 768$, $n_{\text{heads}} = 12$, $d_{\text{mlp}} = 3072$.
- **Optimization:** Both models were trained using AdamW with a learning rate of $5 \times 10^{-4}$, weight decay of $\lambda = 1.0$, and a batch size of 128 to accommodate memory constraints.

**3. Modern LLaMA-Style Architecture:** To validate findings against modern structural primitives, we constructed a 6-layer medium-scale model mirroring the LLaMA architectural family.

- **Dimensions:** $d_{\text{model}} = 512$, $n_{\text{heads}} = 8$.
- **Primitives:** We replaced LayerNorm with RMSNorm ($\epsilon = 1 \times 10^{-5}$). We replaced the standard MLP with a SwiGLU gated architecture, setting the intermediate dimension to $d_{\text{mlp}} = 1376$ (approximately $\frac{8}{3} d_{\text{model}}$). Furthermore, all linear projections in the attention and feed-forward blocks were implemented without biases.
- **Optimization:** Trained using AdamW with a learning rate of $5 \times 10^{-4}$ and weight decay of $\lambda = 0.1$.

Table 4: Summary of Mechanistic Claims and Evidentiary Strength

| Claim | Evidence | Strength |
|---|---|---|
| DP information remains decodable OOD | Linear probes + controls | Correlational |
| OOD L0 representation is functionally usable | OOD activation patching | Causal |
| L1 routing is sufficient for substantial recovery | Oracle intervention | Causal sufficiency |
| Positional geometry limits routing capacity | PI / ALiBi / RPT | Intervention |
| Dependency distance predicts degradation | Fixed-$N$ distance analysis | Controlled association |

## B OOD Activation Patching Protocol

In Section 5.2, we detailed an out-of-distribution (OOD) activation patching matrix to isolate causal sufficiency. The protocol was implemented as follows:

1. **Source Model (Rescued):** The source representation is generated by passing the $N = 16$ OOD sequence through the standard network while applying Linear Position Interpolation (PI) (Chen et al., 2023) at a scale factor of $s = 2.0$. We use PyTorch forward hooks to cache the full residual stream tensor (shape: $[B, T, d_{\text{model}}]$) at the exact output of Transformer Block 0 (Layer 0) and Block 1 (Layer 1).
2. **Target Model (Broken):** The target computation runs the exact same $N = 16$ sequence through the standard network without any inference-time positional interventions (scale $s = 1.0$).
3. **Patching Operation:** During the target model's forward pass, we apply a forward hook that intercepts the output of the specified layer. The hook explicitly overwrites the target's residual stream with the cached source residual stream. Crucially, the attention probability matrices (`att_weights`) calculated during the target's forward pass are bypassed in the patched layer, forcing the network to accept the corrected output state.
4. **Evaluation:** We measure the Sequence-Level Exact Match accuracy on the final autoregressive outputs of the patched target model.

## C Distance-Controlled Evaluation Setup

To map representational degradation to geometric positional requirements (Section 6.1), we explicitly decoupled sequence length from DP dependency distance. For a given LIS sequence, we extract the ground-truth DP path. For every token at index $i$ belonging to the valid LIS, we calculate its required look-back distance $d = i - j$, where $j$ is the index of the immediately preceding token in the valid subsequence.

We generate a massive set of $N = 16$ sequences and route them through the frozen network to extract Layer 0 activations. Instead of computing probe accuracy over the entire sequence, we evaluate the Logistic Regression probe strictly on tokens belonging to specific distance subsets: $d \leq 4$, $4 < d \leq 8$, and $d > 8$. By holding the sequence length constant at $N = 16$, this provides controlled evidence that the observed collapse in decodability (from $65.24\%$ to $18.47\%$) is strongly associated with dependency distance rather than sequence length alone.

## D Probe Training and Baselines

All representational decoding was performed using `scikit-learn` Logistic Regression probes. Because neural network activations vary significantly in magnitude (particularly across modern primitives like RMSNorm and SwiGLU), all extracted residual stream tensors were normalized using a `StandardScaler` prior to probe training. Probes were trained on the in-distribution $N = 8$ datasets utilizing the L-BFGS solver with L2 regularization (max_iter $= 1000$) to guarantee strict convergence.

To prevent data leakage and rigorously evaluate extrapolation, probes were trained across all sequence token positions simultaneously from the in-distribution dataset. The target label was strictly the exact ground-truth DP state value ($L[i]$) for that token. Because OOD sequences ($N = 16, 24, 32$) can theoretically produce DP state values (classes) never

observed during $N = 8$ training, the linear probes must evaluate zero-shot class extrapolation based on the continuous residual stream geometry. To ensure balanced evaluation across all DP-state classes and prevent the dominant frequency of low-value states (e.g., $L[i] = 1$) from artificially inflating extraction performance, all reported probe metrics reflect macro-averaged token-level accuracy.

For the Token-Shuffled Baseline, we applied a random permutation strictly to the input sequence tokens $S$ prior to the network's forward pass. The network still received the properly formatted `[BOS]` $S$ `[SEP]` wrapper, but the algorithmic ordering of $S$ was destroyed, providing evidence that probe performance depends on sequence-order information rather than static token statistics alone.

# E Master Experiment Summary

While our global behavioral scaling metrics are averaged across 5 random seeds to ensure generalizability, targeted causal interventions (e.g., activation patching, Oracle, PI) are evaluated on a fixed, representative checkpoint to cleanly isolate specific internal mechanisms. Because this introduces slightly different local baselines (e.g., $2.08\%$ vs. $1.18\%$), we summarize our core experiments, sequence lengths, and associated baselines in Table 5. This transparently contextualizes the varying OOD baselines reported across specific experimental conditions in the main text.

Table 5: Master Summary of Core LIS Experiments and Baselines

| **Experiment Name** | **Model** | **Train $N$** | **Test $N$** | **Seeds** | **OOD Baseline** | **Intervention / Result** |
|---|---|---|---|---|---|---|
| Behavioral Scaling | 2-Layer LIS | 8 | 16 | 5 (Mean) | $2.08\% \pm 1.49\%$ | None (Observational) |
| OOD Activation Patching | 2-Layer LIS | 8 | 16 | 1 (Fixed) | $1.18\%$ | L1 Patch ($14.50\%$) |
| Oracle Routing Intervention | 2-Layer LIS | 8 | 16 | 1 (Fixed) | $0.60\%$ | Ground-Truth Pointer ($81.00\%$) |
| Positional Interpolation (PI) | 2-Layer LIS | 8 | 16 | 1 (Fixed) | $1.30\%$ | Linear PI ($19.00\%$) |
| Randomized Pos. Training | 2-Layer LIS | 8 | 16 | 5 (Mean) | $2.20\% \pm 1.45\%$ | Freq. Randomization ($28.06\%$) |

## E.1 Architectural Comparisons: ALiBi vs. NTK and Extended Baselines

To further validate the precise relative mapping required by algorithmic routing, we compare the Linear PI intervention ($19.00\%$) against an identical Transformer trained with Attention with Linear Biases (ALiBi) (Press et al., 2021), which retains $11.80\%$ accuracy at $N = 16$, and NTK-Aware scaling ($1.40\%$). We also evaluate Random Positional Jitter (injecting zero-mean Gaussian noise $\sigma = 0.2$ to absolute positional indices) to test reliance on absolute geometries.

The severe failure under Random Positional Jitter ($1.19\%$) indicates that the routing mechanism heavily overfits to exact absolute geometries rather than learning robust, translation-invariant relative distances. Furthermore, the failure of NTK-Aware scaling highlights that algorithmic DP pointers require exact, relative distance mapping (e.g., retrieving a value exactly $j - i$ steps away), which Linear PI uniformly preserves, whereas NTK scaling applies non-uniform frequency distortions.

To mechanistically explain the performance gap between Linear PI and ALiBi, we empirically extracted the pre-softmax query-key logits from the primary routing heads of both models on OOD sequences.

As shown in Figure 8, ALiBi's hardcoded linear distance penalty becomes overwhelmingly suppressive at extended distances, artificially driving the target logits below zero. Rather than completely blinding the model, this imposes a strict distance penalty that can mathematically suppress the exact, long-range pointer retrieval required for dynamic programming.

## E.2 Structural Sanity Check (CKA and RSA Alignment)

To verify that these behavioral and routing failures are not simply the result of a catastrophic fracture in the overall geometric manifold of the network, we utilized Centered Kernel Alignment (CKA) and Representational Similarity Analysis (RSA) to evaluate the global macro-geometry of the residual stream under OOD shifts.

As illustrated in Figure 9, while Layer 1 (routing) exhibits structural degradation under OOD shifts, Layer 0 geometries remain tightly clustered. Furthermore, CKA scores between the standard OOD representations and the PI-intervened representations approach $\sim 0.9999$ across both Layer 0 and Layer 1. These measures capture different aspects of similarity: RSA compares ID–OOD relational structure, whereas the CKA intervention comparison asks whether the PI intervention globally transforms the representation. Thus, high CKA does not imply absence of localized

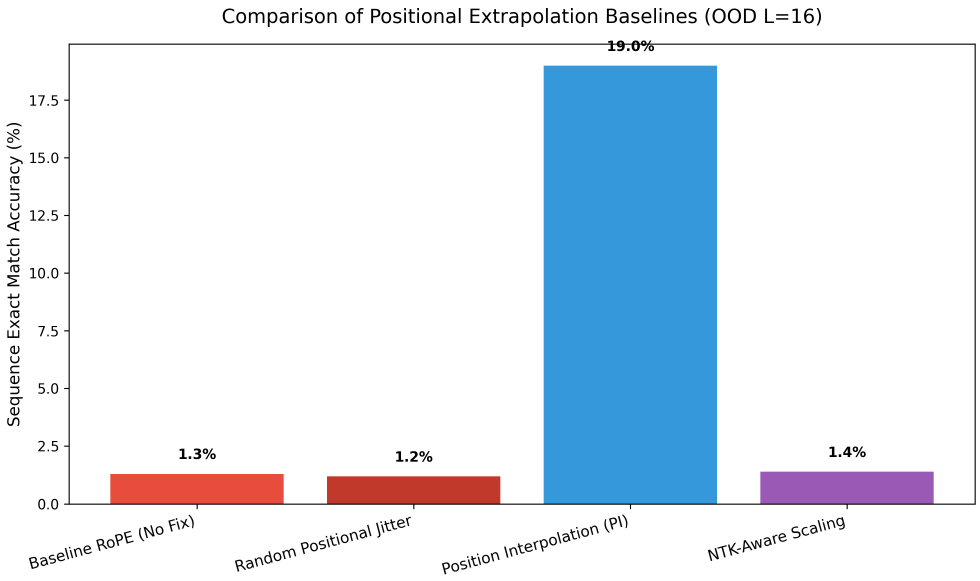


Figure 7: Comparison of Positional Extrapolation Baselines on OOD sequences ($N = 16$). Random jitter causes severe failure, indicating a reliance on absolute geometries. Linear PI outperforms NTK-Aware scaling, highlighting the strict linear distance constraints required for dynamic programming pointers.

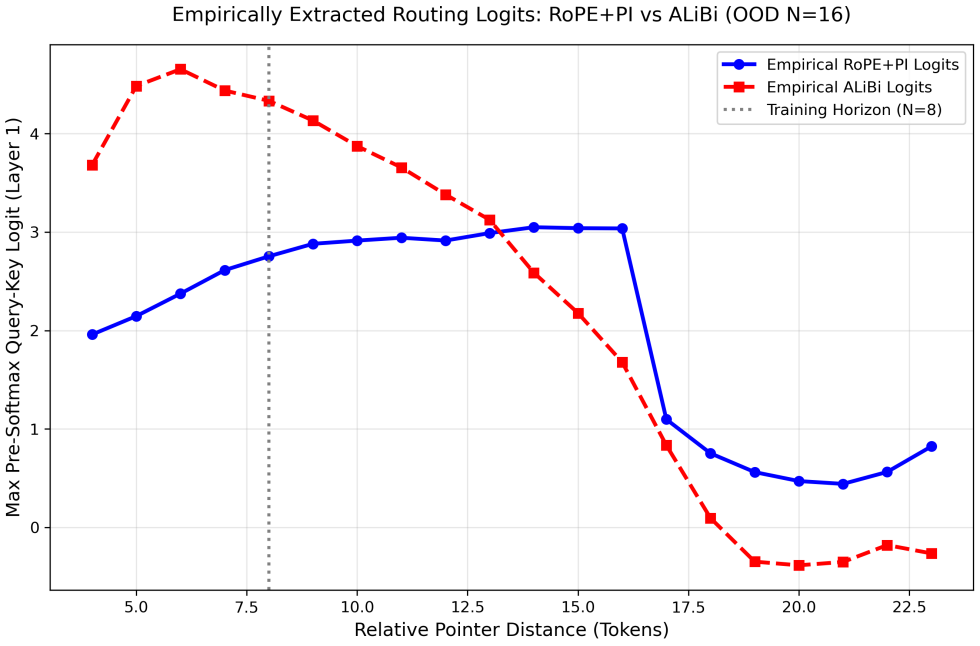


Figure 8: Empirically Extracted Routing Logits (OOD $N = 16$). ALiBi's strict linear penalty severely suppresses distant target logits. RoPE+PI dynamically scales the geometry, preserving high logit magnitudes across the extended sequence.

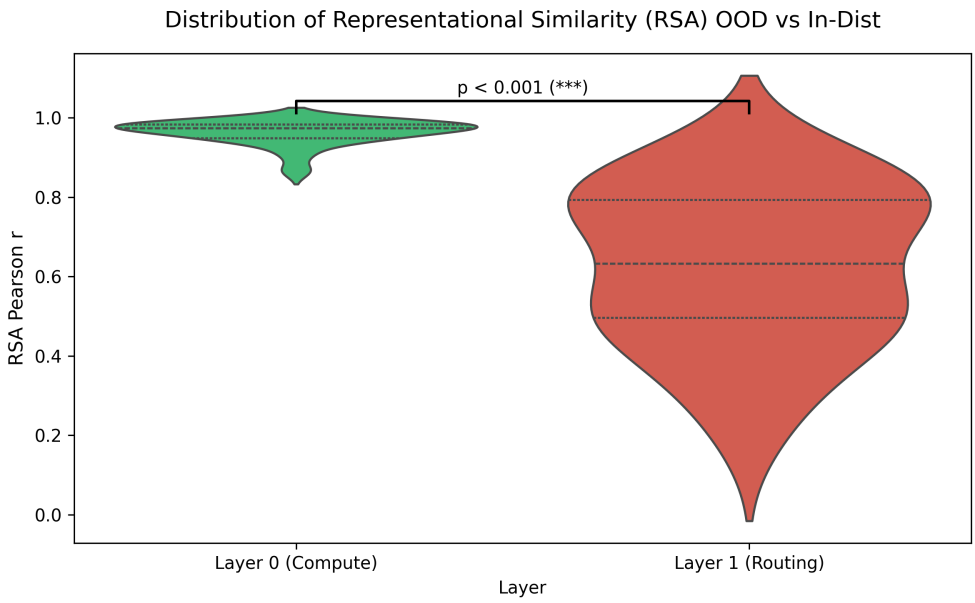


Figure 9: Distribution of Representational Similarity (RSA) between In-Distribution ($N = 8$) and OOD ($N = 16$) lengths across 20 data splits. Layer 0 (Compute) retains tight geometric alignment, while Layer 1 (Routing) exhibits structural degradation.

routing-specific degradation. As a negative control, it simply confirms that the global residual-stream geometry remains highly similar across conditions.

### AI USE STATEMENT

ChatGPT was used solely for language editing and improving the clarity of the manuscript.

### ETHICS STATEMENT

While this study focuses on synthetic algorithmic tasks to ensure tractable mechanistic interpretability, understanding the internal failure modes of length extrapolation has critical implications for the reliable deployment of large language models. As models are increasingly tasked with reasoning over highly extended contexts such as in legal document analysis, medical record synthesis, and automated software engineering silent algorithmic failures pose significant safety and alignment risks. A silent failure occurs when a model successfully computes and retains the correct factual state internally, but fails to appropriately route or express it in the output due to positional constraints, potentially resulting in confidently hallucinated generations. By diagnosing these specific architectural vulnerabilities and isolating the causal role of positional geometries, this work contributes to the broader objective of designing neural architectures with predictable, transparent, and robust out-of-distribution reasoning capabilities.

## REPRODUCIBILITY STATEMENT

We provide complete model configurations, optimization settings, dataset construction procedures, probing protocols, causal intervention procedures, and evaluation details in Sections 3–7 and Appendices A–E. The appendix reports the exact targeted intervention configurations, probe training hyperparameters, additional architectural baselines, and statistical evaluation details. Anonymous source code, dataset generators, and experiment configurations are provided as supplementary material to ensure full reproducibility of all claims.

### ACKNOWLEDGMENTS

None.