TEST TIME DILATION IN SEQUENCE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

A model can have access to earlier information yet fail to use it when a dependency becomes longer. We study *test-time dilation*: expanding the material between an earlier state and its use after training, while preserving the required answer. Controlled experiments separate three requirements: preserving one value, retrieving one of several bindings, and preserving an outer value while computing inside a scope. Attention can retrieve content directly, but positional scoring can make that route unreliable. Selective recurrence can preserve information, but repeated updates can alter it. Targeted interventions restore both forms of access, and scaling allows ordinary recurrence to learn more reliable carry. Neither successful retrieval nor accurate copying alone establishes robustness to the other requirements. A controlled scoped task makes the distinction concrete: models that master an order-sensitive program throughout the training range can still collapse when answer-preserving operations are inserted beyond that range. Dilation robustness is therefore a property of the learned computation, not a guarantee supplied by an attention edge or a long context window.

1 INTRODUCTION

A dependency can become longer without changing the information it requires. Compare "the report is accurate" with "the report that the analysts revised after several meetings is accurate." The expanded relative clause changes the sentence, but the main verb still agrees with the same singular subject. Similar separation can arise when a retrieved passage or a tool response grows between an instruction and its later use. A model may remain inside its familiar context window while the particular dependency exceeds the distances encountered during training.

Does the learned computation survive that change? An attention layer offers a direct connection to an earlier token, whereas a recurrent model updates its state at every step. These observations suggest an advantage for attention, but do not determine the answer. A positional score can suppress an available attention connection. Conversely, a recurrent transition can preserve a memory exactly. **The existence of an information path is not the same as learning a reliable way to use it.**

We test this distinction by moving a source farther from its query without adding tokens, changing the answer, or updating the model's weights (Figure 1). The first experiment asks whether one value survives. The second adds several bindings and asks for a selected older value. The third asks the model to preserve one value while another state undergoes a sequence of operations. These are different requirements, not simply three difficulty settings of the same task.

Each experiment tests the limit of the preceding explanation. Reliable copying helps one value survive, but does not keep every older binding recoverable after later writes. Successful retrieval, in turn, does not establish the ability to compute while protecting another memory.

Our contributions are:

- A test-time dilation protocol that distinguishes longer dependencies from larger inputs, and separates single-value preservation, multi-binding retrieval, and computation within a scope.
- Direct tests of the failure mechanisms: changing positional scoring restores attention-based retrieval, while preventing post-source recurrent updates restores stored information.
- Evidence that successful carry is not sufficient for retrieval or computational extrapolation, with scaling and scoped controls showing how the learned solution changes.

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 WHAT CHANGES IN A DILATION TEST?

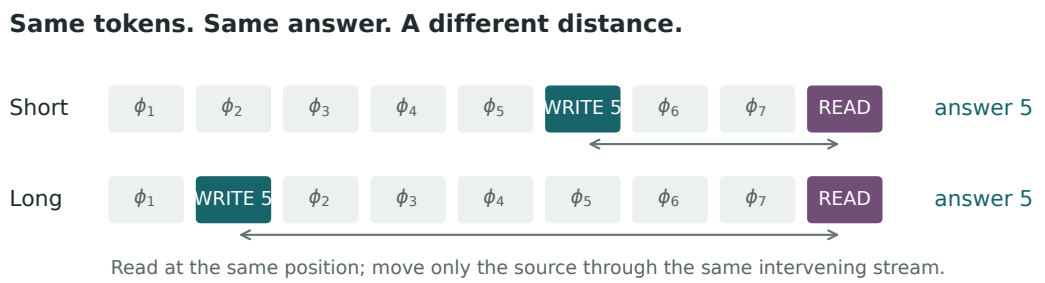


Figure 1: Longer dependency, not a larger input. Both rows contain the same source value, final query, and ordered intervening tokens. Moving the source makes more tokens occur between writing and reading. Models are frozen during this comparison. The diagram is a short illustration; the experiments use 128-token sequences.

Training support versus test-time behavior. In the central experiments, models learn from source–query distances at most 16. We evaluate the frozen checkpoints at distances up to 120. Training and evaluation both use 128-token sequences. The test is not whether an architecture *can learn* broad spacing after seeing it; it is whether the solution learned from short dependencies already works when stretched. Broad-gap training is a useful control for this distinction, not a prerequisite for the definition of dilation.

Exact pairs. Let s and q be the source and final-query positions, with tokens x_s and x_q, and let $\Phi = (\phi_1, \ldots, \phi_N)$ be the ordered intervening stream. For distance $d = q - s$, construct

$$D_d(x) = [\Phi^{(d)}_{\text{before}},\ x_s,\ \Phi^{(d)}_{\text{between}},\ x_q], \qquad |\Phi^{(d)}_{\text{between}}| = d - 1. \tag{1}$$

The source splits the same stream at different locations. Removing source and query recovers exactly Φ in every member of the family. Thus length, token identities, their relative order within Φ, and the label are fixed. For retrieval, Φ includes the other bindings. The query always sees the entire preceding input; we do not equate unseen suffix padding with visible context. A drop along a paired curve therefore means that the same checkpoint answers fewer of the same underlying problems correctly after the source is moved. It does not mean that the longer-gap examples contain more facts to store. Moving the source also changes its preceding context, which motivates the localized intervention in Section 4.1.

Three questions, defined by what must remain available. *Simple dilation* presents one `WRITE(5)` and a later `READ?`; the answer is 5. Unrelated intervening tokens test persistence. A harder variant replaces some of them with explicitly marked distractors carrying candidate values. These are not additional queryable bindings: there is still one distinguishable true source.

Retrieval dilation presents several atomic bindings, such as $A \to 7$, $B \to 2$, and $C \to 5$, and later asks $Q(B)$. The model must keep multiple candidate memories recoverable because different examples query different keys. We stretch the target–query distance at fixed memory load. Presenting other bindings after the target tests whether later writes interfere with an older memory, not merely whether one current value survives. The queried key varies across examples, so success requires selecting the requested binding rather than always returning the most recent value.

Scoped dilation preserves an outer value while an inner state is transformed. Its two queries test the evolving state and the original outer value. Here the extra content is an operation, not ignorable filler. Section 6 uses fixed readout positions and inserts adjacent identity blocks that leave both final answers unchanged. This holds answers and visible context fixed, but changes the number of active operations.

Models and comparisons. Simple and retrieval dilation compare five two-block, width-32 models with approximately 26K–28K parameters: NoPE, RoPE, canonical ALiBi, a diagonal selective recurrence, and its exact-copy variant. The attention variants differ in positional treatment; the

recurrent variants differ in their update gate within the same residual scaffold. ALiBi uses the published head-slope schedule (Appendix A). The recurrent mixer is not full Mamba. These small controlled models are chosen to expose mechanisms rather than rank production language models; extending the resulting diagnostics to pretrained models is a separate empirical question.

Models train from scratch on freshly generated examples. At a query, the loss rewards probability assigned to the correct value; that value is a training target, not an extra input token. Simple and retrieval examples have one scored query each. The scoped task scores both answers.

Simple-dilation curves average six training seeds. Retrieval, interventions, scaling, and the simplified scoped comparison use three. All seeds, including poorly generalizing seeds, are retained. Simple models solve the short reference condition; where a harder task is not mastered, we distinguish acquisition failure from loss of an already learned ability. In the scoped comparison, both short-task acquisition and subsequent dilation behavior are informative. Appendix A specifies training budgets, losses, evaluation counts, and reproducible configurations.

3 Background and connection to existing work

Associative recall has become a useful diagnostic for comparing attention and efficient sequence models. Arora et al. (2024) show that efficient models can succeed on simple recall while lagging on the multi-query demands reflected in language. Arora et al. (2025) identify architecture-specific retrieval algorithms, while Bick et al. (2025) describe a shared gather-and-aggregate organization with implementation-specific bottlenecks. These results motivate comparing mechanisms rather than treating equal accuracy as evidence of equal computation. Our dilation tests ask a complementary question: once a short-range solution has been learned, which parts of that solution survive when the dependency is stretched?

The attention side builds on self-attention, RoPE, ALiBi, and controlled studies of length generalization (Vaswani et al., 2017; Su et al., 2024; Press et al., 2022; Kazemnejad et al., 2023). NoPE retains a causal mask and can therefore learn order-sensitive representations without explicit positional embeddings. For recurrence, Mamba's selective-copying task and analyses of input selectivity establish that selective state-space models can learn which information to preserve (Gu & Dao, 2023; Huang et al., 2025). Copying limitations for specified fixed-state model classes (Jelassi et al., 2024) concern information load, which is distinct from holding one already-established fact over additional steps. Formal-language analyses likewise distinguish state tracking from harder composition under specified assumptions (Sarrof et al., 2024; Merrill et al., 2024).

Our contribution is not another generic long-context benchmark. We hold the answer and visible context fixed while changing what the intervening span demands of the learned computation. This lets us separate distance-sensitive access, recurrent transport, multi-binding addressability, and extension of an ordered computation within one experimental framework.

4 Simple dilation: an available route can become unreliable

The simplest question is also the cleanest: can a model recover the same value after more intervening tokens? We write this transformation as

$$D_d^{\mathrm{simple}}(x) = [\Phi_{\mathrm{before}}^{(d)},\, x_s,\, \Phi_{\mathrm{between}}^{(d)},\, x_q], \qquad d = q - s,$$
$$y\Big(D_d^{\mathrm{simple}}(x)\Big) = y(x),$$

so the target is fixed while only the source–query separation changes. Figure 2 shows two different answers. NoPE remains at 100% throughout the tested range. The ordinary selective SSM becomes less reliable, reaching 65.9% with unrelated tokens and 37.2% when some tokens carry competing values. The exact-copy variant reaches 100% and 82.6%, respectively. Thus preserving one value can fail even though the number of facts to remember never increases.

The two positional attention variants fail differently. RoPE falls, recovers to 100% at distance 64 in every seed, and falls again. The linear-bias variant degrades more steadily. Their errors are not explained by a lack of an attention edge: that edge exists at every tested distance. The competing-value manipulation changes their curves much less than changing distance itself.

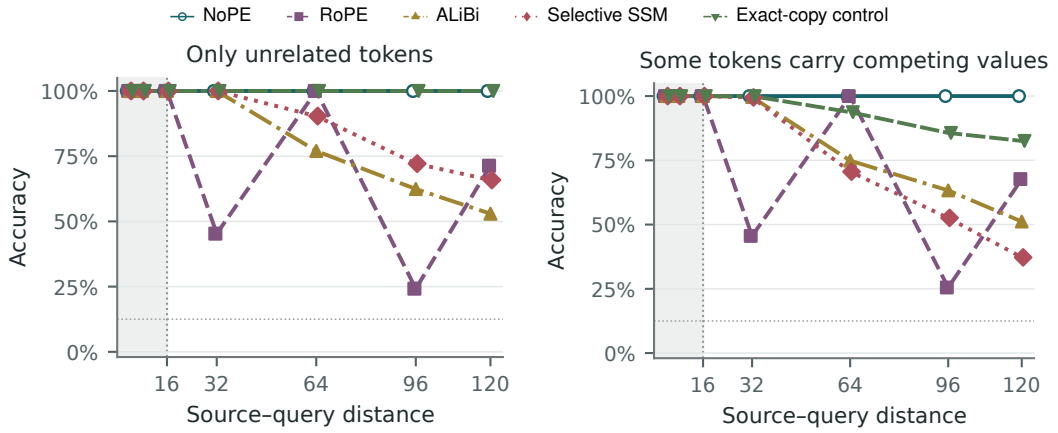


Figure 2: Simple dilation: the same value becomes harder to recover. Training distances end at the dotted boundary, $d = 16$. The right panel adds explicitly distinguishable tokens carrying competing values; it remains a single-memory task. At this small scale, the recurrent models lose more accuracy under added interference, while the attention variants change much less. Lines use true distances; markers are slightly dodged to show overlaps. Means over six seeds; Appendix B reports variability. RoPE's recovery near 64 is consistent with rotary phase realignment.

4.1 Change the suspected computation, not the example

For one attention head, the score on an earlier source has the form

$$\ell_{qs} = \frac{\mathbf{q}_q^\top \mathbf{k}_s}{\sqrt{d_h}} + b(q, s), \qquad \alpha_{qs} = \frac{e^{\ell_{qs}}}{\sum_{j \le q} e^{\ell_{qj}}}. \tag{2}$$

RoPE changes the effective dot product with relative position; ALiBi bias directly penalizes distance. NoPE has neither explicit mechanism. This does not make a deep causal NoPE Transformer position-invariant by construction: its contextual representations can still change. Empirically, the measured NoPE source scores and attention mass remain nearly constant in this task.

We keep the tokens, order, causal mask, and answer unchanged, but bring the source's positional relation back to a distance of 16. For RoPE, we change the source's rotary index at each layer; for ALiBi attention, we change the final query's bias toward the source. These interventions differ algebraically, but test the same question: does the learned content route work when the positional obstacle is removed? Both models return to 100% at distance 120 (Figure 3, left). This identifies positional computation as a cause of their failure in these checkpoints, not a general inability of attention to retrieve.

For the selective recurrence, the corresponding computation is

$$h_t = a_t \odot h_{t-1} + (1 - a_t) \odot u_t, \qquad a_t = \sigma(W_a x_t + b_a). \tag{3}$$

Here u_t is the input-dependent candidate state and a_t controls how much of the previous state survives. A gate of one copies the old state; a gate of zero replaces it with the candidate. The contribution of a coordinate present at the source is multiplied across subsequent filler steps. If each multiplier is at most $1 - \epsilon$, its retained contribution before the query is at most $(1 - \epsilon)^{d-1} \le e^{-\epsilon(d-1)}$. Later updates can also introduce competing content. A declining carry product is therefore a candidate explanation, not by itself proof that usable task information has vanished.

The direct test leaves the source computation normal and replaces only updates strictly *after* it and *before* the query by $h_t = h_{t-1}$ in each recurrent layer. At distance 120, random-token accuracy rises from 70.3% to 100%, and competing-value accuracy from 39.0% to 99.9% (Figure 3, right). A pre-source-only intervention has essentially no effect (Appendix C). The intervention never inserts the correct answer: it preserves whatever was encoded at the source and leaves the query update normal. Its success implicates subsequent processing in the loss of usable information.

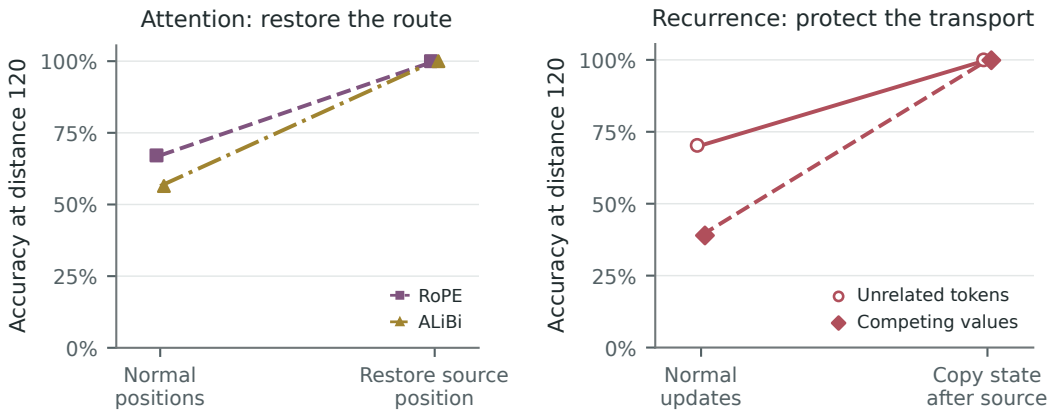


Figure 3: The route exists; changing its use restores the answer. Left: changing the source's positional relation in all layers restores position-sensitive attention. Right: copying recurrent state only after the source restores access to the stored value. The inputs and labels are unchanged. These diagnostic evaluations use three seeds, distinct from Figure 2's six-seed batches. They intervene on known source locations to identify mechanisms, not to supply an inference algorithm.

4.2 Copying and scale change the learned solution

An illustrative recurrent control replaces the sigmoid carry gate with

$$g_t = 1 - e^{-\operatorname{ReLU}(W_g x_t + b_g)}, \qquad h_t = (1 - g_t) \odot h_{t-1} + g_t \odot u_t. \tag{4}$$

A nonpositive preactivation makes $g_t = 0$, so the state is copied exactly at a finite parameter value. This is an instance of an established design idea: preserving state and skipping updates have antecedents in Zoneout, Skip RNN, and selective-update recurrence (Krueger et al., 2017; Campos et al., 2018; Yin et al., 2026). The contribution here is the controlled diagnosis and comparison, not priority for copying state. In real arithmetic, the sigmoid in Eq. 3 approaches identity asymptotically; finite-precision saturation can nevertheless produce exact numerical ones.

The gate must still learn when to copy and which coordinates to protect. Perfect retention through random tokens need not transfer to distractors that resemble useful information: those tokens can trigger writes even when the target should remain unchanged. Providing an identity operation does not specify when to use it.

Scaling the ordinary selective SSM from roughly 26K to 3M parameters removes the observed simple-dilation failure as it learns near-identity channels (Appendix B). NoPE remains perfect across the same three scales, whereas RoPE's distance sensitivity remains non-monotonic. Thus additional capacity can make transport more reliable, while explicit copying achieves robust single-value behavior at smaller scale. This is a capacity-sensitive learned solution, not an unavoidable recurrent limitation or evidence that every harder memory requirement has been solved.

5 Retrieval dilation: carrying one value is not keeping a collection

Does accurate copying solve long dependencies? Retrieval dilation adds eight bindings and a final key selecting which value to return. Let $b_* = (k_*, v_*)$ be the queried binding and let $\mathcal{B}_{-*}$ denote the ordered non-target bindings. For the later-writes placement, a paired member can be written

$$D_d^{\mathrm{ret}}(x) = [\Phi_{\mathrm{before}}^{(d)},\, b_*,\, \Phi_{\mathrm{between}}^{(d)},\, \mathcal{B}_{-*},\, q(k_*)],$$
$$y\big(D_d^{\mathrm{ret}}(x)\big) = v_*,$$

with $\mathcal{B}_{-*}$ preserved while the target–query separation increases. The target-last condition places the same preserved non-target bindings before b_*. Each binding is one atomic token, removing local key–value binding difficulty and concentrating the test on whether the requested association remains recoverable. This is not the multi-query associative-recall benchmark of Arora et al. (2024).

Consider $B \to 2$. If the other seven bindings precede it, the model can establish the target and then carry it through unrelated tokens. If those bindings follow it, the model must incorporate new records while retaining access to $B \to 2$. They concern other keys, not revisions to B: the required answer remains 2. A later write is therefore harmless to the target in the task definition but can still disturb its representation inside the model. Within each condition we vary target–query distance using exact pairs at fixed memory load. Moving other writes from before to after the target is a separate interference manipulation, not merely a distance change.

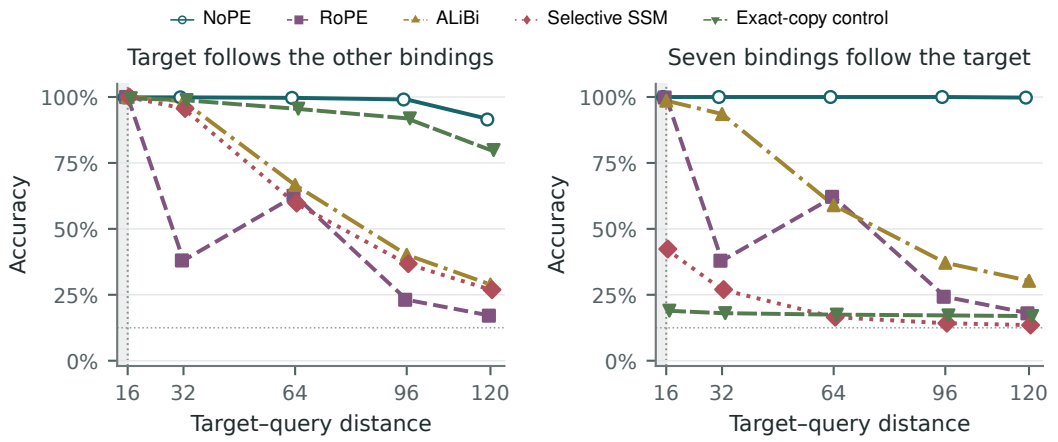


Figure 4: Retrieval dilation separates access from preservation. Left: the target follows its competitors. Right: seven other bindings follow the target. RoPE and canonical ALiBi learn the short-gap retrieval task but lose accuracy with distance; later writes also make recurrent retrieval difficult before dilation. Absolute accuracy, three-seed means; only markers are offset. Uncertainty and query-placement controls are in the appendix.

When the target is written last, all five models are essentially perfect at distance 16. At distance 120, NoPE retains 91.5%, the ordinary selective SSM 26.9%, and the exact-copy control 79.9%. Accurate copying therefore addresses much of the additional transport loss even with several bindings in context.

The attention variants are not interchangeable. RoPE and canonical ALiBi reach essentially 100% at the short gap when the target is last, but fall to 17.1% and 28.9% at distance 120. With later competing writes, their long-gap accuracies are similarly low, 18.0% and 30.3%. The available attention edge therefore does not give every positional treatment NoPE's retrieval robustness.

When seven writes follow the target, the recurrent difficulty is already present at distance 16: ordinary recurrence achieves 42.4% and exact copy 19.0%, compared with 100% for NoPE. These are base-retrieval failures, not errors that should all be attributed to dilation. Among the examples each model answers correctly at distance 16, 34.6% remain correct at 120 for ordinary recurrence and 89.1% for exact copy; NoPE retains 99.8%. Conditional survival measures what additional distance destroys, but is not overall retrieval accuracy and conditions on different solved subsets for different models. The nearly flat, low exact-copy curve on the right is consequently not a successful memory collection: it mostly retains the small subset of bindings it could already recover.

The experiment therefore shows two obstacles rather than one. First, the model must construct a representation from which the queried older binding can be recovered after other writes. Second, that representation must remain usable over a longer path. Copying greatly improves the second obstacle without reliably solving the first. NoPE handles both in this atomic-binding setting.

A query-after-placement control changes the training distribution so no binding receives a target-specific position. Recurrent base-task performance improves substantially, but the distinction persists: ordinary recurrence loses many already-correct examples with distance, while the exact-copy model mostly preserves those it solves. The control changes the placement distribution as well as query selection, so it is a robustness check rather than an isolated causal attribution (Appendix C).

6 SCOPED DILATION: COMPUTE INSIDE, PRESERVE OUTSIDE

Retrieval can leave earlier content in place and look it up later. Scoped dilation instead inserts content that must be processed. We use three inner states with two noncommuting operations, $A = (0\ 1\ 2)$ and $B = (0\ 1)$. Starting from state 1, AB and BA produce different answers, so order is essential. Let P_d be the inner program of width d; then

$$D_d^{\text{scope}}(x) = [o,\ \Phi_{\text{pre}},\ \text{OPEN}(z_0),\ P_d,\ \text{INNER?},\ \text{CLOSE},\ \text{OUTER?}], \qquad |P_d| = d,$$
$$P_d(z_0) = P_{16}(z_0), \qquad y(D_d^{\text{scope}}(x)) = \big(P_{16}(z_0),\ o\big).$$

Training covers scope widths 2–16. Each scope contains one A and one B in random order and positions, with identity operations elsewhere; the model reports the final inner state and then an outer value stored before the scope. At test time, the width-16 example is expanded to 32, 64, 96, or 120 by inserting unmarked identity words such as BB, $ABAB$, and $BABA$ at random boundaries. Each inserted token is an ordinary operation and changes intermediate state, but the complete word composes to identity. Both answers, query positions, and total visible context are therefore unchanged.

Carry-compose is an illustrative two-lane recurrent control: an exact-capable protected carry lane stores persistent information while an LSTM working lane performs state-dependent computation (full formulation in Appendix A). To teach the protected lane what to retain, it receives an auxiliary outer-state trace loss before `OPEN`; the other plotted models use only endpoint losses. It is therefore a sufficiency control for separating persistent and working state, not a parameter-matched architecture ranking.

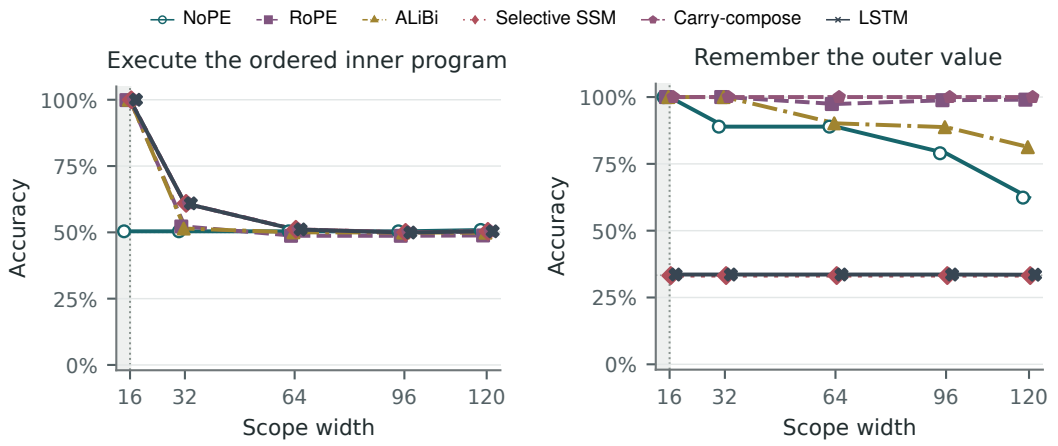


Figure 5: Scoped dilation: training support ends at width 16. Only the training boundary and out-of-support widths are shown. The core attention variants and selective SSM retain the same marker identities as Figures 2 and 4; Carry-compose and the LSTM use distinct markers. RoPE, ALiBi, the selective SSM, Carry-compose, and the LSTM master the ordered inner task at width 16; NoPE remains near the 50% order-blind baseline. Beyond 16, every model that mastered the inner task falls rapidly toward that baseline. Carry-compose preserves the separate outer value through width 120. Three-seed means; markers are slightly dodged to show overlaps.

What is being dilated? Consider an inner state initialized to 0 and the two-operation program AB. Applying $A = (0\ 1\ 2)$ gives $0 \to 1$, and then $B = (0\ 1)$ gives $1 \to 0$, so the correct answer is 0. A dilated version can insert the identity word BB, producing $ABBB$:

$$0 \xrightarrow{A} 1 \xrightarrow{B} 0 \xrightarrow{B} 1 \xrightarrow{B} 0.$$

The inserted operations are not filler: every inserted B changes the intermediate state. Yet the pair BB cancels, so the final answer is unchanged. Scoped dilation therefore asks whether a computation learned over short ordered programs continues to work when additional meaningful state transitions are inserted without changing the required result.

The contrast with the earlier tests is direct. Simple dilation asks whether one fact survives irrelevant material, and retrieval asks whether one of several bindings remains addressable. Here the inserted span performs real state transitions. RoPE is 99.9% accurate on the inner task at width 16 but falls

to 48.8% at width 120; ALiBi, the selective SSM, Carry-compose, and the LSTM show the same qualitative collapse after mastering the training range. The exact-copy control, omitted from Figure 5 for legibility, follows the selective SSM's inner curve on this test (100% at width 16, 60.9% at 32, and 50.4% at 120). Exact copying therefore does not solve computational extrapolation. NoPE shows the complementary failure: it preserves the outer value but never learns the order-sensitive inner computation in this small configuration. Thus NoPE's strong carry and retrieval behavior is not a universal dilation advantage: when success depends critically on positional order, this configuration can fail even inside training support.

The outer panel should be read separately. The attention variants solve the outer query within training support, while the selective SSM and LSTM are already near chance on that long fixed dependency. Carry-compose instead remains at 100% outer accuracy through width 120, showing that an explicitly protected memory can coexist with a working computation even though the computation itself still fails to extrapolate. We therefore do not attribute the ordinary recurrent outer failure to scoped dilation. The useful separation is that preserving a value, retrieving a binding, and extending a learned computation are distinct obligations.

7 Discussion: what dilation separates

The experiments identify a recurring design failure: a model solves the short task using a computation whose unwanted sensitivity is negligible over training distances. Positional scores can make a relevant item appear too far away; small recurrent updates can repeatedly alter a fact that should remain fixed; later writes can disturb an older binding; and an active computation can exceed the range over which an ordered procedure was learned. More context capacity does not directly test any of these behaviors. Dilation asks whether the required relation survives changes to its realization.

Test	What expands?	What must stay invariant?	What failure means
Simple	Tokens between one stored value and its use	One established value remains usable	Transport or positional access becomes unreliable
Retrieval	Distance around several valid bindings and intervening writes	The requested older binding remains independently recoverable	New information disrupts addressability, or distance makes the target inaccessible
Scoped	An ordered computation inside a protected outer dependency	The computation's answer and a separate outer value	A learned computation fails to extend, or active computation disrupts persistent memory

Table 1: Test-time dilation separates obligations that a single long-context score can conflate. The three tasks differ in what the intervening span is allowed or required to do.

These tests are deliberately cumulative. Simple dilation asks whether a model can preserve one established fact. Retrieval prevents a trivial extension of that solution: copying one current state is insufficient when several older states must remain independently addressable. Scoped dilation changes the requirement again. The intervening tokens can no longer be ignored because they are operations whose order affects the computation. The model must update one state while preserving another.

This decomposition changes how the architecture comparison should be read. NoPE is exceptionally robust on the simple and atomic-retrieval tests, but it does not acquire the order-sensitive scoped computation in our small configuration. RoPE and ALiBi do acquire that computation throughout training support, yet it fails when additional answer-preserving operations are inserted beyond that support. The ordinary recurrent baselines readily acquire the ordered computation, while their small state can struggle to preserve an unrelated long-lived value; Carry-compose shows that explicitly protecting that memory does not make the learned computation extrapolate. Exact copying likewise repairs one transport problem but leaves computational extrapolation almost unchanged. Scale can move some of these frontiers—larger recurrence learns more reliable carry—without making the requirements equivalent. *There is therefore no single architecture-level property called "handling long dependencies": different learned operations fail under different forms of dilation.*

The fixes are correspondingly specific. Removing a positional obstacle restores an existing attention route. Exact or sufficiently accurate carry protects a representation that has already been established; it does not create an independently addressable collection of records. The scoped result adds a different limitation: mastering an ordered program throughout training support does not guarantee that the computation extends when more answer-preserving operations are inserted. These controls illustrate complementary design principles rather than an optimal architecture.

Training support is a control, not the main contribution. When expected dilations can be represented during training, ordinary supervision on broader-gap examples nearly eliminates the simple-dilation failures. A matched-support ablation gives essentially the same accuracy with and without a symmetric-KL agreement term (Appendix C). The practical requirement is to understand the kind of expansion expected at deployment and the relation whose target remains valid. This does not answer how a frozen model handles unanticipated dilation; it shows that several observed failures are properties of the learned solution rather than hard expressivity barriers.

From synthetic dilation to language. The controlled tasks make the invariant explicit: we know exactly which answer should survive an expansion. Natural language makes this harder, but the same distinction appears in ordinary model use. An instruction may be separated from its execution by a long tool trace. A question may be separated from the relevant retrieved passage by additional passages that are themselves meaningful. A subject and verb may be separated by a longer embedded clause. In an agentic response, an early commitment or variable may need to survive while many intermediate steps modify other parts of the working state.

These examples also illustrate why "long context" is not synonymous with dilation robustness. The entire interaction may fit comfortably inside the context window. The question is whether a particular relation still works after meaningful material is inserted between its endpoints. Constructing natural-language dilation benchmarks therefore requires more than adding arbitrary text: the expansion must preserve a well-defined target relation while remaining linguistically or computationally meaningful. Recurrent syntax work identifies protected grammatical features in trained models (Lakretz et al., 2019), offering one concrete target for such analysis. Extending dilation tests to pretrained language models is a natural next step, but requires identifying naturally occurring transformations that preserve a well-defined target relation. We therefore leave that extension to future work and focus here on controlled settings in which the invariant, intervention, and failure mechanism can be specified exactly. The model and artifact boundaries are recorded in Appendix D.

8 Conclusion

How do models handle a dependency when its intervening span expands? The answer depends on the learned operation. Simple dilation shows that information can remain present in the input while becoming harder to use: positional scoring obstructs an available attention route, and repeated recurrent updates alter a stored value. Changing the suspected mechanism restores performance, while greater recurrent capacity can make the learned transport reliable without an explicit copying rule.

The other tests identify what those successes leave unsolved. Copying a current representation does not ensure that an older binding remains recoverable after later writes. Mastering an ordered inner program within training support does not ensure that the same computation extrapolates when additional answer-preserving operations are inserted. Dilation therefore exposes different obligations of the same sequence model: access the right information, preserve what should not change, and transform what must. The central lesson is not that one architecture handles distance better, but that distance exposes whichever assumptions the learned computation made about how information would be carried, selected, or transformed. **Robustness to longer dependencies is a property of how these operations are learned and coordinated, not a guarantee supplied by an attention edge or a long context window.**

AI USE STATEMENT

Human work on the project included originating the dilation question and motivating examples, setting the research direction, refining hypotheses and task definitions, prioritizing model comparisons, controls, scaling, ablations, and replication checks, critically reviewing reported results, challenging interpretations and shortcuts, determining the claims and scope, and substantially revising the abstract, narrative, explanations, terminology, conclusions, figures, and appendix organization. Generative AI assisted with literature discovery and synthesis, mathematical formulation, experimental design, synthetic-data generators, code implementation and debugging, tool-mediated training and evaluation, result analysis, figure production, artifact checks, and drafting and revision of prose. These roles were iterative and collaborative; human contribution extended beyond approving AI-generated text. Human review is not represented as independent execution of every experiment, and the recorded computational checks were performed with AI assistance. No human-subject data or translation was involved. Final responsibility for the manuscript and artifacts, including AI-assisted material, is not delegated to the AI system.

REPRODUCIBILITY STATEMENT

Every numerical study presented here has executable code and retained checkpoint or run-level evidence in the supplied replication collection. Figure inputs, evaluation contracts, seed-level summaries, and the build script are packaged with the source. Earlier aggregate-only stress extensions and the later thinking-token screen are not used as evidence in this manuscript. Appendix D maps the retained studies and states what the local reruns establish. The replication archive also preserves the earlier experimental record separately from the current figure inputs.

REFERENCES

Aryaman Arora, Neil Rathi, Nikil Roashan Selvam, Róbert Csordás, Dan Jurafsky, and Christopher Potts. Mechanistic evaluation of transformers and state space models. *arXiv preprint arXiv:2505.15105*, 2025.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations*, 2024.

Aviv Bick, Eric P. Xing, and Albert Gu. Understanding the skill gap in recurrent language models: The role of the gather-and-aggregate mechanism. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 4324–4344, 2025.

Víctor Campos, Brendan Jou, Xavier Giró-i Nieto, Jordi Torres, and Shih-Fu Chang. Skip RNN: Learning to skip state updates in recurrent neural networks. In *International Conference on Learning Representations*, 2018.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Ningyuan Teresa Huang, Miguel Sarabia, Abhinav Moudgil, Pau Rodriguez, Luca Zappella, and Federico Danieli. Understanding input selectivity in mamba: Impact on approximation power, memorization, and associative recall capacity. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 25693–25727, 2025.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 21502–21521, 2024.

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

David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Aaron Courville, and Chris Pal. Zoneout: Regularizing RNNs by randomly preserving hidden activations. In *International Conference on Learning Representations*, 2017.

Yair Lakretz, Germán Kruszewski, Théo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in LSTM language models. In *Proceedings of NAACL-HLT*, pp. 11–20, 2019.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35492–35506. PMLR, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Yash Sarrof, Yana Veitsman, and Michael Hahn. The expressive capacity of state space models: A formal language perspective. In *Advances in Neural Information Processing Systems*, 2024.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

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

Bojian Yin, Shurong Wang, Haoyu Tan, Sander Bohte, Federico Corradi, and Guoqi Li. Efficient sparse selective-update RNNs for long-range sequence modeling. *arXiv preprint arXiv:2603.02226*, 2026.

A ESSENTIAL PROTOCOLS AND MODEL DEFINITIONS

Simple and retrieval data. The synthetic vocabulary distinguishes ordinary filler, value-specific source tokens, marked distractors, and a query. Values are indexed 0–7 in code; the examples in the text use readable labels. Marked distractors can coincide with the source value by chance, but their token type remains distinguishable. Training uses online-generated short gaps; simple evaluation spans $4, 8, 16, 32, 64, 96, 120$. Retrieval trains on one to eight bindings, target distances 8–16, and mixed non-target placements. Each binding is encoded as one token. At evaluation, target distance and the before/after write condition are controlled separately. Only one key is queried per sequence. Exact pairing preserves the residual ordered stream, including other bindings, within a given evaluation family.

Architecture. The small attention and diagonal recurrent models share RMS normalization, residual mixing, a second normalization, and a fourfold-expansion GELU feed-forward network. Recurrent layers additionally have input-dependent output gates. The candidate update is $u_t = \tanh(W_u x_t + b_u)$. NoPE retains the causal mask but has no explicit position embedding. RoPE rotates every adjacent coordinate pair with base 10,000. ALiBi uses the published schedule $m_h = 2^{-8(h+1)/H}$ for $H = 4$ heads, giving $(1/4, 1/16, 1/64, 1/256)$ in the reference ordering (Press et al., 2022). It is applied during training and evaluation. All ALiBi results reported here use fresh canonical-slope checkpoints; the earlier custom-slope runs remain separately labeled in the historical replication archive, not pooled into these results.

Experiment / model	Width	Layers	Updates	Batch
Simple: all attention variants	32	2	300	64
Simple: selective / exact-copy	32	2	220	32
Retrieval: all five models	32	2	1,000	64
Scoped: endpoint-only plotted models	32	2	600	16
Scoped: Carry-compose control	32	1	600	16

Table 2: Training configurations. Simple, retrieval, and simplified scoped runs use AdamW with learning rate 3×10^{-3}, weight decay 10^{-4}, and gradient clipping at 1. The standard scoped models use endpoint losses only. Carry-compose uses the same update budget plus a 0.25 outer-trace auxiliary loss and one LSTM working layer beside its protected carry lane. Attention has four heads.

Simplified scoped computation. The simplified scoped task uses three inner states and two noncommuting operations, $A = (0\ 1\ 2)$ and $B = (0\ 1)$, with the inner start state fixed to 1. A training scope has width $d \in [2, 16]$ and contains exactly one A and one B in random order and positions; the remaining operation slots are explicit identities. The inner target therefore distinguishes AB from BA. The outer value is written before the scope and queried after it. Evaluation anchors a width-16 example and inserts unmarked identity words BB, $ABAB$, or $BABA$ at random boundaries to reach widths $32, 64, 96$, and 120. Each word composes to identity, so both targets remain exactly fixed while the intermediate state changes inside the inserted block. The two query positions and total sequence length are fixed. The standard scoped models use only the two endpoint query losses. Carry-compose additionally receives an outer-state trace loss at positions before `OPEN`, which trains its protected lane; no scratchpad states are supplied as inputs.

Carry-compose formulation. Let $e_t = E(x_t)$ be the token embedding and $u_t = \mathrm{RMSNorm}(e_t)$. Carry-compose splits its recurrent state into a protected lane $p_t \in \mathbb{R}^{d_p}$ and an LSTM working lane $(h_t, c_t) \in \mathbb{R}^{d_w}$. The protected update is

$$r_t = \mathrm{ReLU}(W_g u_t + b_g), \qquad g_t = 1 - \exp(-r_t), \qquad v_t = \tanh(W_p u_t + b_p),$$

$$p_t = (1 - g_t) \odot p_{t-1} + g_t \odot v_t, \qquad (h_t, c_t) = \mathrm{LSTM}(u_t, h_{t-1}, c_{t-1}).$$

Thus any protected coordinate with nonpositive gate preactivation has $r_{t,j} = g_{t,j} = 0$ and copies exactly, $p_{t,j} = p_{t-1,j}$, at a finite parameter value. The two lanes are combined only at readout:

$$z_t = W_o[e_t; p_t; h_t] + b_o, \qquad \ell_t = W_{\mathrm{head}}\,\mathrm{RMSNorm}(z_t).$$

For the scoped experiment $d_p = d_w = 16$ and the model has one LSTM layer. Its objective is

$$\mathcal{L} = \mathcal{L}_{\text{inner}} + \mathcal{L}_{\text{outer}} + \lambda_{\text{trace}} \frac{1}{|T_{\text{pre}}|} \sum_{t \in T_{\text{pre}}} \text{CE}(\ell_t, o), \qquad \lambda_{\text{trace}} = 0.25,$$

where T_{pre} contains positions before `OPEN`. The trace loss is applied through the shared readout, rather than directly to p_t; it encourages the architecture to place the outer value in a representation that remains available while the working lane processes the inner scope. This additional supervision is why Carry-compose is used as a sufficiency control rather than a parameter-matched architecture comparison.

Evaluation. Figures 2–5 use recorded fixed families. Simple curves have six seeds and 384 examples per seed/condition; retrieval has three seeds per model and 1,024 families per condition. The simplified scoped figure has three seeds per plotted model and 512 fixed examples at each width, grouped into shared evaluation families. Three exact-copy SSM runs on the same families are retained as an unplotted comparison. Positional rescue uses three seeds with 512 examples each. Localized recurrent rescue uses 1,024 examples for seeds 0 and 1, and 512 for seed 2. Repeats are averaged within seed; seed means have equal weight. Reruns are not new seeds.

B ENDPOINT UNCERTAINTY AND SCALING

Main figures display seed means without error bars to keep overlapping trajectories legible. The following tables report mean ± sample standard deviation across training seeds. Standard deviation describes variation among trained models, not uncertainty of the mean. Full trajectories and seed-specific values are in the source data and replication records.

Model	Random tokens	Competing values
NoPE	100.0 ± 0.0	100.0 ± 0.0
RoPE	71.4 ± 15.4	67.7 ± 16.5
ALiBi	53.0 ± 17.2	51.2 ± 15.2
Selective SSM	65.9 ± 27.0	37.2 ± 17.7
Exact-copy control	100.0 ± 0.0	82.6 ± 26.2

Table 3: Simple dilation at distance 120, six seeds. Accuracy (%). All six NoPE and exact-copy runs score 100% with random filler. RoPE recovers to 100% at distance 64 in all six runs; a frequency-specific causal explanation was not tested.

	Target last	Seven later writes		
Model	$d = 120$	$d = 16$	$d = 120$	Survival
NoPE	91.5 ± 4.5	100.0 ± 0.0	99.8 ± 0.1	99.8 ± 0.1
RoPE	17.1 ± 1.3	99.8 ± 0.2	18.0 ± 0.9	18.0 ± 1.0
ALiBi	28.9 ± 2.3	98.6 ± 0.8	30.3 ± 5.6	30.7 ± 5.5
Selective SSM	26.9 ± 8.0	42.4 ± 16.2	13.5 ± 1.6	34.6 ± 12.6
Exact-copy control	79.9 ± 25.5	19.0 ± 2.2	16.9 ± 0.7	89.1 ± 8.8

Table 4: Retrieval, three seeds. Accuracy and survival (%). Survival is $P(C_{120} \mid C_{16})$, conditional on each model's own correctly answered reference examples. It is not accuracy on a common solved subset.

The scaling ladder changes width from 32 to 112 to 352, retaining two layers. Batch sizes are 64, 32, and 16. Runs are checked on the short task; most stop at 100 updates, with one small NoPE run reaching 125. The replication manifest gives exact stopping and evaluation settings. This is a capacity-and-training comparison, not a pure width intervention at identical optimization histories. At distance 120, the ordinary SSM's high-retention channels move much closer to identity as capacity increases. The layer-2 99th-percentile carry product is about 0.751, 0.991, and 0.994 across scales. The product alone is not a decoded memory measure; its interpretation is supported by the localized rescue.

Model	∼26K	∼306K	∼3M
NoPE	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
RoPE	68.8 ± 22.0	41.7 ± 10.9	79.9 ± 18.2
Selective SSM	20.8 ± 2.0	92.0 ± 7.1	100.0 ± 0.0
Exact-copy control	83.9 ± 11.8	100.0 ± 0.0	100.0 ± 0.0

Table 5: Fixed-task scaling, three seeds. Distance-120 accuracy (%) with competing values, computed from retained run-level records. All cells have 100% mean reference accuracy. These are separately trained checkpoints, not the six-seed runs in the first table.

Simplified scoped endpoints. At width 16, inner accuracy is $50.4 \pm 1.4\%$ for NoPE, $99.9 \pm 0.2\%$ for RoPE, and $100.0 \pm 0.0\%$ for ALiBi, the selective SSM, Carry-compose, and the LSTM. Because the fixed start state makes AB and BA the only two inner targets, an order-blind predictor scores 50%. At width 120, inner accuracy is 50.9 ± 2.0, 48.8 ± 0.0, 49.9 ± 1.8, 50.4 ± 0.0, 50.4 ± 0.0, and $50.4 \pm 0.0\%$, respectively. Carry-compose keeps outer accuracy at $100.0 \pm 0.0\%$ through width 120; NoPE, RoPE, and ALiBi reach 62.4 ± 24.7, 99.0 ± 1.5, and $81.3 \pm 12.0\%$ at width 120. The unplotted exact-copy SSM follows the selective SSM inner curve exactly on these evaluation families and remains near chance on the outer query.

C CONTROLS THAT DISTINGUISH COMPETING EXPLANATIONS

What the recurrent intervention changes. For source s and query q, the post-source mask replaces every recurrent update with identity only at $s < t < q$, in every recurrent layer. The source and query updates remain normal. A pre-source mask applies only at $t < s$. At distance 120, baseline/pre-source/post-source accuracies are $70.3/70.2/100.0\%$ for random filler and $39.0/39.1/99.9\%$ for competing values. This negative control tests whether the improvement is instead explained by altering the state into which the source is written. The post-source intervention suppresses both contraction and competing writes; it does not separately identify their contributions on decoy examples.

What the position intervention changes. The RoPE intervention substitutes $p_s = q - 16$ for the source's rotary index in the specified layers while retaining the original causal mask and token positions. It changes the source's positional interactions with other tokens in those layers, not only the final edge. The ALiBi intervention instead changes the final query's bias for the source to the distance-16 value. Thus these are targeted but not algebraically identical interventions. Normal/final-layer/all-layer accuracies at distance 120 are approximately $67/82/100\%$ and $56/87/100\%$, respectively.

Source-anchored control. A separately trained family fixes the source and its full prefix and moves the query. Ordinary recurrence still degrades, while NoPE and exact copy remain strong. This removes changed source encoding by construction, but moving the query changes how many tokens it can see; equal physical length is not equal visible context. It is complementary to, not a replacement for, the main fixed-query test.

Query-after-placement retrieval. A separate three-seed audit places all bindings in a shared short-gap band before choosing the queried key. Recurrent mastery improves. With seven later writes, short/long accuracy is $75.9/31.8\%$ for the selective SSM and $64.3/63.7\%$ for exact copy; NoPE is $99.3/99.3\%$. The audit also changes non-target placement and overwrites different filler positions during its evaluation, so it is not an exact stream-paired ablation. It checks training-distribution sensitivity; the primary retrieval curves provide the exact pairing.

Why use the simplified scoped task. The original longer scoped program was not acquired reliably enough by the small one-pass models to make a clean architectural comparison. The revised task keeps noncommutativity but makes in-support acquisition explicit: RoPE, ALiBi, the selective SSM, Carry-compose, and the LSTM solve the inner task throughout widths 2–16. NoPE remains at the 50% order-blind baseline while solving the outer query. Beyond width 16, every model that mastered the inner task falls sharply toward the same order-blind baseline. The selective SSM and LSTM do not master the long outer dependency, so their outer curves are treated as acquisition failures rather than dilation failures; Carry-compose supplies a positive control in which the outer state is explicitly protected.

Broader support without a required consistency penalty. The same base example is realized at $d_s \in [1, 16]$ and $d_\ell \in [32, 96]$ with an exactly shared ordered intervening stream. Both answers receive cross-entropy supervision. The optional penalty is $\lambda\,\mathrm{SKL}(p_s, p_\ell)$, where

$$\mathrm{SKL}(p_s, p_\ell) = \tfrac{1}{2}\{\mathrm{KL}(p_s \| p_\ell) + \mathrm{KL}(p_\ell \| p_s)\}.$$

The $\lambda = 0$ and 0.25 arms use identical initialization, examples, optimizer, and sequence budget; only λ changes. Gap 120 is withheld. Three seeds and 1,024 fixed examples per condition give:

Model	Short only	Paired CE	Paired CE + KL
Selective SSM	40.07 ± 6.18	99.48 ± 0.90	99.67 ± 0.56
RoPE	65.01 ± 21.78	99.93 ± 0.11	99.90 ± 0.17
ALiBi	53.26 ± 12.64	99.80 ± 0.34	100.00 ± 0.00

Table 6: Training-support control. Accuracy (%) at distance 120 with competing values. Both paired arms reach 100% on random filler. Consistency has no material, consistent added accuracy benefit here; the comparison does not isolate pairing from broader support.

D EVIDENCE MAP AND REPRODUCIBILITY

Figure inputs, generation scripts, checkpoints, and evaluation logs accompany the paper. The data-provenance manifest maps current inputs to their recorded experiments; historical studies remain separate from the new canonical-ALiBi runs.

Evidence in this paper	Source-data record
Simple dilation	`simple_raw.csv`; repeats averaged within seed
Attention intervention	`position_rescue_raw.csv`
Recurrent intervention	`transport_rescue.csv`; localized mask records in replication
Retrieval	`retrieval.csv`; `retrieval_raw.csv`, five models
Scoped comparison	`scoped_simplified_summary.csv`; 18 retained runs and checkpoints
Four-model scaling	`scaling_raw.csv`; not aggregate-only extensions
Training-support ablation	`training_ablation.csv`; matched-support arms

Table 7: A short route from claim to evidence. The packaged `PROVENANCE.json` maps each input to its experiment records and hash. Derived summaries use equal weighting of seed means.

Three levels of checking. Replotting recorded measurements checks figure regeneration. Evaluating saved weights checks checkpoint replay. Training again from initialization checks retraining. The retained audit executed 123 fresh training runs and reevaluated 63 original checkpoints, with explicit records of differences. A further 18 matched-support training runs accompany the later studies. The scoped panel in Figure 5 uses 18 fresh simplified-task runs (six models, three seeds), with retained checkpoints, logs, and fixed-family evaluations packaged with this source and replication record.

All retained original checkpoint evaluations matched their recorded rows in the local audit. Fresh training recovered the core simple, primary retrieval, corrected paired-training, and four-model scaling evaluations under reconstructed historical settings; some original scoped and query-audit retraining outcomes differed. Original and fresh results are not silently pooled, and repeated executions of one seed are not treated as independent samples. These are local execution checks, not a claim of independent third-party replication. This revision adds six canonical-ALiBi simple runs, three retrieval runs, and six matched-support training runs. Position interventions use the first three new simple checkpoints. The retained RoPE retrieval models are reevaluated, not counted as new training seeds. New predictions, training logs, checkpoints, and replay checks are packaged separately from historical custom-slope results. The optimized retrieval forward pass was checked against full-forward logits, predictions, and gradients.

Execution settings matter. The historical scaling evaluations used different batch sizes by scale and architecture. Their exact values are stored in the run manifest; default command-line values did not always reproduce the historical batches. The recurrent-intervention sample counts also differ by seed, as stated in Appendix A. Current aggregation first averages within seed, so unequal numbers of repeated batches do not overweight a training seed. Checkpoints support replay and fresh training, not necessarily resuming the exact historical optimizer and random-number-generator state.

Material not used to support this rewrite. The additional ALiBi scaling and Stress A/B extensions retain aggregates but lack their original seed-level artifacts and implementations. They are preserved in the historical archive but removed from this manuscript's numerical tables. The thinking-token screen changes the compute regime and increases the generated trace at test time; its partial records are not used to infer fixed-context dilation behavior. Earlier scoped screens and intermediate diagnostics remain available as project history rather than occupying the paper's appendix.

Reading the scope correctly. These experiments establish concrete failure modes and interventions in the tested models. They do not demonstrate identical circuits in pretrained language models or unbounded-distance invariance. In language applications, a valid augmentation requires knowing what dependency should survive the expansion and verifying its target. That practical problem remains even when a synthetic task admits exact labels.