A Controlled Benchmark for Lag-Structured Dependency Motifs

Bowen Qi [1]

Abstract

Long-context benchmarks often report pooled scores over heterogeneous tasks, making it difficult to identify which dependency structures a model actually recovers. We propose a controlled benchmark chart for lag-structured dependencies. Each task is specified by a normalized causal kernel and represented by a lossy but interpretable descriptor $\Phi(w) = (s, P, T_\eta, D)$, measuring support density, peakiness, tail mass, and dispersion. We instantiate 1021 tasks across eight anchor, bridge, and stress families, and compare same-order lightweight full attention, sliding-window attention, diagonal SSM, and Mamba-like selective SSM heads. The resulting chart reveals architecture-task structure hidden by pooled reporting: a pooled diagnostic summary nearly ties the two best models (0.659 vs. 0.657), while distinct families have sharply different winners. Local neighborhoods in Φ predict held-out winners with 66.7% accuracy, outperforming family-, region-, and single-model baselines; a targeted three-seed rerun preserves winners on 97.5% of mid/high-gap tasks. Finally, two query-dependent bridge probes, QueriedDecay and AddressedDecay, suggest interpretable preference migration beyond the fixed-kernel face rather than immediate collapse. These results argue for benchmark designs that report structured task neighborhoods rather than only aggregate scores.

1. Introduction

Long-context evaluations often pool together tasks that probe qualitatively different dependency structures. Some tasks require exact retrieval from a single lag; others require broad long-tail aggregation; others involve sparse integration across multiple timescales. When these tasks are summarized by a single pooled score, the resulting ranking can obscure the structure that the benchmark is actually measuring. Realistic long-context suites such as SCROLLS, LongBench, RULER, HELMET, and NoLiMa emphasize heterogeneous capability coverage [1; 2; 3; 13; 14]. Controlled synthetic studies such as Zoology/MQAR, Repeat After Me, and controllable-memory-function analyses isolate narrower recall mechanisms [4; 12; 15]. Our goal is complementary: to build a controlled benchmark whose organizing object is the *task kernel* itself.

This paper studies benchmark design rather than model design. We ask whether a benchmark can expose predictable architecture-task interactions before aggregation hides them. Our answer is a controlled benchmark chart: define tasks by causal kernels, map kernels to a lossy but interpretable descriptor, and test whether local descriptor neighborhoods predict model preference. This makes benchmark construction falsifiable through coverage, region structure, predictive signal, and controlled extrapolation across bridge coordinates.

Our starting point is to write the *task*, rather than the model implementation, as the primary object of analysis. Given an input sequence $x = (x_1, \ldots, x_T)$ and a causal kernel $w = (w_0, \ldots, w_L)$, we consider

$$y_t = \sum_{k=0}^{\min(t-1,L)} w_k x_{t-k}. \tag{1}$$

Under this view, task differences are first differences in kernel shape. We then organize tasks with the descriptor

$$\Phi(w) = (s, P, T_\eta, D), \tag{2}$$

which provides a lossy but interpretable chart over task kernels.

Notation guide. Appendix A provides a compact index of the main symbols, family labels, and boundary/bridge probes used throughout the paper.

We do *not* claim that this chart is a complete or injective parameterization of arbitrary memory tasks. Different task kernels can share the same descriptor coordinates, and tasks with identical Φ values need not exhibit identical winner behavior. Instead, we use Φ as an organizing chart for a controlled benchmark over lag-structured dependency motifs. The benchmark contains four anchor families

[1]Shanghai University, Shanghai, China. Correspondence to: Bowen Qi <bowenqi24@gmail.com>.

Accepted at the ICML 2026 Workshop on Combining Theory and Benchmarks (CTB).

(A1/A2/B1/B2) and four bridge/stress families (C1–C4). Compared with the earlier 174-task anchor-only version, the current 1021-task benchmark gives much denser coverage of the descriptor-chart interior. Finally, rather than treating extrapolation only as a projection problem, we build controlled bridge charts outside the fixed-kernel class and ask whether winner behavior remains interpretable as conditionality and query dependence are introduced.

We do not compare memory in the broad semantic sense. Rather, we compare how sequence architectures recover controlled lag-structured dependency rules from raw scalar inputs under a shared end-to-end interface. We view this benchmark as a fixed-kernel face of a broader operator-requirement view; the present paper validates only this local face and uses boundary probes to motivate, not validate, additional axes such as source conditionality, content addressability, selector timing, selective update, aggregation/compression, compositional depth, and match observability.

We use this benchmark to test four falsifiable claims. **Coverage claim**: moving from 174 to 1021 tasks materially fills the interior of this descriptor chart rather than merely inflating task count. **Region claim**: despite the chart's incompleteness, denser coverage still reveals stable shape- and family-conditioned preference regions that pooled means erase. **Predictive claim**: local neighborhoods in the lossy chart support benchmark-local winner prediction better than global, region-only, or family-only summaries. **Bridge claim**: narrow controlled extensions beyond the fixed-kernel face exhibit interpretable winner migration or boundary flips rather than immediate collapse.

2. Benchmark Formulation

2.1. Task kernels and descriptors

Every task in the benchmark is defined by a nonnegative normalized causal kernel. This kernel is not a complete representation of full task semantics. Rather, it is a *lag-wise relative weighting profile*: it describes how strongly different history distances contribute to the target. In other words, we study a controlled family of tasks defined by lag-wise weighting, not arbitrary query-conditioned, content-conditioned, or nonlinearly compositional tasks.

For any kernel w with $w_k \geq 0$ and $\sum_{k=0}^{L} w_k = 1$, the descriptor $\Phi(w)$ is

$$s(w) = \frac{1}{L+1} \sum_{k=0}^{L} \mathbb{I}[w_k > \epsilon], \tag{3}$$

$$P(w) = \max_{0 \leq k \leq L} w_k, \tag{4}$$

$$T_\eta(w) = \sum_{k > \eta L} w_k, \tag{5}$$

$$D(w) = -\frac{1}{\log(L+1)} \sum_{k=0}^{L} w_k \log(w_k + \epsilon), \tag{6}$$

where ϵ is a small numerical threshold and we use $\eta = 0.5$ throughout. Here $s(w)$ measures support density, $P(w)$ peakiness, $T_\eta(w)$ tail mass, and $D(w)$ normalized dispersion.

The descriptor is intentionally lossy. Different kernels can share the same descriptor, benchmark tasks with identical Φ coordinates need not be identical kernels, and identical Φ coordinates do not force identical winner behavior. Moreover, not every point in $[0, 1]^4$ corresponds to a realizable kernel. Our claims are therefore made at the level of descriptor regions, local neighborhoods, and family-conditioned trajectories, not at the level of exact kernel identity.

2.2. From anchors to bridges

The active benchmark contains eight families. A1/A2/B1/B2 define the endpoint anchors:

- **A1**: fixed-lag point retrieval;
- **A2**: sparse multi-delay retrieval/aggregation;
- **B1**: power-law long-tail aggregation;
- **B2**: gamma-decay long-tail aggregation.

We then add C1–C4 as synthetic bridge and stress families that fill the interior regions between these anchors. These families are not intended as a taxonomy of real-world tasks; they are controlled probes that make the benchmark interior directly observable.

Table 1 summarizes the intended role of each family. This family design is important to the paper's logic: the benchmark is not a grab-bag of synthetic tasks, but a deliberately arranged chart with anchors, bridges, and stressors that expose different parts of the lag-structured space.

2.3. From 174 to 1021

The earlier 174-task benchmark was conceptually clean but sparse in the interior of descriptor space. The current benchmark improves that coverage substantially: the number of mid-band tasks increases from 39/174 to 356/1021, and the number of occupied bins in the (P, D) plane increases from

Figure 1. Benchmark-chart pipeline. The benchmark first parameterizes tasks by causal kernels, maps them into a lag-shape descriptor chart, trains four controlled model families, and then asks where each model wins and whether local chart neighborhoods predict the winner.

Table 1. Task families in the active 1021-task suite. The rightmost column summarizes the design role of each family in the benchmark rather than a claimed universal interpretation.

Family	Canonical motif	Benchmark role
A1	Fixed-lag point retrieval	Null sanity-check anchor
A2	Sparse multi-delay retrieval/aggregation	Sparse anchor region
B1	Power-law long-tail aggregation	Diffuse long-tail anchor
B2	Gamma-decay long-tail aggregation	Diffuse decayed-tail anchor
C1	Sparse-to-diffuse interpolation	Weak bridge band
C2	Spike-plus-decay mixtures	Attention-favoring bridge
C3	Log-spaced multi-delay mixtures	Selective-SSM-favoring bridge
C4	Delayed-hump stress kernels	Delayed-mass stress family

31 to 49. We therefore move to 1021 tasks not to make the benchmark larger for its own sake, but to make its interior directly visible.

3. Related Work

Heterogeneous long-context benchmarks such as SCROLLS, LongBench, RULER, HELMET, and NoLiMa emphasize broad capability coverage across summarization, retrieval, and reasoning settings [1; 2; 3; 13; 14]. Our benchmark is narrower and more controlled: instead of maximizing task diversity, it fixes the scalar regression interface and varies the task kernel itself.

Controlled synthetic studies are closer in spirit. Zoology/MQAR and Repeat After Me isolate retrieval-style stressors in synthetic sequence settings [4; 12], while controllable-memory-function analyses study simplified memory operators directly [15]. Our aim is complementary: to organize a larger controlled suite by a compact lag descriptor, while also marking the boundary where fixed kernels stop being sufficient.

Classic memory-augmented architectures such as Neural Turing Machines, End-to-End Memory Networks, Differentiable Neural Computers, and modern Hopfield networks targeted broader content-addressable or external-memory behavior [5; 6; 7; 8]. We do not treat the present lag benchmark as a replacement for that literature. Instead, the boundary probes below are meant to show which extra task axes a future memory-oriented benchmark would need to cover.

A second neighboring thread concerns benchmark methodology rather than task content. Many long-context evaluations ultimately reduce model behavior to pooled scores over heterogeneous task collections. Our intervention is not to replace those suites, but to show that in a controlled setting, reporting by descriptor neighborhood and family can preserve reversals that pooling suppresses. The bridge-family, QueriedDecay, and AddressedDecay results then extend that methodological point one step further: controlled extrapolation can itself be organized, rather than treated only as an informal out-of-distribution stress test.

4. Scope Beyond Fixed Kernels

The fixed-kernel chart studied here is deliberately local: it occupies the low-source-conditionality, non-addressed, non-compositional face of a broader task space. External results from Zoology/MQAR, Repeat After Me, RULER, and NoLiMa identify axes outside that face, including content addressability, selector timing, aggregation/compression, compositional depth, and match observability [4; 12; 3; 14]. Appendix G records the codebook and completed-result endpoints for this broader operator-axis inventory; we use it to motivate future endpoint-level validation, not to score real tasks in the present paper.

5. Experimental Setup

We compare four sequence models: `diag_ssm`, `mamba_like_ssm`, `full_attention`, and `sliding_window_attention`. All are implemented through the same scalar-sequence regression interface and compared at the same order of parameter count. These lightweight heads are controlled representatives of recurrent, selective state-space, and attention-style sequence modeling rather than faithful reproductions of systems such as S4,

Table 2. Shared main-benchmark protocol. Parameter counts differ only within the lightweight controlled-head design; all models use the same train/validation splits and scalar target interface.

Quantity	Value
Sequence length	$T \in \{128, 256, 512\}$
Train/val seqs	64 / 32 per task
Epochs, batch size	12, 16
LR grid	$\{10^{-3}, 3\times10^{-3}, 10^{-2}\}$
Seeds	main 42; rerun 42/43/44
Metric	MSE for A1/A2, NMSE otherwise
Head dims	diag 512; Mamba 288; attention 12
Params	2.9k–7.5k trainable
Attention	4 heads, learned abs. pos.; sliding window 32

Table 3. Benchmark and evaluation footprint.

Quantity	Value
Active tasks	1021
Families	8
Unique descriptor points	567
Mid-band tasks	356 / 1021
Occupied (P, D) bins	49
Main training runs	12,252
Seed-rerun slice	60 tasks, 3 seeds
QueriedDecay bridge	9 τ points, 3 seeds
AddressedDecay bridge	5 λ points, 3 seeds
MQAR diagnostics	3 scales

Hyena, or Mamba [9; 10; 11]. The `mamba_like_ssm` head follows the selective state-space design of Mamba [11], retaining input-dependent state transitions while using a simplified scalar-regression-compatible head; it is not intended as a faithful reimplementation of published Mamba. We use a best-over-LR protocol: for each $(\text{task}, \text{model})$ pair, we report the lowest validation loss across three learning rates. The canonical experiment source is `p4_extended_1024`, which corresponds to the active 1021 tasks and a total of 12,252 training runs. To quantify seed sensitivity where the region claims matter most, we additionally rerun a 60-task slice over A2/B1/C2/C3/C4 with seeds 42/43/44, stratified by low/mid/high best-vs.-second-best gaps under the original seed-42 run. A1 is treated as a null sanity-check family throughout: it is retained for calibration, but it is not used as positive evidence for preference-region claims. For controlled extrapolation beyond fixed kernels, we run separate QueriedDecay and AddressedDecay bridge sweeps, and we report MQAR diagnostics only as a hard associative-recall corner motivating the latter relaxation.

Our headline statistic is not a pooled average but shape-conditioned structure. Operationally, we call a summary *more faithful* if it preserves winner patterns that are erased by pooling. In the present benchmark, this means that if pooled means nearly tie two models but family-conditioned winner profiles split them sharply, then the pooled summary is too coarse for the object being measured. We also distinguish task-descriptor axes from protocol covariates. In future memory-oriented extensions, axes such as conditionality, query dependence, offset, and observability would describe what the task asks the model to recover; quantities such as depth, width, attention window, sequence length T, sample count N, and learning rate affect solver reachability without changing the task geometry itself.

Table 3 collects the main scale and evaluation quantities in one place. This is useful because the paper combines several empirical layers: the 1021-task benchmark itself, the targeted three-seed stability rerun, the controlled bridge sweeps, and MQAR diagnostics used only to motivate the query-selectivity relaxation. All winner claims compare models within the same task and metric; pooled losses are reported only to show how aggregate summaries can hide family-level reversals.

6. Main Results

6.1. Preference regions remain visible under denser coverage

The move from 174 to 1021 tasks instantiates the benchmark-chart pipeline in Figure 1: the benchmark interior is no longer inferred only from endpoints, but directly sampled. Figure 2 then shows that the winning-model partition remains visibly structured under the denser benchmark rather than dissolving into noise.

The original anchor story survives the upgrade. A1 remains a null sanity-check family and is therefore excluded from positive preference-region evidence. By contrast, A2 remains a clear `diag_ssm` region, preserving the earlier point-memory result. Meanwhile, the central long-tail regions of B1 and B2 remain primarily favorable to `mamba_like_ssm`. The new benchmark therefore does not overturn the old one; it makes the older story less fragile.

6.2. Bridge families reveal interior structure

The bridge families do not merely add clutter. They make previously invisible interior structure measurable. At the family level, the picture is clear:

- **C1** forms a weak but visible bridge band, overall biased toward the attention side;
- **C2** is more clearly attention-favoring;
- **C3** forms the most stable selective-SSM-favoring bridge region;
- **C4** forms a delayed-hump stress family in which attention-side models dominate and recurrence-side models, especially `diag_ssm`, perform substantially worse.

Thus, the bridge families are not decorative additions. They turn the benchmark from an endpoint map into an interior-readable map.

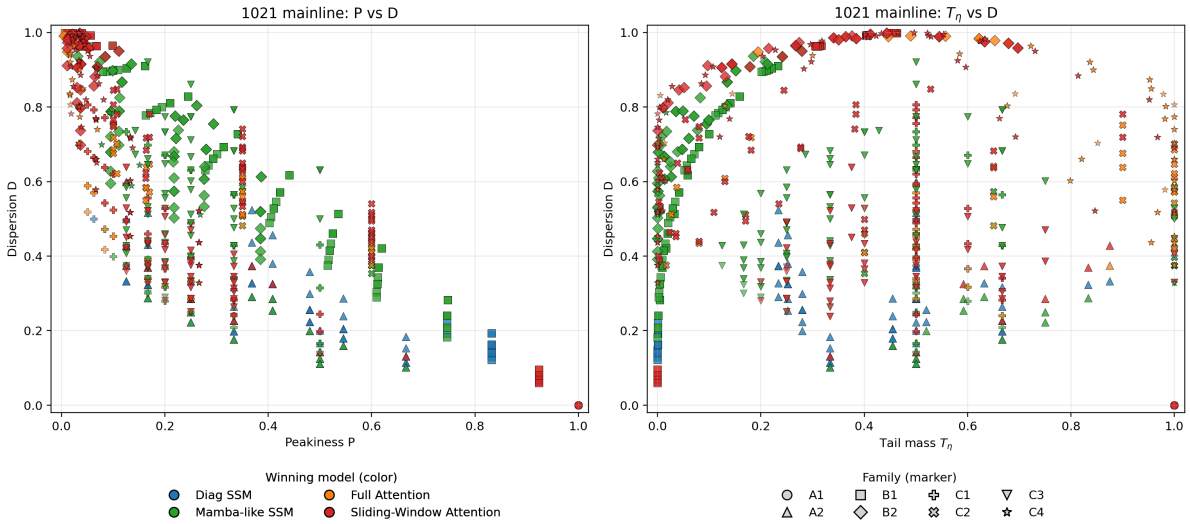


Figure 2. Preference-region map over the active 1021-task benchmark. Denser coverage does not dissolve the winning-model partition into noise; instead, the partition remains visibly structured.

The absolute loss scale in C1/C2/C4 is indeed higher than in the easiest anchor families, so winner counts alone would be insufficient. Best-vs.-second-best gaps separate the bridge families more cleanly: the median gap is 0.011 in C1, 0.021 in C2, 0.062 in C3, and 0.094 in C4. Thus C1 remains the weakest bridge band, while C3/C4 exhibit much stronger interior separation than a pure near-failure story would predict.

6.3. Family-level landscape is more informative than pooled scores

If we average across all tasks, `mamba_like_ssm` and `sliding_window_attention` are nearly tied in mean best validation loss (0.659 vs. 0.657). This pooled value is diagnostic rather than the basis of our winner claims: models are ranked within each task under the same metric, so target-scale variation across tasks does not affect winner identity. The pooled tie is precisely why aggregate reporting is too coarse; the benchmark does not support a simple claim that one architecture is globally superior.

What matters instead is the family-level landscape in Figure 3. A2, B1/B2, C2, C3, and C4 correspond to distinct winner patterns. The two heatmaps jointly show that the models are not competing over a single homogeneous regime, but are favored in different regions of the shape spectrum. For a benchmark of this kind, region-level reporting is more faithful than pooled ranking. More concretely, `diag_ssm` wins 69/90 A2 tasks, `mamba_like_ssm` wins 138/217 C3 tasks, and `sliding_window_attention` wins 105/210 C4 tasks. Any single pooled score suppresses these reversals. The three-seed rerun sharpens the point: winner identity is preserved in 47/60 selected tasks overall and in 39/40 mid/high-gap tasks, with flips concentrated in low-gap near-ties. Thus the main region structure is not a seed artifact, even though the most ambiguous boundary tasks remain fragile.

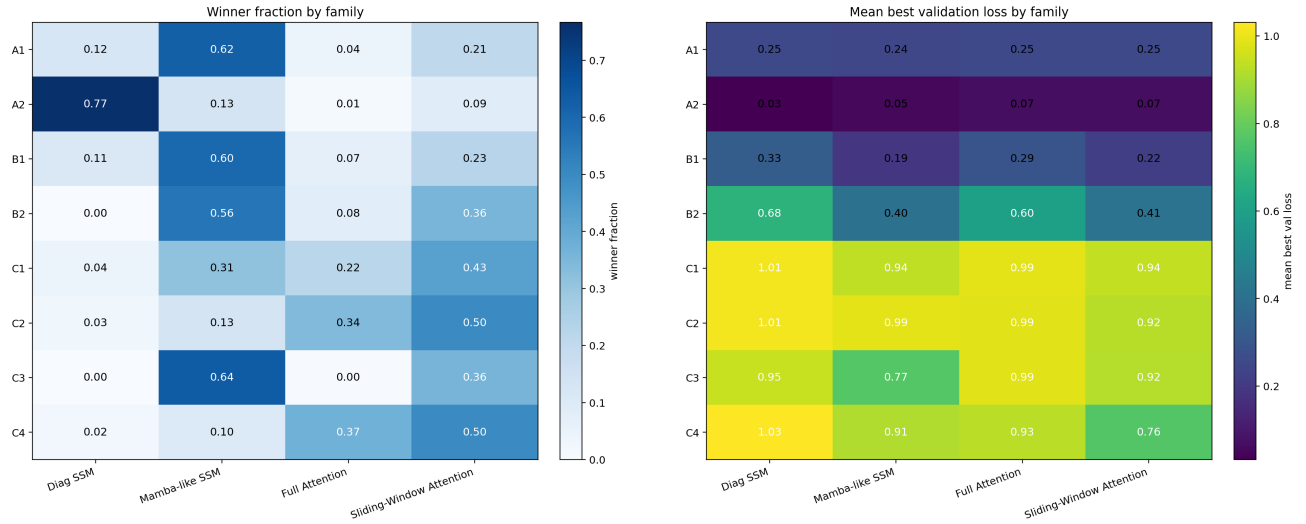
Table 4. Per-family leave-one-shape-out 9-NN accuracy over unique descriptor points. A1 is omitted because it is treated as a null sanity-check family rather than a preference-region family.

Family	Unique shapes	Accuracy
A2	68	69.1%
B1	72	76.4%
B2	72	87.5%
C1	53	49.1%
C2	72	58.3%
C3	102	73.5%
C4	118	55.1%

6.4. Φ supports local winner prediction

The descriptor is not only useful for organizing the benchmark after the fact; it also carries predictive signal. Using the 567 unique descriptor points in the active benchmark, a leave-one-shape-out 9-nearest-neighbor rule in standardized Φ-space predicts the held-out winner model with 66.7% accuracy (binomial 95% CI $\approx [62.8, 70.6]\%$), versus 55.7% for family-majority, 41.1% for region-majority, and 33.7% for a single-model baseline. This remains a benchmark-local result, but it changes the role of Φ: local neighborhoods carry enough information to support winner prediction within the benchmark itself. Appendix E reports cheap sensitivity checks from the same logs: $k \in \{1, 3, 5, 15\}$ gives 63.3–64.6% accuracy, and the $k = 9$ result remains well above a shuffled-label baseline of $30.8 \pm 2.6\%$.

Table 4 shows that this signal is not uniform. Accuracy



(a) Winner fractions by family.

(b) Mean best validation loss by family.

Figure 3. Family-level landscape under the active 1021-task benchmark. The benchmark does not support a single global winner; instead, different models are favored in different shape regions.

is strongest in the cleaner B1/B2/C3 regions, where local winner structure is already visually stable, and weaker in the noisier C1/C4 bridge bands. Even so, the descriptor remains predictive well above the single-model baseline across most non-null families.

6.5. Boundary probes beyond fixed kernels

The benchmark itself stays on the fixed-kernel face. To mark where that face stops, we inspect three already-run D-family probes drawn from a separate variable-lag suite. These probes are not counted as part of the 1021-task benchmark, but they expose directions in which a broader future descriptor would need new coordinates.

Figure 4a shows the cleanest boundary case. In D1 variable-lag retrieval with explicit markers and a late query, `full_attention` reaches 4.7×10^{-7} validation loss and `mamba_like_ssm` reaches 6.1×10^{-4}, whereas `sliding_window_attention`, `diag_ssm`, and a fixed-lag Ridge baseline remain at 0.232, 0.312, and 0.318, respectively. Since no single global lag kernel can solve a task whose source position changes by sample, this result should be read as a scope marker: the fixed-kernel class studied in the main benchmark is a proper subspace of a broader memory-like class.

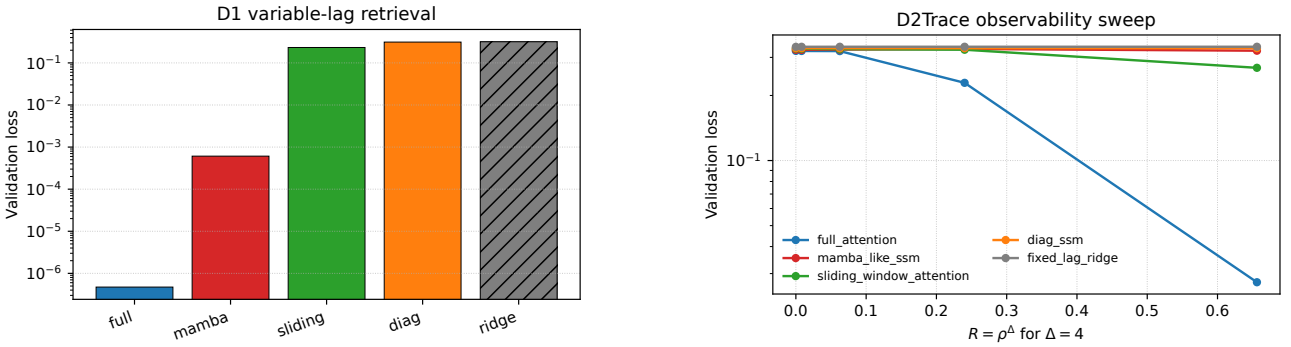
Figure 4b then shows a more graded transition. In D2Trace with offset $\Delta = 4$, raw indirection alone is near-baseline for all models, but increasing trace observability produces a smooth response when plotted against $R = \rho^{\Delta}$. `full_attention` moves from near-baseline at $R = 0$ to 0.027 at $\rho = 0.9$ ($R = 0.6561$), while fixed-lag Ridge remains at 0.336 and the other lightweight models remain much closer to baseline. A separate D4Soft query-key probe shows the same qualitative boundary from another angle: with pairs=4, keys=8, and key_beta=0.5, `full_attention` reaches 0.015 while fixed-lag Ridge remains at 0.421. Together, these probes motivate future descriptor axes for conditionality, candidate pressure, query dependence, selector-value offset, and observability. Just as importantly, they suggest that extrapolation should be tested along controlled bridge families rather than only by disconnected endpoint probes.

6.6. From MQAR corner to continuous bridge charts

The D-family probes above show that fixed kernels stop being sufficient once source position or query dependence becomes sample-conditioned. A natural next endpoint is MQAR-style associative recall [4], where a late query selects among many key-value writes. We do not treat direct MQAR as a positive transfer benchmark for this paper. Instead, Table 5 shows why it is useful as a *hard corner*: full attention solves a small instance, remains weakly above random at a middle scale, and is only barely above random at the preferred full scale under our lightweight from-scratch protocol.

This motivates a different test: rather than jumping directly from fixed kernels to the hard MQAR endpoint, can we define continuous bridge coordinates that expose how architecture preference changes as query dependence is introduced? We report two such charts.

QueriedDecay. The first chart, QueriedDecay(τ), contains



(a) D1 variable-lag retrieval.

(b) D2Trace observability sweep at $\Delta = 4$.

Figure 4. Boundary probes outside the fixed-kernel benchmark. D1 shows a clean failure of global fixed-lag prediction; D2Trace shows that observability, summarized here by $R = \rho^{\Delta}$, creates a smoother transition than raw offset alone.

Table 5. MQAR diagnostics. Direct associative recall is learnable at small scale but becomes a difficulty cliff under the lightweight from-scratch protocol, motivating continuous relaxations such as AddressedDecay.

Setting	Full attn.	Mamba-like	Random
Small (128 vocab, 8 pairs)	0.9998	0.0178	0.0156
Middle (512 vocab, 16 pairs)	0.0293	0.0093	0.0039
Full (8192 vocab, 64 pairs)	0.0016	0.0010	0.00024

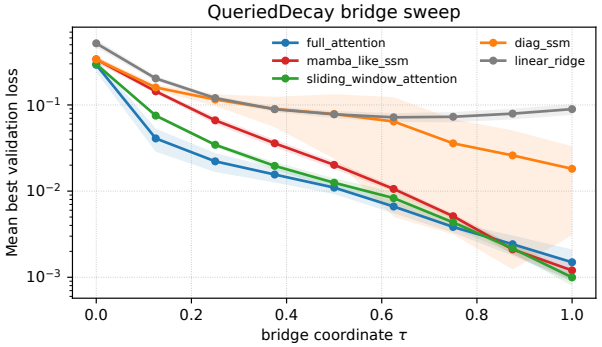


Figure 5. QueriedDecay bridge sweep. Mean best validation loss over three seeds along a continuous family linking latest-key overwrite at $\tau = 0$ to decayed accumulation at $\tau = 1$. Shaded bands denote ± 1 standard deviation across seeds after best-over-LR selection.

keyed write events and a late query while preserving the scalar regression interface. At $\tau = 0$, matching-key writes overwrite a running state, so the target is the latest matching value. At $\tau = 1$, the query becomes irrelevant and the target becomes a query-independent decayed accumulation. Intermediate τ mixes unconditional accumulation into the update while reducing carry from 1.0 to 0.9. Details and a worked example are in Appendix B.

Figure 5 gives the bridge curve, and Table 6 shows the full sweep. `full_attention` wins from $\tau = 0$ through $\tau = 0.75$, `sliding_window_attention` wins at $\tau = 1.0$, and a narrow transition band appears at $\tau = 0.875$, where `mamba_like_ssm`, `sliding_window_attention`, and `full_attention` achieve mean losses 0.00211, 0.00216, and 0.00242. At that same point, each of the three models wins one seed. Equivalently, the mean `full`-vs.-`sliding` gap changes sign between $\tau = 0.75$ (-4.99×10^{-4}) and $\tau = 0.875$ ($+2.56\times10^{-4}$).

AddressedDecay. The second chart is motivated directly by the MQAR corner. Each sample contains scalar write events with keys and a late query key q. For query-selectivity coordinate $\lambda \in [0, 1]$, write t receives weight

$$\mathrm{select}_t(\lambda) = (1 - \lambda) + \lambda\, \mathbb{I}[k_t = q], \tag{7}$$

and the target is a decayed weighted average of written values:

$$y = \frac{\sum_t \rho^{T_q - 1 - t} \mathrm{write}_t\, \mathrm{select}_t(\lambda) v_t}{\max(\sum_t \rho^{T_q - 1 - t} \mathrm{write}_t\, \mathrm{select}_t(\lambda), \epsilon)}. \tag{8}$$

Thus $\lambda = 0$ is a global decayed readout, while $\lambda = 1$ is a scalar-payload keyed retrieval endpoint. In a five-point, three-seed pilot over $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$, `sliding_window_attention` wins every low- and mid-selectivity point, whereas `full_attention` wins the hard keyed endpoint with mean loss 0.0061 versus 0.0167 for sliding attention. Linear Ridge remains much worse throughout, from 0.029 at $\lambda = 0$ to 0.293 at $\lambda = 1$.

Together, MQAR and AddressedDecay clarify the role of continuous bridge charts. Direct MQAR marks a real associative-recall corner, but mostly as a difficulty cliff under this protocol. AddressedDecay turns that corner into a controlled one-dimensional slice where the preference boundary is visible. This does not validate a full memory-task atlas; it shows that more than one independently constructed bridge beyond the fixed-kernel face can produce interpretable preference migration or boundary flips.

Table 6. QueriedDecay full τ-sweep over three seeds. Cells report mean $\pm$ standard deviation of best validation loss after best-over-LR selection.

Bridge coordinate τ	full_attention	mamba_like_ssm	sliding_window_attention	diag_ssm	Ridge
0	0.293 $\pm$ 0.052	0.339 $\pm$ 0.031	0.297 $\pm$ 0.008	0.338 $\pm$ 0.029	0.519 $\pm$ 0.056
0.125	0.041 $\pm$ 0.012	0.144 $\pm$ 0.008	0.075 $\pm$ 0.003	0.160 $\pm$ 0.013	0.203 $\pm$ 0.014
0.25	0.022 $\pm$ 0.005	0.066 $\pm$ 0.006	0.034 $\pm$ 0.002	0.116 $\pm$ 0.018	0.120 $\pm$ 0.007
0.375	0.016 $\pm$ 0.003	0.036 $\pm$ 0.004	0.020 $\pm$ 0.002	0.090 $\pm$ 0.035	0.089 $\pm$ 0.007
0.5	0.011 $\pm$ 0.002	0.020 $\pm$ 0.002	0.013 $\pm$ 0.002	0.079 $\pm$ 0.055	0.078 $\pm$ 0.007
0.625	0.007 $\pm$ 0.001	0.011 $\pm$ 8.66e-4	0.008 $\pm$ 0.002	0.064 $\pm$ 0.059	0.072 $\pm$ 0.009
0.75	0.004 $\pm$ 5.49e-4	0.005 $\pm$ 4.06e-4	0.004 $\pm$ 6.36e-4	0.036 $\pm$ 0.033	0.073 $\pm$ 0.010
0.875	0.0024 $\pm$ 6.39e-4	0.0021 $\pm$ 2.65e-4	0.0022 $\pm$ 2.59e-4	0.026 $\pm$ 0.025	0.079 $\pm$ 0.011
1	0.001 $\pm$ 6.09e-4	0.001 $\pm$ 2.88e-4	9.97e-4 $\pm$ 2.03e-4	0.018 $\pm$ 0.015	0.089 $\pm$ 0.013

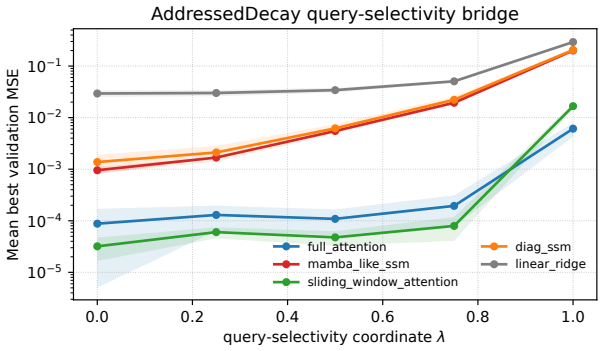


Figure 6. AddressedDecay query-selectivity bridge. Mean best validation MSE over three seeds. The coarse five-point pilot flips from sliding-window attention at low λ to full attention at the hard keyed endpoint $\lambda = 1$.

7. Discussion

The object of this paper is a class of *lag-structured controlled dependency motifs*, not a unified semantics of arbitrary real-world tasks or a broad evaluation of memory in the semantic sense. Our kernel representation emphasizes lag-wise weighting and therefore does not capture query-conditioned, content-conditioned, compositional, or nonlinear interaction tasks. The D1, D2Trace, D4Soft, MQAR, QueriedDecay, and AddressedDecay probes make this boundary concrete: once source position, observability, or query dependence becomes sample-conditioned, fixed-lag summaries cease to be sufficient. The benchmark should therefore be read as measuring end-to-end recovery of fixed causal lag rules from raw scalar inputs under a shared interface.

Within that scope, the methodological message is consistent: denser coverage reveals shape-conditioned preference regions, pooled summaries erase family-level reversals, local neighborhoods retain benchmark-local predictive signal, and narrow controlled bridge charts support this message within limited extensions beyond the fixed-kernel view.

7.1. Limitations

First, $\Phi(w)$ is a lossy chart rather than a complete task space. Different kernels can share the same descriptor coordinates, and tasks that agree in Φ need not agree in winner behavior. The present descriptor should therefore be used to organize a controlled benchmark, not to claim a universal parameterization of memory tasks.

Second, our best-over-LR protocol partially conflates architecture bias with optimization sensitivity. As a sanity check, the raw logs at fixed LR 10^{-3} preserve 94.5% of best-over-LR winners and have mean task-wise rank Spearman $\rho = 0.92$; higher fixed LRs drift more strongly, so this is partial reassurance rather than a replacement for a full optimizer robustness study.

Third, the boundary and bridge results remain intentionally narrow. D1, D2Trace, D4Soft, and MQAR are scope markers; QueriedDecay and AddressedDecay are one-dimensional bridge charts, not validation of a broader memory-task space. Direct scalar real-task projections through autocorrelation or predictive-kernel estimates are treated as negative-control boundaries: without reliable source, query, candidate, or composition metadata, a broader operator map should abstain rather than force a high-confidence coordinate.

8. Conclusion

Pooled scores can erase the architecture-task structure that a controlled benchmark is supposed to expose. Our 1021-task lag-kernel chart shows that family-conditioned reporting and local descriptor neighborhoods recover part of that structure, while narrow bridge charts mark how to extend the approach beyond fixed kernels without claiming broad real-task validity.

References

[1] Uri Shaham et al. SCROLLS: Standardized CompaRison Over Long Language Sequences. In *EMNLP*, 2022.

[2] Yushi Bai et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. *arXiv preprint arXiv:2308.14508*, 2023.

[3] Cheng-Ping Hsieh et al. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv preprint arXiv:2404.06654*, 2024.

[4] Simran Arora et al. Zoology: Measuring and Improving Recall in Efficient Language Models. *arXiv preprint arXiv:2312.04927*, 2023.

[5] Alex Graves, Greg Wayne, Ivo Danihelka. Neural Turing Machines. *arXiv preprint arXiv:1410.5401*, 2014.

[6] Sainbayar Sukhbaatar et al. End-To-End Memory Networks. In *NeurIPS*, 2015.

[7] Alex Graves et al. Hybrid Computing Using a Neural Network with Dynamic External Memory. *Nature*, 538(7626):471–476, 2016.

[8] Hubert Ramsauer et al. Hopfield Networks is All You Need. In *ICLR*, 2021.

[9] Albert Gu, Karan Goel, Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces. In *ICLR*, 2022.

[10] Michael Poli et al. Hyena Hierarchy: Towards Larger Convolutional Language Models. In *ICML*, 2023.

[11] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *COLM*, 2024.

[12] Samy Jelassi et al. Repeat After Me: Transformers Are Better Than State Space Models at Copying. In *ICML*, 2024.

[13] Howard Yen et al. HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly. In *ICLR*, 2025.

[14] Ali Modarressi et al. NoLiMa: Long-Context Evaluation Beyond Literal Matching. In *ICML*, 2025.

[15] Haotian Jiang, Zeyu Bao, Shida Wang, Qianxiao Li. Numerical Investigation of Sequence Modeling Theory using Controllable Memory Functions. *arXiv preprint arXiv:2506.05678*, 2025.

A. Notation Index

This appendix is included as a reader aid; it does not introduce additional claims.

Table 7. Compact notation index for recurring symbols and labels.

Notation	Meaning in this paper
$x = (x_1, \ldots, x_T)$	Raw scalar input sequence.
y_t	Scalar target at time t.
$w = (w_0, \ldots, w_L)$	Nonnegative normalized causal lag kernel defining a fixed-kernel task.
L	Maximum lag/support length of the kernel.
$\Phi(w)$	Lossy task-side lag-shape descriptor, not a learned model feature.
$s(w)$	Support density: fraction of kernel entries above threshold ϵ.
$P(w)$	Peakiness: largest kernel mass.
$T_\eta(w)$	Far-tail mass beyond lag fraction η; we use $\eta = 0.5$.
$D(w)$	Normalized dispersion/entropy of the kernel.
A1/A2/B1/B2	Anchor fixed-kernel families.
C1–C4	Bridge and stress fixed-kernel families that fill the descriptor interior.
D1/D2Trace/D4Soft	Boundary probes outside the fixed-kernel benchmark.
MQAR	Associative-recall hard corner used diagnostically, not as a main transfer result.
QueriedDecay(τ)	Query-dependent bridge from latest-key overwrite ($\tau = 0$) to query-independent decayed accumulation ($\tau = 1$).
AddressedDecay(λ)	Query-selectivity bridge from global decayed readout ($\lambda = 0$) to keyed retrieval endpoint ($\lambda = 1$).

B. QueriedDecay Details

We repeat the formal QueriedDecay definition from the main text here for self-containment, and add a worked example that is easier to inspect line by line than the main-text summary.

QueriedDecay keeps the scalar-sequence interface of the main benchmark while moving outside the fixed-kernel class through sample-dependent key matching. For bridge coordinate $\tau \in [0, 1]$, the generated task uses

$$c(\tau) = 1 - 0.1\tau, \tag{9}$$

$$\beta(\tau) = \tau, \tag{10}$$

where $c(\tau)$ is the carry coefficient and $\beta(\tau)$ the unconditional accumulation weight. Matching writes use coefficient 1, while nonmatching writes use coefficient $\beta(\tau)$. For query key q, keyed write event (k_t, v_t), and latent state h_t, the update rule can be written as

$$h_t = \begin{cases} c(\tau)h_{t-1}, & \text{if no write occurs at } t, \\ c(\tau)h_{t-1} + \beta(\tau)v_t, & \text{if } k_t \neq q, \\ v_t + \beta(\tau)c(\tau)h_{t-1}, & \text{if } k_t = q. \end{cases} \tag{11}$$

Thus $\tau = 0$ recovers latest-key overwrite, while $\tau = 1$ recovers query-independent decayed accumulation with carry 0.9.

Worked example. Consider $T = 16$, $\tau = 0.5$, hence $c = 0.95$ and $\beta = 0.5$, with query key $q = 2$. Suppose the only writes are:

$$(t, k_t, v_t) \in \{(11, 5, 0.6), (12, 2, -0.4), (13, 3, 1.0)\}.$$

Starting from $h_{10} = 0$, the nonmatching write at $t = 11$ gives $h_{11} = 0.95 \cdot 0 + 0.5 \cdot 0.6 = 0.3$. The matching write at $t = 12$ gives $h_{12} = -0.4 + 0.5 \cdot 0.95 \cdot 0.3 = -0.2575$. The final nonmatching write at $t = 13$ gives $h_{13} = 0.95 \cdot (-0.2575) + 0.5 \cdot 1.0 = 0.255375$. The query at $t = 14$ does not change the state, so the answer slot at $t = 15$ receives target value $y_{15} = 0.255375$.

C. MQAR and AddressedDecay Details

MQAR diagnostics. The MQAR diagnostic uses token sequences with key-value pairs followed by query tokens. Accuracy is measured only at query positions. The small diagnostic uses vocabulary 128, 8 pairs, sequence length 64, two layers, and 60 epochs; the middle diagnostic uses vocabulary 512, 16 pairs, sequence length 128, two layers, and 60 epochs; the preferred full diagnostic uses vocabulary 8192, 64 pairs, sequence length 256, and 20 epochs. These runs are reported only as hard-corner diagnostics, not as evidence of broad transfer.

AddressedDecay. AddressedDecay keeps a scalar regression interface while adding keyed write events and a late query key. The reported pilot fixes $T = 128$, key count 8, write probability 0.15, four guaranteed matching writes, $\rho = 0.9$, 2048 training samples, 512 validation samples, 40 epochs, three seeds, two learning rates, and six-way parallel execution. The sweep varies only $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$, making it a coarse bridge chart rather than a full grid over all query-dependent task axes.

D. Boundary Probe Details

D1 variable-lag retrieval. The D1 probe uses the channelized scalar interface from `run_d1_variable_lag.py`: input channels are [value, is_marker, is_query, is_answer], sequence length is $T = 64$, marker position is sampled uniformly from $\{4, \ldots, 60\}$, the query token is placed at $T - 2$, and the answer token at $T - 1$. The target is the scalar value stored at the marked position. The reported run uses hidden size 64, four heads, 100 epochs, 4096 training samples, 512 validation samples, and best-over-LR selection over $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$.

D2Trace. The D2Trace probe uses the shared D-family generator with $T = 64$, fixed selector-value offset $\Delta = 4$, and a trace channel whose amplitude decays as ρ^{step} along the selector-to-value path. The target is the scalar value stored at the offset position. The reported sweep fixes hidden

Table 8. D1 variable-lag retrieval: validation loss across learning rates.

Model	10^{-3}	3×10^{-3}	10^{-2}
`diag_ssm`	0.318	0.312	0.313
`mamba_like_ssm`	8.27e-4	6.09e-4	0.032
`full_attention`	5.44e-7	4.72e-7	8.52e-6
`sliding_window_attention`	0.232	0.233	0.233
Ridge		0.318	

size 64, four heads, one layer, 50 epochs, 2048 training samples, and 512 validation samples, while scanning $\rho \in \{0, 0.1, \ldots, 0.9, 0.95\}$ and selecting best-over-LR across $\{10^{-3}, 3 \times 10^{-3}\}$.

D4Soft. The D4Soft probe uses keyed query-value pairs with $T = 64$, pairs=4, keys=8, and a soft key-contrast parameter `key_beta`. The query is placed at $T-2$, the answer slot at $T-1$, and the target is the value attached to the queried key. The available sweep fixes hidden size 64, four heads, 25 epochs, 1024 training samples, and 256 validation samples, and scans `key_beta` $\in \{0, 0.25, 0.5, 0.75, 0.9\}$. Unlike D1 and D2Trace, this pilot sweep was run only at learning rate 3×10^{-3}; the appendix table reports exactly that available run rather than a nonexistent multi-LR sweep.

E. Robustness Checks

The following checks use existing logs only; no additional model training is introduced. Table 11 probes whether the local prediction result depends on the chosen k or on a single descriptor coordinate. Table 12 probes how much the best-over-LR protocol changes the winner map relative to fixed learning rates.

Table 11. Robustness checks for benchmark-local winner prediction. All rows use the same 567 unique descriptor points as the main 9-NN result.

Check	Setting	Accuracy
k sensitivity	$k = 1$	63.5%
k sensitivity	$k = 3$	64.4%
k sensitivity	$k = 5$	64.6%
k sensitivity	$k = 9$ (main)	66.7%
k sensitivity	$k = 15$	63.3%
Descriptor ablation	remove s	63.0%
Descriptor ablation	remove P	60.1%
Descriptor ablation	remove T_η	55.9%
Descriptor ablation	remove D	64.6%
Shuffled labels	$k = 9$, 200 shuffles	30.8±2.6%

Table 12. Fixed-learning-rate sanity check from the raw 1021-task logs. Pattern columns report the fraction of tasks won by the named model.

LR	Winner match	Rank ρ	A2 diag	C3 Mamba-like / C4 sliding
0.001	94.5%	0.92	83.3%	64.1% / 49.0%
0.003	78.7%	0.65	53.3%	43.3% / 52.9%
0.01	60.2%	0.29	55.6%	13.4% / 66.2%

F. Data Provenance

Table 13 indexes every benchmark-scale quantity quoted in the main text to a concrete CSV/JSON artifact, and Table 14 does the same for the probe and bridge numbers. The more detailed working trace, including paper-location pointers and direct value extraction notes, is kept in `notes/v2-number-trace.md` in the repository.

G. Operator-Axis Evidence Artifacts

The operator-axis bridge in the main text is supported by a coded evidence inventory rather than a new benchmark result. Table 15 gives the compact codebook. The full machine-readable endpoint table, `results/operator_axis_lit_pilot/operator_axis_endpoint_evidence.csv`, contains 25 endpoints with coordinates, source paper, result location, reported models, metric, reported trend, result kind, confidence, and paper-use flag. The companion contrast table, `results/operator_axis_lit_pilot/literature_task_coding_expanded.csv`, contains 40 literature-coded contrast directions. We include these as scope and positioning artifacts only: they motivate the broader operator view but are not used to claim real-task architecture selection.

Table 9. D2Trace sweep at offset $\Delta = 4$. Each cell reports best validation loss for a single learning rate.

Trace ρ	full_attention @ 10^{-3}	full_attention @ 3×10^{-3}	mamba_like_ssm @ 10^{-3}	mamba_like_ssm @ 3×10^{-3}	sliding_window_attention @ 10^{-3}	sliding_window_attention @ 3×10^{-3}	diag_ssm @ 10^{-3}	diag_ssm @ 3×10^{-3}	Ridge
0	0.320	0.321	0.331	0.325	0.330	0.329	0.331	0.331	0.336
0.1	0.320	0.322	0.331	0.325	0.330	0.329	0.331	0.331	0.336
0.2	0.320	0.322	0.331	0.325	0.330	0.329	0.331	0.331	0.336
0.3	0.320	0.321	0.331	0.325	0.330	0.329	0.331	0.331	0.336
0.4	0.320	0.320	0.331	0.325	0.330	0.328	0.331	0.331	0.336
0.5	0.319	0.321	0.331	0.326	0.331	0.327	0.331	0.331	0.336
0.6	0.317	0.287	0.331	0.327	0.331	0.327	0.331	0.331	0.336
0.7	0.313	0.229	0.331	0.328	0.327	0.325	0.331	0.331	0.336
0.8	0.306	0.153	0.331	0.327	0.323	0.300	0.331	0.331	0.336
0.9	0.062	0.027	0.331	0.322	0.306	0.268	0.331	0.331	0.336
0.95	0.141	0.067	0.331	0.320	0.308	0.260	0.331	0.331	0.336

Table 10. D4Soft key-ambiguity sweep for pairs=4 and keys=8. The available run used a single learning rate 3×10^{-3}.

Key β	full_attention	mamba_like_ssm	sliding_window_attention	diag_ssm	Ridge
0	0.001	0.362	0.339	0.370	0.432
0.25	0.002	0.366	0.360	0.372	0.430
0.5	0.015	0.369	0.359	0.367	0.428
0.75	0.304	0.371	0.360	0.368	0.423
0.9	0.305	0.371	0.358	0.368	0.420

Table 13. Provenance index for benchmark-scale and benchmark-result numbers in the main text. Paths are relative to the repository root.

Main-text quantity	Source artifact	Relative path
Mid-band coverage $39/174 \rightarrow 356/1021$	task-point CSVs	main_archive/04_experiments/benchmark/results/task_points.csv; main_archive/04_experiments/benchmark/results/task_points_extended_1021.csv
Occupied (P, D) bins $31 \rightarrow 49$	task-point CSVs	main_archive/04_experiments/benchmark/results/task_points.csv; main_archive/04_experiments/benchmark/results/task_points_extended_1021.csv
Unique descriptor points 567	predictive summary JSON	main/results/predictive_probe/phi_predictive_summary.json
Pooled means 0.659 vs. 0.657	pilot benchmark CSV	main/results/benchmark/p4_extended_1024/pilot_best_over_lr.csv
Winner counts $69/90$, $138/217$, $105/210$	pilot benchmark CSV	main/results/benchmark/p4_extended_1024/pilot_best_over_lr.csv
C-family median gaps $0.011, 0.021, 0.062, 0.094$	pilot benchmark CSV	main/results/benchmark/p4_extended_1024/pilot_best_over_lr.csv
9-NN accuracies and CI inputs	predictive summary JSON	main/results/predictive_probe/phi_predictive_summary.json
Seed stability $47/60$, $39/40$, 78.3%, 97.5%	seed-stability CSV/JSON	main/results/seed_stability/seed_stability_per_task.csv; main/results/seed_stability/seed_stability_summary.json

Table 14. Provenance index for probe and bridge numbers in the main text.

Main-text quantity	Source artifact	Relative path
D1 values 4.7×10^{-7}, 6.1×10^{-4}, $0.232, 0.312, 0.318$	D1 summary/raw CSV	main/results/variable_lag_probe_channels_t64_4k_e100/d1_summary.csv; main/results/variable_lag_probe_channels_t64_4k_e100/d1_raw_results.csv
D2Trace value 0.027 at $R = 0.6561$ and Ridge 0.336	D-family summary/raw CSV	main/results/smooth_d2trace_delta4_t64_2k_e50_best2lr/d_family_summary.csv; main/results/smooth_d2trace_delta4_t64_2k_e50_best2lr/d_family_raw_results.csv
D4Soft values 0.015 and 0.421	D-family summary/raw CSV	main/results/d4soft_e_sweep_light_t64_1k_e25_lr003/d_family_summary.csv; main/results/d4soft_e_sweep_light_t64_1k_e25_lr003/d_family_raw_results.csv
QueriedDecay $\tau = 0.875$ losses $0.00211, 0.00216, 0.00242$	bridge curve summary CSV	main/results/queried_decay_overnight_tau9_seed3_t256_1k_e25/queried_decay_curve_summary.csv
QueriedDecay gap sign change between $\tau = 0.75$ and 0.875	bridge curve summary CSV	main/results/queried_decay_overnight_tau9_seed3_t256_1k_e25/queried_decay_curve_summary.csv
QueriedDecay Ridge 0.519 at $\tau = 0$ and 0.089 at $\tau = 1$	bridge curve summary CSV	main/results/queried_decay_overnight_tau9_seed3_t256_1k_e25/queried_decay_curve_summary.csv
MQAR diagnostic accuracies in Table 5	MQAR summary CSVs	main/runs/mqar_diagnostic_v128_p8_l2_e60/mqar_model_summary.csv; main/runs/mqar_diagnostic_v512_p16_l2_e60/mqar_model_summary.csv; main/runs/mqar/mqar_model_summary.csv
AddressedDecay winner flip and Ridge values	AddressedDecay curve summary CSV	main/runs/addressed_decay_pilot_lambda5_seed42_43_44_t128_2k_e40/addressed_decay_curve_summary.csv

Table 15. Codebook for the broader operator-axis scaffold. Values are coded as absent/partial/dominant when task metadata is available; categorical axes such as selector timing and match observability are treated separately.

Axis	Question encoded	Example high-endpoint
H_{dist}	How far is the relevant evidence?	RULER retrieval, NoLiMa
L_{local}	Is a local window sufficient?	CTB local A2 subset
A_{card}	How many evidence sites matter?	RULER aggregation
A_{comp}	Can evidence compress to a small state?	Fixed decays / word-frequency aggregation
C_{src}	Does source identity vary by sample?	MQAR, variable-lag retrieval
K_{addr}	Is source selection key/content addressed?	MQAR, RULER NIAH
T_{sel}	Is selector pre-, co-, or post-evidence?	Prefix/suffix lookup
S_{upd}	Is input-dependent write/update required?	Selective copying / Mamba probes
D_{comp}	Is nested or multi-hop computation required?	ListOps, RULER variable tracking
P_{cand}	How many plausible candidates/distractors?	Multi-key NIAH, phone-book lookup
O_{match}	Is matching literal, symbolic, or latent?	NoLiMa no-literal retrieval
Γ_{conf}	How reliable is the coding provenance?	Generator metadata vs. scalar proxy

Table 16. Representative completed-result endpoints in the broader operator-axis view. The table is used as a scope bridge: it connects the fixed-kernel face studied in this paper to established task primitives, but it is not used as a validated real-task winner predictor.

Endpoint	Dominant operator axes	Completed result	Interpretation
CTB A2	Fixed sparse lags	`diag_ssm` wins 69/90 tasks	Φ_{lag} refines fixed-lag face
CTB C3	Fixed multiscale mixtures	`mamba_like_ssm` wins 138/217	Interior bridge region, not pooled tie
CTB C4	Delayed fixed mass	`sliding_window_attention` wins 105/210	Nonlocal stress region
AddressedDecay $\lambda = 1$	Keyed late retrieval	`full_attention` wins; gap 0.0106	Hard addressability endpoint
Zoology AR hits	Content addressability, candidate pressure	AR hits explain 82% of attention gap	External support for K_{addr}
Repeat phone-book	Key-value lookup, candidate pressure	Pythia $>$ Mamba across sizes	External retrieval stress result
RULER aggregation	Evidence cardinality, compression	Aggregation is separate from retrieval	External $A_{\mathrm{card}}/A_{\mathrm{comp}}$ endpoint
NoLiMa	Latent match observability	No-literal retrieval degrades strongly	External O_{match} endpoint