# Block-Coherent Delta: A Parallel Pairwise Fast-Weight Operator for Continuous Spatial Data Recall

Dian Yu

**Abstract**

We consider spatial observations encoded so that nearby coordinates produce similar keys, regardless of access order. A query at the current coordinate can then combine information stored at that address with evidence from similar historical keys. Standard Delta memory corrects only one address per event. We introduce Block-Coherent Delta (BC-Delta), a fixed-capacity fast-weight rule that forms a two-key update block from the current and preceding keys, with the latter serving as causal context without implying spatial adjacency. A $2 \times 2$ Gram pseudoinverse separates their shared key component, while squared cosine coherence attenuates residual transport at low similarity. BC-Delta retains Delta's single learning rate and adds no learned gate, window length, or decay parameter. Its rank-two affine update admits associative prefix evaluation. Under equal-budget tuning, both methods select $\eta = 0.025$. Across six route–field conditions and three horizons from 16K to 262K events, BC-Delta lowers blind historical-recall MSE by 4.43% (95% confidence interval 4.11–4.82) and wins 90/90 primary comparisons. Tuned controls, coherence strata, and a signed key–value diagnostic distinguish residual transport from Gram preconditioning and expose the boundary of the spatial-continuity assumption. Code, frozen seeds, and evaluation records support reproducibility.

**Keywords:** associative memory; Delta rule; fast weights; spatial data analysis; similarity-based recall; parallel prefix scan.

# 1 Introduction

Spatial observations often arrive as a sequence even when acquisition order and physical distance disagree. A raster scan, a shuffled patch schedule, or a moving sensor may place nearby coordinates many steps apart. Recurrent and attention-based processors see the order, but the step index does not identify which earlier observations lie near the current coordinate. We therefore use the coordinate as the memory address. Similar positional keys let one read recover evidence stored at the current location and at nearby locations that were visited earlier.

Fast-weight memories compress key–value associations into a matrix whose size does not increase with sequence length.[1–3] Delta writing stores the current residual instead of adding the same association repeatedly.[3;4] A Delta event still has only one write direction. When a causal context key resembles the current key, their directions overlap, but ordinary Delta neither separates them nor uses the amount of overlap. Consecutive coordinates need not be neighbors, so any context correction must also fade when the keys are unrelated.

Simply adding a second Delta write does not solve this problem. Correlated keys share a component, and independent writes along both keys count that component twice. Nor should the current value always be propagated to the context address: a route can jump between rows or distant regions. The update therefore needs separate directions for the two readouts and a transport strength determined by their key geometry.

BC-Delta builds a two-column block from the current and preceding keys. A Gram pseudoinverse gives one write direction for each readout. The current value corrects its own residual as in Delta and also corrects the context-address residual in proportion to squared key coherence. That second residual is measured against the current value, not against the preceding observation. The only tuned scalar is the same learning rate used by Delta; there is no chosen block size, learned gate, or temporal trace.

Although the reference code runs events in order, one BC-Delta step is an affine map of the incoming state. Such maps compose associatively, so a tree scan can recover every prefix state. The operator can therefore supply a history-conditioned read to a recurrent, transformer, or other scanning model without keeping an ever-growing map.

The contributions are fourfold:

- a two-address Delta rule that separates correlated key directions before transporting a residual;
- exact contraction identities, Delta boundary cases, and an associative rank-two affine representation;
- a fully paired evaluation in which Delta, BC-Delta, and every structural control receive the same candidate rates and validation budget; and
- a complete $6 \times 3$ blind test matrix, mechanism ablations, coherence strata, and a signed key–value diagnostic that maps the boundary of the continuity assumption.

# 2 Related Work

## 2.1 Fast weights and residual memory

Fast weights were introduced as rapidly changing parameters for short-term associative storage.[1;2] Linear attention later made the matrix-memory interpretation explicit,[5] and fast-weight programming connected these recurrences to learned key, value, and update streams.[3] Delta-style writing subtracts the answer already stored at the incoming key and therefore avoids unrestricted repeated accumulation. Parallel DeltaNet further shows how Delta recurrences can be evaluated over sequence length without equating recurrence with serial execution.[4] BC-Delta changes the elementary write event rather than the surrounding sequence architecture.

## 2.2 Projection, preconditioning, and spatial address codes

Affine projection algorithms use a short input block to cope with correlated regressors in adaptive filtering.[6] Curvature-aware sequence preconditioning pursues a different objective by improving the conditioning of a learned recurrence.[7] Our $2 \times 2$ Gram operation has a narrower role: it creates dual address directions, after which two residuals receive different geometry-determined strengths. The experiment includes a current-only Gram control so that preconditioning alone cannot be mistaken for residual transport.

Random Fourier features approximate stationary kernels,[8] and sinusoidal coordinate features are widely used to preserve high-frequency spatial information.[9] Explicit spatial memories such as Neural Map and MapNet allocate structured maps for navigation.[10;11] BC-Delta instead retains a lossy, fixed-size associative state. It is designed to recall historical evidence near a query, not to reconstruct a metric map or an exact trajectory.

Recent intelligent data analysis research makes local structure explicit through similarity measures, spatial co-location patterns, and feature-interaction operators.[12–14] Those studies pursue batch or task-level analyses. We instead study an online update: BC-Delta maintains a fixed-capacity associative state as observations arrive and provides a causal recall signal to a downstream system.

# 3 Causal Spatial Recall

Let a stream contain coordinates and observed values

$$\mathcal{S} = \{(\boldsymbol{p}_t, \boldsymbol{v}_t)\}_{t=1}^{T}, \qquad \boldsymbol{p}_t \in \Omega \subset \mathbb{R}^{d_p}, \quad \boldsymbol{v}_t \in \mathbb{R}^{d_v}. \tag{1}$$

A positional encoder gives $\boldsymbol{k}_t = \phi(\boldsymbol{p}_t) \in \mathbb{R}^{d_k}$. The memory state is $\boldsymbol{M}_{t-1} \in \mathbb{R}^{d_k \times d_v}$. Its rows are not map cells; together they form a fixed-size associative surface over the key space. Before the value at step $t$ is revealed, the model reads

$$\widehat{\boldsymbol{v}}_t = \boldsymbol{M}_{t-1}^{\mathsf{T}} \boldsymbol{k}_t. \tag{2}$$

The read-before-write order prevents the current observation from entering its own evaluation.

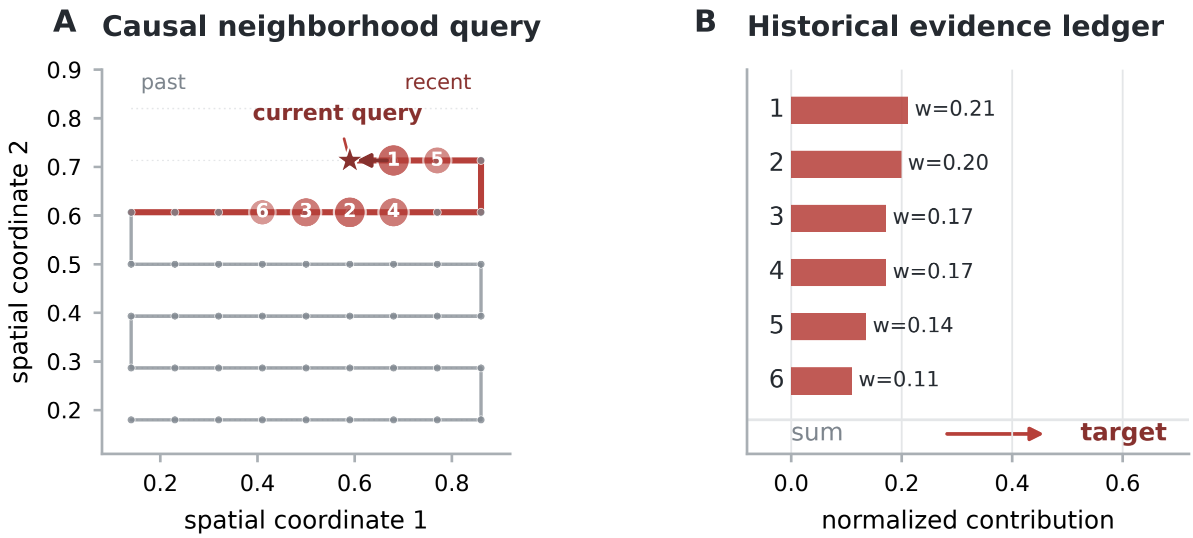


Figure 1: Construction of a historical neighborhood target. (a) One illustrative acquisition order, the current query, and previously observed nearby coordinates; the nearby observations need not be consecutive in that order. Marker area and opacity encode contribution. (b) The six largest normalized contributions are combined into a causal target. The plot shows sampled evidence rather than a continuous kernel surface.

The desired answer is not necessarily the last value written at the exact coordinate. It is a causal local recollection: observations close to the query should contribute more, but no future or current value may contribute. Let $\mathcal{H}_{t^-}$ be the set of coordinates observed before step $t$, and let $\ell_{t^-}(i)$ be the most recent value observed at coordinate $i$. With radial weight $g_\sigma$, the target is

$$\boldsymbol{v}_t^\star = \frac{\sum_{i\in\mathcal{H}_{t^-}} g_\sigma(\boldsymbol{p}_i - \boldsymbol{p}_t)\ell_{t^-}(i)}{\sum_{i\in\mathcal{H}_{t^-}} g_\sigma(\boldsymbol{p}_i - \boldsymbol{p}_t)}, \qquad g_\sigma(\boldsymbol{r}) = \exp\left(-\frac{\|\boldsymbol{r}\|^2}{2\sigma^2}\right). \tag{3}$$

At an evaluation event, $\widehat{\boldsymbol{v}}_t$ is compared with $\boldsymbol{v}_t^\star$. Only after scoring is the newly observed $\boldsymbol{v}_t$ supplied to the update; the neighborhood target $\boldsymbol{v}_t^\star$ is never written into memory. Because $g_\sigma$ is positive and evaluation begins after observations exist, the denominator in Eq. (3) is nonzero. Exact returns and neighborhood-only queries are reported separately. Figure 1 represents Eq. (3) as a finite causal ledger rather than a continuous kernel surface: only observed locations contribute.

# 4 Block-Coherent Delta

## 4.1 Pointwise Delta

For $L$ sinusoidal frequency pairs, a unit-norm positional key can be written as

$$\boldsymbol{k}_t = \frac{1}{\sqrt{L}}[\cos(\boldsymbol{\omega}_1^\top \boldsymbol{p}_t), \sin(\boldsymbol{\omega}_1^\top \boldsymbol{p}_t), \ldots, \cos(\boldsymbol{\omega}_L^\top \boldsymbol{p}_t), \sin(\boldsymbol{\omega}_L^\top \boldsymbol{p}_t)]^\top. \tag{4}$$

The frequency vectors $\boldsymbol{\omega}_j$ are fixed by the benchmark. Their role is to turn spatial proximity into key overlap: nearby coordinates tend to have a large inner product even if they are far apart in the acquisition order. The operator itself only requires an encoder with this property. With learning rate $\eta > 0$, normalized Delta writes

$$\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \eta \frac{\boldsymbol{k}_t}{\|\boldsymbol{k}_t\|^2}(\boldsymbol{v}_t - \boldsymbol{M}_{t-1}^\top \boldsymbol{k}_t)^\top. \tag{5}$$

The parenthesized term is what the memory still gets wrong at the current address. Delta writes that residual along $\boldsymbol{k}_t$, so the current readout moves toward $\boldsymbol{v}_t$ without storing a separate record. This economy is also its limitation here: one write direction cannot independently control two overlapping address readouts.

## 4.2 Minimal address block

The smallest extension that exposes a second causal readout is a pair. Set $\boldsymbol{h}_t = \boldsymbol{k}_{t-1}$ and define

$$\boldsymbol{K}_t = [\boldsymbol{k}_t, \boldsymbol{h}_t], \qquad \boldsymbol{G}_t = \boldsymbol{K}_t^\top \boldsymbol{K}_t, \qquad \boldsymbol{D}_t = \boldsymbol{K}_t \boldsymbol{G}_t^\dagger = [\boldsymbol{d}_{0,t}, \boldsymbol{d}_{1,t}], \tag{6}$$

where $\dagger$ denotes the Moore–Penrose pseudoinverse. The Gram matrix records how much the two keys overlap. Multiplication by its pseudoinverse removes that overlap and converts the original keys into dual write directions. We use the immediately preceding key because it is causally available without a search or learned selector; it is not assumed to be the nearest spatial neighbor. For a rank-two block, the resulting directions satisfy

$$\boldsymbol{k}_t^\top \boldsymbol{d}_{0,t} = 1, \quad \boldsymbol{k}_t^\top \boldsymbol{d}_{1,t} = 0, \quad \boldsymbol{h}_t^\top \boldsymbol{d}_{0,t} = 0, \quad \boldsymbol{h}_t^\top \boldsymbol{d}_{1,t} = 1. \tag{7}$$

These four identities are the reason for the Gram step: $\boldsymbol{d}_{0,t}$ changes the readout at $\boldsymbol{k}_t$ without changing the readout at $\boldsymbol{h}_t$, while $\boldsymbol{d}_{1,t}$ does the converse. The method can therefore assign different corrections to correlated addresses instead of double counting their shared component. The pseudoinverse also defines the degenerate rank-one case. The block always has two columns and introduces no block-size hyperparameter.

### 4.3 Coherence-controlled residual transport

Separating the addresses does not by itself justify changing the context address. That decision should depend on whether the keys say the two addresses are related. We therefore reuse their normalized overlap, rather than introduce a learned gate, and define squared cosine coherence together with two residuals against the current observation:

$$\rho_t = \frac{(\boldsymbol{k}_t^\top \boldsymbol{h}_t)^2}{\|\boldsymbol{k}_t\|^2 \|\boldsymbol{h}_t\|^2} \in [0,1], \tag{8}$$

$$\boldsymbol{e}_{0,t} = \boldsymbol{v}_t - \boldsymbol{M}_{t-1}^\top \boldsymbol{k}_t, \qquad \boldsymbol{e}_{1,t} = \boldsymbol{v}_t - \boldsymbol{M}_{t-1}^\top \boldsymbol{h}_t. \tag{9}$$

Squaring makes the coefficient sign-invariant: parallel and antiparallel keys receive the same coherence. The first residual asks what is missing at the current address; the second asks how incompatible the current observation is with what the memory returns at the context address. It is deliberately not the error of the preceding observation. Section 7 tests the resulting failure boundary explicitly. Combining address separation with this parameter-free transport strength gives

$$\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \eta \left( \boldsymbol{d}_{0,t} \boldsymbol{e}_{0,t}^\top + \rho_t \boldsymbol{d}_{1,t} \boldsymbol{e}_{1,t}^\top \right). \tag{10}$$

The first term preserves the familiar Delta correction at the current address. The second treats the current value as local evidence for the context address, but only to the extent indicated by key coherence. Thus the operator does not copy the preceding observation: it transports the newly observed residual. When the two keys are unrelated, $\rho_t$ suppresses this transport.

**Proposition 1.** *If $\boldsymbol{K}_t$ has rank two, the post-update residuals of the current observation at the two block addresses obey*

$$\boldsymbol{v}_t - \boldsymbol{M}_t^\top \boldsymbol{k}_t = (1-\eta)\boldsymbol{e}_{0,t}, \qquad \boldsymbol{v}_t - \boldsymbol{M}_t^\top \boldsymbol{h}_t = (1-\eta\rho_t)\boldsymbol{e}_{1,t}. \tag{11}$$

*Proof.* Transpose Eq. (10) and evaluate the updated state at $\boldsymbol{k}_t$ or $\boldsymbol{h}_t$. The dual identities in Eq. (7) remove the cross term in each readout. Subtracting the resulting readouts from $\boldsymbol{v}_t$ gives Eq. (11). □

Equation (11) makes the intuition precise. The current-address error contracts exactly as in Delta, whereas the context-address error contracts more slowly when the keys are less coherent. At the first event, ordinary Delta is used because no context key exists. When $\rho_t = 0$, the transported term vanishes. If the two keys coincide, the pseudoinverse divides the repeated direction equally between the two columns; their sum is exactly one Delta write. These are structural reductions, not separately tuned modes. The contraction factors describe the two block readouts only; they do not guarantee global Euclidean stability for an arbitrary ill-conditioned stream.

# 5 Parallel Form and Cost

Expanding the residuals in Eq. (10) shows that each event is linear in the old state plus a data-dependent write:

$$\boldsymbol{M}_t = \boldsymbol{A}_t \boldsymbol{M}_{t-1} + \boldsymbol{B}_t, \tag{12}$$

where $I$ is the $d_k \times d_k$ identity and

$$\boldsymbol{A}_t = I - \eta(\boldsymbol{d}_{0,t}\boldsymbol{k}_t^\top + \rho_t \boldsymbol{d}_{1,t}\boldsymbol{h}_t^\top), \tag{13}$$

$$\boldsymbol{B}_t = \eta(\boldsymbol{d}_{0,t} + \rho_t \boldsymbol{d}_{1,t})\boldsymbol{v}_t^\top. \tag{14}$$

The coefficients depend only on $(\boldsymbol{k}_{t-1}, \boldsymbol{k}_t, \boldsymbol{v}_t)$ and can be prepared in parallel. The pair $(\boldsymbol{A}_t, \boldsymbol{B}_t)$ records the effect of one event on any incoming state. Two adjacent pairs combine as

$$(\boldsymbol{A}_2, \boldsymbol{B}_2) \circ (\boldsymbol{A}_1, \boldsymbol{B}_1) = (\boldsymbol{A}_2\boldsymbol{A}_1, \boldsymbol{A}_2\boldsymbol{B}_1 + \boldsymbol{B}_2). \tag{15}$$

Here the first affine map is applied before the second. Matrix associativity and distributivity make the composition associative, so pairwise, blocked, and tree groupings give the same prefix transformations. A balanced reduction and downsweep therefore has $O(\log T)$ parallel depth. An implementation should keep the rank-two factors instead of materializing one dense $d_k \times d_k$ transition per event.

Retrieval and residual writing cost $O(d_k d_v)$ per sample. The Gram entries and the closed two-dimensional inverse add $O(d_k)$ work. The persistent state is $O(d_k d_v)$ plus one previous key, independent of $T$. BC-Delta approximately doubles the low-rank residual work of Delta. The supplied NumPy implementation is intentionally sequential; the paper establishes algebraic scan compatibility but makes no unmeasured accelerator-speed claim.

# 6 Experimental Protocol

## 6.1 Generated scan family

The benchmark uses a $96 \times 96$ coordinate lattice, six-dimensional values, and 32 sine–cosine frequency pairs ($d_k = 64$). Observation noise has standard deviation 0.015, and the historical target uses $\sigma = 0.045$. Evaluation starts halfway through a stream, occurs every eight events and on forced revisits, and always precedes the current write. The supplementary code specifies the route generators, fields, frequencies, and random seeds exactly.

Three path families are crossed with static and dynamic value fields. *Simple* paths contain persistent short steps over smooth fields. *Mixed* paths combine local runs with relocations and heterogeneous fields. *Complex* paths add multiscale components, piecewise regions, stochastic jumps, local changes, and revisits spanning short to ultra-long lags. All six conditions are evaluated at 16,384, 65,536, and 262,144 events.

Five development seeds are used only for rate selection. Five disjoint seeds (82009, 98299, 114689, 131101, and 147481) form the blind test. For the set $\mathcal{Q}$ of evaluation events, the primary metric is the mean squared Euclidean recall error (abbreviated MSE in the figures and tables)

$$\mathrm{MSE} = \frac{1}{|\mathcal{Q}|}\sum_{t\in\mathcal{Q}}\|\widehat{\boldsymbol{v}}_t - \boldsymbol{v}_t^\star\|^2. \tag{16}$$

Exact returns, neighborhood-only queries, and consecutive-pair coherence strata are also retained.

## 6.2 Equal-budget selection and controls

Every write rule receives the identical global grid

$$\eta \in \{0.0125, 0.025, 0.05, 0.1, 0.2\}. \tag{17}$$

Each candidate is run on the same six development conditions, the same 16K horizon, and the same five seeds. For every method–condition–seed triple, its MSE is divided by the lowest MSE attained over the five candidate rates. The mean of these ratios is the condition-balanced selection score. One global rate is chosen per method; environment-specific rates are not allowed. A second rule averages the normalized Delta and BC-Delta scores at each rate and chooses one shared rate. Both procedures select $\eta = 0.025$ for the two primary methods (Fig. 2), so their blind comparison uses the same learning rate.

Four controls isolate the write structure. *Historical pair* uses Gram duals but assigns the preceding observation, rather than the current observation, to the context address. *Unweighted transport* propagates the current observation with coefficient one. *Transport without Gram* preserves $\rho_t$ but writes along the original normalized keys. *Gram current-only* keeps the first dual direction and removes the transported residual. Each control receives the same grid and validation budget; transport without Gram selects 0.0125 and all other controls select 0.025.

All comparisons are paired by condition, horizon, and seed. Confidence intervals resample the five seed clusters 20,000 times, preserving the repeated conditions within a seed. The primary experiment contains $6 \times 3 \times 5 = 90$ paired units.

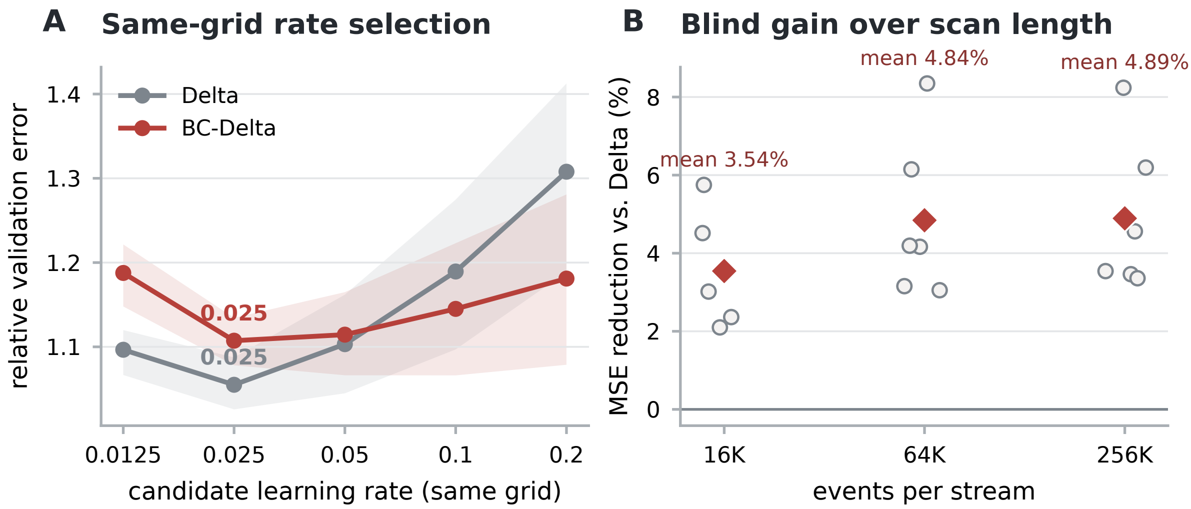


Figure 2: Fair rate selection and blind length scaling. (a) Delta and BC-Delta are evaluated on the same five-rate grid; bands span the 10th–90th percentiles of development-seed scores. Both minima occur at 0.025. Labels are positioned above the curves. (b) Each open point is one route–field condition, diamonds are means, and all six conditions improve at every blind horizon.

### 6.3 Signed similarity diagnostic

The main benchmark makes nearby values statistically related. To test this assumption directly, a two-write diagnostic varies the cosine similarity of the key pair and the cosine similarity of the value pair independently. The context association is first brought near convergence. One current write is then applied, and retention error is measured again at the context address. This diagnostic includes positive, zero, and negative relationships and is not used for parameter selection.

## 7 Results

### 7.1 Primary blind comparison

Across all 90 paired units, BC-Delta reduces error by 4.43% relative to Delta (seed-cluster bootstrap 95% confidence interval 4.11–4.82%) and wins 90/90 comparisons. Table 1 shows that the improvement increases from 3.54% at 16K to 4.84% at 65K and 4.89% at 262K. The absolute mean error stays nearly constant for both methods as the horizon grows, while the structural gap persists.

Figure 3 reports the complete factorial result. There are no untested cells. Gains range from 2.10% in the short simple-static case to 8.35% in the 65K complex-dynamic case. Dynamic fields benefit more than their static partners within every path family at 65K and 262K. The absolute-error panel confirms that percentage gains are not caused by an anomalously small denominator.

| Horizon | Delta | BC-Delta | Paired gain | Wins |
|---|---|---|---|---|
| 16K | 0.3222 | **0.3118** | 3.54% [3.21, 3.93] | 30/30 |
| 65K | 0.3262 | **0.3116** | 4.84% [4.52, 5.25] | 30/30 |
| 262K | 0.3265 | **0.3117** | 4.89% [4.58, 5.30] | 30/30 |

Table 1: Blind historical-recall error pooled over six conditions (lower is better).

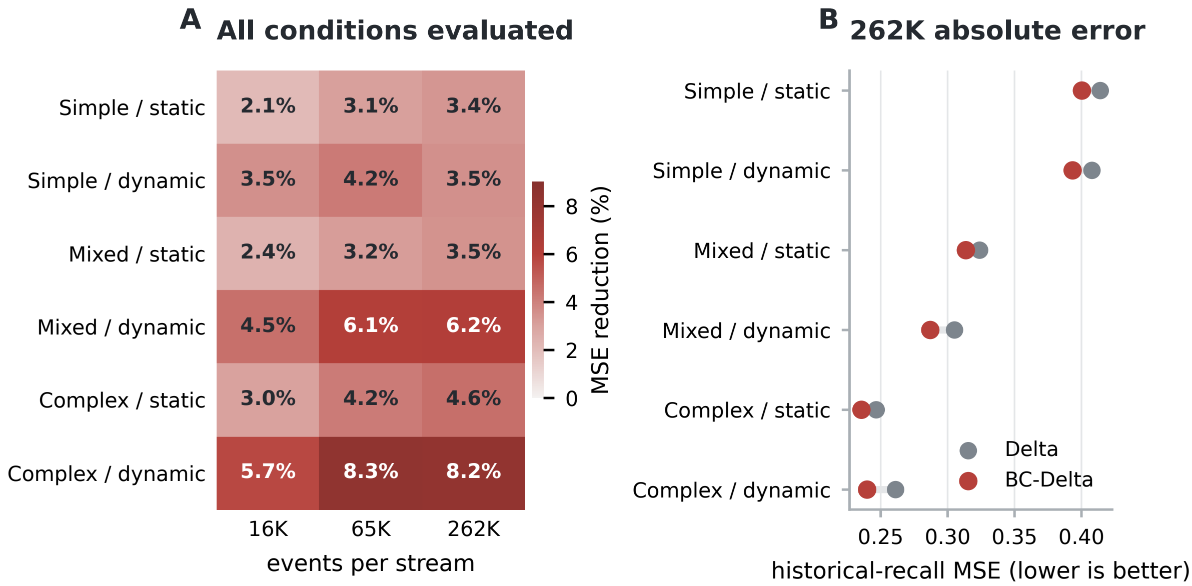


Figure 3: Complete blind test matrix. (a) Five-seed paired MSE reduction in all 18 condition–horizon cells; no cell is missing or imputed. (b) Absolute 262K means for the same six conditions. Connecting segments show the within-condition difference.

Exact-return error falls by 4.04% (95% interval 3.72–4.46; 88/90 wins). Among the 64 units containing neighborhood-only queries, error falls by 6.22% (4.29–7.56; 63/64 wins). The larger neighborhood effect is consistent with the intended fuzzy spatial recall rather than simple overwriting at identical coordinates.

## 7.2 Mechanism tests

Table 2 compares complete BC-Delta with controls whose rates were selected independently under the same budget. Historical-pair projection and unconditional propagation do not reproduce the result. Removing dualization creates a 5.12% disadvantage despite being allowed to select the smaller rate 0.0125. Conversely, the current-only Gram direction is much worse, showing that the second residual, rather than a Gram correction alone, is essential.

Adjacent transitions are partitioned by squared key cosine: low ($< 0.25$), medium (0.25–0.75), and high ($\geq 0.75$). Gains are 3.67%, 3.51%, and 5.47%, respectively. Figure 4 shows all five seed aggregates and confidence intervals. The largest effect at high coherence agrees with Eq. (10); positive low-coherence performance shows that route jumps do not

| Comparator | Advantage | 95% interval | Wins |
|---|---|---|---|
| Delta | 4.43% | [4.11, 4.82] | 90/90 |
| Historical pair | 4.07% | [3.57, 4.63] | 84/90 |
| Unweighted transport | 5.00% | [3.97, 6.25] | 90/90 |
| Transport without Gram | 5.12% | [4.87, 5.43] | 90/90 |
| Gram current-only | 19.88% | [19.02, 20.73] | 90/90 |

Table 2: Paired tuned-control ablations. Positive values favor complete BC-Delta.

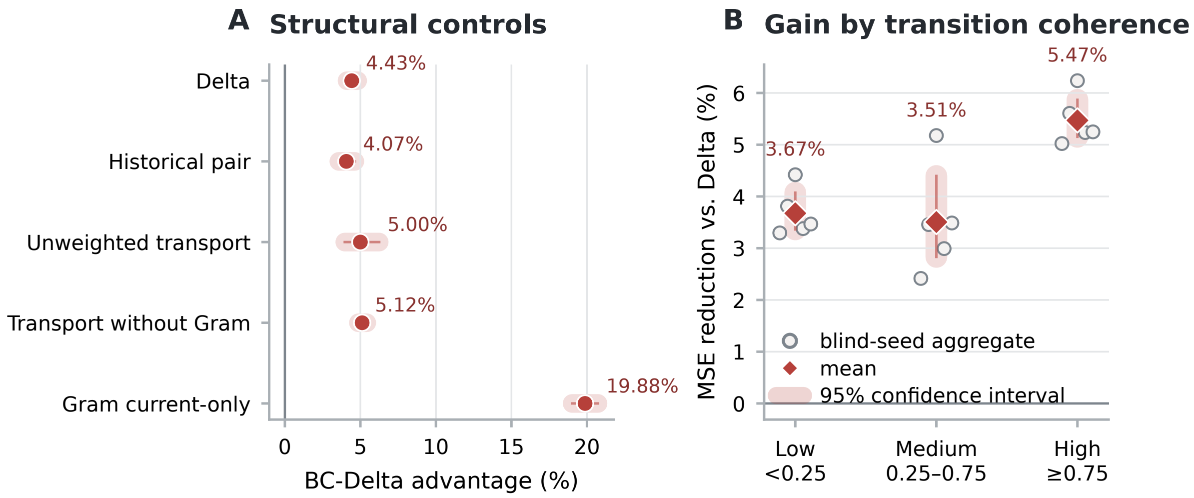


Figure 4: Mechanism evidence. (a) Complete BC-Delta versus equally tuned structural controls. (b) Gains in three coherence strata. Red circles and diamonds denote paired means, open circles denote the five blind-seed aggregates, and translucent red bands denote seed-cluster bootstrap 95% confidence intervals.

destabilize the average.

### 7.3 Where the continuity assumption breaks

Figure 5 varies key similarity and value similarity independently. The gray cells are important because they mark cases in which Delta is better. With positive key cosine and a sharp change in value, BC-Delta usually preserves the previous address more accurately: dualization removes Delta's correlated cross-write, while transport is scaled by $\rho_t$. When both keys and values already point in similar directions, Delta's implicit spill can help slightly in this isolated two-write test. The clearest failure occurs for strongly negative key cosine. Squaring the cosine treats antipodal keys as coherent, so the outcome then depends on how their values are related. This is a specific limitation of the present coefficient.

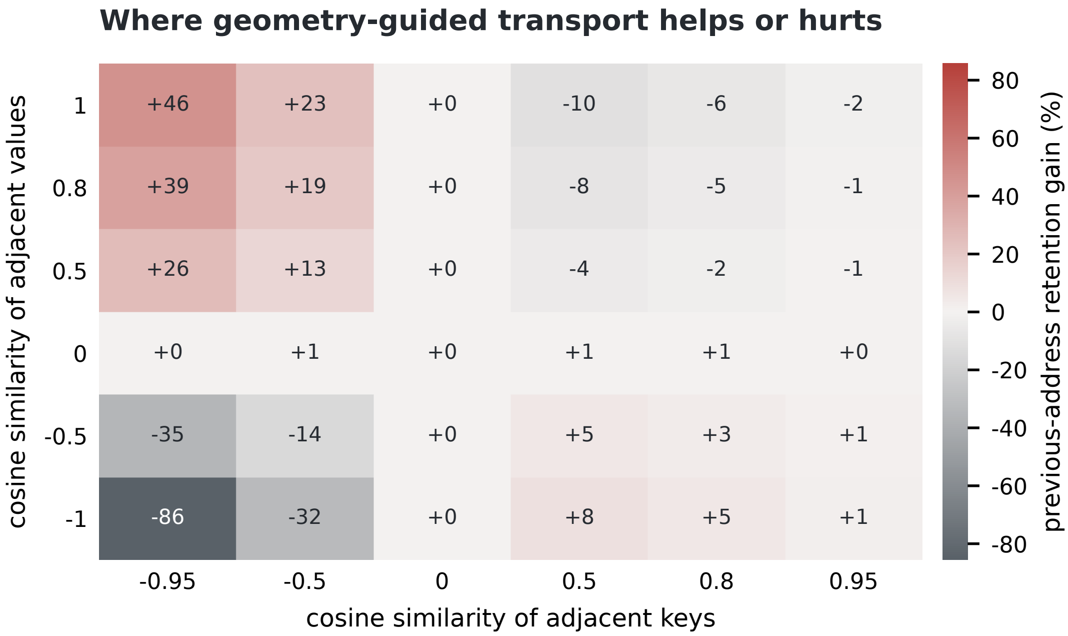


Figure 5: Signed key–value boundary diagnostic. Entries give the percentage change in context-address retention error after one current-key write; red favors BC-Delta and gray favors Delta. The test independently varies key and value cosine, making favorable and adverse regimes visible.

# 8 Discussion

Delta and BC-Delta receive identical search spaces, data, seeds, selection scores, and trial counts. Independent selection and the shared-rate rule both return 0.025, and the blind matrix contains every planned condition at every horizon. The reported difference therefore does not arise from giving the two methods different learning-rate budgets.

The controls separate the two parts of the rule. Gram dualization provides directions that alter the correlated readouts independently; coherence sets the size of the context correction. Removing either part loses the full gain. Both quantities come from the two keys already present at the update.

A practical use case is spatially keyed acquisition in either regular or irregular order: line or raster data, patch sequences, slice collections, or moving-sensor observations. A classifier or controller could combine its current feature with $\boldsymbol{M}_{t-1}^{\top}\phi(\boldsymbol{p}_t)$ as a historical cue. The read may retrieve evidence from similar stored keys even when neighboring coordinates were acquired far apart in the sequence. The preceding key is used only to form the minimal causal update pair, and coherence reduces its influence after an access jump. Our experiments test this memory signal in isolation; they do not establish end-to-end gains on images, robot logs, or point clouds. Tasks requiring exact storage, obstacle-aware topology, or an auditable trajectory still call for an explicit map or database.

The signed diagnostic identifies a further boundary. Coordinate similarity must be a reasonable proxy for value continuity. A wall, semantic edge, or feature collision can violate that relation. Content- or topology-aware keys may address this issue, but evaluating them would introduce a learned front end and move beyond the operator-only question studied here.

Equation (15) establishes scan compatibility, not a wall-clock speedup. That claim would require a fused blocked implementation. Learned key distributions and end-to-end scanning tasks also remain to be tested.

# 9 Conclusion

BC-Delta changes one Delta event from a point write to a two-key block. The Gram pseudoinverse separates the two readouts, and key coherence sets how much of the current residual reaches the causal context address. Successive accesses need not be spatial neighbors: similar keys receive transport, whereas unrelated pairs receive little. The rule remains fixed-capacity, uses no learned gate, selected window, or extra decay, and has an associative rank-two affine form. Delta and BC-Delta select the same learning rate under equal-budget tuning, and BC-Delta lowers historical-recall error in all 90 blind primary comparisons. The ablations and signed boundary test narrow the claim: the gain appears when key geometry is informative about local values, not for arbitrary key–value streams.

# Statements and Declarations

**Ethical considerations.** Not applicable. This study uses only synthetic data generated by deterministic computer programs and involves no human participants, human data, animals, or biological material.

**Consent to participate.** Not applicable.

**Consent for publication.** Not applicable.

**Declaration of conflicting interest.** The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding.** The author received no financial support for the research, authorship, and/or publication of this article.

**Data availability.** All observations are synthetic. The deterministic generators, frozen seeds, and archived result files are included in the supplementary reproducibility package.

**Code availability.** Source code, rate-selection records, per-seed blind results, analysis scripts, and figure scripts are included in the supplementary reproducibility package and will be deposited in a public repository upon publication.

# References


[1] Schmidhuber J. Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. Neural Comput. 1992;4(1):131-9.

[2] Ba J, Hinton GE, Mnih V, Leibo JZ, Ionescu C. Using Fast Weights to Attend to the Recent Past. In: Advances in Neural Information Processing Systems. vol. 29; 2016. .

[3] Schlag I, Irie K, Schmidhuber J. Linear Transformers Are Secretly Fast Weight Programmers. In: Proceedings of the 38th International Conference on Machine Learning; 2021. p. 9355-66.

[4] Yang S, Wang B, Zhang Y, Shen Y, Kim Y. Parallelizing Linear Transformers with the Delta Rule over Sequence Length. In: Advances in Neural Information Processing Systems. vol. 37; 2024. .

[5] Katharopoulos A, Vyas A, Pappas N, Fleuret F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In: Proceedings of the 37th International Conference on Machine Learning; 2020. p. 5156-65.

[6] Haykin S. Adaptive Filter Theory. 4th ed. Prentice Hall; 2002.

[7] Tumma N, Loo N, Rus D. Preconditioned DeltaNet: Curvature-Aware Sequence Modeling for Linear Recurrences; 2026. ArXiv:2604.21100 [cs.LG].

[8] Rahimi A, Recht B. Random Features for Large-Scale Kernel Machines. In: Advances in Neural Information Processing Systems. vol. 20; 2007. .

[9] Tancik M, Srinivasan PP, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, et al. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In: Advances in Neural Information Processing Systems. vol. 33; 2020. p. 7537-47.

[10] Parisotto E, Salakhutdinov R. Neural Map: Structured Memory for Deep Reinforcement Learning. In: International Conference on Learning Representations; 2018. .

[11] Henriques JF, Vedaldi A. MapNet: An Allocentric Spatial Memory for Mapping Environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 8476-84.

[12] Bao X, Huang Y, Yuan S, Chang L. Discovering worthy spatial co-location patterns based on pattern distributions through clustering. Intell Data Anal. 2026;30(2):317-38.

[13] Wu B, Pang A, Yang W, Zhao Q. A similarity measurement method with sliding window approach based on transformer for multivariate time-series. Intell Data Anal. 2026;30(2):354-72.

[14] Zhou YQ, Yang W, Ma ZM, Xu Z. Weighted Bonferroni mean-based classifiers with factorization machines for binary classification. Intell Data Anal. 2025;29(5):1085-105.