Memory by Design: Probabilistic Sequence Layers

Paper discussion: covariance as a memory-writing rule

Paper: Matthew Dowling, Hyungju Jeon,
Cristina Savin, Il Memming Park

arXiv:2605.31163v1 (29 May 2026)
Reading: 22 September 2026

The question is where a new write should go

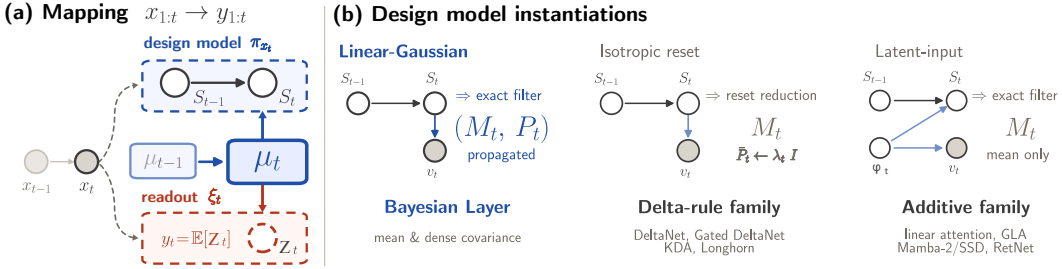
- Fixed-size recurrent memory can confuse nearby addresses.
- A smaller scalar gain slows both overwriting and new learning.
- Propagated covariance can redirect the write away from resolved directions.

Central claim

Store the association and the uncertainty geometry that should govern future writes.

Sec. 1–2, PDF pp. 1–5.

A design model separates writing from reading



- Inputs parameterize an auxiliary model; its filter updates memory.
- A query-dependent readout produces the deterministic layer output.

Fig. 1, PDF p. 2. Original author figure.

Bayesian here describes the layer recurrence

- Inputs x_t are controls, not samples generated by the design model.
- Memory belief: $\mathrm{vec}(S_t) \sim \mathcal{N}(\mathrm{vec}(M_t), I_m \otimes P_t)$.
- $M_t \in \mathbb{R}^{D \times m}$ stores content; $P_t \in \mathbb{R}^{D \times D}$ stores directional uncertainty.
- Task training learns the maps that produce keys, values and dynamics.

Sec. 2–2.1, PDF pp. 2–4. This is not a Bayesian posterior over all network weights.

Linear–Gaussian assumptions determine the write

$$S_t \mid S_{t-1} \sim \mathcal{N}(A_t S_{t-1}, \ell_t^2 I), \qquad v_t \mid S_t \sim \mathcal{N}(S_t^\top k_t, r_t^2 I).$$

- A_t: memory dynamics; $k_t \in \mathbb{R}^D$: write key.
- $v_t \in \mathbb{R}^m$: pseudo-observed value; r_t^2: distrust of the write.
- ℓ_t^2: process noise replenishes uncertainty.
- Matrix-Gaussian shorthand: columns share a key-space covariance.

Definition 1, Sec. 2.1, PDF pp. 3–4.

Prediction creates an uncertainty-weighted gain

$$\bar{M}_t = A_t M_{t-1}, \qquad \bar{P}_t = A_t P_{t-1} A_t^\top + \ell_t^2 I,$$
$$u_t = \bar{P}_t k_t, \qquad \beta_t = (r_t^2 + k_t^\top u_t)^{-1},$$
$$K_t = \beta_t u_t.$$

- u_t changes the direction, not just the magnitude, of a write.
- β_t is innovation precision; K_t is the vector Kalman gain.
- Initialize $M_0 = 0$ and $P_0 = p_0 I$.

Proposition 1, Eqs. (9)–(11), PDF p. 4; Appendix B, pp. 13–14.

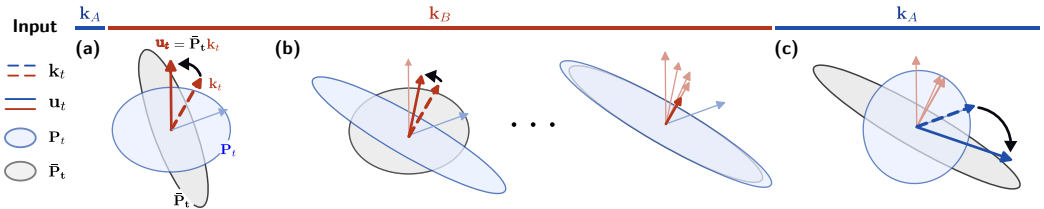
The mean corrects errors; covariance records evidence

$$
\begin{aligned}
e_t &= v_t - \bar{M}_t^\top k_t, \\
M_t &= \bar{M}_t + K_t e_t^\top, \\
P_t &= \bar{P}_t - \beta_t u_t u_t^\top, \qquad y_t = M_t^\top q_t.
\end{aligned}
$$

- Conditioning reduces uncertainty along the observed direction.
- The query reads the mean and does not enter the write update.
- Output uncertainty is not used by the evaluated downstream losses.

Eqs. (8), (12)–(13), PDF p. 4; Appendix B; limitations, p. 10.

Covariance rotates writes away from confident memories



$$k_A = \rho k_B + \sqrt{1-\rho^2}\, k_\perp, \quad u_A \approx \rho \bar{\sigma}_B^2 k_B + \sqrt{1-\rho^2}\, \bar{\sigma}_\perp^2 k_\perp.$$

Fig. 2 and Eq. (15), PDF p. 5. Original author figure.

Two existing families require two different explanations

Family	Mechanism
Bayesian Layer	Propagate covariance; correct along $\bar{P}_t k_t$.
Delta-rule	Reset $\bar{P}_t = \lambda_t I$; correct along raw k_t.
Additive	Observe a transient write coefficient in a different design model.

- DeltaNet / GDN / KDA differ in the transition structure.
- Linear Attention / RetNet / diagonal GLA / SSD arise from additive means.

Table 1, PDF p. 6. These are core-recurrence correspondences, not full-network equivalences.

Covariance reset recovers the Delta-rule form

$$\eta_t = \frac{\lambda_t}{r_t^2 + \lambda_t \|k_t\|^2}, \qquad M_t = \bar{M}_t + \eta_t k_t (v_t - \bar{M}_t^\top k_t)^\top.$$

- $A_t = I$: DeltaNet; $A_t = \alpha_t I$: Gated DeltaNet.
- $A_t = \mathrm{diag}(\alpha_t)$: KDA core.
- The reset preserves error correction but removes covariance history.

Proposition 2 and Eq. (17), PDF p. 6.

Three evaluations form an evidence ladder

1. Controlled collisions: test the predicted mechanism with fixed addresses, then train and extrapolate.
2. MQAR: learn associations from discrete tokens; stress updates and delayed values.
3. A 340M pretrained model: replace four layers, distill, and measure retrieval plus language-modeling cost.

Sec. 4, PDF pp. 7–9; full protocols in Appendices E–G.

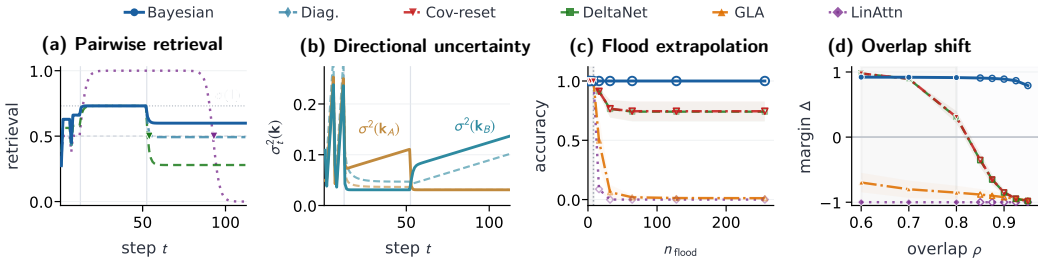
Boost B, flood A, then query B

- Two unit keys overlap by $\rho = 0.92$; values are one-hot.
- $D = 16$, $m = 6$, $P_0 = 3I$, $r^2 = \ell^2 = 0.05$, $A = I$.
- 12 seed writes, 40 B-boost writes, 60 A-flood writes.
- Compare full covariance with reset, raw-key updates and a diagonal ablation.

$$p_B^{AB} = \frac{\exp(y_B)}{\exp(y_A) + \exp(y_B)}$$ (pairwise probability, not classification accuracy).

Sec. 4.1, PDF p. 7; Appendix E.3–E.4, pp. 18–20.

The controlled curves support directional protection



- Left: retrieval and directional uncertainty. Right: learned extrapolation.
- Panel (c) is labeled accuracy; panel (d) is labeled margin.

Fig. 3, PDF p. 7. Original figure; v1 metric definitions are not fully consistent.

Exact limits show protection with a small residual loss

Pairwise B probability	Bayesian	Reset / Delta
Infinite B-boost limit	0.73106	0.73106
A-flood limit from actual pre-flood state	0.72309	0.35138

- $0.73106 = \text{sigmoid}(1)$ comes from the one-hot logit gap.
- Flood gain approaches 0.618, not zero; the innovation vanishes.
- At $\rho = .92$, cross-direction variance grows by 0.00768 per step asymptotically.

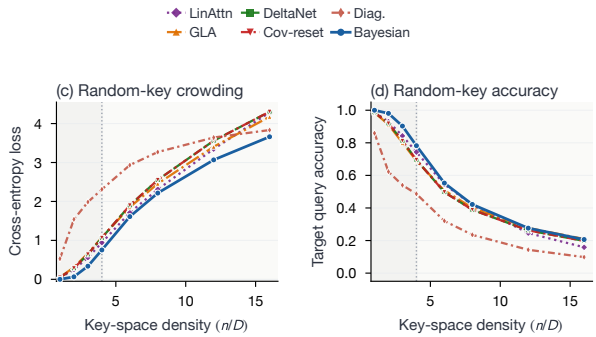
Appendix E.1–E.3, PDF pp. 16–19. Conflicting tables on p. 20 are not used here.

Learned tests stress length and address overlap

- Train: $n_f \leq 8$, $\rho \in [.60, .80]$; 8 pairs in 16 dimensions.
- Test in Fig. 3: $n_f = 256$ ($32\times$) and overlap up to .95.
- Addresses bypass learned projections; values are re-assigned each sequence.
- 2 layers, width 64; 2,500 steps; 3 seeds with one-standard-deviation bars.
- BL retains its advantage; reset and Delta-style fail at high overlap.

Fig. 3, PDF pp. 7–8; Appendix E.5, pp. 21–24. Appendix lists only 16/32/64 for flood test settings.

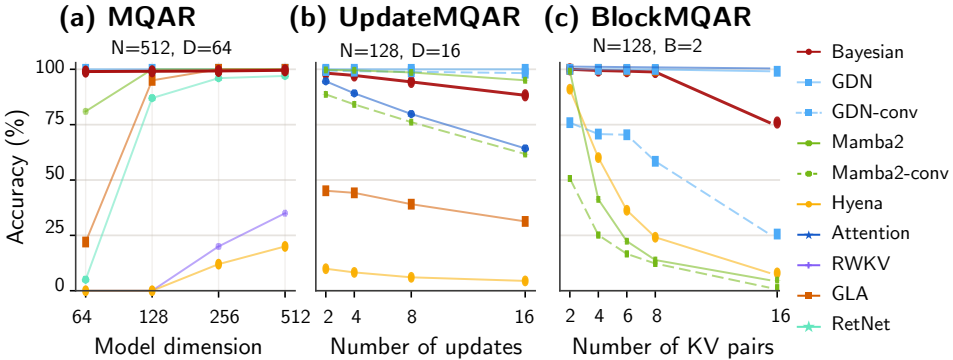
Random-key crowding still degrades every finite memory



- Fixed projected addresses, $D = 16$; test occupancy $n/D = 6, 8, 12, 16$.
- Lower query CE favors BL; this does not remove the linear interpolation limit.

Fig. 7, PDF p. 24; Appendix E.5.4, p. 23. Original author figure.

MQAR tests learned binding, updating and delayed values



- Base: competitive near-perfect recall at larger widths.
- Update: approximately 88% BL vs. 60% GDN/Mamba-2 in the caption.
- Block: both convolution-free BL and GDN bridge delayed values.

Fig. 4, PDF p. 8. Approximate percentages, not an average across all configurations.

Four Bayesian layers are distilled into a 340M GDN

- Replace layers $\{5, 11, 17, 23\}$; training context is 2,048.
- Phase A: about 100M tokens per layer, parallel hidden-state matching.
- Phase B: about 300M tokens, teacher-logit KL plus boundary MSE.
- GDN-FT control: same reported 400M-token budget and trainable mask, but next-token CE.
- One training seed; token matching is not a reported end-to-end FLOP benchmark.

Sec. 4.3, PDF pp. 8–9; Appendix G, pp. 25–27.

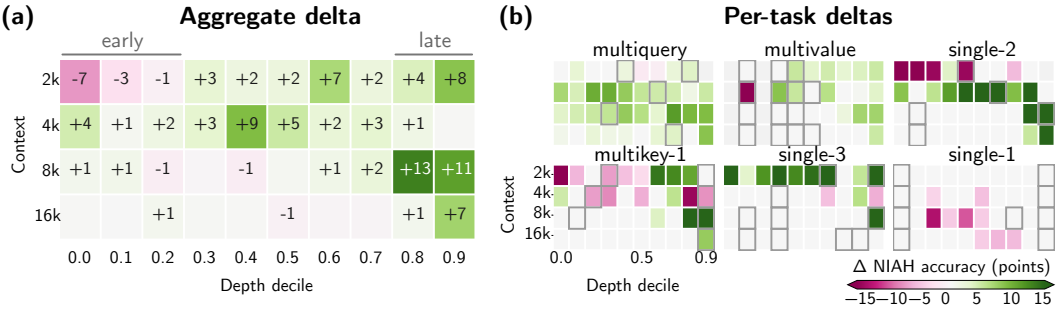
Mean retrieval improves, but 16k performance remains low

Context	GDN-zero	GDN-FT	BL	BL−zero
2k	49.52	48.46	50.81	+1.29
4k	31.08	31.58	34.02	+2.94
8k	16.81	16.06	20.00	+3.19
16k	7.85	8.36	8.81	+0.96

- Mean across six RULER NIAH tasks; CWE excluded.
- Scores are percentages; differences are percentage points.
- At 2k, single-2 drops from 96.50 to 89.00 (7.50 points).

Table 4 (Appendix Table A1), PDF p. 26.

Position-dependent gains are based on sparse cells



- Green favors BL; depth increases from early to late context.
- 208 of 240 task/length/depth cells contain fewer than 30 samples.

Fig. 5, PDF p. 9; sample counts, Appendix G, p. 27. Avoid treating peaks as precise population effects.

Retrieval gains have a language-modeling cost

8k context	GDN-zero PPL	BL PPL	Relative increase
WikiText-103	14.57	14.97	2.75%
SlimPajama-heldout	12.15	12.43	2.30%
PG-19 validation	19.29	19.81	2.70%

- PPL is lower-is-better; the gap is roughly flat across context lengths.
- Table-derived range is about 2.3–2.75%, broader than the prose summary.
- This is a retrieval/quality trade-off, not a universal improvement.

Table 5 (Appendix Table A2), PDF p. 27. Percentages recalculated from printed PPL values.

The strongest result is the memory-writing mechanism

- Covariance history changes where errors are written.
- Resetting that history recovers familiar Delta-rule dynamics.
- Controlled tests are the clearest evidence; pretrained-model gains are modest and task-specific.
- Open questions: state cost, throughput, larger scales, multi-seed replication and uncertainty-aware readout.

Sec. 5, PDF pp. 9–10; interpretation of the evidence in this reading.

Reference and scope of this discussion

Memory by Design: Probabilistic Sequence Layers

Matthew Dowling, Hyungju Jeon, Cristina Savin, Il Memming Park.

arXiv:2605.31163v1, submitted 29 May 2026.

Based on the version-matched 27-page PDF, author TeX and original figures.

Main text and Appendices A–G were read. No model retraining was performed.

Companion Chinese notes contain full protocols, derivations and v1 consistency checks.

Prepared for paper discussion on 22 September 2026. Later arXiv versions are outside scope.

Backup: a scalar Riccati map gives the gain floor

$$x_{t+1} = \frac{r^2 x_t}{r^2 + x_t} + \ell^2, \qquad x_\infty = \frac{\ell^2 + \sqrt{\ell^4 + 4r^2\ell^2}}{2},$$

$$g_\infty = \frac{x_\infty}{r^2 + x_\infty}, \qquad \Delta_B \longrightarrow (1 - \rho^2)\ell^2.$$

- Assumptions: $A = I$, a repeated unit key, constant positive noise.
- Cross-covariance with the orthogonal subspace contracts to zero.
- A positive gain floor is compatible with a vanishing innovation.

Appendix E.1–E.2, PDF pp. 15–17; Eq. (42).

Backup: efficient filtering still carries a covariance cost

- Base state and per-token work: $O(Dm + D^2)$ with diagonal dynamics.
- Grouped noise: G covariance blocks, state $O(Dm + GD^2)$.
- Compute gain trajectories, then apply affine mean-update machinery.
- Chunks inherit boundary states sequentially; covariance arithmetic uses fp32.
- The implementation also describes damped rotations and real-row noise injection.

Appendix D, PDF pp. 14–15. Reported implementation details are not a measured throughput comparison.

Backup: important v1 inconsistencies remain unresolved

- Gain table says about 2.0, but $g = x/(r^2 + x) < 1$ (p. 20 vs. p. 17).
- Flood margins +0.20 / -0.44 conflict with the exact-limit section and overlap table.
- Pairwise vs. full-vocabulary softmax margins differ; Fig. 3(c) says accuracy.
- The additive mean closes without covariance, but latent write noise still implies uncertain memory.

PDF pp. 6, 8, 17–20, 24. Reader checks; these are not claimed author corrections.