CQFA: Channel-Wise Query Feedback for Gated DeltaNet-2

Anonymous authors
Paper under double-blind review

Abstract

Linear attention offers an efficient alternative to softmax attention by enabling sequence modeling with linear-time computation and constant-size recurrent states. Within this framework, compressing sequence history into fixed-capacity memory makes selective state correction essential for retaining useful context while continuously incorporating new information. GDN-2 supports flexible memory updates by decoupling erasure and writing, but its correction remains key-driven and does not explicitly incorporate the query used for memory readout. Q-Delta incorporates query feedback into state correction, yet its per-head scalar coefficient applies the same feedback strength across channels, limiting channel-specific control over state correction. We therefore propose CQFA, which augments GDN-2 with input-dependent channel gates to selectively modulate query-based memory predictions on the erase side, independently of the writing of new values. This design allows feedback to be strengthened in some channels and suppressed in others while preserving the state size and rank-one update structure. We establish a sufficient condition for one-step residual contraction and implement efficient chunkwise parallel training with throughput comparable to that of GDN-2. Systematic analyses and experiments demonstrate that CQFA trains stably and achieves substantial gains over strong baselines on language modeling and information retrieval tasks.

1 Introduction

Causal linear attention represents sequence history with a matrix-valued recurrent state, enabling linear-time sequence processing and constant-size memory during decoding (Katharopoulos et al., 2020). Viewed as a fast-weight associative memory, this state stores key–value associations that queries read out (Schlag et al., 2021). This efficiency depends on compressing an expanding history into a fixed capacity. As new tokens arrive, the model must incorporate their information while preserving associations that remain useful for prediction. The update rule therefore determines how effectively the model uses its limited memory.

A useful update must control both the persistence of old information and the integration of new associations. Gated Linear Attention (GLA) and selective state space models, including Mamba-2 and Mamba-3, address retention through input-dependent decay (Yang et al., 2024a; Dao & Gu, 2024; Lahoti et al., 2026). Decay alone, however, does not adjust a write according to the association already stored. DeltaNet introduces such prediction-based correction: it retrieves the value associated with the current key and updates the state using the residual between this prediction and the incoming value (Schlag et al., 2021), with chunkwise parallelization enabling efficient training (Yang et al., 2024b). This interpretation of memory updates as online learning also underlies Test-Time Training (TTT), which represents the state as the parameters of a linear or nonlinear learner trained on incoming tokens through a self-supervised objective (Sun et al., 2025).

For linear memories, Gated DeltaNet brings selective decay and residual correction into a single update, allowing outdated information to be attenuated while individual associations are revised (Yang et al., 2025). Kimi Delta Attention (KDA) refines the retention component with channel-wise decay, but its update still ties erasure and writing to a shared coefficient (Kimi Team, 2025). Gated DeltaNet-2 (GDN-2) removes this coupling through separate channel-wise gates for key-side erasure and value-side writing (Hatamizadeh et al., 2026). This gives the model independent control over modifying existing associations and storing new values.

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

Even with this separation, GDN-2 forms its correction from a key-conditioned memory prediction. The query determines how the state is read for the layer's output, but does not enter that prediction. Its readout may expose complementary information in the same memory, motivating its use in state correction. Q-Delta introduces query feedback with an input-dependent scalar coefficient per head (Park et al., 2026). A scalar only rescales the combined query readout, however, and cannot independently adjust its channel contributions. We therefore examine whether query readouts provide useful information beyond key readouts after decay and erase gating, and whether exploiting it benefits from channel-wise feedback.

To address these questions, we introduce *Channel-wise Query Feedback Attention* (**CQFA**), which equips GDN-2 with input-dependent feedback strengths for individual key-side channels. Reweighting query contributions before they are combined in the memory readout lets CQFA alter the direction as well as the magnitude of feedback. A separate query gate provides this flexibility while retaining GDN-2's distinct controls over key-side erasure and value-side writing. CQFA reuses the existing query and preserves the recurrent state size and diagonal-plus-rank-one transition structure, enabling efficient chunkwise parallel computation. Feedback also changes the recurrent transition, motivating analysis of both local residual correction and information propagation.

Our contributions are threefold:

- We analyze the role of query readouts in state correction and investigate the need for channel-wise feedback. This leads to CQFA, which selectively controls query feedback across memory channels while retaining GDN-2's decoupled erase and write mechanism.
- We establish a sufficient condition for one-step residual contraction and provide a systematic analysis connecting key–query signal complementarity, residual alignment, feedback strength, and update dynamics. This analysis examines both local residual correction and the propagation of stored information through the recurrent state.
- We implement efficient chunkwise parallel computation for CQFA, preserving linear-time sequence processing and the original recurrent state size. Experiments under matched training conditions show improvements over GDN-2 on language modeling and in-context information retrieval tasks, alongside controlled comparisons of scalar and channel-wise feedback.

2 Background

2.1 Recurrent Linear Attention and Delta Updates

Consider one head with $q_t, k_t \in \mathbb{R}^{d_k}$, $v_t \in \mathbb{R}^{d_v}$, and state $S_t \in \mathbb{R}^{d_k \times d_v}$ at position t. The output is $o_t = S_t^\top q_t$; state rows and columns index key-side and value-side channels. We use $\odot$ for elementwise multiplication, $\mathrm{Diag}(x)$ for a diagonal matrix, and $\langle \cdot, \cdot \rangle_F$ for the Frobenius inner product. Feature maps and normalization are absorbed into q_t, k_t. Table 1 compares the recurrent cores, omitting output projections and gates.

Linear attention accumulates key–value associations from $S_0 = 0$ (Katharopoulos et al., 2020):

$$S_t = S_{t-1} + k_t v_t^\top, \qquad o_t = S_t^\top q_t = \sum_{i=1}^{t} v_i (k_i^\top q_t). \tag{1}$$

This fast-weight associative memory (Schlag et al., 2021) takes $O(T d_k d_v)$ time for T tokens and $O(d_k d_v)$ recurrent memory.

Decay controls retention. RetNet uses fixed rates (Sun et al., 2023), whereas Mamba and Gated Linear Attention (GLA) make them input-dependent (Gu & Dao, 2024; Yang et al., 2024a). For channel-wise decay, $\bar{S}_t = D_t S_{t-1}$ with $D_t = \mathrm{Diag}(\boldsymbol{\alpha}_t)$. Mamba-2 uses $D_t = \alpha_t I$ and relates its recurrence to attention through structured state space duality (Dao & Gu, 2024); write $P_t = \alpha_t S_{t-1}$, absorbing discretization scaling into v_t. Mamba-3 extends this state-space route through exponential-trapezoidal integration, complex transitions implemented by query/key rotations, and multi-input, multi-output (MIMO) computation (Lahoti et al., 2026). Its single-input, single-output write is $J_t = \eta_t \tilde{k}_{t-1} v_{t-1}^\top + \zeta_t \tilde{k}_t v_t^\top$, where $\tilde{k}_t$ is the rotated key and η_t, ζ_t are integration weights.

Table 1: Local objectives and recurrent updates in a common state orientation. Prior methods follow the order of their first public reports; Scalar-Q and CQFA use the GDN-2 backbone.

Method	Local objective $\mathcal{L}_t(S)$	State update
Linear attention	$\|S - S_{t-1}\|_F^2 - 2\langle S^\top k_t, v_t\rangle$	$S_t = S_{t-1} + k_t v_t^\top$
DeltaNet	$\|S - S_{t-1}\|_F^2 - 2\langle S^\top k_t, \beta_t(v_t - S_{t-1}^\top k_t)\rangle$	$S_t = (I - \beta_t k_t k_t^\top)S_{t-1} + \beta_t k_t v_t^\top$
Mamba-2	$\|S - P_t\|_F^2 - 2\langle S^\top k_t, v_t\rangle$	$S_t = \alpha_t S_{t-1} + k_t v_t^\top$
Gated DeltaNet	$\|S - P_t\|_F^2 - 2\langle S^\top k_t, \beta_t(v_t - P_t^\top k_t)\rangle$	$S_t = \alpha_t(I - \beta_t k_t k_t^\top)S_{t-1} + \beta_t k_t v_t^\top$
Mamba-3	$\|S - P_t\|_F^2 - 2\langle S, J_t\rangle_F$	$S_t = \alpha_t S_{t-1} + \eta_t \tilde{k}_{t-1} v_{t-1}^\top + \zeta_t \tilde{k}_t v_t^\top$
KDA	$\|S - \bar{S}_t\|_F^2 - 2\langle S^\top k_t, \beta_t(v_t - \bar{S}_t^\top k_t)\rangle$	$S_t = (I - \beta_t k_t k_t^\top)D_t S_{t-1} + \beta_t k_t v_t^\top$
GDN-2	$\|S - \bar{S}_t\|_F^2 - 2\langle S^\top k_t, y_t - \bar{S}_t^\top (b_t \odot k_t)\rangle$	$S_t = (I - k_t(b_t \odot k_t)^\top)\bar{S}_t + k_t y_t^\top$
Q-Delta	$\|S - P_t\|_F^2 - 2\langle S^\top k_t, \beta_t[v_t - P_t^\top(k_t + \lambda_t q_t)]\rangle$	$S_t = (I - \beta_t k_t(k_t + \lambda_t q_t)^\top)P_t + \beta_t k_t v_t^\top$
Scalar-Q	$\|S - \bar{S}_t\|_F^2 - 2\langle S^\top k_t, y_t - \bar{S}_t^\top a_t^{\mathrm{s}}\rangle$	$S_t = (I - k_t(a_t^{\mathrm{s}})^\top)\bar{S}_t + k_t y_t^\top$
CQFA (ours)	$\|S - \bar{S}_t\|_F^2 - 2\langle S^\top k_t, y_t - \bar{S}_t^\top a_t\rangle$	$S_t = (I - k_t a_t^\top)\bar{S}_t + k_t y_t^\top$

These mechanisms regulate retention and input integration; delta updates add correction using the association already stored.

The delta rule retrieves the value associated with the current key and writes the prediction residual:

$$S_t = \bar{S}_t + \beta_t k_t \big(v_t - \bar{S}_t^\top k_t\big)^\top, \qquad \bar{S}_t = D_t S_{t-1}. \tag{2}$$

This is a gradient step of size β_t on $\ell_t(S) = \frac{1}{2}\|S^\top k_t - v_t\|^2$ at $\bar{S}_t$ (Schlag et al., 2021; Yang et al., 2024b). DeltaNet uses $D_t = I$; Gated DeltaNet adds scalar decay, and Kimi Delta Attention (KDA) makes the decay channel-wise (Yang et al., 2025; Kimi Team, 2025). In each case, β_t scales both erasure and writing. RWKV-7 further separates removal and replacement keys with vector-valued gates (Peng et al., 2025); DeltaProduct composes multiple generalized Householder updates per token (Siems et al., 2025).

The regression view also connects recurrent memory to online learning. Longhorn derives an implicit associative-regression update and a diagonal approximation (Liu et al., 2025); Test-Time Training (TTT) updates linear or nonlinear state learners (Sun et al., 2025), and Titans adds momentum and adaptive forgetting (Behrouz et al., 2025). Following GDN-2 and Q-Delta (Hatamizadeh et al., 2026; Park et al., 2026), Table 1 expresses each recurrent step as $S_t = \arg\min_S \mathcal{L}_t(S)$. A quadratic term anchors the state to its prior; a linear term specifies the write using predictions at that fixed prior. Appendix C derives these objectives by completing the square.

Chunkwise computation combines recurrence between chunks with parallel matrix operations within them (Sun et al., 2023; Yang et al., 2024a; Dao & Gu, 2024). For additive attention, let Q, K, V stack the chunk's vectors as rows and M be its lower-triangular causal mask:

$$S_{\text{out}} = S_{\text{in}} + K^\top V, \qquad O = QS_{\text{in}} + \big(QK^\top \odot M\big)V. \tag{3}$$

For delta updates, compact products of rank-one transitions yield triangular solves and matrix multiplications (Yang et al., 2024b; 2025). Preserving this structure enables efficient query feedback.

2.2 GDN-2 and Query Feedback

Gated DeltaNet-2 (GDN-2) decouples erasure and writing with sigmoid gates $b_t \in (0,1)^{d_k}$ and $w_t \in (0,1)^{d_v}$, while retaining channel-wise decay (Hatamizadeh et al., 2026):

$$\begin{aligned} &\bar{S}_t = D_t S_{t-1}, \qquad y_t = w_t \odot v_t, \\ &r_t^0 = y_t - \bar{S}_t^\top (b_t \odot k_t), \qquad S_t = \bar{S}_t + k_t (r_t^0)^\top. \end{aligned} \tag{4}$$

The erase gate weights the old memory prediction; the write gate independently scales the incoming value. Setting $b_t = \beta_t \mathbf{1}$ and $w_t = \beta_t \mathbf{1}$ recovers Equation (2). The correction is key-conditioned, while q_t reads the updated state.

Q-Delta replaces the prediction with $\bar{S}_t^\top(k_t + \lambda_t q_t)$ using one input-dependent scalar per head (Park et al., 2026); its original rule retains scalar decay and a shared erase/write coefficient. Transferring

this feedback to GDN-2 gives our Scalar-Q control, $a_t^s = b_t \odot (k_t + \lambda_t q_t)$. CQFA instead uses $a_t = b_t \odot (k_t + \boldsymbol{\lambda}_t \odot q_t)$, with $\boldsymbol{\lambda}_t \in \mathbb{R}^{d_k}$. This distinction motivates the next section: what the query adds to memory prediction, and how channel-specific feedback controls that contribution.

3 CQFA: Channel-Wise Query Feedback

3.1 Query Readout over Accumulated Memory

A recurrent state combines past writes after their subsequent transformations. Following the memory-expansion view of Park et al. (2026), let A_t denote the transition and $y_t = w_t \odot v_t$ the independently gated write. For $S_0 = 0$,

$$S_t = A_t S_{t-1} + k_t y_t^\top, \qquad g_{\tau,t} = A_{t-1} \cdots A_{\tau+1} k_\tau, \quad \tau < t,$$
$$S_{t-1} = \sum_{\tau<t} g_{\tau,t} y_\tau^\top, \qquad \bar{S}_t = D_t S_{t-1}. \tag{5}$$

The propagated key $g_{\tau,t}$ determines how an earlier write is addressed at position t. Empty products equal I; Appendix C derives the expansion including a nonzero initial state.

Reading the same memory with the current key and query gives

$$\hat{v}_t = S_{t-1}^\top k_t, \qquad \hat{o}_t = S_{t-1}^\top q_t,$$
$$S_{t-1}^\top x_t = \sum_{\tau<t} (g_{\tau,t}^\top x_t) y_\tau, \qquad p_t^x = \bar{S}_t^\top (b_t \odot x_t), \quad x_t \in \{k_t, q_t\}. \tag{6}$$

Both readouts aggregate stored values, but their coefficients address different combinations of the accumulated memory. The query is particularly relevant because it also determines the layer's output. We examine $\hat{o}_t, \hat{v}_t$ before the current update, and use p_t^q, p_t^k to track their contributions after decay and erase weighting.

Figure 1 shows directional variation in a trained 340M CQFA model; Table 2 quantifies PCA coverage and gated energy. The readouts share dominant value-space directions, while their per-token combinations differ. Thus, the distinction lies in how a common memory is combined. To measure its magnitude in the gated prediction, define, for nonzero readouts,

$$\rho_t = \frac{\langle p_t^q, p_t^k \rangle}{\|p_t^q\| \, \|p_t^k\|}, \qquad P_t^k = \frac{p_t^k (p_t^k)^\top}{\|p_t^k\|^2}, \qquad \frac{\|(I - P_t^k) p_t^q\|^2}{\|p_t^q\|^2} = 1 - \rho_t^2. \tag{7}$$

Across FineWeb, NQ, and SQuAD, respectively, CQFA places 67.13%, 71.33%, and 63.12% of its gated query energy outside the contemporaneous key-readout direction. Its total query-to-key energy ratios are 0.072, 0.105, and 0.059. The query therefore exposes a distinct, non-negligible contribution to the prediction used for state correction.

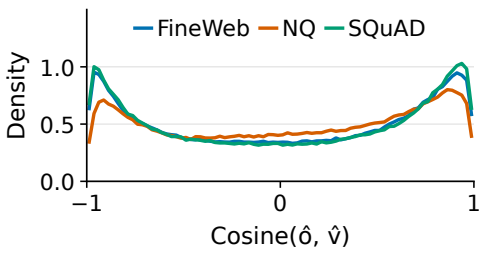

Figure 1: Prior-state readout cosine on three sources, using one CQFA checkpoint.

Table 2: Query-readout geometry. Own/Key: PCA basis fitted to $\hat{o}/\hat{v}$. PCA and $\eta_{q|k}$ are percentages.

	Rank-16 PCA			
Source	Own	Key	$\eta_{q\|k}$	E_q/E_k
FineWeb	76.61	75.02	67.13	0.072
NQ	83.32	81.18	71.33	0.105
SQuAD	73.42	71.47	63.12	0.059

3.2 Channel-Wise Feedback and the CQFA Update

A scalar gate can vary the strength of query feedback but couples all of its channel contributions. Let $M_t = \bar{S}_t^\top \operatorname{Diag}(b_t \odot q_t) \in \mathbb{R}^{d_v \times d_k}$. Then scalar and channel-wise feedback take the forms

$$u_t^{\text{scalar}} = \lambda_t M_t \mathbf{1}, \qquad u_t^{\text{channel}} = M_t \boldsymbol{\lambda}_t = \sum_{i=1}^{d_k} \lambda_{t,i} b_{t,i} q_{t,i} \bar{S}_{t,i,:}^\top. \tag{8}$$

A shared coefficient scales the sum of memory-row contributions. Separate coefficients reweight these contributions before summation, allowing the feedback direction to change together with its magnitude. CQFA implements this control directly on the query channels.

Figure 2 shows the complete CQFA layer and token mixer. The orange branch adds query feedback to GDN-2's normalized features and independent decay, erase, and write gates (Hatamizadeh et al., 2026). The erase gate weights both key and gated-query predictions; the write gate controls the incoming value. After state correction, the query reads the memory, followed by normalization, output gating, and projection. Query feedback preserves the $d_k \times d_v$ state and the rank-one correction.

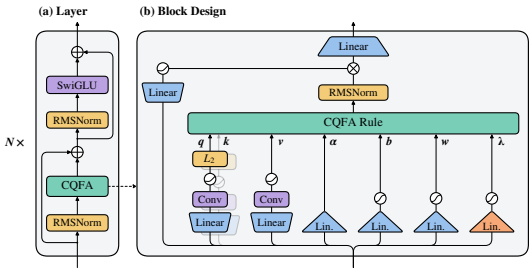

Figure 2: CQFA architecture. Orange marks the query-feedback gate.

The resulting correction is

$$\begin{aligned} a_t &= b_t \odot (k_t + \boldsymbol{\lambda}_t \odot q_t), \qquad u_t = M_t \boldsymbol{\lambda}_t, \\ r_t^q &= y_t - \bar{S}_t^\top a_t = r_t^0 - u_t, \qquad S_t = \bar{S}_t + k_t (r_t^q)^\top. \end{aligned} \tag{9}$$

This separation lets the model adjust query feedback without jointly changing the key coefficient or the new-value write.

For the prenormalized layer input $h_t \in \mathbb{R}^d$, we parameterize the feedback logits with two bias-free linear maps and no intermediate activation. With $m = Hd_k$ channels across H heads,

$$\boldsymbol{\lambda}_t^{\text{all}} = L\,\sigma(W_{\text{up}} W_{\text{down}} h_t + \delta_\lambda \mathbf{1}), \quad W_{\text{down}} \in \mathbb{R}^{r \times d}, \quad W_{\text{up}} \in \mathbb{R}^{m \times r}. \tag{10}$$

Each head takes its corresponding d_k entries. The default implementation sets $L = 0.4$ and $\delta_\lambda = -0.9$, so each gate lies in $(0, 0.4)$. The offset is fixed, separate from the bias-free projections. The module uses the same query for memory feedback and downstream readout.

Setting $\boldsymbol{\lambda}_t = 0$ recovers GDN-2; setting $\boldsymbol{\lambda}_t = \lambda_t \mathbf{1}$ gives Scalar-Q. Our scalar control retains the same low-rank projection and averages its logits within each head before the sigmoid. Section 4.2 compares the resulting feedback variants across downstream tasks.

The trained gates also exhibit this distinction geometrically. We decompose the actual feedback into components parallel and orthogonal to p_t^q, with full statistics in Appendix E.1. The orthogonal component accounts for 9.05–13.34% of CQFA's feedback energy across the three natural-text sources, whereas Scalar-Q has $u_t = \lambda_t p_t^q$ and hence zero orthogonal energy. Channel-wise control therefore changes the feedback used in the forward computation.

3.3 Conditional Stability and Feedback Strength

The feedback direction determines how the correction acts on its own prediction residual. Hold the current target y_t and direction a_t fixed, and define $e_t^- = y_t - \bar{S}_t^\top a_t$ and $e_t^+ = y_t - S_t^\top a_t$. Substituting Equation (9) gives

$$\begin{aligned} e_t^+ &= (1 - c_t) e_t^-, \qquad c_t = k_t^\top a_t = c_t^k + c_t^q, \\ c_t^k &= \sum_i b_{t,i} k_{t,i}^2, \qquad c_t^q = \sum_i b_{t,i} \lambda_{t,i} k_{t,i} q_{t,i}. \end{aligned} \tag{11}$$

Consequently, $0 < c_t < 2$ guarantees strict one-step contraction for a nonzero residual. Appendix C derives the identity and bounds its coefficient under the gate constraints.

The decomposition separates the nonnegative key term from the signed query term, identifying how channel-specific feedback affects the contraction margin. Figure 3 evaluates the contraction coefficient on the trained CQFA model. More than 99.9998% of CQFA observations satisfy $0 < c_t < 2$ on each source. The small negative tail occurs when the signed query term opposes and exceeds the key term. FP64 replay verifies the residual identity to a maximum absolute error of 1.24×10^{-14}; Appendix E.2 reports the full counts.

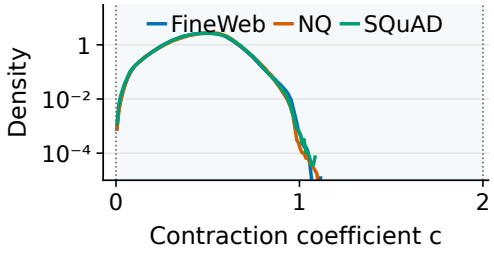

Figure 3: Contraction coefficients. Shading indicates $0 < c_t < 2$.

Feedback strength also controls the residual presented to this update. At the retained state $\bar{S}_t$, the exact change is

$$\Delta E_t(u_t) = \|r_t^0\|^2 - \|r_t^0 - u_t\|^2 = 2\langle r_t^0, u_t\rangle - \|u_t\|^2. \tag{12}$$

The first term rewards alignment with the key-only residual; the second is the quadratic feedback cost. Together with Equation (11), $\|e_t^+\|^2 = (1-c_t)^2(\|r_t^0\|^2 - \Delta E_t)$ separates prediction adjustment from state correction. On FineWeb, NQ, and SQuAD, alignment exceeds the feedback cost, reducing aggregate residual energy by 0.048%, 0.100%, and 0.036%, respectively, with positive 95% intervals (Appendix E.2). Thus, the learned channel-wise feedback improves the local prediction through its direction and strength. Matched training and task results in Section 4.2 assess the resulting task benefit.

3.4 Chunkwise Parallelization and Implementation

Chunkwise formulations replace tokenwise execution with parallel work inside contiguous chunks and recurrence across their boundaries. DeltaNet and its gated extensions achieve this through compact WY/UT representations of rank-one updates (Yang et al., 2024b; 2025). GDN-2 additionally absorbs channel-wise decay into asymmetric erase and write factors (Hatamizadeh et al., 2026). CQFA preserves this structure:

$$A_t = (I - k_t a_t^\top)D_t = D_t - k_t(D_t a_t)^\top, \qquad S_t = A_t S_{t-1} + k_t y_t^\top. \tag{13}$$

All gates, including query feedback, are generated from the layer input before the recurrent computation. Hence, the new prediction direction can enter the same blockwise factorization.

For a chunk of length C with incoming state S_{in} and strictly positive channel decays, define

$$\begin{aligned} &\gamma_i = \boldsymbol{\alpha}_1 \odot \cdots \odot \boldsymbol{\alpha}_i, \qquad \gamma_0 = \mathbf{1}, \\ &\tilde{k}_i = k_i \oslash \gamma_i, \qquad \tilde{a}_i = \gamma_i \odot a_i, \qquad \tilde{q}_i = \gamma_i \odot q_i. \end{aligned} \tag{14}$$

Stack these factors and y_i as the rows of $\widetilde{K}, \widetilde{E}, \widetilde{Q}, Y$. The strictly causal interactions form a unit lower-triangular system:

$$\begin{aligned} &\mathcal{T} = I + \text{tril}(\widetilde{E}\widetilde{K}^\top, -1), \qquad \mathcal{T}[\,U\ W\,] = [\,Y\ \widetilde{E}\,], \\ &R = U - W S_{\text{in}}, \end{aligned} \tag{15}$$

where $R, U \in \mathbb{R}^{C \times d_v}$ and $W \in \mathbb{R}^{C \times d_k}$. With the diagonal-inclusive causal mask M, the block state and outputs are

$$\begin{aligned} S_{\text{out}} &= \text{Diag}(\gamma_C)\big(S_{\text{in}} + \widetilde{K}^\top R\big), \\ O &= \widetilde{Q} S_{\text{in}} + \big(\widetilde{Q}\widetilde{K}^\top \odot M\big) R. \end{aligned} \tag{16}$$

CQFA changes the erase factor to $\tilde{a}_i = \gamma_i \odot b_i \odot (k_i + \boldsymbol{\lambda}_i \odot q_i)$. The system dimensions, write target, and block-boundary state are unchanged.

The implementation constructs local coefficients and solves for U, W in parallel, propagates chunk states, and evaluates the outputs by matrix multiplication. Accumulating decay in log space expresses

the required factors through differences of cumulative log decays. Backpropagation reverses this factorization and distributes the gradient of a_t to the key, query, erase gate, and query gate; the query also receives its ordinary readout gradient. Appendix D derives both passes and provides reference and CUDA-path code. For fixed head dimensions and chunk length, computation is linear in sequence length and the recurrent state size is unchanged. The additional work is the feedback projection and elementwise composition, whose measured cost is reported in Section 4.3.

4 Experiments

4.1 Experimental Setup

We train 340M and 1.7B models on 15B and 40B FineWeb-Edu tokens, respectively (Penedo et al., 2024), using bfloat16 on 16 NVIDIA RTX PRO 6000 GPUs or 16 Ascend NPUs. Models share data order and evaluation protocols at each scale. We use three independent training seeds and evaluate each run; tables report the best results. RULER uses three shared evaluation-input seeds, with the same best-score rule for all models. Appendix A details the protocols.

Baselines and feedback variants. We compare with Mamba-2 (Dao & Gu, 2024), Mamba-3 (Lahoti et al., 2026), Gated DeltaNet (Yang et al., 2025), KDA (Kimi Team, 2025), and Q-Delta (Park et al., 2026). GDN-2 (Hatamizadeh et al., 2026), Scalar-Q, and CQFA share depth, hidden size, and state dimensions. At the 340M scale, their parameter counts are 340.7M, 345.4M, and 345.4M, respectively. Scalar-Q pools the feedback logits within each head, whereas CQFA retains their channel-wise resolution. Appendix B specifies the gates.

4.2 Language Modeling and Task Performance

Language modeling and commonsense reasoning. Table 3 reports held-out cross-entropy, WikiText perplexity, and eight zero-shot accuracies, including BoolQ and OpenBookQA. CQFA reaches 47.88% and 58.39% average accuracy at 340M and 1.7B, exceeding the strongest baseline by 1.40 and 1.41 points, with strong gains on WinoGrande and OpenBookQA. Bold and underlined numbers denote the best and second-best distinct values, respectively.

Table 3: Language-modeling and reasoning results. Accuracy is in %; ^{n}: length-normalized scoring; Avg.: eight accuracy tasks. Mamba-3 uses SISO.

Model	Test NLL↓	Wiki. PPL↓	Lamb. Acc.↑	ARC-E ↑	ARC-C^n ↑	Hella.^{n} ↑	PIQA ↑	Wino. ↑	BoolQ ↑	OBQA^n ↑	Avg. ↑
340M parameters, 15B training tokens											
Mamba-2	2.4625	29.07	30.48	54.13	23.50	36.19	63.69	50.70	58.79	28.91	43.30
Mamba-3	2.4364	28.64	32.35	56.43	25.27	37.88	65.33	50.99	60.11	31.54	44.99
Gated DeltaNet	2.4321	28.12	31.29	56.26	25.34	37.70	65.25	52.43	59.08	30.88	44.78
KDA	2.4214	<u>27.54</u>	33.24	<u>57.90</u>	<u>26.99</u>	39.02	<u>67.10</u>	52.88	60.09	31.97	46.15
Q-Delta	2.4196	28.04	<u>33.42</u>	57.45	26.81	39.15	66.81	<u>53.43</u>	**62.07**	<u>32.69</u>	<u>46.48</u>
GDN-2	2.4162	**27.51**	33.38	56.78	26.40	<u>39.32</u>	66.03	52.26	60.72	32.49	45.92
Scalar-Q	<u>2.4138</u>	28.04	33.28	57.11	26.09	38.48	66.02	49.65	59.19	32.34	45.27
CQFA	**2.4087**	27.67	**35.33**	**59.93**	**28.08**	**39.60**	**67.56**	**55.95**	<u>60.78</u>	**35.79**	**47.88**
1.7B parameters, 40B training tokens											
Mamba-2	2.1141	19.19	48.73	63.94	32.08	48.92	70.98	56.75	62.45	35.60	52.43
Mamba-3	2.0811	17.83	51.14	66.30	34.87	51.98	73.45	59.73	64.14	38.06	54.96
Gated DeltaNet	2.0996	18.34	50.66	65.75	33.42	49.55	71.70	59.39	63.83	37.68	54.00
KDA	2.0735	18.21	53.63	<u>68.88</u>	35.96	53.15	74.51	60.96	66.29	41.02	56.80
Q-Delta	2.0693	17.85	<u>54.29</u>	68.37	36.18	53.34	73.74	<u>62.16</u>	66.25	<u>41.49</u>	<u>56.98</u>
GDN-2	2.0679	17.96	53.22	68.27	<u>36.36</u>	53.10	**74.96**	60.99	65.72	40.11	56.59
Scalar-Q	<u>2.0622</u>	**17.69**	54.20	68.21	35.79	<u>53.40</u>	<u>74.52</u>	61.58	**66.46**	41.37	56.94
CQFA	**2.0604**	<u>17.70</u>	**56.45**	**70.45**	**36.65**	**54.02**	74.05	**64.88**	<u>66.30</u>	**44.31**	**58.39**

Natural retrieval. Table 4 uses the single-read cloze protocol of Arora et al. (2024) with a 2K input limit. CQFA leads on SWDE, SQuAD, and NQ. It reaches 21.26% average accuracy, exceeding the next-best baseline by 1.15 points. The tasks use shared inputs and answer-span accuracy.

Table 4: Natural retrieval at 340M/15B (%). Avg.: six tasks.

Model	SWDE	SQuAD	FDA	TriviaQA	NQ	DROP	Avg.
Mamba-2	7.54	22.22	2.43	43.36	6.66	16.20	16.40
Mamba-3	9.46	25.29	2.43	47.16	8.16	16.96	18.24
Gated DeltaNet	9.44	24.99	3.47	45.68	7.63	16.60	17.97
KDA	11.88	27.51	**4.11**	47.91	8.35	18.32	19.68
Q-Delta	12.94	28.22	3.84	46.22	10.53	18.92	20.11
GDN-2	11.16	26.07	3.94	**48.16**	8.74	17.35	19.24
Scalar-Q	9.79	26.39	3.69	45.60	8.73	**19.35**	18.93
CQFA	**16.86**	**30.03**	3.35	47.53	**11.74**	18.07	**21.26**

Controlled associative retrieval. Tables 5–6 evaluate RULER (Hsieh et al., 2024) at 1K–4K for S-NIAH and 2K for MK1/MV/MQ. Each condition uses three shared input sets of 384 examples, reporting the best score. CQFA leads the multi-association average at 23.13%, compared with 21.89% for Q-Delta and 15.51% for GDN-2. Its strongest multi-association gains are on MV and MQ.

Table 5: S-NIAH recall (%), 340M: best of three shared evaluation-input seeds.

	S-NIAH-1			S-NIAH-2			S-NIAH-3		
Model	1K	2K	4K	1K	2K	4K	1K	2K	4K
Mamba-2	98.63	91.37	70.57	53.23	24.34	9.18	4.29	0.65	0.26
Mamba-3	99.63	95.59	84.40	70.60	33.28	15.99	10.59	4.21	0.61
Gated DeltaNet	99.86	95.96	88.09	70.71	33.18	14.01	9.64	3.55	0.21
KDA	**100.00**	97.73	91.20	76.99	40.89	18.60	14.91	5.27	1.17
Q-Delta	99.76	97.29	88.77	81.62	41.66	19.80	14.53	5.57	0.53
GDN-2	99.98	**100.00**	**96.29**	76.37	38.99	17.41	**16.92**	5.35	0.70
Scalar-Q	99.35	93.32	75.41	67.23	39.35	**20.75**	6.13	2.68	0.00
CQFA	**100.00**	98.78	91.73	**85.22**	**43.97**	19.99	15.71	**6.53**	**1.29**

4.3 Training Throughput

Figure 4 compares 340M models on one RTX PRO 6000 at 16,384 tokens per step. From 2K to 16K sequence length, CQFA retains 95.99–96.34% of GDN-2 throughput, decreasing from 67.58 to 61.21 Ktokens/s versus 70.40 to 63.54. CQFA preserves the scaling profile with a modest feedback cost. Each of three repeats times ten forward and backward steps after warm-up (Appendix D).

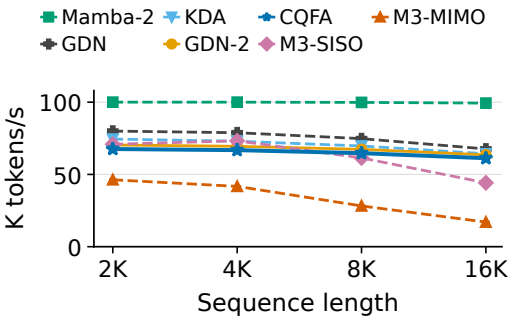

Figure 4: Mean training throughput; bars: SD. M3: Mamba-3.

Table 6: Multi-association recall (%), 340M/2K. Best of three seeds; Avg.: MK1/MV/MQ.

Model	MK1	MV	MQ	Avg.
Mamba-2	18.04	10.14	6.83	11.67
Mamba-3	22.57	15.85	10.73	16.38
Gated DeltaNet	19.95	12.83	9.20	13.99
KDA	25.58	18.95	15.92	20.15
Q-Delta	**27.18**	21.09	17.39	21.89
GDN-2	23.06	13.38	10.10	15.51
Scalar-Q	25.45	21.90	16.99	21.45
CQFA	25.58	**23.06**	**20.76**	**23.13**

5 Conclusion

We presented CQFA, which incorporates channel-wise query feedback into the decoupled erase-and-write update of GDN-2. The method lets the query participate in state correction through input-dependent channel gates while retaining a constant-size recurrent state and a rank-one update. Our analysis connects the geometry of query and key readouts to the feedback direction, characterizes its contribution to the local residual, and establishes a sufficient condition for one-step contraction. An equivalent chunkwise formulation supports parallel training with a small measured throughput cost, linking finer memory control to a practical implementation.

Our experiments cover models up to 1.7B parameters and training budgets up to 40B tokens. Computational budget constraints prevented experiments at larger model and data scales.

AI Use Statement

AI tools assisted with writing and polishing the manuscript and with literature retrieval and discovery. They also supported experimental planning, code and figure preparation, and checks of derivations and analyses. The authors take responsibility for the paper's content and conclusions.

Reproducibility Statement

Section 4.1 and Appendices A–E describe the data construction, training setup, evaluation protocols, theoretical derivations, and implementation. Code will be released.

References

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language models enable simple systems for generating structured views of heterogeneous data lakes. *Proceedings of the VLDB Endowment*, 17(2):92–105, 2023. URL https://www.vldb.org/pvldb/vol17/p92-arora.pdf.

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. Just read twice: Closing the recall gap for recurrent language models. *arXiv preprint arXiv:2407.05483*, 2024. URL https://arxiv.org/abs/2407.05483.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4ca07aa108036f80cbb5b82285fd4b1-Paper-Conference.pdf.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, 2019. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: A unified solution for structured web data extraction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 775–784, 2011. doi: 10.1145/2009916.2010020. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/StructedDataExtraction_SIGIR2011.pdf.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026. URL https://arxiv.org/abs/2605.22791.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025. URL https://arxiv.org/abs/2510.26692.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.15569.

Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. Longhorn: State space models are amortized online learners. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8jOqCcLzeO.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Sumin Park, Seojin Kim, and Noseong Park. Q-Delta: Beyond key–value associative state evolution. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.08804. arXiv:2606.08804v1.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. RWKV-7 "goose" with expressive dynamic state evolution. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayB1PACN5j.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. DeltaProduct: Improving state-tracking in linear RNNs via Householder products. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2502.10297.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023. URL https://arxiv.org/abs/2307.08621.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 2024a. URL https://proceedings.mlr.press/v235/yang24ab.html.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, volume 37, pp. 115491–115522, 2024b. doi: 10.52202/079017-3668. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d13a3eae72366e61dfdc7eea82eeb685-Abstract-Conference.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

A Datasets and Evaluation Tasks

A.1 Pretraining Data

FineWeb-Edu selects educational English web text from the FineWeb corpus (Penedo et al., 2024). We use the same pretraining corpus and held-out data for the compared models at each scale. Validation loss tracks optimization; held-out language-modeling loss and downstream tasks assess the learned sequence representations. The two training budgets are given in Section 4.1. Training uses three independent seeds, with each trained model evaluated under the same task protocol. The displayed 340M results use completed 15B-token checkpoints. Language modeling and natural retrieval use fixed evaluation sets. For synthetic retrieval, all models receive the same three independently generated sets of 384 examples per condition; each cell reports the highest of these three scores. Each model uses one checkpoint throughout the task tables.

A.2 Language Modeling and Reasoning

WikiText (Merity et al., 2017) measures next-token prediction on natural text, while LAMBADA (Paperno et al., 2016) requires predicting a final word from its preceding passage. The zero-shot reasoning suite covers scientific knowledge, event completion, physical reasoning, coreference, and reading comprehension. Table 7 summarizes the tasks. Multiple-choice answers are scored by their conditional log likelihood; task-specific length normalization is held fixed across models.

Table 7: Language-modeling and reasoning datasets. Accuracy uses the task-specific candidate scoring indicated in Table 3.

Dataset	Capability	Metric
WikiText (Merity et al., 2017)	Natural-text language modeling	NLL / PPL
LAMBADA (Paperno et al., 2016)	Passage-level final-word prediction	Accuracy
ARC-E / ARC-C (Clark et al., 2018)	Scientific question answering	Accuracy
HellaSwag (Zellers et al., 2019)	Commonsense event continuation	Accuracy
PIQA (Bisk et al., 2020)	Physical commonsense reasoning	Accuracy
WinoGrande (Sakaguchi et al., 2020)	Commonsense coreference	Accuracy
BoolQ (Clark et al., 2019)	Passage-conditioned yes/no answers	Accuracy
OpenBookQA (Mihaylov et al., 2018)	Fact and commonsense composition	Accuracy

A.3 Natural-Context Retrieval

We evaluate the 340M models using the single-read cloze tasks released by Arora et al. (2024), with the same examples and a 2K-token input limit across models. SWDE extracts attributes from web pages (Hao et al., 2011); FDA extracts fields from the medical-device reports used by Evaporate (Arora et al., 2023). SQuAD uses the answerable examples of SQuAD 2.0 (Rajpurkar et al., 2018). TriviaQA, Natural Questions (NQ), and DROP cover evidence-based factual answers, web questions, and paragraph-level reasoning, respectively (Joshi et al., 2017; Kwiatkowski et al., 2019; Dua et al., 2019). These tasks test whether the recurrent state preserves the information needed to complete a context-grounded answer.

For example j, let g_j be the generated answer and $\mathcal{A}_j$ the set of accepted reference strings. The released answer-span metric is

$$\mathrm{Acc}_{\mathrm{span}} = \frac{100}{N} \sum_{j=1}^{N} \max_{a \in \mathcal{A}_j} \mathbf{1}\{\mathrm{lower}(a) \text{ is a substring of } \mathrm{lower}(g_j)\}. \tag{17}$$

A match to any accepted reference counts as a correct recovery. We keep the released prompts and generation limits fixed across models and report the unweighted mean of the six task scores.

A.4 Controlled Associative Retrieval

RULER (Hsieh et al., 2024) inserts key–value facts into a background and queries their values. Table 8 specifies the six variants used in Tables 5 and 6. The 340M models receive identical examples

at each length: S-NIAH uses total budgets of 1,024, 2,048, and 4,096 tokens; the multi-association tasks use 2,048. Each budget includes the official 128-token generation allowance. S-NIAH isolates retrieval under different backgrounds and value formats; MK-NIAH introduces competing associations; MV-NIAH and MQ-NIAH require the recovery of several values from one or several keys.

Table 8: Controlled retrieval tasks from RULER. Numeric values have seven digits; UUIDs are random identifiers. Each row is a separate evaluation variant.

Variant	Background	Required retrieval
S1	Repeated noise sentences	One numeric value for a word key
S2	Natural essays	One numeric value for a word key
S3	Natural essays	One UUID value for a word key
MK1	Natural essays	One of four word-key/numeric-value associations
MV	Natural essays	All four numeric values linked to one word key
MQ	Natural essays	Four numeric values, one for each queried word key

Let $\mathcal{V}_j$ contain the values required by example j. The official case-insensitive recall score is

$$\text{Recall} = \frac{100}{N}\sum_{j=1}^{N}\frac{1}{|\mathcal{V}_j|}\sum_{v\in\mathcal{V}_j}\mathbf{1}\{\text{lower}(v) \text{ is a substring of } \text{lower}(g_j)\}. \tag{18}$$

S-NIAH and MK-NIAH have one target per example. MV-NIAH and MQ-NIAH have four, so their scores retain information about partial recovery. We report each S-NIAH variant separately at each length, and average the three multi-association scores in Table 6.

B Channel-Wise Feedback Parameterization

Let $h_t \in \mathbb{R}^d$ denote the normalized layer input, H the number of heads, and $m = Hd_k$ the total key-side width. CQFA constructs the feedback logits through a factorized projection:

$$\begin{aligned} z_t &= W_{\text{down}}h_t \in \mathbb{R}^r, \qquad \ell_t = W_{\text{up}}z_t \in \mathbb{R}^m, \\ \boldsymbol{\lambda}_t^{\text{all}} &= L\,\sigma(\ell_t + \delta_\lambda \mathbf{1}_m), \qquad W_{\text{down}} \in \mathbb{R}^{r\times d}, \quad W_{\text{up}} \in \mathbb{R}^{m\times r}. \end{aligned} \tag{19}$$

The two projections have no intervening activation or learned bias. For head h, the corresponding d_k entries form $\boldsymbol{\lambda}_{t,h} \in (0, L)^{d_k}$. The default values $L = 0.4$ and $\delta_\lambda = -0.9$ give $\boldsymbol{\lambda}_t^{\text{all}} = 0.4\,\sigma(\ell_t - 0.9\mathbf{1}_m)$. The offset is fixed rather than a learned bias; the projection learns relative channel strengths. This branch adds $r(d+m)$ parameters per layer and leaves the recurrent state dimension unchanged.

Scalar-Q uses the same projection and pools its logits within each head:

$$\bar{\ell}_{t,h} = \frac{1}{d_k}\sum_{i=1}^{d_k}\ell_{t,h,i}, \qquad \lambda^{\text{s}}_{t,h} = L_{\text{s}}\,\sigma(\bar{\ell}_{t,h} + \delta_\lambda), \qquad \boldsymbol{\lambda}^{\text{s}}_{t,h} = \lambda^{\text{s}}_{t,h}\mathbf{1}_{d_k}. \tag{20}$$

Both feedback variants allocate the same parameter budget to the low-rank projection. Scalar-Q uses one coefficient per head, whereas CQFA retains channel-specific coefficients. The evaluated Scalar-Q control uses $L_{\text{s}} = 1.0$ and $\delta_\lambda = -0.9$. With head indices suppressed, the three updates are specified by

$$a_t = \begin{cases} b_t \odot k_t, & \text{GDN-2,} \\ b_t \odot (k_t + \lambda^{\text{s}}_t q_t), & \text{Scalar-Q,} \\ b_t \odot (k_t + \boldsymbol{\lambda}_t \odot q_t), & \text{CQFA.} \end{cases} \tag{21}$$

They share the independent write target $y_t = w_t \odot v_t$. GDN-2 omits the feedback projection and preserves the same recurrent state dimensions. Section 4.2 compares task performance across the three updates.

C Theoretical Derivations

We use $S_t \in \mathbb{R}^{d_k\times d_v}$ and the single-head notation of Section 3. Vector norms are Euclidean. The gates, addressing vectors, and target at a position are fixed when analyzing its state correction.

C.1 Memory Expansion and Readout Geometry

The recurrence and its ordered transition product are

$$S_t = A_t S_{t-1} + k_t y_t^\top, \qquad A_t = (I - k_t a_t^\top) D_t, \qquad \Phi_{j:i} = A_j A_{j-1} \cdots A_i, \tag{22}$$

with $\Phi_{j:i} = I$ for $j < i$. Expanding two successive steps gives

$$\begin{aligned} S_n &= A_n\big(A_{n-1} S_{n-2} + k_{n-1} y_{n-1}^\top\big) + k_n y_n^\top \\ &= A_n A_{n-1} S_{n-2} + A_n k_{n-1} y_{n-1}^\top + k_n y_n^\top \\ &= \Phi_{n:1} S_0 + \sum_{\tau=1}^{n} \Phi_{n:\tau+1} k_\tau y_\tau^\top. \end{aligned} \tag{23}$$

For completeness, the induction step follows by substituting the expansion at $n-1$ and using $A_n \Phi_{n-1:i} = \Phi_{n:i}$; the term at $\tau = n$ uses the empty product. Thus the formula holds for arbitrary S_0, with the transition order inherited directly from the recurrence.

Set $g_{\tau,t} = \Phi_{t-1:\tau+1} k_\tau$. For any addressing vector $x_t \in \mathbb{R}^{d_k}$,

$$\begin{aligned} S_{t-1} &= \Phi_{t-1:1} S_0 + \sum_{\tau<t} g_{\tau,t} y_\tau^\top, \\ S_{t-1}^\top x_t &= S_0^\top \Phi_{t-1:1}^\top x_t + \sum_{\tau<t} (g_{\tau,t}^\top x_t) y_\tau. \end{aligned} \tag{24}$$

Taking $x_t = k_t$ and $x_t = q_t$ yields $\hat{v}_t$ and $\hat{o}_t$. This recovers the historical-memory interpretation used by Q-Delta (Park et al., 2026), here with independently gated values y_τ and channel-wise decay. For $S_0 = 0$, both readouts lie in the span of the historical writes, with respective coefficients $g_{\tau,t}^\top k_t$ and $g_{\tau,t}^\top q_t$.

Current decay and erase gating modify the addressing vector before the feedback readout. Since $D_t = D_t^\top$,

$$\begin{aligned} p_t^x &= \bar{S}_t^\top (b_t \odot x_t) = S_{t-1}^\top D_t (b_t \odot x_t) \\ &= S_0^\top \Phi_{t-1:1}^\top D_t (b_t \odot x_t) + \sum_{\tau<t} \big[g_{\tau,t}^\top D_t (b_t \odot x_t)\big] y_\tau. \end{aligned} \tag{25}$$

This gives the coefficients used by the gated energy analysis, while the cosine and PCA analyses use the prior-state readouts $\hat{o}_t, \hat{v}_t$.

For nonzero p_t^k, let $P_t^k = p_t^k (p_t^k)^\top / \|p_t^k\|^2$. The identities $(P_t^k)^\top = P_t^k$ and $(P_t^k)^2 = P_t^k$ give

$$\begin{aligned} \|(I - P_t^k) p_t^q\|^2 &= (p_t^q)^\top (I - P_t^k)^\top (I - P_t^k) p_t^q \\ &= \|p_t^q\|^2 - \frac{\langle p_t^q, p_t^k \rangle^2}{\|p_t^k\|^2}, \\ \frac{\|(I - P_t^k) p_t^q\|^2}{\|p_t^q\|^2} &= 1 - \frac{\langle p_t^q, p_t^k \rangle^2}{\|p_t^q\|^2 \|p_t^k\|^2} = 1 - \rho_t^2. \end{aligned} \tag{26}$$

The last line also requires $p_t^q \neq 0$; zero-norm cases are counted separately. Over observations $\mathcal{I}$, the aggregate energy fraction is

$$\mathcal{E}_{q|k} = \frac{\sum_{t\in\mathcal{I}} \|(I - P_t^k) p_t^q\|^2}{\sum_{t\in\mathcal{I}} \|p_t^q\|^2} = \frac{\sum_{t\in\mathcal{I}} \|p_t^q\|^2 (1 - \rho_t^2)}{\sum_{t\in\mathcal{I}} \|p_t^q\|^2}. \tag{27}$$

Thus high-energy readouts contribute proportionally to the reported fraction.

C.2 Channel-Wise Feedback and the Local Objective

The contribution of memory row i and the resulting feedback matrix are

$$m_{t,i} = b_{t,i} q_{t,i} \bar{S}_{t,i,:}^\top, \qquad M_t = [m_{t,1}, \ldots, m_{t,d_k}] = \bar{S}_t^\top \operatorname{Diag}(b_t \odot q_t). \tag{28}$$

Consequently,

$$p_t^q = M_t\mathbf{1}, \qquad u_t^{\text{scalar}} = \lambda_t M_t\mathbf{1}, \qquad u_t^{\text{channel}} = M_t\boldsymbol{\lambda}_t. \tag{29}$$

To isolate gate granularity at a fixed state, consider a common bound L and the closures of the attainable gate ranges:

$$\mathcal{U}_t^{\text{s}} = \{\lambda M_t\mathbf{1} : 0 \le \lambda \le L\}, \qquad \mathcal{U}_t^{\text{c}} = \{M_t\lambda : \lambda \in [0, L]^{d_k}\}. \tag{30}$$

The embedding $\lambda \mapsto \lambda\mathbf{1}$ proves $\mathcal{U}_t^{\text{s}} \subseteq \mathcal{U}_t^{\text{c}}$. If $\operatorname{rank}(M_t) \ge 2$ and $M_t\mathbf{1} \ne 0$, at least one column $m_{t,i}$ lies outside $\operatorname{span}(M_t\mathbf{1})$: otherwise all columns would span at most one dimension. Hence

$$Lm_{t,i} = M_t(Le_i) \in \mathcal{U}_t^{\text{c}}, \qquad Lm_{t,i} \notin \mathcal{U}_t^{\text{s}}, \qquad \mathcal{U}_t^{\text{s}} \subsetneq \mathcal{U}_t^{\text{c}}. \tag{31}$$

If $M_t\mathbf{1} = 0$ and $M_t \ne 0$, a nonzero column gives the same strict inclusion because $\mathcal{U}_t^{\text{s}} = \{0\}$. The sigmoid approaches the endpoints by limits. These sets describe control at a fixed state; the shared projection in Appendix B specifies how gates are generated across inputs.

The additional direction selected by a learned gate can be isolated directly. For $p_t^q \ne 0$, define $P_t^q = p_t^q(p_t^q)^\top/\|p_t^q\|^2$. Then

$$\begin{aligned} u_t &= P_t^q M_t\boldsymbol{\lambda}_t + (I - P_t^q)M_t\boldsymbol{\lambda}_t, \\ \|(I - P_t^q)u_t\|^2 &= \boldsymbol{\lambda}_t^\top M_t^\top (I - P_t^q)M_t\boldsymbol{\lambda}_t, \\ (I - P_t^q)u_t^{\text{scalar}} &= 0. \end{aligned} \tag{32}$$

This decomposition measures the component introduced by unequal channel weighting in the actual feedback.

We next derive the local objectives in Table 1. For a fixed prior P and a fixed write matrix B, completing the square yields

$$\begin{aligned} \mathcal{L}(S; P, B) &= \|S - P\|_F^2 - 2\langle S, B\rangle_F \\ &= \|S - (P + B)\|_F^2 - \|B\|_F^2 - 2\langle P, B\rangle_F, \\ \nabla_S\mathcal{L} = 2(S - P - B), \qquad \nabla_S^2\mathcal{L} = 2I, \qquad &\arg\min_S \mathcal{L} = P + B. \end{aligned} \tag{33}$$

The identity $\langle S, kr^\top\rangle_F = \langle S^\top k, r\rangle$ converts each rank-one write to the table's notation. The substitutions are

$$(P, B) = \begin{cases} (S_{t-1},\ k_t v_t^\top), & \text{linear attention,} \\ (\alpha_t S_{t-1},\ k_t v_t^\top), & \text{Mamba-2,} \\ (\alpha_t S_{t-1},\ J_t), & \text{Mamba-3,} \\ (D_t S_{t-1},\ \beta_t k_t(v_t - (D_t S_{t-1})^\top k_t)^\top), & \text{delta family,} \\ (\bar{S}_t,\ k_t(y_t - \bar{S}_t^\top a_t)^\top), & \text{GDN-2 / Scalar-Q / CQFA.} \end{cases} \tag{34}$$

The delta family uses $D_t = I$ for DeltaNet, $D_t = \alpha_t I$ for Gated DeltaNet, and diagonal D_t for KDA. Q-Delta is recovered with $P = \alpha_t S_{t-1}$ and $B = \beta_t k_t[v_t - P^\top(k_t + \lambda_t q_t)]^\top$. These substitutions give the local-objective view of GDN-2 and Q-Delta (Hatamizadeh et al., 2026; Park et al., 2026) in a common state orientation.

For CQFA specifically, $r_t^q = y_t - \bar{S}_t^\top a_t$ is evaluated at the prior and held fixed in the objective. Therefore

$$\begin{aligned} \mathcal{L}_t(S) &= \|S - \bar{S}_t\|_F^2 - 2\langle S^\top k_t, r_t^q\rangle, \\ \nabla_S\mathcal{L}_t(S) &= 2(S - \bar{S}_t) - 2k_t(r_t^q)^\top, \qquad S_t = \bar{S}_t + k_t(r_t^q)^\top. \end{aligned} \tag{35}$$

The distinction between prediction and writing addresses is explicit in the corresponding regression gradients:

$$\begin{aligned} \nabla_S \tfrac{1}{2}\|S^\top k_t - v_t\|^2 &= k_t(S^\top k_t - v_t)^\top, \\ \bar{S}_t - \beta_t\nabla_S \tfrac{1}{2}\|S^\top k_t - v_t\|^2\big|_{S=\bar{S}_t} &= \bar{S}_t + \beta_t k_t(v_t - \bar{S}_t^\top k_t)^\top, \\ \nabla_S \tfrac{1}{2}\|S^\top a_t - y_t\|^2 &= a_t(S^\top a_t - y_t)^\top. \end{aligned} \tag{36}$$

The first two lines recover the delta gradient step. CQFA instead retains k_t as its write address while using a_t for prediction, as expressed by Equation (35).

C.3 Residual Energy and Feedback Strength

Fix $\bar{S}_t, y_t$ and write $r_t^0 = y_t - p_t^k$, $u_t = M_t \boldsymbol{\lambda}_t$, and $r_t^q = r_t^0 - u_t$. Expanding the square gives

$$\begin{aligned} \|r_t^q\|^2 &= (r_t^0 - u_t)^\top (r_t^0 - u_t) \\ &= \|r_t^0\|^2 - 2\langle r_t^0, u_t\rangle + \|u_t\|^2, \\ \Delta E_t(u_t) &= \|r_t^0\|^2 - \|r_t^q\|^2 = 2\langle r_t^0, u_t\rangle - \|u_t\|^2. \end{aligned} \tag{37}$$

The gate-dependent form separates channel alignment from interactions between channel contributions:

$$\begin{aligned} \Delta E_t(M_t\boldsymbol{\lambda}_t) &= 2\boldsymbol{\lambda}_t^\top M_t^\top r_t^0 - \boldsymbol{\lambda}_t^\top M_t^\top M_t \boldsymbol{\lambda}_t \\ &= 2\sum_i \lambda_{t,i}\langle m_{t,i}, r_t^0\rangle - \sum_{i,j} \lambda_{t,i}\lambda_{t,j}\langle m_{t,i}, m_{t,j}\rangle. \end{aligned} \tag{38}$$

Thus feedback is governed jointly by alignment with the remaining residual and by reinforcement or cancellation among memory-row contributions.

For a fixed direction u_t, scaling it by $\eta \geq 0$ gives

$$\Delta E_t(\eta u_t) = 2\eta\langle r_t^0, u_t\rangle - \eta^2\|u_t\|^2 = \eta\big(2\langle r_t^0, u_t\rangle - \eta\|u_t\|^2\big). \tag{39}$$

For $u_t \neq 0$ and positive alignment,

$$\Delta E_t(\eta u_t) > 0 \quad \Longleftrightarrow \quad 0 < \eta < \frac{2\langle r_t^0, u_t\rangle}{\|u_t\|^2}. \tag{40}$$

For nonpositive alignment, every $\eta > 0$ gives nonpositive gain; for $u_t = 0$, the gain is zero. The upper endpoint follows from the crossing of the linear alignment term and the quadratic energy term.

C.4 Conditional Residual Contraction

For the same target y_t and addressing vector a_t, define $e_t^- = y_t - \bar{S}_t^\top a_t$ and $e_t^+ = y_t - S_t^\top a_t$. Substituting the rank-one update yields

$$\begin{aligned} S_t^\top a_t &= \big(\bar{S}_t + k_t(e_t^-)^\top\big)^\top a_t = \bar{S}_t^\top a_t + e_t^-(k_t^\top a_t), \\ e_t^+ &= (1 - c_t)e_t^-, \qquad c_t = k_t^\top a_t, \\ \|e_t^+\|^2 &= (1 - c_t)^2\|e_t^-\|^2. \end{aligned} \tag{41}$$

For a nonzero residual, strict contraction is equivalent to $|1 - c_t| < 1$, or $0 < c_t < 2$. At $c_t = 1$ the residual vanishes; the endpoints $c_t = 0, 2$ preserve its norm. A zero residual produces no write.

For CQFA, separate the key contribution and the signed query contributions:

$$c_t = c_t^k + \sum_i \lambda_{t,i} z_{t,i}, \qquad c_t^k = \sum_i b_{t,i} k_{t,i}^2, \qquad z_{t,i} = b_{t,i} k_{t,i} q_{t,i}. \tag{42}$$

For $0 \leq \lambda_{t,i} \leq L$, each term satisfies $L\min(z_{t,i}, 0) \leq \lambda_{t,i} z_{t,i} \leq L\max(z_{t,i}, 0)$. Summing these inequalities gives the exact box envelope

$$c_t^k + L\sum_i \min(z_{t,i}, 0) \;\leq\; c_t \;\leq\; c_t^k + L\sum_i \max(z_{t,i}, 0). \tag{43}$$

The lower endpoint selects $\lambda_{t,i} = L$ on negative $z_{t,i}$ and zero on positive $z_{t,i}$; reversing these choices gives the upper endpoint. Hence all gates in the box satisfy the contraction condition whenever

$$c_t^k + L\sum_i \min(z_{t,i}, 0) > 0, \qquad c_t^k + L\sum_i \max(z_{t,i}, 0) < 2. \tag{44}$$

The measured coefficient c_t gives the condition for each realized gate.

Combining the energy and contraction identities gives

$$\|e_t^+\|^2 = (1 - c_t)^2\big(\|r_t^0\|^2 - \Delta E_t(u_t)\big). \tag{45}$$

The inner term is the residual after forming the mixed prediction, and the outer term is its change under the state correction.

To make the role of changing targets explicit, define the drift between successive prediction problems by

$$\begin{aligned} d_t &= (y_t - y_{t-1}) - S_{t-1}^\top (D_t a_t - a_{t-1}), \\ e_t^- &= e_{t-1}^+ + d_t, \qquad e_t^+ = (1 - c_t)(e_{t-1}^+ + d_t). \end{aligned} \tag{46}$$

This follows by adding and subtracting $y_{t-1} - S_{t-1}^\top a_{t-1}$; the current decay enters through $D_t a_t$. If $|1 - c_j| \le \xi < 1$ and $\|d_j\| \le d_{\max}$ for $j = 2, \dots, t$, repeated substitution and the triangle inequality give

$$\begin{aligned} \|e_t^+\| &\le \xi \|e_{t-1}^+\| + \xi \|d_t\| \\ &\le \xi^{t-1} \|e_1^+\| + \sum_{j=2}^{t} \xi^{t-j+1} \|d_j\| \\ &\le \xi^{t-1} \|e_1^+\| + \frac{\xi(1 - \xi^{t-1})}{1 - \xi} \, d_{\max}. \end{aligned} \tag{47}$$

This is a residual bound under a uniform contraction margin and bounded target drift. The single-step identity requires neither additional assumption.

D Chunkwise Parallelization and Implementation

Consider a chunk of length C with incoming state S_{in} and local indices $i = 1, \dots, C$. Its gates and addressing vectors are computed from the layer inputs. We derive the forward transformation, its reverse-mode derivatives, and their correspondence to the implementation.

D.1 Removing the Diagonal Decay

The sequential computation is

$$\begin{aligned} &\bar{S}_i = D_i S_{i-1}, \qquad r_i^q = y_i - \bar{S}_i^\top a_i, \\ &S_i = \bar{S}_i + k_i (r_i^q)^\top, \qquad a_i = b_i \odot (k_i + \boldsymbol{\lambda}_i \odot q_i), \qquad y_i = w_i \odot v_i. \end{aligned} \tag{48}$$

For strictly positive decay components, define

$$\gamma_i = \bigodot_{j=1}^{i} \boldsymbol{\alpha}_j, \qquad F_i = \mathrm{Diag}(\gamma_i), \qquad F_0 = I, \qquad \widehat{S}_i = F_i^{-1} S_i. \tag{49}$$

Since $D_i F_{i-1} = F_i$, the prediction and state update become

$$\begin{aligned} &\bar{S}_i = F_i \widehat{S}_{i-1}, \qquad r_i^q = y_i - \widehat{S}_{i-1}^\top F_i a_i, \\ &\widehat{S}_i = F_i^{-1} \big(F_i \widehat{S}_{i-1} + k_i (r_i^q)^\top \big) = \widehat{S}_{i-1} + \tilde{k}_i (r_i^q)^\top, \\ &\tilde{k}_i = F_i^{-1} k_i, \qquad \tilde{a}_i = F_i a_i, \qquad \tilde{q}_i = F_i q_i. \end{aligned} \tag{50}$$

Starting from $\widehat{S}_0 = S_{\text{in}}$, this additive recurrence gives

$$\widehat{S}_{i-1} = S_{\text{in}} + \sum_{j<i} \tilde{k}_j (r_j^q)^\top. \tag{51}$$

Substituting it into the residual expresses every dependency through a scalar coefficient $c_{ij} = \tilde{a}_i^\top \tilde{k}_j$.

D.2 Triangular Solution and the Compact Transition

The residual equation at position i is

$$r_i^q + \sum_{j<i} c_{ij} r_j^q = y_i - S_{\text{in}}^\top \tilde{a}_i. \tag{52}$$

The first three positions make the triangular dependence explicit:

$$\begin{aligned} r_1^q &= y_1 - S_{\text{in}}^\top \tilde{a}_1, \\ c_{21} r_1^q + r_2^q &= y_2 - S_{\text{in}}^\top \tilde{a}_2, \\ c_{31} r_1^q + c_{32} r_2^q + r_3^q &= y_3 - S_{\text{in}}^\top \tilde{a}_3. \end{aligned} \tag{53}$$

Stack $\tilde{k}_i, \tilde{a}_i, y_i, r_i^q$ as rows of $\widetilde{K}, \widetilde{E}, Y, R$, respectively. Then

$$B = \text{tril}(\widetilde{E}\widetilde{K}^\top, -1), \qquad \mathcal{T} = I + B, \qquad \mathcal{T}R = Y - \widetilde{E} S_{\text{in}}. \tag{54}$$

Here $\mathcal{T} \in \mathbb{R}^{C \times C}$ is unit lower triangular. As $B^C = 0$, its inverse has the finite expansion

$$\mathcal{T}^{-1} = \sum_{j=0}^{C-1} (-B)^j,$$

$$(I + B) \sum_{j=0}^{C-1} (-B)^j = I + (-1)^{C-1} B^C = I. \tag{55}$$

Thus the system has a unique solution. To separate the incoming state from block-local quantities, solve

$$\mathcal{T}U = Y, \qquad \mathcal{T}W = \widetilde{E}, \qquad R = U - W S_{\text{in}}. \tag{56}$$

The shapes are $U, R, Y \in \mathbb{R}^{C \times d_v}$ and $W, \widetilde{E} \in \mathbb{R}^{C \times d_k}$.

Restoring the decay gives the boundary state

$$\begin{aligned} S_{\text{out}} &= F_C \left(S_{\text{in}} + \sum_{j=1}^{C} \tilde{k}_j (r_j^q)^\top \right) = F_C \big(S_{\text{in}} + \widetilde{K}^\top R \big) \\ &= F_C \big[(I - \widetilde{K}^\top W) S_{\text{in}} + \widetilde{K}^\top U \big]. \end{aligned} \tag{57}$$

The homogeneous recurrence in normalized coordinates has transitions $H_i = I - \tilde{k}_i \tilde{a}_i^\top$. Setting $Y = 0$ in the preceding derivation, for arbitrary S_{in}, gives

$$H_C \cdots H_1 = I - \widetilde{K}^\top \mathcal{T}^{-1} \widetilde{E} = I - \widetilde{K}^\top W. \tag{58}$$

This derives the compact rank-one-product representation underlying the WY/UT organization of delta-rule kernels (Yang et al., 2024b; 2025; Hatamizadeh et al., 2026).

Since readout follows the current write,

$$\begin{aligned} o_i &= S_i^\top q_i = \widehat{S}_i^\top \tilde{q}_i \\ &= S_{\text{in}}^\top \tilde{q}_i + \sum_{j \le i} (\tilde{q}_i^\top \tilde{k}_j) r_j^q. \end{aligned} \tag{59}$$

Stacking outputs as rows and letting M include the diagonal gives

$$O = \widetilde{Q} S_{\text{in}} + \big(\widetilde{Q} \widetilde{K}^\top \odot M \big) R. \tag{60}$$

The prediction system uses $j < i$, whereas the output uses $j \le i$. These two masks reproduce the order of prediction, writing, and readout in the sequential update.

D.3 Decay-Weighted Factors and Implementation

The normalized derivation contains $F_i^{-1} k_i$, but only products with decayed factors are needed. Let $G_i = \sum_{j \le i} \log \boldsymbol{\alpha}_j$ and $d_{ij} = \exp(G_i - G_j)$, with elementwise exponential. The required pairwise factors are

$$\begin{aligned} B_{ij} &= \mathbf{1}[j < i] \sum_{c=1}^{d_k} a_{i,c} k_{j,c} d_{ij,c}, \\ P_{ij} &= s\, \mathbf{1}[j \le i] \sum_{c=1}^{d_k} q_{i,c} k_{j,c} d_{ij,c}, \qquad \mathcal{T} = I + B. \end{aligned} \tag{61}$$

We restore the fixed output scale $s = d_k^{-1/2}$ used in the implementation; the query entering a_i is unscaled. For $0 < \alpha_{i,c} \leq 1$ and $j \leq i$, $G_{i,c} - G_{j,c} \leq 0$. Evaluating the difference before exponentiation forms the causal products without materializing inverse cumulative decays.

Define the remaining row factors by

$$E_i = a_i \odot \exp(G_i), \qquad Q_i^0 = q_i \odot \exp(G_i), \qquad K_i^{\text{tail}} = k_i \odot \exp(G_C - G_i), \qquad f = \exp(G_C). \tag{62}$$

As $E = \widetilde{E}$ and $(K^{\text{tail}})^\top = F_C \widetilde{K}^\top$, Equations (56)–(60) become

$$\begin{aligned} \mathcal{T}[U\ W] &= [Y\ E], \qquad R = U - W S_{\text{in}}, \\ S_{\text{out}} &= \text{Diag}(f) S_{\text{in}} + (K^{\text{tail}})^\top R, \\ O &= s Q^0 S_{\text{in}} + P R. \end{aligned} \tag{63}$$

The coefficients and auxiliary matrices are formed independently for every chunk; only the boundary states connect successive chunks. CQFA enters this decomposition through a_i, preserving the shapes of the triangular system, output multiplication, and recurrent state.

Listing 1 implements these equations in PyTorch. The inputs have sequence and channel as their last two dimensions, with normalized queries and keys and per-token log decay g. The causal mask is applied before exponentiation.

```
def cqfa_chunkwise(q, k, v, g, erase, write, feedback, state,
                   chunk_size=64, scale=None):
    # q is unscaled in feedback; scale is applied only to the output.
    scale = q.shape[-1] ** -0.5 if scale is None else scale
    a, y = erase * (k + feedback * q), write * v
    outputs = []
    for start in range(0, q.shape[-2], chunk_size):
        sl = slice(start, start + chunk_size)
        qc, kc, ac, yc = [x[..., sl, :] for x in (q, k, a, y)]
        G = g[..., sl, :].cumsum(dim=-2)
        n = qc.shape[-2]
        index = torch.arange(n, device=q.device)
        causal = index[:, None] >= index[None, :]

        # Mask before exponentiation: noncausal ratios can overflow.
        delta = G.unsqueeze(-2) - G.unsqueeze(-3)
        decay = delta.masked_fill(~causal[..., None], 0).exp()
        B = (ac.unsqueeze(-2) * kc.unsqueeze(-3) * decay).sum(-1)
        P = (qc.unsqueeze(-2) * kc.unsqueeze(-3) * decay).sum(-1)
        T = torch.eye(n, dtype=q.dtype, device=q.device) + B.tril(-1)
        E = ac * G.exp()
        rhs = torch.cat((yc, E), dim=-1)
        UW = torch.linalg.solve_triangular(
            T, rhs, upper=False, unitriangular=True)
        U, W = UW.split((v.shape[-1], q.shape[-1]), dim=-1)

        R = U - W @ state
        outputs.append((qc * G.exp()) @ state + P.tril() @ R)
        tail = (G[..., -1:, :] - G).exp()
        state = (G[..., -1, :].exp().unsqueeze(-1) * state
                 + (kc * tail).transpose(-1, -2) @ R)
    return torch.cat(outputs, dim=-2) * scale, state
```

Listing 1: Differentiable chunkwise reference for CQFA.

The CUDA path tiles the pairwise factors and uses the GDN-2 state and triangular-transform kernels. Its `Aqk`, `Akk`, `u_wy/w_wy`, `kg`, and `v_new` correspond to P, $\mathcal{T}^{-1}$, U/W, K^{tail}, and R, respectively. The effective direction a is supplied through the kernel argument named `b`. Base-2 log accumulation and `exp2` implement the same decay factors. Listing 2 shows the forward orchestration and the trainable entry; the source files retain the original attribution and license.

```
def cqfa_chunk_forward(q, k, v, log_decay, b, w, lam, state0=None):
    a = fused_erase(q, k, b, lam)
    chunk_size = 64
    scale = k.shape[-1] ** -0.5
    g = chunk_local_cumsum(
        g=log_decay, scale=RCP_LN2, chunk_size=chunk_size)
    W, U, Qg, Kg, Aqk, Tinv = chunk_gdn2_fwd_intra(
        q=q, k=k, v=v, gk=g, b=a, w_gate=w, scale=scale,
        chunk_size=chunk_size, disable_recompute=True)
    states, residuals, state_out = chunk_gated_delta_rule_fwd_h(
        k=Kg, w=W, u=U, gk=g, initial_state=state0,
        output_final_state=True, chunk_size=chunk_size)
    outputs = chunk_gla_fwd_o_gk(
        q=q, v=residuals, g=g, A=Aqk, h=states,
        scale=scale, chunk_size=chunk_size)
    return outputs, state_out


# Use this analytic-backward wrapper for training.
def cqfa_cuda(q, k, v, g, b, w, lam, state0=None):
    a = fused_erase(q, k, b, lam)
    return chunk_query_gdn2(
        q, k, v, g, a, w, initial_state=state0,
        output_final_state=True,
        use_qk_l2norm_in_kernel=False,
        disable_recompute=True)
```

Listing 2: CUDA forward decomposition and the analytic-backward entry.

The reference computation matches the sequential recurrence in outputs, final states, and input gradients in double precision, including nonzero initial states, unequal key/value dimensions, and partial final chunks.

D.4 Backward Through the Triangular System

Write $\overline{X} = \partial \mathcal{J}/\partial X$ for an adjoint; $\overline{S}_{\text{out}}$ includes the gradient from later chunks. Differentiating the output and boundary-state equations gives

$$\begin{aligned} \overline{R} &= P^\top \overline{O} + K^{\text{tail}} \overline{S}_{\text{out}}, \\ \overline{P} &= (\overline{O} R^\top) \odot M, \qquad \overline{Q}^0 = s \overline{O} S_{\text{in}}^\top, \\ \overline{K}^{\text{tail}} &= R \overline{S}_{\text{out}}^\top, \qquad \overline{f} = (\overline{S}_{\text{out}} \odot S_{\text{in}}) \mathbf{1}_{d_v}. \end{aligned} \tag{64}$$

To differentiate $\mathcal{T} R = Y - E S_{\text{in}}$, take its total differential:

$$\mathcal{T}\, dR = dY - dE\, S_{\text{in}} - E\, dS_{\text{in}} - d\mathcal{T}\, R. \tag{65}$$

Solve $\mathcal{T}^\top Z = \overline{R}$. The Frobenius product then becomes

$$\begin{aligned} \langle \overline{R}, dR \rangle_F &= \langle Z, \mathcal{T}\, dR \rangle_F \\ &= \langle Z, dY \rangle_F - \langle Z S_{\text{in}}^\top, dE \rangle_F - \langle E^\top Z, dS_{\text{in}} \rangle_F - \langle Z R^\top, d\mathcal{T} \rangle_F. \end{aligned} \tag{66}$$

Combining the direct state paths with these terms yields

$$\begin{aligned} \overline{Y} &= Z, \qquad \overline{E} = -Z S_{\text{in}}^\top, \qquad \overline{B} = -\operatorname{tril}(Z R^\top, -1), \\ \overline{S}_{\text{in}} &= s (Q^0)^\top \overline{O} + \operatorname{Diag}(f) \overline{S}_{\text{out}} - E^\top Z. \end{aligned} \tag{67}$$

The strict mask follows from the fixed unit diagonal of $\mathcal{T}$.

We give the remaining derivatives explicitly to connect the triangular solution to the gates. Let $g_i^a, g_i^{q,\text{readout}}, g_j^{k,\text{other}}$ denote gradients before differentiating the definition of a_i. From Equations (61)

and (62),

$$
\begin{aligned}
g_i^a &= \exp(G_i) \odot \overline{E}_i + \sum_{j<i} \overline{B}_{ij}\, k_j \odot d_{ij}, \\
g_i^{q,\text{readout}} &= \exp(G_i) \odot \overline{Q}_i^0 + s \sum_{j\le i} \overline{P}_{ij}\, k_j \odot d_{ij}, \\
g_j^{k,\text{other}} &= \exp(G_C - G_j) \odot \overline{K}_j^{\text{tail}} + \sum_{i>j} \overline{B}_{ij}\, a_i \odot d_{ij} + s \sum_{i\ge j} \overline{P}_{ij}\, q_i \odot d_{ij}.
\end{aligned} \tag{68}
$$

For the decay derivatives, define for $i > j$

$$
h_{ij} = \overline{B}_{ij}\, a_i \odot k_j \odot d_{ij} + s\overline{P}_{ij}\, q_i \odot k_j \odot d_{ij}. \tag{69}
$$

Each pair contributes $+h_{ij}$ to G_i and $-h_{ij}$ to G_j. Diagonal output terms have $G_i - G_i = 0$. Adding the direct factor paths gives

$$
\begin{aligned}
\overline{G}_\ell = \overline{E}_\ell \odot E_\ell + \overline{Q}_\ell^0 \odot Q_\ell^0 - \overline{K}_\ell^{\text{tail}} \odot K_\ell^{\text{tail}} + \sum_{j<\ell} h_{\ell j} - \sum_{i>\ell} h_{i\ell} \\
+ \mathbf{1}[\ell = C] \left(\overline{f} \odot f + \sum_i \overline{K}_i^{\text{tail}} \odot K_i^{\text{tail}} \right), \\
g_j^{\log \alpha} = \sum_{\ell \ge j} \overline{G}_\ell.
\end{aligned} \tag{70}
$$

The final line is the reverse cumulative sum induced by $G_\ell = \sum_{j\le\ell} \log \boldsymbol{\alpha}_j$.

Finally, differentiating $a_i = b_i \odot (k_i + \boldsymbol{\lambda}_i \odot q_i)$ and $y_i = w_i \odot v_i$ gives

$$
\begin{aligned}
g_i^q &= g_i^{q,\text{readout}} + b_i \odot \boldsymbol{\lambda}_i \odot g_i^a, & g_i^k &= g_i^{k,\text{other}} + b_i \odot g_i^a, \\
g_i^b &= (k_i + \boldsymbol{\lambda}_i \odot q_i) \odot g_i^a, & g_i^\lambda &= b_i \odot q_i \odot g_i^a, \\
g_i^v &= w_i \odot \overline{y}_i, & g_i^w &= v_i \odot \overline{y}_i.
\end{aligned} \tag{71}
$$

The query therefore receives both readout and feedback gradients. For $\lambda = L\sigma(\ell + \delta_\lambda)$, the projection receives

$$
g^\ell = \boldsymbol{\lambda} \odot \left(\mathbf{1} - \frac{\boldsymbol{\lambda}}{L} \right) \odot g^\lambda. \tag{72}
$$

Scalar-Q first sums g^λ within a head, applies the scalar sigmoid derivative, and distributes the resulting logit gradient uniformly through the mean in Equation (20).

D.5 Computational Cost

For one head and one chunk, pairwise coefficient construction costs $O(C^2 d_k)$, the triangular solves cost $O(C^2(d_k + d_v))$, and the state/output products cost $O(Cd_k d_v + C^2 d_v)$. Multiplying by the T/C chunks gives

$$
\frac{T}{C}\big[O(C^2 d_k) + O(C^2(d_k + d_v)) + O(Cd_k d_v + C^2 d_v)\big] = O\big(Td_k d_v + TC(d_k + d_v)\big). \tag{73}
$$

The implementation that materializes the triangular inverse adds $O(C^3)$ per chunk, giving $O(Td_k d_v + TC(d_k + d_v) + TC^2)$ overall. Both forms are linear in sequence length for fixed head dimensions and chunk size. The recurrent state contains $d_k d_v$ elements per head; training also stores or recomputes chunk-local intermediates. Across all heads, CQFA adds $O(Tr(d+m))$ for the feedback projection and $O(Tm)$ for assembling a_t, with no increase in the triangular-system or boundary-state dimensions.

Throughput measurement. The benchmark uses FP32 weights with BF16 autocast on one RTX PRO 6000 Blackwell Server Edition. Each synchronized wall-clock measurement includes forward computation, fused cross-entropy, backward computation, and gradient clearing, with 16,384 tokens per step. After warm-up, three repeats of ten steps provide the mean and standard deviation; optimizer updates and data loading are outside the timing window. GDN in Figure 4 denotes Gated DeltaNet. The Mamba-3 MIMO path uses rank four and a chunk size of eight to accommodate the device's shared-memory capacity.

E Additional Results and Analyses

E.1 Readout Geometry and Energy

The trained 340M CQFA model is evaluated on 64 fitting and 128 confirmation inputs per source: FineWeb, NQ, and SQuAD 1.1 (Penedo et al., 2024; Kwiatkowski et al., 2019; Rajpurkar et al., 2016). We sample all 18 layers and 8 heads at 15 interval centers after 64 prompt tokens and at the final position. The same CQFA checkpoint is used throughout, with the gate parameterization in Appendix B. Scalar-Q provides the analytic directional control $u_t = \lambda_t p_t^q$.

For each layer/head and readout $x \in \{o, v\}$, stack fitting vectors as rows of Z_x:

$$\mu_x = \frac{1}{n}\sum_{j=1}^{n} z_j^x, \qquad \widehat{Z}_x = Z_x - \mathbf{1}\mu_x^\top, \qquad \widehat{\Sigma}_x = \frac{1}{n}\widehat{Z}_x^\top \widehat{Z}_x = U^x \Lambda^x (U^x)^\top. \tag{74}$$

The first r eigenvectors U_r^o fitted to $\hat{o}$ give held-out coverage

$$C_{x\leftarrow o}(r) = \frac{\sum_j \|(U_r^o)^\top (z_j^x - \mu_x)\|^2}{\sum_j \|z_j^x - \mu_x\|^2}, \qquad z_j^o = \hat{o}_j, \quad z_j^v = \hat{v}_j, \quad x \in \{o, v\}. \tag{75}$$

Each target uses its own fitting mean. Swapping o, v gives the reverse coverage in Table 9; both readouts share a dominant principal space.

Table 9: CQFA geometry on confirmation inputs: rank-16 centered PCA coverage and gated query energy outside the key-readout direction (percentages).

Source	$\mathbb{E}\lvert\cos(\hat{o}, \hat{v})\rvert$	$C_{o\leftarrow o}$	$C_{o\leftarrow v}$	$C_{v\leftarrow v}$	$C_{v\leftarrow o}$	p^q outside p^k
FineWeb	0.594	76.61	75.02	75.56	73.58	67.13
NQ	0.552	83.32	81.18	78.59	76.15	71.33
SQuAD	0.605	73.42	71.47	75.40	73.60	63.12

With $P_j^x = p_j^x (p_j^x)^\top / \|p_j^x\|^2$, aggregate energy ratios are

$$\eta_{q|k} = \frac{\sum_j \|(I - P_j^k) p_j^q\|^2}{\sum_j \|p_j^q\|^2}, \qquad \eta_{u|q} = \frac{\sum_j \|(I - P_j^q) u_j\|^2}{\sum_j \|u_j\|^2}. \tag{76}$$

Scalar-Q has $u_j = \lambda_j p_j^q$ and thus $\eta_{u|q} = 0$. Zero-denominator cases are counted separately.

Table 10 reports non-scalar feedback energy, its uncertainty, and gate dispersion. The positive intervals across all three sources show consistent use of non-scalar feedback directions.

Table 10: CQFA channel control: direction-outside feedback energy with 95% cluster bootstrap intervals, and mean within-head gate standard deviation.

Source	Energy (%) [95% CI]	Gate std.
FineWeb	9.62 [9.39, 9.85]	0.1339
NQ	13.34 [13.06, 13.62]	0.1330
SQuAD	9.05 [8.85, 9.25]	0.1340

E.2 Feedback and Stability Diagnostics

For each source, Table 11 normalizes energy terms by the key-only residual:

$$A = \frac{2\sum_j \langle r_j^0, u_j\rangle}{\sum_j \|r_j^0\|^2}, \qquad B = \frac{\sum_j \|u_j\|^2}{\sum_j \|r_j^0\|^2}, \qquad G = A - B = \frac{\sum_j \Delta E_j}{\sum_j \|r_j^0\|^2}. \tag{77}$$

Intervals use 2,000 cluster bootstrap draws over 128 FineWeb documents, 128 NQ articles, and 14 SQuAD articles, measuring input variation for the fixed model.

Table 11: CQFA energy decomposition (% of key-only residual energy, 95% intervals) and contraction exceptions (% of observations outside $0 < c_t < 2$).

	Energy decomposition (%)			
Source	A	B	G [95% CI]	Exceptions (%)
FineWeb	0.259	0.211	$0.048\ [0.031, 0.065]$	0.000097
NQ	0.353	0.254	$0.100\ [0.086, 0.114]$	0.000058
SQuAD	0.231	0.195	$0.036\ [0.019, 0.051]$	0.000107

Scanning all prompt tokens after position 64 gives 8,257,536 / 56,856,240 / 2,794,320 observations on FineWeb / NQ / SQuAD. The respective exception counts are 8, 33, and 3, all with $c_t < 0$. FP64 replay from FP32 chunk-entry states yields maximum absolute errors of 1.24×10^{-14} (contraction) and 3.59×10^{-14} (energy), and relative readout error 1.91×10^{-5} against native outputs.