IN-CONTEXT REGRET: SEQUENCE MODELS ARE ONLINE LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

The search for post-transformer architectures has converged on *test-time-learning* layers—linear attention, DeltaNet, gated variants, Titans, mesa/ridge layers—each from a hand-chosen objective and update rule. What is missing is a *performance calculus*: a way to say, before training at scale, how good a memory rule is and how it must fail. We supply one. Every causal sequence-mixing layer is an *online learner* over its own key–value stream, and its *in-context regret* organizes the design space: linear attention is Hebbian regularized follow-the-leader ($\Theta(T)$ under bounded predictions), DeltaNet is online gradient descent ($O(\sqrt{T})$), gating is discounting, and the mesa layer is *exactly* the Vovk–Azoury–Warmuth forecaster ($O(d_k \log T)$). Against the *memorizing* comparator that attention estimates, we prove a *memory–regret tradeoff*: any layer whose state passes an m-bit bottleneck suffers per-query recall error at least $h^{-1}\big((1 - m/T_s d_v)_+\big)$, with h the binary entropy—while its regret against the *linear* comparator can be zero. What an architecture pays in state is the size of the comparator class it can chase. The theory is generative: parameter-free and strongly-adaptive online learning yield two layers without learned step-size or time-scale gates, and *regret probes* measure trained networks label-free. On enwik8 from 14M to 340M parameters it pays off twice. The coin-betting layer's gap to the tuned delta rule closes monotonically with scale and flips sign: at 340M it posts the best test bpc of any recurrent rule, at 2–3$\times$ the delta and gated baselines' throughput. And length extrapolation mirrors the hierarchy at every scale—bounded-regret rules evaluate below their training-length bpc at 16$\times$ context, while Hebbian's interference diverges.

1 INTRODUCTION

What computational principle should replace—or explain—attention? The most active answer treats a sequence layer as a small learning system of its own: fast weights updated at test time as tokens stream in—linear attention and fast weight programmers (Katharopoulos et al., 2020; Schlag et al., 2021), the delta-rule family (Yang et al., 2024b; 2025; Peng et al., 2025), test-time training and deep memories (Sun et al., 2025; Behrouz et al., 2025), ridge layers (von Oswald et al., 2023b; 2026), unified as regression over an associative memory (Wang et al., 2026; Liu et al., 2025; Behrouz et al., 2026b). Yet the unifications are *descriptive*, not *predictive*: they identify which objective a layer optimizes, not how good that choice is, what it must fail at, or which knobs are necessary. New architectures still ship with hand-designed gates learned by outer training and justified empirically; concurrent work even argues a broad class of TTT layers collapses to learned linear attention, questioning the online-learning reading altogether (Liu et al., 2026).

The online-learning reading becomes *exact, quantitative, and generative* once we ask: **what is the regret?** A forward pass is a repeated game—receive k_t, predict the value, observe v_t, suffer squared loss, update—and its *in-context regret* is its excess cumulative loss over the best fixed associative map in hindsight (Definition 2.1). This quantity organizes the design space (Table 1):

- **A regret hierarchy for existing layers** (§2–3). Building on the regression unification of Wang et al. (2026), we identify *which classical online algorithm* each update rule is—exactly, not by analogy: the mesa layer *is* the Vovk–Azoury–Warmuth forecaster, $O(d_k \log T)$ (Theorem 3.1); DeltaNet *is* online gradient descent, $O(\sqrt{T})$ (Theorem 3.2); linear attention is Hebbian, up to

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.

Table 1: **The design space, organized by regret.** Each sequence-mixing layer is an online learner over its own key–value stream; the update rule determines the in-context regret (unit keys, protocol of Definition 2.1; "dynamic" = tracking bounds). †: worst case unclipped; $\Theta(T)$ clipped (Prop. B.2). KT-Delta/SA-Ladder (ours, §4) carry no learned step-size or time-scale parameters, only the input filter α_t (ablated in §5.2). $*$: SA-Ladder's fixed-rate read-out is competitive with every rung (Thm. 4.2b); strong adaptivity uses the restart form with block-tuned base steps and a parameter-free meta-rule (Thm. 4.2a).

Layer	Online algorithm	State	Static regret	Gates
Linear attention	Hebbian (FTRL, linear loss)	$d_k d_v$	$\Omega(T^3)^\dagger$	—
DeltaNet	OGD (Widrow–Hoff)	$d_k d_v$	$O(\sqrt{T})$	β_t
Gated DeltaNet	discounted OGD	$d_k d_v$	dynamic	β_t, γ_t
Mesa layer	VAW forecaster (FTRL)	$d_k d_v + d_k^2$	$O(d_k \log T)$	λ
Softmax attention	kernel FTL (Nadaraya–Watson)	$O(T(d_k + d_v))$	nonparametric	—
KT-Delta (ours)	coin betting (KT)	$d_k d_v + 1$	$\tilde{O}(\|U\|\sqrt{T})$, $\forall U$ (Huber)	filter
SA-Ladder-R / -γ (ours)	sleeping-experts HEDGE, horizon prior	$O(M d_k d_v)$	per-rung + SA*	filter

$\Omega(T^3)$ unclipped; gates are discount factors whose tracking role is a theorem; softmax attention is the nonparametric reference (Nadaraya–Watson).

- **A memory–regret tradeoff** (§3), stated against the *memorizing comparator* that attention estimates: any causal layer whose state fits in m bits incurs per-query squared recall error $\geq h^{-1}\big((1 - m/T_s d_v)_+\big)$ on adversarial recall streams; any *matrix* memory $\geq 1 - d_k/T_s$ (Theorem 3.3). The same layers can have near-*zero* linear-comparator regret there (Remark 3.4): the hierarchy of architectures is a hierarchy of comparator classes, and attention's growing state is the price of chasing the largest one.
- **New architectures, derived rather than designed** (§4). If gates are learning rates, online learning says they are *removable*. Coin betting (Orabona & Pál, 2016) yields KT-Delta: a delta rule with no step-size parameters, its per-token learning rate emerging from a wealth recursion (Theorem 4.1). Strongly-adaptive online learning (Daniely et al., 2015) yields SA-Ladder: a geometric ladder of time-scales whose prior-weighted hedged read-out is competitive with every rung at once, and whose restart form with block-tuned base steps is strongly adaptive under a parameter-free meta-rule (Theorem 4.2)—deriving the multi-time-scale gate structure that gated DeltaNet, Mamba and Titans learn empirically.
- **Regret probes** (§5). Regret is defined on the layer's own (k_t, v_t) stream, hence measurable on *trained* networks without task labels: it detects interference pathologies at two orders of magnitude, delineates when layer regret certifies task success, and shows production LLMs exhibit slow ICL regret growth over the measured horizon (Appendix C.4).

Our experiments (§5) check the bounds' *non-asymptotic constants*, trace the memory frontier, and compare trained MQAR and byte-LM models with only the online rule varying. KT-Delta and SA-Ladder match or beat gated counterparts under in-context distribution shift; the core suite reproduces on a laptop CPU.[1] An enwik8 sweep from 14M to 340M (§5.5) then delivers the theory's two scale predictions: the parameter-free KT-Delta *overtakes* the tuned delta rule—best test bpc of any recurrent rule at 340M, at 2–3$\times$ the delta and gated baselines' throughput—and length extrapolation follows the regret hierarchy at every scale, out to 16$\times$ the training context.

2 SEQUENCE MIXING AS ONLINE LEARNING

Setting. Following the modern memory-layer template (Schlag et al., 2021; Yang et al., 2024b; Sun et al., 2025; Wang et al., 2026), a causal sequence-mixing layer projects its input into per-head triples

[1] Anonymous code is included in the supplementary materials.

$(q_t, k_t, v_t) \in \mathbb{R}^{d_k} \times \mathbb{R}^{d_k} \times \mathbb{R}^{d_v}$, maintains a *fast state* S_t, and computes

$$S_t = \mathrm{update}(S_{t-1}, k_t, v_t), \qquad o_t = \mathrm{read}(S_t, q_t), \tag{1}$$

with unit keys and $S_0 = 0$. Everything here concerns update. Strip away the network and (1) is a repeated prediction game between the layer and its own stream: at round t the layer observes k_t, forms the strictly-causal prediction $\hat{v}_t = \mathrm{pred}(S_{t-1}, k_t)$, sees v_t, suffers $\ell_t = \frac{1}{2}\|\hat{v}_t - v_t\|^2$, and updates. This is online regression over the key–value stream, and it licenses the classical yardstick:

Definition 2.1 (In-context regret)**.** *Fix a comparator class $\mathcal{F}$ of maps $\mathbb{R}^{d_k} \to \mathbb{R}^{d_v}$. The in-context regret of a layer on a stream $(k_t, v_t)_{t \le T}$ is*

$$R_T(\mathcal{F}) \;=\; \sum_{t=1}^{T} \ell_t \;-\; \min_{f \in \mathcal{F}} \sum_{t=1}^{T} \tfrac{1}{2}\|f(k_t) - v_t\|^2. \tag{2}$$

For an interval $I = [a, b)$, the interval regret R_I *restricts both sums to I; the* dynamic regret *replaces the fixed f by a comparator sequence of bounded variation.*

Two comparator classes. The choice of $\mathcal{F}$ is load-bearing, and we use exactly two. The *linear* class $\mathcal{W}_{\mathrm{lin}} = \{W : \|W\|_F \le D\}$ is the default (written R_T), matching the linear read-outs of fixed-state layers; the upper bounds of §3 are stated against it. The *memorizing* class $\mathcal{F}_{\mathrm{np}}$ of all measurable maps defines $R_T^{\mathrm{np}} \ge R_T$; on distinct-key streams it attains zero loss, so R_T^{np} *equals* cumulative loss there. Softmax attention estimates precisely this comparator (Proposition 2.2); the lower bound of §3.2 is stated against it (Remark 3.4).

Regret is the right abstraction here for a specific reason: the stream is produced by the surrounding network and natural data—neither i.i.d. nor stationary—and regret bounds hold for every individual sequence. It also resolves a conceptual knot in the test-time-learning debate (Liu et al., 2026): whether or not a layer "really" learns, its excess loss over the best fixed associative map is well defined, and that quantity is what our theorems and probes address.

2.1 Existing layers are classical online learners

That these layers all perform regression over an associative memory is the unification of Wang et al. (2026); Liu et al. (2025); Behrouz et al. (2026b); we take it as the starting point. Proposition 2.2 adds the *algorithmic* identification—which classical online-learning algorithm each update rule *is*—because that is what carries regret consequences. The displayed update rules in items 1–4 and 6 give exact algebraic identities (general gated/momentum cases in Appendix A); softmax attention (item 5) enters as the *nonparametric reference*, whose role §3.2 makes precise.

Proposition 2.2 (Correspondences)**.** *With unit keys and the protocol above:*

1. ***Linear attention*** *$S_t = S_{t-1} + v_t k_t^\top$ (Katharopoulos et al., 2020) is regularized follow-the-leader on the* linear *surrogate $-\langle Sk_t, v_t\rangle$ (Hebbian* OGD*). It never corrects: unclipped worst-case regret $\Omega(T^3)$ from a single repeated pair, a tight $\Theta(T)$ under clipped predictions (Prop. B.2).*
2. ***DeltaNet*** *$S_t = S_{t-1} - \beta_t(S_{t-1}k_t - v_t)k_t^\top$ (Yang et al., 2024b) is online gradient descent (Zinkevich, 2003) on ℓ_t—Widrow–Hoff LMS—with per-round step size β_t.*
3. ***Gated DeltaNet*** *(Yang et al., 2025) is* discounted OGD*: decay $S \mapsto \gamma_t S$, then delta-correct—exponential forgetting, the classical device for tracking (Herbster & Warmuth, 1998; Zhang et al., 2018). RWKV-7 and Mamba use related rank-one and selective forgetting mechanisms (Peng et al., 2025; Gu & Dao, 2024).*
4. ***Mesa layer*** *(von Oswald et al., 2023b; 2026), the per-step ridge solution applied to the query, coincides—scored by the online protocol—with the* Vovk–Azoury–Warmuth forecaster *(Vovk, 2001; Azoury & Warmuth, 2001): FTRL with the current key in the regularizer.*
5. ***Softmax attention*** *stores every pair and predicts by Nadaraya–Watson kernel regression: the plug-in estimator of $\mathcal{F}_{\mathrm{np}}$, state growing linearly in t ("kernel* FTL*" names this role, not a regret claim; cf. Theorem 3.3c).*
6. ***TTT / Titans*** *(Sun et al., 2025; Behrouz et al., 2025): mini-batch* OGD *with momentum and weight decay;* ***Longhorn*** *(Liu et al., 2025): the proximal form of item 2 (Appendix A).*

Remark 2.3 (Two footnotes that matter)**.** Read-out vs. prediction: *regret is defined on the strictly-causal* $\mathrm{pred}(S_{t-1}, k_t)$*, layers output* $\mathrm{read}(S_t, q_t)$*; the distinction matters exactly once (*VAW*'s regularized prediction, Theorem 3.1's logarithm).* Negative regret is tracking: *on nonstationary streams an adaptive learner can beat every fixed comparator (Fig. 2); under drift the yardsticks are interval and dynamic regret—what §4 optimizes.*

3 THE REGRET HIERARCHY AND THE MEMORY–REGRET TRADEOFF

Throughout, $\|k_t\| = 1$, $\|v_t\| \le Y$, $\|U\|_F \le D$, gradients bounded by G; proofs in Appendix B. The guarantees below are inherited by the corresponding *layers* verbatim, with constants tight enough to validate numerically (Fig. 3).

3.1 UPPER BOUNDS: WHAT EACH UPDATE BUYS

Theorem 3.1 (Mesa layer = VAW: logarithmic regret)**.** *The mesa layer with ridge $\lambda > 0$, scored by the online protocol, satisfies, for every stream and every $U \in \mathbb{R}^{d_v \times d_k}$,*

$$R_T(U) \;\le\; \frac{\lambda}{2}\|U\|_F^2 \;+\; \frac{d_v Y^2}{2}\,\log\det\Big(I + \tfrac{1}{\lambda}\textstyle\sum_{t\le T} k_t k_t^\top\Big) \;\le\; \tfrac{\lambda}{2}\|U\|_F^2 + \tfrac{d_v d_k Y^2}{2}\log\Big(1 + \tfrac{T}{\lambda d_k}\Big).$$

Theorem 3.2 (Delta rule = OGD: $\sqrt{T}$ regret)**.** *The delta rule with constant step $\beta > 0$ satisfies, with no projection or boundedness assumption on the iterates, $R_T \le \frac{D^2}{2\beta} + \frac{\beta}{2}\sum_{t\le T}\|g_t\|_F^2 \le \frac{D^2}{2\beta} + \frac{\beta G^2 T}{2}$; with steps $\beta_t = D/(G\sqrt{t})$ and projection onto the D-ball, $R_T \le 3DG\sqrt{T}$. The discounted rule with $\gamma \in [\frac{1}{2}, 1)$ and the step of Lemma B.4 tracks with $R_T^\gamma(U) \le 2DG\sqrt{\tau_{\mathrm{eff}}}$ at every horizon, where $\tau_{\mathrm{eff}} = (1-\gamma)^{-1}$; its worst-case* static *regret is $\Omega(T/\tau_{\mathrm{eff}})$ for $T \gg \tau_{\mathrm{eff}}$.*

These formalize the folklore of the DeltaNet line: ridge layers dominate stationary streams, gated layers dominate under drift, neither dominates both—the tension §4 resolves. Hebbian linear attention pays $\Omega(T)$ on repeated keys (Prop. 2.2), the regret-theoretic content of its recall failures (Arora et al., 2024a).

3.2 THE LOWER BOUND: MEMORY BUYS REGRET

The upper bounds share two silent restrictions: a fixed-size matrix state and a *linear* comparator. Softmax attention refuses both, and the next theorem shows the refusal is necessary: against the memorizing comparator, no bounded-state layer has sublinear regret, whatever its update rule. Consider the *store-then-recall* stream: T_s pairs with keys independent of values (distinct a.s.), values uniform on $\{\pm 1/\sqrt{d_v}\}^{d_v}$; after storing, each key is queried once without value feedback. Write Q for this recall phase and restrict regret to Q. The memorizing comparator attains zero loss (Lemma B.1), so average squared recall error is twice per-query nonparametric regret. Let $h^{-1} : [0,1] \to [0, \frac{1}{2}]$ invert the binary entropy.

Theorem 3.3 (Memory–regret tradeoff, against the memorizing comparator)**.** (a) Bit-bounded states. *Any causal algorithm (deterministic or randomized, any update rule) whose state after the store phase is representable in m bits satisfies*

$$\frac{2}{T_s}\,\mathbb{E}\,R_Q^{\mathrm{np}} \;=\; \frac{1}{T_s}\sum_{i=1}^{T_s}\mathbb{E}\,\|\hat{v}_i - v_i\|^2 \;\ge\; h^{-1}\Big(\big(1 - \tfrac{m}{T_s\, d_v}\big)_+\Big).$$

(b) Matrix memories. *Any layer whose answers are linear read-outs $\hat{v}_i = W k_i$ of a single frozen matrix $W \in \mathbb{R}^{d_v \times d_k}$—however W was computed, including in hindsight—satisfies, for isotropic random values, $\frac{1}{T_s}\sum_i \mathbb{E}\|\hat{v}_i - v_i\|^2 \ge \big(1 - {}^{d_k}/_{T_s}\big)_+$.* (c) Attention escapes. *Kernel* FTL *with state $O(T_s(d_k + d_v))$ and sufficiently separated keys attains $R_Q^{\mathrm{np}} = o(T_s)$ on the same streams.*

Remark 3.4 (The comparator class is the currency)**.** *Theorem 3.3 is a statement about R_Q^{np}, and it* must *be: against the linear comparator a fixed-state learner can be essentially optimal on these very streams (the best hindsight matrix itself errs $\ge 1 - {}^{d_k}/_{T_s}$, and the mesa layer tracks it), so $R_Q^{\mathrm{lin}} \approx 0$ coexists with total recall failure. The hierarchy of architectures is a hierarchy of* comparator classes*—fixed-state layers saturate $\mathcal{W}_{\mathrm{lin}}$, attention chases $\mathcal{F}_{\mathrm{np}}$, which by (a)* requires *growing memory. What an architecture pays in state is the size of the comparator class it can chase.*

The parts have different scopes. Part (b) governs *single-matrix linear read-outs* (linear attention, delta variants, KT-Delta, the mesa read; SA-Ladder's hedged mixture and nonlinear deep memories are not covered)—and it is nearly *achieved*: the mesa read approaches it within $1.01\times$ by $T_s{=}8d_k$ (Fig. 4). Part (a) is the architecture-agnostic backstop: any m-bit state, however nonlinear, obeys the entropy bound. Part (c) says why attention survives every efficiency wave: it alone buys recall with memory rather than with loss—a quantitative account of the recall gaps observed at scale (Arora et al., 2024a; Jelassi et al., 2024; Arora et al., 2024b) and of why hybrid stacks retain attention layers.

Corollary 3.5 (Design frontier)**.** *In the (state bits, worst-case average squared recall error) plane, every bit-bounded architecture lies on or above the entropy curve of (a), every single-matrix linear read-out on or above the stronger curve of (b); only added state lowers these frontiers. Within the single-matrix family—the family current subquadratic LMs draw from—architecture search is a one-dimensional tradeoff, not a zoo.*

4 Regret-derived architectures

The hierarchy of §3 is not only a classification; it is a compiler. A delta gate plays two roles, $\beta_t = \alpha_t \cdot \eta_t$: an *input filter* α_t (is this token a training example for the memory at all?) and a *step size* η_t; the decay γ_t selects a forgetting horizon. The filter is the representation's job and can remain learned; the step size and horizon are the *learner's* job—and online learning sets both optimally, per token, with guarantees. Instantiating this gives, to our knowledge, the first sequence layers with provable in-context guarantees and *no learned step-size or time-scale parameters*: a single sigmoid filter $\alpha_t \in [0, 1]$ remains learned, with guarantees on the corresponding filtered convex losses (§5.2 ablates the filter too).

4.1 KT-Delta: the delta rule with betting instead of gates

Coin-betting online learning (Orabona & Pál, 2016) replaces the learning rate by a gambler's wealth; we instantiate the vectorial Krichevsky–Trofimov bettor over the fast-weight matrix. The layer keeps a gradient accumulator $\Theta_t \in \mathbb{R}^{d_v \times d_k}$ and a scalar wealth w_t, and *plays*

$$S_t = \frac{\Theta_{t-1}}{t}\, w_{t-1}, \qquad \Theta_t = \Theta_{t-1} - g_t, \qquad w_t = w_{t-1} - \langle g_t, S_t \rangle, \tag{3}$$

where $g_t = \tilde{r}_t k_t^\top$, $\tilde{r}_t = \alpha_t r_t / \max(1, \|\alpha_t r_t\|)$ is the filtered, clipped residual $r_t = S_t k_t - v_t$, with $\alpha_t = 1$ without the filter and $w_0 = \varepsilon$. Clipping enforces $\|g_t\|_F \le 1$, which keeps the wealth positive (Lemma B.3) and makes the in-context regression *robust* (Huber). Expanding (3) shows it is a delta rule: the update writes the clipped, α_t-filtered error onto the key direction exactly like DeltaNet, but the effective per-token step size $\eta_t = \frac{w_t}{t+1}$ comes from the wealth recursion—no learned step size anywhere, one scalar of extra state per head. And it is parameter-free in the technical sense:

Theorem 4.1 (KT-Delta guarantee)**.** *For every stream with unit keys and every comparator $U \in \mathbb{R}^{d_v \times d_k}$* simultaneously*, the predictions of* (3) *satisfy*

$$R_T^{\mathrm{Hub}}(U) \;\le\; \|U\|_F \sqrt{T\, \log\!\big(1 + \tfrac{24\, T^2 \|U\|_F^2}{\varepsilon^2}\big)} \;+\; \varepsilon,$$

where R^{Hub} uses the convex loss $H_{\alpha_t}(r) = H(\alpha_t r)/\alpha_t$ for $\alpha_t > 0$, $H_0 = 0$, and H is the unit-threshold Huber loss. No step-size, horizon, or scale parameter of the layer depends on $\|U\|$, T, or the data.

The contrast with Theorem 3.2 is the point: OGD needs β tuned to the unknown D and T—why DeltaNet must *learn* $\beta_t(x)$—whereas (3) pays a $\sqrt{\log}$ factor for adapting to all comparator scales at once.

4.2 SA-Ladder: a time-scale ladder with a hedged read-out

Theorem 3.2 leaves a dilemma: the discount γ trades static optimality for tracking, and the right γ depends on the stream's unknown drift—hence learned, token-dependent decay gates. Strongly-adaptive online learning (Daniely et al., 2015; Jun et al., 2017) removes the dilemma wholesale. SA-Ladder maintains $M{+}1$ delta learners on a geometric grid of time-scales, in two forms sharing

one read-out: SA-Ladder-**R** (restart; the theorem's object) runs level m as a fresh delta learner on each dyadic block of length 2^m; SA-Ladder-γ (its smooth constant-state relaxation) replaces restarts by decays $\gamma_m = 1 - 2^{-m}$, $1 \le m \le M$, plus an undiscounted rung. Both combine the levels' strictly-causal predictions with prior-weighted HEDGE (Freund & Schapire, 1997), with a *geometric prior over horizons* $\pi_m \propto \tau_m$. The prior matters when rung losses carry no signal (never-repeated keys): the read-out then defaults to the slowest memory instead of diluting across levels, at a cost of only $\log(1/\pi_m)/\eta$ against faster ones (Lemma B.6). Beyond the shared filter α_t, all quantities are fixed functions—no learned decay, meta-rate, or prior; the state is $M+1$ matrices and scalars per head, $M = O(\log T)$.

Two bounded sets make the meta-level compose with the played loss, both in the implementation: rung predictions are norm-clipped to a ball of radius B before they are mixed or scored (inner delta writes use the raw residuals), and the read-out consumes truncated losses $\min(\ell, \frac{1}{2})$, α-weighted with the filter. Squared loss is convex, so the mixture's loss is at most the hedge-controlled average of the rungs'; on value-bounded streams ($\|v_t\| \le Y$) run with $Y \le B$ and $B + Y \le 1$, projection cannot increase loss and truncation is the identity: the chain closes *for every stream*. The filter gives the same guarantees on α_t-weighted losses; activation rates are measured in §5.1.

Theorem 4.2 (Adaptivity of the ladder mechanism)**.** *(a)* Restart ladder, parameter-free meta-rule. *Under the bounded-loss conditions above, use dyadic coverage through T and block steps $\beta_m = D/(G\sqrt{2^m})$. Combine these specialists with a parameter-free sleeping-experts algorithm—prior-weighted AdaNormalHedge (Luo & Schapire, 2015) with restart resets (implemented as the layer's* ANH *read-out), or CBCE (Jun et al., 2017)—whose per-interval regret is $O(\sqrt{|b|\log(eMT)})$ for* every *block b at once with no meta learning rate. The ladder is then strongly adaptive: for every interval $I \subseteq [T]$ simultaneously, $R_I \le O\big((DG + \sqrt{\log(eMT)})\sqrt{|I|\log(e|I|)}\big)$. (b)* Discount ladder, fixed meta-rate. *At every time T simultaneously: with the Lemma-tuned rung steps $\beta_m = \frac{D}{G}\sqrt{2/(\gamma_m\tau_m)}$ each discounted rung is optimal for its own discount level, $R_T^{\gamma_m}(U) \le 2DG\sqrt{\tau_m}$ (Lemma B.4); and—for* any *rung steps—the prior-weighted average of bounded scores at fixed rate η is within $\log(1/\pi_m)/\eta + \eta\tau_\rho/8$ of every rung under ρ-discounted evaluation (Lemma B.6). The geometric grid places a level within a factor 2 of any drift time-scale up to 2^M.*

Scope. Both read-outs are implemented and measured (§5.1): ANH supplies (a)'s meta-guarantee and, run head-to-head, improves the fixed-rate hedge in every simulated regime; the fixed-rate discount read-out carries (b)'s hedge clause exactly, with an explicit horizon-uniform price for tracking the best rung. The full interval guarantee in (a) uses block-tuned steps; trained rungs write with the exact-recall step η=1 (the choice §5.2 rewards). Both theoretical step schedules are available in the reference learner. The reading is twofold: a single fixed discount cannot cover all interval scales—multi-scale constructions escape the stationary-vs-tracking dilemma, at a $\log T$ state factor—and multi-time-scale decay is therefore not a trick for training to discover but the *derived* solution to nonstationary streams, a first-principles account of the learned forget-gate spectra of gated linear RNNs (Yang et al., 2024a; Gu & Dao, 2024; Yang et al., 2025) and multi-horizon memories (Behrouz et al., 2025; Zancato et al., 2024).

Design principle; efficiency. Both constructions follow one recipe: *take the regret-optimal online algorithm for the failure mode you fear, and read its state recursion as a layer*. FTRL gave the mesa layer (von Oswald et al., 2026); coin betting and geometric covers give KT-Delta and SA-Ladder; optimistic and variance-adaptive learners remain unclaimed. Exact sequence-parallel forms (fused KT scan on GPU) train everything without a Python time loop, parity-tested for values and gradients (Appendix C.1).

5 EXPERIMENTS

We hold backbone, projections, data and optimization fixed and vary *only* the online-learning rule: from theorem validation and the store/recall frontier to trained associative recall and byte-level LM, including a 14M–340M enwik8 sweep (§5.5). Label-free regret probes also cover production LLMs (Appendix C.4). Levels 1–3 reproduce on a 4-core CPU; configurations are in Appendix C.

5.1 THE HIERARCHY AND ITS BOUNDS, MEASURED

Simulating every learner of Table 1 on stationary, switching, and drifting key–value streams recovers the theoretical hierarchy—VAW logarithmic, delta at $\sqrt{T}$, Hebbian off-scale ($R_T \approx 1.9 \times 10^4$), trackers *negative* under drift (Remark 2.3) while the never-forgetting learners accumulate tracking error—and the measured regret of delta, VAW, and KT-Delta sits below the bounds of Theorems 3.2, 3.1 and 4.1 *evaluated with the measured comparator norms and their stated constants* (Figs. 2–3, Table 6, Appendix C.3). The same streams calibrate the guarantees' boundedness devices: KT-Delta's residual clipping (rate 0.0%, max residual $0.75<1$), the ladder's loss truncation at $1/2$ (rate 0.0035%, worst rung loss 0.63) and its prediction projection (rate 0.0%, worst norm 0.82). The inactive KT-Delta clipping makes its Huber and squared losses coincide on these streams; the ladder's truncation is accounted for in Appendix B. The parameter-free ANH meta-rule of Theorem 4.2(a), applied to the same unit-step ladder, beats the fixed-rate hedge in 4/4 regimes ($\mathrm{SA}_{\max}$ 0.81 vs. 0.84). Store/recall streams then realize the frontier of Theorem 3.3 (Fig. 4): kernel-FTL attains the memorizing comparator's zero loss at $O(T_s)$ state while the mesa read converges onto the matrix bound ($1.01\times$ by $T_s{=}8d_k$), delta's stale-write overshoot and Hebbian's divergence lying above it.

5.2 TRAINED RECALL: THE RULE IS THE ONLY VARIABLE

We train 2-block models ($\sim$0.1M parameters, identical backbones, the mixing rule the only variable) on multi-query associative recall (MQAR; Arora et al., 2024a) in two regimes. *Capacity* stores 48 pairs against $d_{\mathrm{head}}{=}32$ over $T{=}144$ tokens—long-horizon retention stress (Appendix C.2). *Rebind* re-binds 75% of keys mid-sequence and queries the *current* binding—an in-context distribution shift that punishes memories that cannot revise. Table 3 (Appendix C.3) reports query accuracy.

Three findings. *First: revision requires the delta family.* Under rebind, every rule that *corrects* (delta, gated, KT-Delta, SA-Ladder: 99–100%) beats every rule that only *accumulates* (mesa 93.0, Hebbian 90.3): averaging and superposition fail exactly as their online algorithms predict. *Second: learned time-scales are initialization-fragile.* On capacity, gated DeltaNet swings 28 points with the decay initialization alone (69.6% at $\gamma_0{\approx}0.95$, 97.8 at 0.999): a mis-set decay erases early bindings and outer training never repairs it (the gradient through 144-step decay products vanishes). Plain delta (99.8) has no decay to mis-set; the ladders' slowest level exists by construction—fixed time-scale structure removes a 28-point initialization risk. *Third: a delta gate entangles filtering, stepping, and horizon selection—and only the first remains learned.* The SA-Ladder ablations dissect this: no input filter, 40.6% (writes at query positions overwrite the memory being read); a *uniform* hedge prior, 29.7% (on never-repeated keys the hedge has no signal and dilutes the slow rung by $1/M$); the geometric prior, learning nothing, recovers 86.5/99.9, and SA-Ladder-R does better still (89.8/99.6). KT-Delta needed none of this care (95.1/99.6 with the filter, 95.5/99.8 without): the wealth-scaled step is self-attenuating, the most plug-and-play of the derived rules.

5.3 REGRET PROBES ON TRAINED NETWORKS

Because in-context regret is defined on the layer's *own* (k, v) stream, it is measurable on a trained network with no task labels: run held-out inputs, record each layer's strictly-causal predictions, compare to hindsight ridge on the same stream. We report the scale-free version restricted to *representation events*—a retention probe—for all 16 trained models of §5.2 (Fig. 5, Appendix C.3). The verdict is *class-level*: Hebbian memories sit two orders of magnitude above every other rule (37–172 vs. ≤ 1.1), the signature of $\Omega(T)$ interference, while ridge and attention memories certify near-zero retention regret exactly where recall is perfect. The probe certifies the *mechanism's* online-learning quality; its connection to task success depends on the probed stream carrying the task's key–value semantics, as in Appendix C.4.

5.4 BYTE-LEVEL LANGUAGE MODELING

Natural data next: byte-level tinyshakespeare with matched 2-block backbones (up to 3 seeds; Table 4, Appendix C.3). Both regret-derived layers train stably exactly as derived, and the two ladders take the *top two places*: SA-Ladder-γ at $2.371 \pm .039$ bpc and SA-Ladder-R at $2.377 \pm .030$, ahead of DeltaNet ($2.431 \pm .006$) and softmax ($2.433 \pm .005$), with pairwise dominance at every seed—

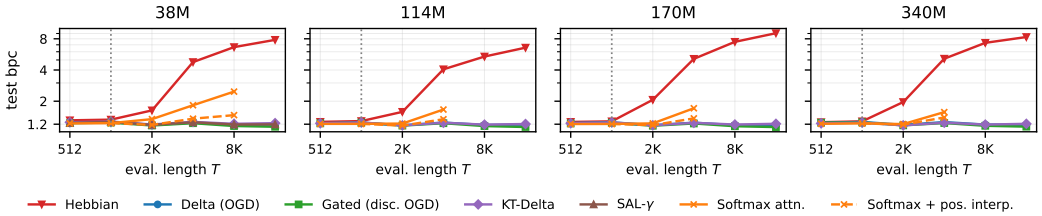

Figure 1: **Length extrapolation at four scales** (trained at T=1024, dotted; no retraining; curves end where evaluation leaves the 24GB budget). Bounded-regret rules stay below their training-length bpc at 16× at every scale; attention degrades at 4× and RoPE position interpolation (dashed) recovers much of that loss while remaining above its own 1K value; Hebbian diverges. SA-Ladder-γ appears in the 38M panel. Log–log.

language is a multi-time-scale stream, and a read-out hedging across horizons with no learned decays is what the theory says it rewards. Hebbian is worst (2.624). §5.5 scales the comparison 700×.

5.5 The hierarchy at scale: 14M–340M

We ran the byte-LM comparison at five scales (Table 2): *small* (d=384, T=512, 2×10^8 tokens; KT-Delta at 3×10^7), then *base* through *xl* (38M/114M/170M/340M parameters, T=1024, 3×10^8 tokens each) on enwik8—identical backbones, the online rule the only variable, three seeds at base and a fixed protocol seed above.

The hierarchy persists across the four larger scales. The ordering the theory prices is measured from base through xl: softmax attention leads in distribution (the comparator §3.2 prices at $O(T)$ state), the best delta baseline follows 0.01–0.02 bpc behind on constant state, and Hebbian's interference floor is unmoved (trailing the family by 0.03–0.09 bpc).

The parameter-free rule overtakes its tuned parent. With the fused Triton scan of Appendix C.1 (84× over the eager loop), KT-Delta runs the full equal-token protocol from base through xl as the fastest recurrent rule measured, at the lowest memory—and its validation gap to DeltaNet closes monotonically with scale, then flips sign: +0.015 bpc at 38M, +0.002 at 114M, −0.001 at 170M, $\mathbf{-0.010}$ at 340M (1.2102 ± .0013 over two seeds vs. 1.2200). At 340M KT-Delta posts the best test bpc of *any* recurrent rule, gated included and both at two seeds (1.2267 ± .0008 vs. gated 1.2295 ± .0002, delta 1.2390), at 2–3× their throughput. Wealth was derived in §4.1 to replace learned step-size gates; at scale it is a strict upgrade—no learned step-size gates, less memory, more speed, better bpc (kernel details in Appendix C.1). SA-Ladder-γ likewise matches its OGD parent at base within seed variation (1.2477 ± .0003 vs. 1.2456 ± .0020) with no time-scale tuning, as designed.

Length extrapolation empirically mirrors the regret hierarchy across four scales. Evaluating the trained checkpoints far beyond their training length, with no retraining (Fig. 1), separates two effects. *Bounded state is what survives the length change.* Every bounded-regret rule still evaluates below its training-length bpc at T=16,384, gated best (1.13–1.14 bpc at 16K), in the same constant memory as at 1K, while attention leaves the 24GB budget past 8K having stored every key and value—the memory–regret tradeoff of §3.2, made physical. *Attention's bpc degradation at* 4× (1.57–1.83) mixes that story with positional extrapolation, so we re-evaluate the same checkpoints with RoPE position interpolation (Chen et al., 2023) (positions rescaled by $1024/T$; dashed). At 4× it recovers 1.829 → 1.357 (38M), 1.662 → 1.340 (114M), 1.710 → 1.366 (170M), 1.570 → 1.394 (340M)—and two things survive: interpolated attention still sits *above* its own 1K bpc (1.229 → 1.357 → 1.468 at 38M) while the gated rule sits below its own (1.227/1.232 vs. 1.357/1.394 at 4×), and interpolation does nothing for the cache. Hebbian is the unambiguous failure: it *diverges* (6.6–9.1 bpc at 16K) on constant state with no positional mechanism involved, exactly as Proposition B.2 prices $\Omega(T)$ interference. We read the grid as a strong empirical correspondence with, not a derivation from, the per-token guarantees.

Table 2: **GPU-scale enwik8** at matched backbones and equal tokens per column (38M–340M parameters, T=1024, 3×10^8; mean $\pm$ sd where seeds repeat: 3 at base, 2 for KT-Delta and Gated at xl; small preset in Table 5). Bold: best validation bpc per column. †: reduced-token.

	validation bpc by scale				at 340M	
Rule	38M	114M	170M	340M	test bpc	tok/s
Linear attn (Hebbian)	1.3209 $\pm$0.0005	1.2649	1.2415	1.2523	1.2555	12k
DeltaNet (OGD)	1.2456 $\pm$0.0020	1.2172	1.1990	1.2200	1.2390	7k
Gated DeltaNet	1.2374 $\pm$0.0052	1.2109	1.1803	1.2048 $\pm$0.0009	1.2295 $\pm$0.0002	6k
KT-Delta (ours)	1.2607 $\pm$0.0012	1.2195	1.1984	1.2102 $\pm$0.0013	1.2267 $\pm$0.0008	15k
SA-Ladder-γ (ours)	1.2477 $\pm$0.0003	–	–	–	–	–
SA-Ladder-R (ours)	1.3133†	–	–	–	–	–
Softmax attention	**1.2260** $\pm$0.0015	**1.1914**	**1.1689**	**1.1912**	1.2086	13k
KT-Delta − *DeltaNet*	+0.0151	+0.0023	−0.0006	−0.0098	−0.0123	2.2×

6 Related work

Test-time-learning architectures and their unifications. Fast weight programmers and linear attention (Schlag et al., 2021; Katharopoulos et al., 2020; Irie et al., 2021) began the modern line; DeltaNet, gating, and expressive state evolution (Yang et al., 2024b;a; 2025; Peng et al., 2025; Grazzi et al., 2025), selective SSMs (Gu & Dao, 2024; Dao & Gu, 2024), TTT and deep memories (Sun et al., 2025; Behrouz et al., 2025; 2026a), and ridge layers (von Oswald et al., 2023b; 2026) followed. Test-time regression (Wang et al., 2026), Longhorn (Liu et al., 2025), and Miras (Behrouz et al., 2026b) unify the family by each layer's objective; Liu et al. (2026) challenge the online-learning narrative; we sharpen all four by attaching *guarantees*, *impossibilities*, *measurements*, and *new constructions*. Closest in spirit, OSDN (Zhou et al., 2026) and Preconditioned DeltaNet (Tumma et al., 2026) import online preconditioning into the delta rule—one cell of Table 1—whereas we chart the table and prove its outer limits.

Recall capacity. Zoology and BASED measure the attention-vs-efficiency recall gap (Arora et al., 2024a;b); Jelassi et al. (2024) prove a copying separation for SSMs. Theorem 3.3 recasts recall capacity as *online regret under a memory budget*, connecting it to memory-bounded online learning (Srinivas et al., 2022; Peng & Zhang, 2023; Marsden et al., 2022), with both a bit bound and the matrix bound our frontier experiment shows is tight.

Online learning theory. We import OGD (Zinkevich, 2003), VAW (Vovk, 2001; Azoury & Warmuth, 2001), tracking and adaptivity (Herbster & Warmuth, 1998; Daniely et al., 2015; Luo & Schapire, 2015; Jun et al., 2017), and coin betting (Orabona & Pál, 2016; Orabona, 2026; Orabona & Tommasi, 2017); KT-Delta is, to our knowledge, its first use as an inner (test-time) layer. ICL-as-learning (von Oswald et al., 2023a; Akyürek et al., 2023) asks what attention *learns to do*; we ask what update rules *provably do*. Park et al. (2025) measure LLM regret at the output level; our probes measure it inside the architecture.

7 Outlook

The enwik8 sweep (§5.5) carries the mechanisms to 340M—the hierarchy's ordering holds across the four larger scales, KT-Delta ends it as the quality-per-throughput frontier of the recurrent rules, and the four-scale grid measures the memory–regret tradeoff at every size; 1B+ long-context training and kernelized comparator classes are next. "Layer = online learner" makes architecture design algorithm selection; each classical learner still unclaimed is a post-transformer architecture with a theorem attached.

AI USE STATEMENT

Generative AI tools were used throughout this project under author direction and review: for research ideation and execution, including formulating theoretical models and mathematical claims, developing proofs, implementing the layers, kernels, synthetic-data generators, visualization scripts and experiment harness, proposing and scripting experiment configurations, and analyzing and interpreting results; for retrieval and discovery of relevant literature; and for drafting sections of the paper and revising and polishing the writing. The authors reviewed the AI-assisted text, references, code, mathematical statements and proofs, and empirical claims, and take full responsibility for the final content of this work.

ETHICS STATEMENT

This is foundational work on sequence-architecture theory using public corpora (enwik8, tinyshakespeare); it involves no human-subject research or collection of personal data and raises no concerns beyond those generic to language modeling.

REPRODUCIBILITY STATEMENT

Proofs and their assumptions are given in Appendix B; experimental configurations and evaluation protocols are described in Appendix C. The anonymous supplementary code includes learner and layer implementations, experiment scripts, result summaries, and tests, together with instructions for regenerating tables and figures and rerunning experiments. Kernel implementations are parity-tested against recurrent references for values and gradients.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *International Conference on Machine Learning (ICML)*, 2024b.

Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. ATLAS: Learning to optimally memorize the context at test time. In *International Conference on Machine Learning (ICML)*, 2026a.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. In *International Conference on Learning Representations (ICLR)*, 2026b.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning (ICML)*, 2015.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg K. H. Franke, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *International Conference on Learning Representations (ICLR)*, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning (ICML)*, 2024.

Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Improved strongly adaptive online learning using coin betting. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

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 (ICLR)*, 2025.

Junchen Liu, Sven Elflein, Or Litany, Zan Gojcic, and Ruilong Li. Test-time training with KV binding is secretly linear attention. In *International Conference on Machine Learning (ICML)*, 2026.

Haipeng Luo and Robert E. Schapire. Achieving all with no parameters: AdaNormalHedge. In *Conference on Learning Theory (COLT)*, 2015.

Annie Marsden, Vatsal Sharan, Aaron Sidford, and Gregory Valiant. Efficient convex optimization requires superlinear memory. In *Conference on Learning Theory (COLT)*, 2022.

Francesco Orabona. Online learning: A modern introduction using convex optimization. arXiv preprint arXiv:1912.13213v10, 2026.

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Chanwoo Park, Xiangyu Liu, Asuman Ozdaglar, and Kaiqing Zhang. Do LLM agents have regret? A case study in online learning and games. In *International Conference on Learning Representations (ICLR)*, 2025.

Binghui Peng and Fred Zhang. Online prediction in sub-linear space. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023.

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 (COLM)*, 2025.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021.

Vaidehi Srinivas, David P. Woodruff, Ziyu Xu, and Samson Zhou. Memory bounds for the experts problem. In *ACM Symposium on Theory of Computing (STOC)*, 2022.

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 *International Conference on Machine Learning (ICML)*, 2025.

Neehal Tumma, Noel Loo, and Daniela Rus. Preconditioned DeltaNet: Curvature-aware sequence modeling for linear recurrences. In *International Conference on Machine Learning (ICML)*, 2026.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning (ICML)*, 2023a.

Johannes von Oswald, Maximilian Schlegel, Alexander Meulemans, Seijin Kobayashi, Eyvind Niklasson, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Blaise Agüera y Arcas, Max Vladymyrov, Razvan Pascanu, and João Sacramento. Uncovering mesa-optimization algorithms in transformers. arXiv preprint arXiv:2309.05858, 2023b.

Johannes von Oswald, Nino Scherrer, Seijin Kobayashi, Luca Versari, Songlin Yang, Sarthak Mittal, Maximilian Schlegel, Kaitlin Maile, Yanick Schimpf, Oliver Sieberling, Alexander Meulemans, Rif A. Saurous, Guillaume Lajoie, Charlotte Frenkel, Razvan Pascanu, Blaise Agüera y Arcas, and João Sacramento. MesaNet: Sequence modeling by locally optimal test-time training. In *International Conference on Learning Representations (ICLR)*, 2026.

Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Ke Alexander Wang, Jiaxin Shi, and Emily B. Fox. Test-time regression: A unifying framework for designing sequence models with associative memory. *Journal of Machine Learning Research*, 27 (169):1–41, 2026.

Bernard Widrow and Marcian E. Hoff, Jr. Adaptive switching circuits. In *1960 IRE WESCON Convention Record, Part 4*, pp. 96–104, 1960.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning (ICML)*, 2024a.

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 (NeurIPS)*, 2024b.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025.

Luca Zancato, Arjun Seshadri, Yonatan Dukler, Aditya Golatkar, Yantao Shen, Benjamin Bowman, Matthew Trager, Alessandro Achille, and Stefano Soatto. B'MOJO: Hybrid state space realizations of foundation models with eidetic and fading memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Chenyu Zhou, Hongpei Li, Yuerou Liu, Jianghao Lin, Dongdong Ge, and Yinyu Ye. OSDN: Improving delta rule with provable online preconditioning in linear attention. arXiv preprint arXiv:2605.13473, 2026.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pp. 928–936, 2003.

A EXTENDED CORRESPONDENCES

This appendix records the layer ↔ online-learner identifications of Proposition 2.2 in update-rule form, together with the members of the family not discussed in the main text. Throughout, $g_t = \nabla_S \frac{1}{2}\|Sk_t - v_t\|^2 = (Sk_t - v_t)k_t^\top$ denotes the in-context regression gradient at the current state.

Linear attention / fast weight programmers (Katharopoulos et al., 2020; Schlag et al., 2021). $S_t = S_{t-1} + v_t k_t^\top$. Since $\nabla_S(-\langle Sk_t, v_t\rangle) = -v_t k_t^\top$, this is OGD with step 1 on the *linear* similarity loss, or equivalently FTRL: $S_t = \arg\min_S \sum_{s\le t} -\langle Sk_s, v_s\rangle + \frac{1}{2}\|S\|_F^2$. On the squared loss it never corrects; re-presenting (k, v) yields response $2v$ (Section 5.1).

DeltaNet (Yang et al., 2024b). $S_t = S_{t-1} - \beta_t g_t$ with g_t evaluated at S_{t-1}: exactly OGD/LMS (Widrow & Hoff, 1960; Zinkevich, 2003). The token-dependent $\beta_t = \sigma(w_\beta^\top x_t)$ is a *learned step-size schedule*.

Gated DeltaNet and related forgetting mechanisms. Gated DeltaNet (Yang et al., 2025) uses $S_t = \gamma_t S_{t-1}(I - \beta_t k_t k_t^\top) + \beta_t v_t k_t^\top = \gamma_t S_{t-1} - \beta_t \nabla \ell_t(\gamma_t S_{t-1})$: *discounted* OGD (decay, then correct). Mamba's selective decay (Gu & Dao, 2024) supplies per-coordinate forgetting with an additive write; RWKV-7 (Peng et al., 2025) uses data-dependent rank-1 state transitions. The exact discounted-OGD identity above applies to the residual-correcting delta update.

Mesa layer (von Oswald et al., 2023b; 2026). $S_t = \arg\min_S \sum_{s\le t} \frac{1}{2}\|Sk_s - v_s\|^2 + \frac{\lambda}{2}\|S\|_F^2 = A_t P_t$ with $A_t = \sum_{s\le t} v_s k_s^\top$, $P_t = (\lambda I + \sum_{s\le t} k_s k_s^\top)^{-1}$. Scored causally with the current key in the inverse, its prediction $\hat{v}_t = A_{t-1} P_t k_t$ is the VAW forecaster (Vovk, 2001; Azoury & Warmuth, 2001); the layer read $o_t = A_t P_t q_t$ is the plain ridge (mesa) read (Remark 2.3). Sherman–Morrison gives the $O(d_k^2)$ recurrent form; our implementation also provides an exact parallel form via batched per-step solves.

TTT (Sun et al., 2025). Inner mini-batch SGD on $\frac{1}{2}\|f_S(k) - v\|^2$ with linear (TTT-Linear) or MLP fast functions: OGD on a (possibly nonlinear) comparator class; our regret protocol covers the linear case exactly and the nonlinear case through Theorem 3.3(a).

Titans (Behrouz et al., 2025). $M_t = (1 - \alpha_t)M_{t-1} + m_t$, $m_t = \eta_t m_{t-1} - \theta_t g_t$: discounted OGD *with momentum* (heavy-ball); α_t is the discount, η_t the momentum coefficient. The momentum recursion provides an additional adaptation mechanism for correlated streams; its regret analysis depends on the momentum and step-size schedules.

Longhorn (Liu et al., 2025). $S_t = \arg\min_S \|S - S_{t-1}\|_F^2 + \lambda_t\|Sk_t - v_t\|^2$: the *implicit* (proximal) form of OGD, i.e. online proximal point; its closed form is a delta rule with effective step $\lambda_t/(1 + \lambda_t\|k_t\|^2)$.

ATLAS / Miras variants (Behrouz et al., 2026a;b). The Miras taxonomy (attentional bias = loss choice; retention = regularizer/discount choice) maps onto Definition 2.1 by replacing the squared loss and the comparator regularizer; e.g. ℓ_p attentional biases give online ℓ_p regression, and Muon-style preconditioned updates (Behrouz et al., 2026a) alter the gradient geometry. For convex objectives, mirror-descent analysis supplies regret bounds adapted to that geometry and its gradient bounds; the loss also determines whether exp-concavity is available.

Softmax attention. With stored pairs $(k_s, v_s)_{s<t}$ and unit keys, $\hat{v}_t = \sum_{s<t} p_s v_s$, $p \propto \exp(\langle k_t, k_s\rangle/\tau)$: Nadaraya–Watson regression with a spherical softmax kernel—FTL over the non-parametric class of local averagers. Its state is the full prefix: the extreme point of Corollary 3.5.

OSDN / Preconditioned DeltaNet (Zhou et al., 2026; Tumma et al., 2026). Delta rules with online (diagonal) preconditioning: second-order OGD variants; in regret terms they improve constants (and condition-number dependence) within the $O(\sqrt{T})$ cell of Table 1, complementary to the cells this paper adds.

B PROOFS

B.1 PRELIMINARIES

Throughout, $\|k_t\| = 1$, $\ell_t(W) = \frac{1}{2}\|Wk_t - v_t\|^2$, g_t denotes the gradient of the relevant loss at the played state, and all losses are convex in W, so for any comparator U,

$$\ell_t(W_t) - \ell_t(U) \;\le\; \langle g_t, W_t - U\rangle, \tag{4}$$

with $\langle\cdot,\cdot\rangle$ the Frobenius inner product. We repeatedly use $\|g_t\|_F = \|r_t\|\|k_t\| = \|r_t\|$ for rank-one gradients $g_t = r_t k_t^\top$.

B.2 PROOF OF THEOREM 3.1 (MESA = VAW)

Reduction to scalar VAW. Write $U = (u_1, \dots, u_{d_v})^\top$ row-wise and likewise for the layer's state. The loss separates, $\ell_t(W) = \sum_{j=1}^{d_v} \frac{1}{2}(w_j^\top k_t - v_{t,j})^2$, and the mesa prediction separates into d_v scalar predictions sharing the design matrix, each equal to the scalar VAW forecaster

$$\hat{y}_{t,j} = a_{t-1,j}^\top A_t^{-1} k_t, \qquad a_{t-1,j} = \textstyle\sum_{s<t} v_{s,j} k_s, \qquad A_t = \lambda I + \sum_{s\le t} k_s k_s^\top.$$

It therefore suffices to prove the scalar bound

$$R_T(u_j) \;\le\; \frac{\lambda}{2}\|u_j\|^2 + \frac{Y_j^2}{2}\sum_{t=1}^{T} k_t^\top A_t^{-1} k_t, \qquad Y_j = \max_t |v_{t,j}|, \tag{5}$$

and to sum over j using $\sum_j Y_j^2 \le d_v Y^2$ and the log-det lemma below.

Proof of (5). This is the classical analysis of the Vovk–Azoury–Warmuth forecaster (Vovk, 2001; Azoury & Warmuth, 2001); we reproduce the compact potential form (Orabona, 2026, Thm. 7.44) (see also Cesa-Bianchi & Lugosi, 2006, Thm. 11.8). Let $\Phi_t = \min_w \left[\frac{\lambda}{2}\|w\|^2 + \frac{1}{2}\sum_{s\le t}(w^\top k_s - y_s)^2\right]$ denote the optimal regularized cumulative loss. Because the VAW prediction is the minimizer of $w \mapsto \Phi_{t-1}$-objective$(w) + \frac{1}{2}(w^\top k_t)^2$ evaluated at k_t (i.e. the current key enters the quadratic before y_t is revealed), completing the square in the quadratic objectives with $A_t = A_{t-1} + k_t k_t^\top$ yields the per-step inequality

$$\tfrac{1}{2}(\hat{y}_t - y_t)^2 \;\le\; \Phi_t - \Phi_{t-1} \;+\; \tfrac{y_t^2}{2}\, k_t^\top A_t^{-1} k_t.$$

The omitted term is $-\frac{1}{2}\hat{y}_t^2 k_t^\top A_{t-1}^{-1} k_t \le 0$. Summing over t telescopes to $\sum_t \frac{1}{2}(\hat{y}_t - y_t)^2 \le \Phi_T + \sum_t \frac{y_t^2}{2} k_t^\top A_t^{-1} k_t$, and $\Phi_T \le \frac{\lambda}{2}\|u\|^2 + \sum_t \frac{1}{2}(u^\top k_t - y_t)^2$ for every u gives (5).

Log-det lemma. For $A_t = A_{t-1} + k_t k_t^\top$, $k_t^\top A_t^{-1} k_t = 1 - \det(A_{t-1})/\det(A_t) \le \log \frac{\det A_t}{\det A_{t-1}}$, so the sum telescopes to $\log\det(A_T/\lambda) = \log\det\left(I + \frac{1}{\lambda}\sum_t k_t k_t^\top\right) \le d_k \log(1 + T/(\lambda d_k))$, the last step by AM–GM over eigenvalues with $\operatorname{tr}\sum_t k_t k_t^\top = T$. □

B.3 PROOF OF THEOREM 3.2 (DELTA RULE = OGD)

With $W_{t+1} = W_t - \beta g_t$ and any U, $\|W_{t+1} - U\|_F^2 = \|W_t - U\|_F^2 - 2\beta\langle g_t, W_t - U\rangle + \beta^2\|g_t\|_F^2$, so $\langle g_t, W_t - U\rangle \le \frac{\|W_t - U\|^2 - \|W_{t+1} - U\|^2}{2\beta} + \frac{\beta}{2}\|g_t\|^2$. Summing, telescoping, and applying (4) gives $R_T \le \frac{D^2}{2\beta} + \frac{\beta}{2}\sum_t \|g_t\|_F^2$ (constant steps need no projection: the telescoped term uses only $\|W_1 - U\| \le D$). For decreasing steps $\beta_t = D/(G\sqrt{t})$ the varying-step telescoping requires $\sup_t \|W_t - U\| \le 2D$, which projection onto the D-ball supplies. The potential terms sum to at most $2D^2/\beta_T$, while $\frac{1}{2}G^2\sum_t \beta_t \le DG\sqrt{T}$, giving $3DG\sqrt{T}$ (Zinkevich, 2003; Orabona, 2026). Figure 3 evaluates the constant-step bound, which requires no projection. For the stationary-stream lower bound, take scalar $k_t = v_t = 1$ and the played discounted recursion of Lemma B.4. Its limiting prediction is $z_\infty = \gamma\beta/(1 - \gamma + \gamma\beta)$. With $\beta = \Theta(\tau_{\text{eff}}^{-1/2})$, $(1 - z_\infty)^2 = \Theta(1/\tau_{\text{eff}})$, whereas $U = 1$ has zero loss. After the geometric transient this gives $\Omega(T/\tau_{\text{eff}})$ static regret for $T \gg \tau_{\text{eff}}$. □

B.4 Proof of Theorem 3.3 (memory–regret tradeoff)

Lemma B.1 (The memorizing comparator attains zero loss)**.** *On any stream with distinct stored keys (an event of probability one for keys drawn from a continuous distribution) and with every repeated presentation of a key carrying the same value—as in the store-then-recall stream—the map $f(k) = v_i$ for $k = k_i$ (and 0 elsewhere) lies in $\mathcal{F}_{\mathrm{np}}$ and has zero cumulative loss. Hence $R_Q^{\mathrm{np}} = \sum_{t\in Q} \ell_t$ on the recall phase.*

Proposition B.2 (Hebbian rates)**.** *On the stream that presents one fixed unit-norm pair (k, v), $\|v\| = 1$, at every step, unclipped linear attention predicts $\hat{v}_t = (t-1)v$ and suffers $\sum_{t\le T} \frac{1}{2}(t-2)^2 = \Theta(T^3)$ regret against the zero-loss comparator $U = vk^\top$ (both linear and memorizing). If predictions are clipped to a fixed ball of radius $B \ge 1$ and values have norm at most 1, the clipped worst-case regret is $\Theta(T)$.*

Proof. $S_{t-1} = (t-1)vk^\top$ by induction, so $\hat{v}_t = (t-1)v$ and $\ell_t = \frac{1}{2}(t-2)^2$; summing gives the cubic. For the clipped lower bound, take scalar $k_t = 1$ and values alternating $+1, -1$. Hebbian predictions alternate $0, 1$, so clipping at any $B \ge 1$ is inactive. Each two rounds incur loss $\frac{1}{2} + 2$, versus 1 for $U = 0$, giving regret $3T/4$ for even T. Every bounded prediction suffers at most $\frac{1}{2}(B+1)^2$ per round, proving the matching upper bound. □

(a) Bit-bounded states. Let $V = (v_1, \dots, v_{T_s})$ with i.i.d. coordinates $v_{i,j} \in \{\pm a\}$, $a = 1/\sqrt{d_v}$, let K denote the keys and query order, drawn independently of V, let R be the algorithm's internal randomness, and let $Z \in \{0,1\}^m$ be the state after the store phase. The query phase provides no value feedback, so every answer is a (measurable) function $\hat{v}_i = f_i(Z, K, R)$.

Step 1: sign errors lower-bound squared error. Decode $\hat{s} = +1$ for $\hat{v} \ge 0$ and $\hat{s} = -1$ otherwise. For $v \in \{\pm a\}$, a decoded sign error implies $|\hat{v} - v| \ge a$. Hence

$$\frac{1}{T_s}\sum_i \|\hat{v}_i - v_i\|^2 \;\ge\; a^2 \cdot \frac{N}{T_s} \;=\; \frac{N}{T_s d_v}, \qquad N = \#\{(i,j) : \hat{s}_{i,j} \ne \mathrm{sign}(v_{i,j})\}. \tag{6}$$

Step 2: Fano. Since $V \perp (K, R)$ and Z has m bits, $I(V; Z, K, R) = I(V; Z \mid K, R) \le H(Z) \le m$, so $H(V \mid Z, K, R) \ge T_s d_v - m$ (bits). The decoded signs $\hat{s}_{i,j} \in \{-1, +1\}$ are functions of (Z, K, R), so $\sum_{i,j} H(v_{i,j} \mid \hat{s}_{i,j}) \ge H(V \mid \hat{s}) \ge H(V \mid Z, K, R) \ge T_s d_v - m$. By Fano's inequality for binary variables, $H(v_{i,j} \mid \hat{s}_{i,j}) \le h(p_{i,j})$ with $p_{i,j} = \Pr[\hat{s}_{i,j} \ne \mathrm{sign}(v_{i,j})]$. Averaging and using concavity of h, $h\big(\bar{p}\big) \ge \frac{1}{T_s d_v}\sum_{i,j} h(p_{i,j}) \ge 1 - \frac{m}{T_s d_v}$, i.e. $\bar{p} \ge h^{-1}\big((1 - \frac{m}{T_s d_v})_+\big)$. Taking expectations in (6) with $\mathbb{E}N = T_s d_v\, \bar{p}$ completes part (a).

(b) Matrix memories. Condition on the keys and fix any (data-dependent) W. Stack keys as $K \in \mathbb{R}^{T_s \times d_k}$ and values as $V \in \mathbb{R}^{T_s \times d_v}$. Then $\sum_i \|Wk_i - v_i\|^2 = \|KW^\top - V\|_F^2 \ge \min_\Omega \|K\Omega - V\|_F^2 = \|(I - P_K)V\|_F^2$, where P_K projects onto the column space of K, of rank $r \le \min(d_k, T_s)$. Keys are independent of V; each output coordinate has covariance $a^2 I$, so $\mathbb{E}\|(I - P_K)V_{:,j}\|^2 = (T_s - r)\, a^2$. Summing over j and dividing by T_s gives $\mathbb{E}\frac{1}{T_s}\sum_i \|Wk_i - v_i\|^2 \ge 1 - \frac{r}{T_s} \ge \big(1 - \frac{d_k}{T_s}\big)_+$.

(c) Attention escapes. Suppose the stored keys satisfy $\max_{i\ne j}\langle k_i, k_j\rangle \le 1 - \delta$. Kernel FTL with temperature τ answers query k_i with weights $p_j \propto e^{\langle k_i, k_j\rangle/\tau}$, so the total weight off the true index is at most $T_s e^{-\delta/\tau}$, and $\|\hat{v}_i - v_i\| \le 2T_s e^{-\delta/\tau}$ (values are unit-norm). Choosing $\tau \le \delta/\log(T_s/\epsilon)$ makes the error $\le 2\epsilon$; with $\epsilon = o(1)$, $R_Q^{\mathrm{np}} \le 2T_s\epsilon^2 = o(T_s)$. Random unit keys in $\mathbb{R}^{d_k}$ have $\delta = \Omega(1)$ with high probability for $T_s \le e^{cd_k}$. Its state stores all T_s pairs: $\Theta(T_s(d_k + d_v))$ scalars, consistent with the growing memory in part (a). □

B.5 KT-Delta: wealth positivity and Theorem 4.1

Lemma B.3 (Wealth positivity)**.** *Run* (3) *with $w_0 = \varepsilon > 0$ and any gradients with $\|g_t\|_F \le 1$. Then $w_t > 0$ for all t.*

Proof. $S_t = \frac{\Theta_{t-1}}{t} w_{t-1}$ gives $w_t = w_{t-1} - \langle g_t, S_t\rangle = w_{t-1}\big(1 - \frac{1}{t}\langle g_t, \Theta_{t-1}\rangle\big)$. Since $\|\Theta_{t-1}\|_F \le \sum_{s<t}\|g_s\|_F \le t - 1$, $|\langle g_t, \Theta_{t-1}\rangle| \le t - 1 < t$, so the bracket lies in $(0, 2)$ and positivity follows by induction. □

Proof of Theorem 4.1. Let $H(r) = \frac{1}{2}\|r\|^2$ for $\|r\| \le 1$ and $H(r) = \|r\| - \frac{1}{2}$ otherwise. With the filter, set $\ell_t^{\mathrm{Hub}}(W) = H(\alpha_t(Wk_t - v_t))/\alpha_t$ for $\alpha_t > 0$ and $\ell_t^{\mathrm{Hub}} = 0$ for $\alpha_t = 0$. These losses are convex with $\nabla \ell_t^{\mathrm{Hub}}(S_t) = \tilde{r}_t k_t^\top = g_t$: the clipped filtered residual *is* the exact gradient of this Huberized loss. At $\alpha_t = 1$ this is ordinary Huber regression. By (4), $R_T^{\mathrm{Hub}}(U) \le \sum_t \langle g_t, S_t - U\rangle$, the linearized regret of the play sequence (3) against gradients with $\|g_t\|_F \le 1$. The play (3) is the vectorial Krichevsky–Trofimov coin-betting algorithm over the Hilbert space $(\mathbb{R}^{d_v \times d_k}, \langle\cdot,\cdot\rangle)$ with initial wealth ε; by the reward–regret duality and the KT potential bound (Orabona & Pál, 2016, Thm. 3 and Cor. 5) (see also Orabona, 2026, Ch. 13),

$$\textstyle\sum_t \langle g_t, S_t - U\rangle \;\le\; \|U\|_F \sqrt{T \log\!\big(1 + \frac{24T^2\|U\|_F^2}{\varepsilon^2}\big)} + \varepsilon \qquad \text{for all } U \text{ simultaneously.}$$

Combining the two displays proves the theorem. When $\|\alpha_t r_t\| \le 1$, the learner's Huberized loss equals its α_t-weighted squared loss, while $H(\alpha_t r)/\alpha_t \le \alpha_t \|r\|^2/2$ for every comparator residual when $\alpha_t > 0$ (both losses are zero at $\alpha_t = 0$). Thus the bound also controls weighted squared-loss regret whenever the learner stays below the clipping threshold. The experiments of §5.1 confirm the squared-loss regret tracks the bound. □

B.6 SA-Ladder: THEOREM 4.2

We prove the two components stated in §4.2: (a) the restart ladder, combined with a parameter-free sleeping-experts meta-rule, is strongly adaptive over intervals; (b) each discounted rung is optimal for its own discount level and the fixed-rate hedged read-out of the implemented layer is simultaneously competitive with *every* rung under discounted evaluation.

Bounded predictions, bounded losses, and the mixture step. Rung predictions are norm-clipped to a ball of radius B (`SAL_PRED_CLIP`) before they are mixed or scored; the read-out consumes $\tilde{\ell}_{m,t} = \min(\ell_{m,t}, \frac{1}{2}) \in [0, 1]$. The inner delta writes use the raw residuals, preserving the closed forms and Lemma B.4. The layer plays the mixture of *clipped* predictions, $\hat{v}_t = \sum_m p_{m,t}\hat{v}_{m,t}$; Jensen's inequality gives $\ell_t(\hat{v}_t) \le \sum_m p_{m,t}\ell_{m,t}$. If $\|v_t\| \le Y \le B$, projection onto the B-ball cannot increase a rung's squared loss. If also $B + Y \le 1$, every rung loss is at most $\frac{1}{2}(B+Y)^2 \le \frac{1}{2}$, so truncation is the identity. The base-learner bounds, the meta-regret bounds, and Jensen then compose for every stream. With the filter, apply the same argument to $\alpha_t \ell_{m,t}$ and use gradients $\alpha_t g_{m,t}$.

The default $B = 16$ preserves a wide range of memory predictions and retains the meta-guarantees on the bounded scores. More generally, writing $\delta_{m,t} = (\ell_{m,t} - \frac{1}{2})_+$ gives the explicit square-loss comparison

$$\ell_t(\hat{v}_t) - \ell_{m^\star,t} \;\le\; \langle p_t, \tilde{\ell}_t\rangle - \tilde{\ell}_{m^\star,t} + \langle p_t, \delta_t\rangle.$$

The same relation holds after multiplying by α_t or discount weights. The simulations quantify these devices: truncation rate 0.0035%, worst rung loss 0.63; projection rate 0.0%, worst prediction norm 0.82 (§5.1).

(a) Restart ladder with a parameter-free meta-rule. Take $M = \max\{1, \lceil \log_2 T \rceil\}$ and dyadic blocks $I_{m,j} = [j2^m, (j+1)2^m)$, $1 \le m \le M$. Specialist (m, j) runs fresh OGD with $\beta_m = D/(G\sqrt{2^m})$ on its block and sleeps otherwise. Under the bounded-gradient assumption, each complete block b pays at most $DG\sqrt{|b|}$ by the constant-step bound of Theorem 3.2; projection of its predictions does not increase loss under the conditions above.

To identify the implemented restart resets with sleeping AdaNormalHedge, regard every block as a distinct expert. Give expert (m, j) prior $q_{m,j} = \pi_m/\mathcal{Z}_T$, where $\mathcal{Z}_T = \pi_\infty + \sum_m \lceil T/2^m \rceil \pi_m$ also includes the never-restarting expert. The common normalizer cancels from the active prediction weights. A fresh block starts with zero regret and variation; its predecessor sleeps forever, so the same level can reuse its state exactly. For $\pi_m \propto 2^m$ and the top horizon proxy 2^{M+1}, $\log(1/q_{m,j}) = O(\log(eMT))$. The sleeping-expert guarantee of Luo & Schapire (2015, Thm. 3) therefore gives $O(\sqrt{|b| \log(eMT)})$ meta-regret on every block simultaneously, with no meta learning rate. CBCE (Jun et al., 2017) supplies an alternative parameter-free geometric-cover meta-rule.

Any interval I decomposes into $O(\log(e|I|))$ complete dyadic blocks and at most two boundary rounds (Daniely et al., 2015). The latter cost at most 1 in total under the bounded-loss conditions.

Summing base and meta bounds over the cover, using $\sum_b \sqrt{|b|} \le \sqrt{\#b\,|I|}$, gives

$$R_I \;\le\; O\big((DG + \sqrt{\log(eMT)})\sqrt{|I|\log(e|I|)}\big) \qquad \text{for every } I \text{ simultaneously.}$$

(a′) Both read-outs are implemented. The fixed-rate restart read-out updates level log-weights by $-\eta\tilde{\ell}_{m,t}$ and re-injects them at $\log\pi_m + \text{logsumexp}$ of the current weights at block boundaries, with fresh fast-weight state. The parameter-free read-out is prior-weighted AdaNormalHedge with restart resets (`meta='anh'` in both the reference learner and the trained layer, with recurrent/parallel parity tests). Each level carries cumulative signed regret R_m to the mixture and absolute variation C_m, and plays $p_m \propto \pi_m[\Phi(R_m+1, C_m+1) - \Phi(R_m-1, C_m+1)]$, where $\Phi(R, C) = \exp([R]_+^2/(3C))$; fresh blocks reset both statistics. This supplies the meta-regret in (a) for arbitrary bounded rung scores. Block-tuned base steps and dyadic coverage through T give the full strongly-adaptive guarantee; the reference learner provides this instantiation. The trained layers and main simulations use the $\eta = 1$ recall write, whose performance is measured in Table 6.

Measured at the online level, ANH improves on the fixed-rate restart hedge in every regime (regret and $\mathrm{SA}_{\max}$, Table 6), and trained end-to-end it matches it under in-context distribution shift (MQAR rebind 99.96 vs. 99.65). The read-outs offer two useful behaviors: when rung losses carry no signal, the fixed-rate hedge defaults to the horizon prior and favors the slowest memory, while ANH's potential adapts to each specialist's accumulated regret and variation. Near zero regret its potential difference decays as $\Delta(0, C) \sim 1/C$, favoring fresh specialists; the capacity comparison gives 47.7% for ANH. The trained comparisons use the prior-weighted hedge for its retention behavior (§5.2).

(b1) Discounted rungs are optimal at their level. For decay $\gamma \in [\frac{1}{2}, 1)$ define the γ-discounted regret $R_T^\gamma(U) = \sum_{t\le T} \gamma^{T-t}\big(\ell_t(y_t) - \ell_t(U)\big)$ and $\tau = (1-\gamma)^{-1}$.

Lemma B.4 (Discounted OGD)**.** *The rung $y_{t+1} = \gamma(y_t - \beta g_t)$, $y_1 = 0$, with $\|g_t\| \le G$, $\|U\| \le D$ and $\beta = \frac{D}{G}\sqrt{2/(\gamma\tau)}$ guarantees,* at every T *simultaneously, $R_T^\gamma(U) \le 2DG\sqrt{\tau}$.*

Proof. Let $\Phi_t = \|y_t - U\|^2$ and $z_t = \|y_t - \beta g_t - U\|$. Writing $y_{t+1} - U = \gamma(y_t - \beta g_t - U) - (1-\gamma)U$ and using convexity of the square, $(\gamma z + (1-\gamma)D)^2 \le \gamma z^2 + (1-\gamma)D^2$:

$$\Phi_{t+1} \;\le\; \gamma\big(\Phi_t - 2\beta\langle g_t, y_t - U\rangle + \beta^2 G^2\big) + (1-\gamma)D^2.$$

Rearranged, $\langle g_t, y_t - U\rangle \le \frac{\gamma\Phi_t - \Phi_{t+1}}{2\beta\gamma} + \frac{\beta G^2}{2} + \frac{(1-\gamma)D^2}{2\beta\gamma}$. Multiply by γ^{T-t} and sum: the first term telescopes to $\frac{\gamma^T\Phi_1 - \Phi_{T+1}}{2\beta\gamma} \le \frac{D^2}{2\beta\gamma}$, and $\sum_t \gamma^{T-t} \le \tau$ bounds the rest, giving $R_T^\gamma \le \frac{D^2}{\beta\gamma} + \frac{\beta G^2\tau}{2}$ via (4). The stated β and $\gamma \ge \frac{1}{2}$ yield $2DG\sqrt{\tau}$. □

Remark B.5 (The analyzed recursion is gated DeltaNet in play coordinates)**.** *Gated DeltaNet decays then corrects at the decayed state: $W_{t+1} = \gamma W_t - \beta\nabla\ell_t(\gamma W_t)$, and the layer* predicts *with the decayed state $z_t = \gamma W_t$. Substituting, $z_{t+1} = \gamma W_{t+1} = \gamma\big(z_t - \beta\nabla\ell_t(z_t)\big)$: exactly the recursion of Lemma B.4 in the played variable, with the same γ and β. (Our simulation objects and layers both predict at the decayed state, matching this identification.)*

Lemma B.6 (Prior-weighted discounted HEDGE)**.** *Let M experts incur losses $\ell_{m,t} \in [0,1]$, take $\rho \in (0,1)$, let $L_{m,t}^\rho = \sum_{s\le t}\rho^{t-s}\ell_{m,s}$, fix a prior $\pi \in \Delta_M$, and play $p_t(m) \propto \pi_m \exp(-\eta\rho L_{m,t-1}^\rho)$. Then for every T and every $m^\star$ simultaneously,*

$$\sum_t \rho^{T-t}\big(\langle p_t, \ell_t\rangle - \ell_{m^\star,t}\big) \;\le\; \frac{\log(1/\pi_{m^\star})}{\eta} + \frac{\eta\tau_\rho}{8}, \qquad \tau_\rho = (1-\rho)^{-1}.$$

With the uniform prior this is the usual $\log M$ bound; with the geometric horizon prior $\pi_m \propto \tau_m$ the cost against the slowest rung is $O(1)/\eta$ and against rung m at most $O(1 + \log(\tau_{\max}/\tau_m))/\eta$.

Proof. Let $W_t = \sum_m \pi_m e^{-\eta L_{m,t}^\rho}$, so $W_0 = 1$. Since $L_{m,t}^\rho = \rho L_{m,t-1}^\rho + \ell_{m,t}$ and, by Jensen's inequality for the concave map $x \mapsto x^\rho$ under the weights π, $\sum_m \pi_m a_m^\rho \le (\sum_m \pi_m a_m)^\rho$ for $a_m \ge 0$,

$$W_t = \sum_m \pi_m \big(e^{-\eta L_{m,t-1}^\rho}\big)^\rho e^{-\eta\ell_{m,t}} \;\le\; W_{t-1}^\rho\, \mathbb{E}_{p_t}\big[e^{-\eta\ell_{m,t}}\big] \;\le\; W_{t-1}^\rho\, e^{-\eta\langle p_t, \ell_t\rangle + \eta^2/8},$$

where the middle step uses that p_t is exactly the Gibbs measure with weights $\pi_m e^{-\eta\rho L^{\rho}_{m,t-1}}$ (normalizing constant $\sum_m \pi_m e^{-\eta\rho L_{m,t-1}} \le W^{\rho}_{t-1}$ by the same Jensen step), and the last is Hoeffding's lemma. Taking logs, $\log W_t \le \rho \log W_{t-1} - \eta\langle p_t, \ell_t\rangle + \eta^2/8$; unrolling with weights ρ^{T-t} (note $\log W_0 = 0$) and lower-bounding $\log W_T \ge \log \pi_{m^\star} - \eta L^{\rho}_{m^\star,T}$ gives the claim after dividing by η. □

(b2) Composition. Apply Lemma B.6 to the bounded rung scores. At fixed η, their hedged average is within $\log(1/\pi_m)/\eta + \eta\tau_\rho/8$ of every rung under ρ-discounted evaluation, simultaneously at every horizon. In the bounded-loss regime above, Jensen transfers this guarantee to the mixture's squared loss; in general the displayed truncation term gives the corresponding square-loss comparison. Each discounted rung with its Lemma-tuned step is separately $2DG\sqrt{\tau_m}$-optimal under its own γ_m discount (Lemma B.4). The horizons $\{2, 4, \ldots, 2^M\}$ place a rung within a factor 2 of every drift time-scale up to 2^M—the discount-space analogue of the dyadic cover in (a)—alongside the undiscounted rung. The implementation uses $p_t(m) \propto \pi_m e^{-\eta_{\text{code}} L^{\rho}_{m,t-1}}$, exactly Lemma B.6 with $\eta = \eta_{\text{code}}/\rho$. □

C Experimental details and additional results

C.1 Implementation

All learners are implemented twice from one specification: (i) as pure functional online learners over (k, v) streams (float64; used for §5.1), and (ii) as trainable multi-head layers inside a pre-norm residual backbone [embedding → $N\times$(mixer + SwiGLU) → head] where the mixer is the only varying component (float32; used for §5.2–5.4). Every mixer—softmax attention included—receives identical depthwise causal short convolutions (kernel 4) on its $q/k/v$ paths, the standard local-shift pathway of the DeltaNet/GLA/Mamba family (it supplies previous-token access, which MQAR's [key value] layout requires); softmax attention additionally uses RoPE, the standard for its family. Keys and queries are SiLU'd and L2-normalized per head, matching the theory's unit-key assumption. DeltaNet rules carry learned per-token gates $\beta_t = \sigma(w_\beta^\top x_t)$ (and γ_t for the gated variant); KT-Delta and SA-Ladder carry no learned step-size or time-scale parameters—their only learned gate is the input filter α_t (Table 3 counts it). The test suite enforces, among others: exact equality of the mesa recurrence with per-step ridge solutions; causality of every layer; wealth positivity; and the regret-bound sanity checks of Fig. 3.

Exact parallel forms. Training uses sequence-parallel closed forms verified (values and gradients) against the recurrent reference:

- *Delta / gated delta:* the WY-style pseudo-value system $\big(I + \mathrm{tril}_{\text{strict}}(\mathrm{diag}(\beta)\,\Gamma \odot KK^\top)\big)U = \mathrm{diag}(\beta)V$ with $\Gamma_{ts} = \prod_{r=s+1}^{t}\gamma_r$, solved by one batched unit-triangular solve (cf. Yang et al., 2024b); outputs and strictly-causal predictions are masked matmuls against the pseudo-values U.
- *Mesa:* per-step ridge solutions for all t at once via batched $d_k \times d_k$ solves of $(\lambda I + G_t)[\,q_t\ k_t\,]$ with cumulative Grams G_t; the VAW prediction uses the strict value mask, the mesa read the inclusive one.
- SA-Ladder*:* each rung is the $\beta = 1$, constant-γ case of the first item; hedge statistics are discounted cumulative losses computed by one lower-triangular weighting, detached from the graph (the hedge is a no-learn component).
- KT-Delta*:* inherently sequential in its wealth scalar; we reduce each step to a matrix–vector product (Θk_t for prediction) and a rank-one update of Θ, plus $O(d_v)$ scalar work, using $\langle g_t, S_t\rangle = \langle \tilde{r}_t, S_t k_t\rangle$.

On a 4-core CPU these forms give the following per-step training times at a reference configuration (batch 32, T=80, 2 blocks, d=64): softmax 0.18s, Hebbian 0.22s, gated delta 0.24s, delta 0.32s, SA-Ladder 0.54s, mesa 0.83s, KT-Delta 1.2s — against 1.7–3.5s for the recurrent reference on the parallelizable rules.

C.2 CONFIGURATIONS

Level 1 (§5.1). B=16 streams, T=768, d_k=16, d_v=8, keys uniform on the sphere, $\|W^\star\|_F$=1, observation noise 0.1; switching: 6 or 32 equal segments with independent $W^\star$; drifting: Gaussian random walk on the sphere of matrices with step 0.03. Learners: Hebbian (η=1); delta (β=0.5); delta with $\beta_t = 1/\sqrt{t}$; discounted delta (β=0.5, γ=0.97); VAW (λ=1); KT-Delta (ε=1); SA-Ladder (M=6 rungs + undiscounted, β=1, η_{meta}=4, ρ=0.98); kernel-FTL (τ=0.05). Static regret uses the exact prefix ridge (λ=10^{-6}), scored by its unpenalized squared loss; dynamic regret uses per-segment ridge. Bound overlays use each stream's prefix comparator norm and measured value and gradient bounds before averaging. Seeds are 1 for the regime grid, 2 for bound overlays, and 3 for boundedness diagnostics.

Level 2 (§5.1). d_k=32, d_v=16, 8 streams, $T_s \in \{8, \ldots, 256\}$; values $\pm 1/\sqrt{d_v}$; recall scored with the read protocol (Remark 2.3) on the query phase, with state frozen after storage and no value feedback or updates during recall.

Levels 3–4 (§5.2–5.3). Backbone d_{model}=64, 2 heads (d_{head}=32), 2 blocks; AdamW, lr 3×10^{-3} cosine with warmup 50, weight decay 0.01, grad-clip 1, batch 32, 1000 steps, $\sim 10^5$ parameters. MQAR-*capacity*: 64 key and 64 value tokens, 48 pairs, 24 queries (T=144), with n_{pairs}=48 > d_{head}=32. Theorem 3.3(b) bounds a *single continuous matrix* readout; the trained network has 2 heads $\times$ 2 layers of state, a discrete 64-way argmax readout that error-corrects analog noise, and MLPs—these support near-perfect discrete recall by fixed-state rules (in bits, $48 \times 6 = 288$ bits fit easily in 4 float matrices), and the per-matrix frontier is measured directly in §5.1. What the long-T setting does stress end-to-end is *retention over the full horizon*, which is what separates the rules in Table 3. MQAR-*rebind*: 16 pairs, 75% re-bound mid-sequence, 16 queries demanding the current binding (T=88). Final accuracy over 32 fresh batches; probes over 8 held-out batches per model, regret from float64 prefix ridge on each layer's own stream.

Level 5 (§5.4). tinyshakespeare bytes, d_{model}=128, 2 heads, 2 blocks, T=160, batch 16, 1200 steps, AdamW lr 10^{-3}; validation split 10%; bpc over 16 batches.

C.3 ADDITIONAL RESULTS

Level-1 summary and detail. Table 6 is the summary cited in §5.1; Table 7 adds dynamic regret on the switching stream and total losses in all regimes.

Table 3: **Trained MQAR query accuracy** (§5.2; mean of final evaluation over 32 fresh batches; identical backbones, the online rule is the only variable). "Gate params" counts *all* learned gate parameters; "(α only)" marks rules whose sole gate is the input filter. ‡: best over the decay-init sweep $\gamma_0 \in \{0.95, 0.993, 0.999\}$; other rules have no such axis to sweep. Bottom rows: ablations.

Rule	gate params	capacity acc.	rebind acc.
Linear attn (Hebbian)	0	0.997	0.903
DeltaNet (OGD)	260	0.998	**1.000**
Gated DeltaNet‡	520	0.978	0.996
Mesa (VAW)	4	0.999	0.930
KT-Delta (ours)	260 (α only)	0.951	0.996
SA-Ladder-γ (ours)	260 (α only)	0.865	0.999
SA-Ladder-R (ours)	260 (α only)	0.898	0.996
Softmax attention	0	**1.000**	0.997
KT-Delta (ours) – *no filter*	0	0.955	0.998
SA-Ladder-γ (ours) – *no filter*	0	0.406	0.982
SA-Ladder-γ (ours) – *no horizon prior*	260 (α only)	0.297	0.991

Gated decay-initialization sweep. The capacity column for gated DeltaNet in Table 3 reports the best of $\gamma_0 \in \{0.95, 0.993, 0.999\}$ (accuracies 0.696/0.958/0.978;

Table 4: **Byte-level LM** (§5.4): validation bits-per-character on tinyshakespeare at matched backbones and steps (lower is better). Three seeds for delta, both ladders and softmax; one for Hebbian and KT-Delta.

Rule	params	val. bpc
Linear attn (Hebbian)	413,312	2.624
DeltaNet (OGD)	413,828	2.431 $\pm$0.006
KT-Delta (ours)	413,828	2.485
SA-Ladder-γ (ours)	413,828	**2.371** $\pm$0.039
SA-Ladder-R (ours)	413,828	2.377 $\pm$0.030
Softmax attention	413,312	2.433 $\pm$0.005

Table 5: **GPU small preset** (§5.5; d=384, T=512, 2×10^8 tokens, enwik8): the sweep's first scale point. †: reduced-token KT-Delta run predating the fused kernel of Appendix C.1.

Rule	tokens	val. bpc	test bpc	tok/s
Linear attn (Hebbian)	2e8	1.4234	1.4126	250k
DeltaNet (OGD)	2e8	1.3384 $\pm$0.0031	1.3292	139k
Gated DeltaNet	2e8	**1.3278** $\pm$0.0016	1.3252	78k
KT-Delta (ours)	3e7†	1.6788	1.6697	2k
SA-Ladder-γ (ours)	2e8	1.3396 $\pm$0.0014	1.3337	27k
SA-Ladder-R (ours)	2e8	1.3435	1.3384	14k
Softmax attention	2e8	1.3315 $\pm$0.0015	1.3170	178k

`results/exp3b_gated_sweep.json`), so the comparison is against the strongest fair version of the learned-decay baseline; the 28-point spread across initializations is itself the second finding of §5.2.

Probe scatter. Figure 5 is the probe-vs-accuracy scatter discussed in §5.3; per-layer values for every model are in `results/exp4.json`, trajectories in `results/exp3_*.json`.

C.4 The in-context regret of production LLMs

If the online-learning reading of sequence models is right, it should leave a fingerprint at the other end of the stack: the in-context learning of a *trained frontier LLM*, probed purely through its token interface, should trace a regret curve comparable to the classical learners'. We stream scalar linear-regression problems ($v_t = \langle w^\star, k_t \rangle$ + noise, d_k=4, $\|w^\star\|$=2, noise 0.05, T=16, 8 streams) through chat models one example at a time, eliciting a numeric prediction of v_t *before* revealing it (temperature 0, answers parsed from $\leq$ 16 completion tokens), and score exactly the static regret of Definition 2.1 against the same hindsight ridge used everywhere else. Reference learners run on identical streams. Figure 6 shows the result: all three production models exhibit slow regret growth over the evaluated horizon; in-context learning *is* measurably an online learner. At T=16: `gemini-2.5-flash-lite` R_T=4.6, `deepseek-v3` R_T=5.2, `gpt-4o-mini` R_T=8.8, against delta/OGD 2.5, VAW 3.3, Hebbian 33.9 on the identical streams (zero parse failures across 384 predictions): the LLMs sit within a small factor of the classical no-regret learners and well below the Hebbian interference line. We regard this as a proof of concept that regret probes extend beyond our own trained models to black-box systems; the full protocol ($\sim$400 cached API calls) ships with the repository.

C.5 GPU-scale configuration (§5.5)

Scale labels are rounded parameter counts; exact counts are recorded in the run summaries. Single node, 4×RTX4090 (24GB), bf16 autocast (linalg solves in fp32 per autocast policy), AdamW (wd 0.1), peak lr $3 \cdot 10^{-3} \cdot 256/d_{\text{model}}$ with warmup+cosine over the token budget, batch$\times T\times$accum $= 24 \times 512$ (small) / $12 \times 1024 \times 2$ (base), byte vocab 256. Runs use per-GPU lane scheduling, checkpointing and resume. Two provenance notes on Table 2. The Hebbian 114M cell was trained

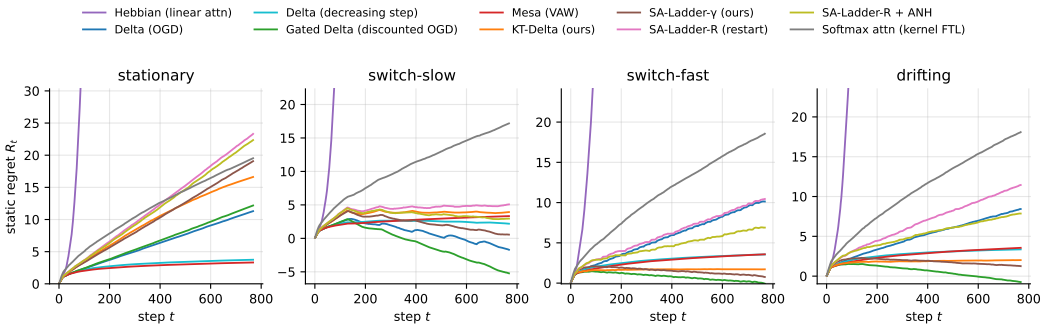

Figure 2: **In-context regret of every learner in Table 1** (§5.1) on stationary, slow-switching, fast-switching, and drifting streams; mean over 16 streams, Hebbian off-scale (axis capped). The hierarchy, the negative-regret tracking phenomenon, and the failure of never-forgetting learners (VAW, kernel-FTL) under nonstationarity are all visible.

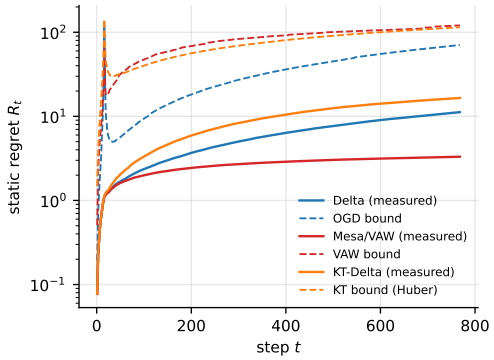

Figure 3: **Bounds with their real constants.** Measured *squared-loss* regret (solid) vs. the non-asymptotic bounds (dashed) of Thms. 3.2, 3.1, 4.1 on the stationary stream (log scale). KT-Delta's guarantee is for the Huber loss (Thm. 4.1); on this stream clipping never activates (rate 0.0%, max residual $0.75 < 1$), so its Huber and squared losses coincide exactly and the comparison is faithful.

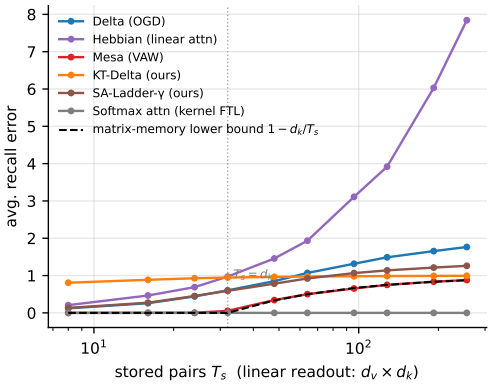

Figure 4: **The memory–regret frontier is real and tight** (§5.1; d_k=32, d_v=16, T_s from 8 to 256). Average recall error vs. number of stored pairs T_s. The mesa read hugs the matrix-memory lower bound $(1 - d_k/T_s)_+$ of Theorem 3.3(b); kernel-FTL (softmax attention) escapes it with $O(T)$ state; Hebbian interference grows unboundedly. Below capacity ($T_s < d_k$) the mesa read has $\leq 10^{-2}$ error while the plain delta rule already pays key-correlation interference; at larger loads delta overshoots (error > 1: stale writes corrupt the readout).

after its block and reports final-checkpoint validation re-evaluated under the block's protocol. The 170M block was additionally retrained end-to-end on different hardware to regenerate evaluation checkpoints (extrapolation and position-interpolation curves in Fig. 1); the retrain reproduces the block's rule ordering exactly, with KT-Delta again beating DeltaNet on test (-0.006), and its summaries ship in the repository alongside the originals. The gated decay kernels compute masked decays and chunked carry ratios in exponent space with an exact clamp at 0, preserving finite values under bf16 arithmetic; values and gradients are parity-tested against the recurrent forms.

Fused and chunked kernels used at scale. KT-Delta runs the Triton scan of `icr/kt_triton.py`: one program per (batch, head), the T-step recurrence and its full adjoint chain inside single forward/ backward kernels with θ on-chip; measured $84\times$ the eager loop (small preset) and 76k tok/s at base, the fastest recurrent rule at the lowest peak memory (3–7

Table 6: **Level-1 summary** (mean over 16 streams). Final static regret on the stationary stream; total in-context loss per regime; and the *measured* strongly-adaptive statistic $\mathrm{SA}_{\max} = \max_I R_I/\sqrt{|I|}$ over dyadic intervals ($|I| \geq 48$), worst regime. Bold: best per column. Reported for every learner and regime; KT-Delta/SA-Ladder reach their cells with no tuned or learned step-size/decay parameters, the property the trained tasks of §5.2 reward.

Learner	R_T (stat.)	L_T (stat.)	L_T (sw-slow)	L_T (sw-fast)	L_T (drift)	$\mathrm{SA}_{\max}$
Linear attn (Hebbian)	1.9×10^4	1.9×10^4	5460.41	2268.80	2523.08	834.77
DeltaNet (OGD, β=.5)	11.30	41.28	47.60	62.75	60.87	0.41
DeltaNet (β_t=$1/\sqrt{t}$)	3.74	33.73	51.49	56.17	55.80	**0.40**
Gated Delta (γ=.97)	12.18	42.16	**44.08**	**52.55**	**51.70**	0.44
Mesa (VAW)	**3.32**	**33.30**	52.63	56.19	55.99	0.42
Softmax attn (kernel FTL)	19.53	49.51	66.48	71.17	70.53	0.70
KT-Delta (ours)	16.61	46.60	53.23	54.33	54.45	0.60
SA-Ladder-γ (ours)	19.07	49.05	49.86	53.37	53.69	0.69
SA-Ladder-R (ours)	23.31	53.30	54.38	63.06	63.89	0.84
SA-Ladder-R+ANH	22.35	52.34	52.25	59.51	60.30	0.81

Table 7: Level-1 detail (mean over 16 streams): per-regime regret against the strongest applicable comparator (dynamic per-segment on switching regimes, static otherwise) and per-regime SA statistic.

	stationary		switch-slow		switch-fast		drifting	
Learner	R_T	SA	R_T^{dyn}	SA	R_T^{dyn}	SA	R_T	SA
Linear attn (Hebbian)	1.9×10^4	834.77	5433.64	213.01	2258.64	86.16	2470.65	96.93
DeltaNet (OGD, β=.5)	11.30	0.41	20.83	0.35	52.59	0.37	8.43	0.34
DeltaNet (β_t=$1/\sqrt{t}$)	3.74	0.24	24.72	0.40	46.01	0.30	3.36	0.30
Gated Delta (γ=.97)	12.18	0.44	17.31	0.26	42.38	0.26	-0.74	0.24
Mesa (VAW)	3.32	0.23	25.86	0.42	46.03	0.30	3.56	0.30
Softmax attn (kernel FTL)	19.53	0.70	39.71	0.62	61.00	0.67	18.09	0.65
KT-Delta (ours)	16.61	0.60	26.46	0.38	44.17	0.28	2.01	0.28
SA-Ladder-γ (ours)	19.07	0.69	23.09	0.34	43.21	0.27	1.26	0.28
SA-Ladder-R (ours)	23.31	0.84	27.61	0.40	52.90	0.39	11.45	0.41
SA-Ladder-R+ANH	22.35	0.81	25.48	0.38	49.35	0.34	7.86	0.35

GiB). The delta family and the ladders use the exact chunked closed forms of `icr/chunked.py` ($O(C^2)$ blocks, state carried across chunks; parity-tested for values and gradients), with per-level activation checkpointing for the ladders at $T \geq 768$ — base-preset SA-Ladder peak memory drops from out-of-memory to 6.4 GiB.

Wealth in finite precision. Wealth positivity (Lemma B.3) is one-sided: nothing upper-bounds the recursion, and on a highly self-predictable stream log-wealth grows linearly within a single sequence. Probing trained checkpoints across all scales and evaluation lengths to $16\times$ places the measured operating range at $w \leq 10^{4.7}$, so the kernel rails fp32 wealth at $[10^{-12}, 10^{12}]$—seven orders beyond that measured range—with valid clamp subgradients: backward reads w_{t-1} from the saved wealth series (the algebraic inverse is wrong on railed steps) and zeroes the wealth adjoint wherever the saved value sits on a rail, a valid zero subgradient at the rails; parity is tested in both regimes. The rails are inactive in every run reported in the paper. The two 340M seeds reach $1.2093/1.2112$ validation and $1.2272/1.2261$ test bpc, agreeing within 0.002; both are reported in Table 2.

C.6 COMPUTE

The CPU levels run on a 4-core VM: levels 1–2 in under two minutes, the trained grids and probes in a few hours per grid. The GPU levels used one $4\times$RTX4090 spot node for $\sim$65 hours across five scale presets ($\approx$ USD 80 of cloud credit in total), covering every training run, the deep length-extrapolation grid, calibration, and kernel validation. The LLM probe of Appendix C.4 uses $\sim 6 \times 10^4$ API tokens.

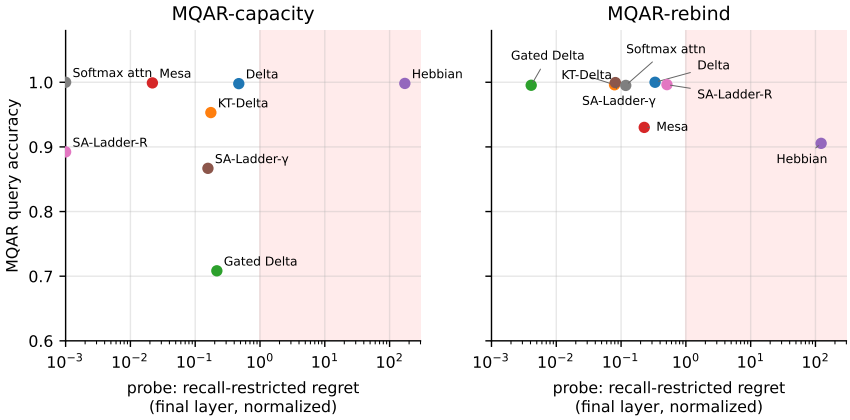

Figure 5: **Regret probes on trained networks.** Recall-restricted, normalized probe regret (final layer, log scale, label-free) vs. MQAR query accuracy, all 16 models. The probe isolates the interference class (Hebbian, shaded region, $\geq 30\times$ every other rule) and certifies near-zero retention regret for ridge/attention memories; §5.3 relates the connection to task accuracy to the semantics of the probed key–value stream.

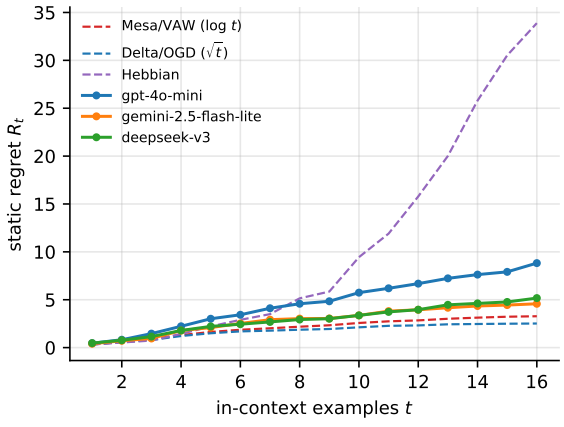

Figure 6: **Static in-context regret of production LLMs** on streamed linear-regression examples, against classical learners on identical streams (dashed). LLM in-context learning is measurably an online learner over the evaluated horizon.