FAST WEIGHTS IN FAST WEIGHTS: TOWARDS BRIDGING PARALLEL AND SEQUENTIAL TEST-TIME LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time training (TTT) layers store context in the weights of a small inner network that is trained by gradient descent as the sequence is read. For hardware efficiency, practical TTT evaluates every gradient in a chunk at the chunk-entrance weights, discarding the effect of earlier within-chunk updates on later gradients. What this mini-batch approximation loses, and which part is worth recovering, is not well understood. Using block-level backward error analysis, we show that the leading-order gap between sequential and mini-batch TTT is the causal interaction $\mathcal{A} = \sum_t H_t \sum_{i<t} g_i$, which is also the leading perturbation of the inner loop's modified vector field. For a two-layer fast-weight MLP, $\mathcal{A}$ splits into four layer-wise interactions with closed forms. This motivates `FF-TTT`, a fast-weights-in-fast-weights design that keeps the first layer as a chunk-level fast weight and makes the output layer a token-level memory, while retaining chunk parallelism. With features fixed, the output layer follows the delta rule, which a chunkwise WY solve computes exactly. The influence of earlier first-layer updates reduces to *feature transport*, a causal linear-attention pass that shifts the key and query features before the solve. For small step sizes, we prove that `FF-TTT` matches the sequential output-layer update through second order, leaving the two first-layer interactions as the entire second-order residual. From 340M to 1B parameters, `FF-TTT` improves on the chunk-parallel baseline `LaCT` in perplexity and average zero-shot accuracy at every scale. On long-context tasks, it improves the average score by 8.5 points over `LaCT` and by 3.8 points over an ablation without feature transport.

1 INTRODUCTION

Test-time training (TTT) turns sequence modeling into online learning (Sun et al., 2020; 2025; Zhang et al., 2026a). A small inner network $f_{\boldsymbol{W}}$ is trained on the sequence as it is read, taking gradient steps on a self-supervised reconstruction loss, and later tokens query the updated network. The weights $\boldsymbol{W}$ are therefore a recurrent state, a set of *fast weights*, and the inner network determines how expressive this memory can be. Replacing a linear inner map with an MLP increases the capacity of the memory, but it also makes the inner loop harder to parallelize. The natural TTT update is token-sequential. When $f_{\boldsymbol{W}}$ is linear and the loss is the squared reconstruction error, the update is the delta rule (Schlag et al., 2021), and its sequential dynamics admit an exact chunkwise-parallel form based on the WY representation (Yang et al., 2024c). No such form is known for deeper fast weights, so practical implementations evaluate all token gradients in a chunk at the fast weights from the start of the chunk (Sun et al., 2020; Zhang et al., 2026a). This mini-batch update is more than an implementation detail. It removes the dependence of each token's gradient on the updates made by earlier tokens in the same chunk, and so changes the learning dynamics of the inner loop. We ask the following research questions. *RQ: What is lost by the removal, and what can be recovered while keeping chunk parallelism?*

We answer the first question with block-level backward error analysis (BEA) (Hairer et al., 2006; Barrett & Dherin, 2021; Smith et al., 2021). To second order in the step size, the endpoints of sequential and mini-batch TTT over a block differ by $\eta^2\mathcal{A}$ with $\mathcal{A} = \sum_t H_t \sum_{i<t} g_i$. Here the prefix gradient $\sum_{i<t} g_i$ represents what earlier tokens have written, and the Hessian H_t converts that displacement into a change in the gradient of token t. The same term is the leading difference between

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.

the modified vector fields of the two block maps. To answer the second question, we specialize to a two-layer fast-weight MLP with parameters $(\boldsymbol{W}_1, \boldsymbol{W}_2)$. There $\mathcal{A}$ splits into four layer-wise interactions, $\mathcal{A}_{11} + \mathcal{A}_{12} + \mathcal{A}_{21} + \mathcal{A}_{22}$, indexed by which layer's gradient is perturbed and which layer made the earlier update. The two interactions that act on the output layer $\boldsymbol{W}_2$ are structured. $\mathcal{A}_{22}$ comes from the dynamics of $\boldsymbol{W}_2$ with the features held fixed, which again follow the delta rule, now with the hidden features as keys. $\mathcal{A}_{21}$ describes how earlier $\boldsymbol{W}_1$ updates change the features that later $\boldsymbol{W}_2$ writes consume. It depends on the $\boldsymbol{W}_1$ trajectory only through a feature shift that causal linear attention can compute.

This structure leads to our *fast-weights-in-fast-weights* construction, which starts from a deliberate choice of timescales. $\boldsymbol{W}_1$ is a chunk-level feature map that takes one mini-batch step per chunk, and $\boldsymbol{W}_2$ is a token-level fast-weight memory nested inside the TTT layer. Under this choice, $\mathcal{A}_{22}$ and $\mathcal{A}_{21}$ describe the within-chunk dynamics of $\boldsymbol{W}_2$, and we recover both. A WY triangular solve runs the $\boldsymbol{W}_2$ delta rule exactly on the chunk's features, which recovers $\mathcal{A}_{22}$ together with every higher-order interaction within $\boldsymbol{W}_2$. Before the solve, *feature transport* maps the virtual $\boldsymbol{W}_1$ prefix into feature space, which recovers $\mathcal{A}_{21}$ without making $\boldsymbol{W}_1$ token-level. Applying the same map to the queries lets intra-block reads see that prefix as well. The resulting update matches the sequential $\boldsymbol{W}_2$ endpoint through second order. We implement the construction as `FF-TTT`, a TTT layer with a SwiGLU fast-weight network, and evaluate it on language modeling from 340M to 1B parameters.

Our contributions are:

- We identify the leading causal interaction removed by chunk-parallel TTT and show that it is also the leading perturbation of the modified inner-loop dynamics (Section 3.1).
- For two-layer fast-weight MLPs, we decompose the causal interaction into four layer-wise interactions and give closed forms and token-space factorizations that make each one directly computable (Section 3.2 and Section C).
- With the first layer kept at chunk granularity by design, we recover both output-layer interactions in parallel. We prove that the combined update matches the sequential output-layer endpoint through second order, and that intra-block reads match the sequential reads through first order up to a single-token term (Section 4 and Section A).
- **`FF-TTT`.** We build a TTT layer on this construction with a SwiGLU fast-weight network and a sign-preserving initialization. At 340M, 760M, and 1B parameters, it improves on `LaCT` (Zhang et al., 2026a) in perplexity and average zero-shot accuracy at every scale. It is competitive with or better than `Transformer`, linear-attention, and state-space baselines (Table 1).

2 Background and Related Work

2.1 Test-Time Training

We begin by revisiting TTT (Sun et al., 2020; 2025; Zhang et al., 2026a), which formulates sequence modeling as online learning driven by a self-supervised reconstruction objective. TTT maintains *fast weights* $\boldsymbol{W}$ that parameterize a neural network $f_{\boldsymbol{W}} : \mathbb{R}^d \to \mathbb{R}^d$. Unlike standard model parameters, which are frozen after training, $\boldsymbol{W}$ acts as a dynamic memory state: it is updated online on the test sequence itself and thereby stores contextual information that subsequent predictions can retrieve. Given an input sequence $\boldsymbol{x} = [x_1, \ldots, x_T]$, each token $x_t \in \mathbb{R}^d$ is linearly projected into a key k_t, value v_t, and query q_t.

Gradient update. At step t, the fast weights are updated by one gradient descent step on a token-dependent self-supervised loss, e.g., the reconstruction loss $\ell(\boldsymbol{W}; x_t) = \|f_{\boldsymbol{W}}(k_t) - v_t\|^2$:

$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} - \eta \nabla_{\boldsymbol{W}} \ell(\boldsymbol{W}_{t-1}; x_t), \tag{1}$$

where η is the inner-loop step size. Intuitively, this step writes the information in x_t into $f_{\boldsymbol{W}}$. The updated function is then queried to produce the output $o_t = f_{\boldsymbol{W}_t}(q_t)$. Since $\boldsymbol{W}_t$ depends on all inputs up to time t, o_t aggregates context from both the current token and its predecessors.

Mini-batch update. In practice, TTT partitions the sequence into blocks of size b and evaluates all gradients within a block at the fast weights from the start of that block for computational efficiency,

$\boldsymbol{W}_{t'}$ with $t' = b\lfloor (t-1)/b \rfloor$. This removes the sequential dependency within each block, so the gradients can be computed in parallel with dense matrix operations, but it makes the inner-loop dynamics differ from those of token-wise online GD. Although this formulation captures the core mechanism of TTT, implementations vary considerably. A substantial body of work explores this design space through different training objectives, optimizers, and parameterizations of $f_{\boldsymbol{W}}$, aiming to improve both modeling quality and efficiency.

Variants of TTT. Beyond this basic form, TTT layers vary along several axes. The *inner objective* has been extended from key–value reconstruction (Sun et al., 2024) to other attentional biases and retention terms (Behrouz et al., 2025b), sliding-window objectives (Behrouz et al., 2025a), locally optimal least-squares updates (von Oswald et al., 2026), implicit online-learning updates (Liu et al., 2024a), and next-token prediction (Tandon et al., 2025; Feng et al., 2026); Wang et al. (2025) give a unifying regression view. The *inner optimizer* has been augmented with momentum and weight decay (Behrouz et al., 2024), Muon (Behrouz et al., 2025a; Zhang et al., 2026a), and multiple steps per token (Siems et al., 2025). The *fast-weight parameterization* ranges from linear and MLP models (Sun et al., 2024) to deep MLP memories (Behrouz et al., 2024), SwiGLU fast weights (Zhang et al., 2026a), and repurposed MLP projections of a pretrained LLM (Feng et al., 2026). Closest to our setting, several works revisit the *block size* itself: Zhang et al. (2026b) use chunks of 2K–1M tokens to raise hardware utilization, and Li et al. (2026) observe that large chunks accelerate training but degrade quality, and decouple training and inference chunk sizes through a hierarchical memory. Neither characterizes how the mini-batch dynamics deviate from sequential GD as a function of b, which is the focus of this work.

2.2 Backward Error Analysis

For gradient descent (GD), $\theta_{k+1} = \theta_k - \eta\,\nabla\ell(\theta_k)$, where θ denotes the model parameters, ℓ the training loss, and η the learning rate. This update is the first-order (explicit) Euler discretization, with step size η, of the gradient flow $\dot\theta(t) = -\nabla\ell(\theta(t))$. The discrete iterates do not lie on the continuous trajectory: each step incurs a local error $\|\theta_1 - \theta(\eta)\| = \mathcal{O}(\eta^2)$, and these errors accumulate to a global error $\|\theta_k - \theta(k\eta)\| = \mathcal{O}(\eta)$ over a fixed time horizon. Backward error analysis (BEA) takes the opposite view (Hairer et al., 2006; Barrett & Dherin, 2021; Smith et al., 2021; Rosca, 2023). Instead of asking how far the iterates drift from the gradient flow, it constructs a nearby *modified* flow $\dot\theta = -\nabla\ell(\theta) + \eta\, f_1(\theta) + \eta^2 f_2(\theta) + \cdots$ whose time-η map agrees with the discrete update to a prescribed order in η, so that the iterates follow the modified flow more closely than the original one. For GD, truncating at first order gives

$$\dot\theta = -\nabla\ell(\theta) \;\underbrace{-\;\frac{\eta}{2}\,H(\theta)\,\nabla\ell(\theta)}_{\text{implicit correction}} + \mathcal{O}(\eta^2), \qquad H(\theta) \coloneqq \nabla^2\ell(\theta), \tag{2}$$

where the correction term equals $\nabla\big(\frac{\eta}{4}\,\|\nabla\ell(\theta)\|^2\big)$, i.e., GD implicitly penalizes the squared gradient norm (Barrett & Dherin, 2021). In this work, we leverage BEA to study the inner-loop online learning dynamics of test-time training (TTT) to characterize the implicit regularizer they induce.

3 Backward Error Analysis of TTT

3.1 BEA Gap of Sequential and Mini-Batch

Mini-batching does more than change how TTT is implemented. It also removes the dependence of each token's gradient on the updates made by earlier tokens in the same block. We use BEA to isolate this missing causal interaction. Without loss of generality, consider a single block of size b with entrance state $\boldsymbol{W}_0$, and write $\ell_t(\boldsymbol{W}) = \ell(\boldsymbol{W}; x_t)$ for $t = 1, \ldots, b$. We identify $\boldsymbol{W}$ with its vectorization and denote

$$g_t(\boldsymbol{W}) = \nabla_{\boldsymbol{W}}\ell_t(\boldsymbol{W}), \qquad H_t(\boldsymbol{W}) = \nabla^2_{\boldsymbol{W}}\ell_t(\boldsymbol{W}). \tag{3}$$

Sequential TTT and mini-batch TTT map the block entrance to

$$\boldsymbol{W}_b^{\mathrm{seq}} = \boldsymbol{W}_0 - \eta\sum_{t=1}^{b} g_t(\boldsymbol{W}_{t-1}^{\mathrm{seq}}), \qquad \boldsymbol{W}_b^{\mathrm{mb}} = \boldsymbol{W}_0 - \eta\sum_{t=1}^{b} g_t(\boldsymbol{W}_0), \tag{4}$$

respectively. The two maps agree to first order in η. Their leading difference therefore comes from how the updates of earlier tokens perturb the gradient of a later token. Hereafter, g_t and H_t without an argument are evaluated at $\boldsymbol{W}_0$.

Assumption 3.1 (Local smoothness)**.** *Each ℓ_t is three times continuously differentiable on a common neighborhood of $\boldsymbol{W}_0$ with uniformly bounded derivatives, and $b\eta$ is small enough that the sequential trajectory remains in this neighborhood.*

Proposition 3.2 (Causal interaction removed by mini-batching)**.** *Let $G_{<t} = \sum_{i<t} g_i$ be the prefix gradient and $\mathcal{A} = \sum_{t=1}^{b} H_t\, G_{<t}$ the causal interaction. Under Assumption 3.1,*

$$\boldsymbol{W}_b^{\mathrm{seq}} - \boldsymbol{W}_b^{\mathrm{mb}} = \eta^2 \mathcal{A} + \mathcal{O}(b^3\eta^3). \tag{5}$$

The proof is given in Appendix A.1. Equation 5 makes the missing mechanism explicit. The term $-\eta G_{<t}$ is the displacement accumulated from earlier tokens in the block, and H_t maps this displacement to the resulting change in the gradient of token t. Mini-batching removes this interaction by evaluating every g_t at the same entrance state $\boldsymbol{W}_0$.

$\mathcal{A}$ also has a direct BEA interpretation. Viewing one block as a time interval of length $b\eta$, let $\widetilde{f}_{\mathrm{seq}}$ and $\widetilde{f}_{\mathrm{mb}}$ denote the modified vector fields whose time-$b\eta$ flows match the two maps in equation 4. Modified-equation matching (Appendix B.1) then gives

$$\widetilde{f}_{\mathrm{seq}} - \widetilde{f}_{\mathrm{mb}} = \frac{\eta}{b}\,\mathcal{A} + \mathcal{O}(\eta^2). \tag{6}$$

Thus $\mathcal{A}$ is not merely an endpoint Taylor remainder. It is the leading-order perturbation that mini-batching induces in the effective test-time learning dynamics. Decomposing these modified fields into conservative and order-dependent parts aids interpretation but is not needed for our algorithm, so we defer it to Appendix B.

3.2 A Layer-wise Decomposition of TTT-MLP

The interaction $\mathcal{A}$ identifies what mini-batching removes, but it lives in the full parameter space. To guide algorithm design, we specialize it to a two-layer fast-weight MLP and ask: *which part of the missing interaction acts on the feature map, which part acts on the output memory, and could we recover some parts while still preserving the parallel efficiency?* We do not consider single-layer TTT-Linear. Under the reconstruction loss, its sequential update is the delta rule, which already admits an exact chunkwise-parallel implementation (Yang et al., 2024c; Liu et al., 2026), so mini-batching is unnecessary there.

For token t in a block, let

$$u_t = \boldsymbol{W}_1 k_t, \qquad h_t = \phi(u_t), \qquad r_t = \boldsymbol{W}_2 h_t - v_t, \qquad \ell_t = \tfrac{1}{2}\left\|r_t\right\|^2, \tag{7}$$

where ϕ is the non-linear activation function and acts elementwise. With $e_t = \phi'(u_t) \odot (\boldsymbol{W}_2^\top r_t)$, the per-token gradients are $g_{1,t} = e_t k_t^\top$ and $g_{2,t} = r_t h_t^\top$. As in Section 3.1, all quantities are evaluated at the block entrance.

Layer-wise decomposition. For $\alpha, \beta \in \{1, 2\}$, let $H_{\alpha\beta,t}$ denote the Hessian block of ℓ_t that maps a perturbation of $\boldsymbol{W}_\beta$ to the induced change in $g_{\alpha,t}$, and define $G_{\beta,<t} = \sum_{i<t} g_{\beta,i}$ and $\mathcal{A}_{\alpha\beta} = \sum_{t=1}^{b} H_{\alpha\beta,t}\, G_{\beta,<t}$. The layer-$\alpha$ component of $\mathcal{A}$ is then $\mathcal{A}_{\alpha 1} + \mathcal{A}_{\alpha 2}$. Repeating the expansion of Proposition 3.2 with layer-wise step sizes η_1, η_2 gives

$$\boldsymbol{W}_{1,b}^{\mathrm{seq}} - \boldsymbol{W}_{1,b}^{\mathrm{mb}} = \eta_1^2 \mathcal{A}_{11} + \eta_1\eta_2 \mathcal{A}_{12} + \mathcal{O}(\eta_{\max}^3 b^3), \tag{8}$$

$$\boldsymbol{W}_{2,b}^{\mathrm{seq}} - \boldsymbol{W}_{2,b}^{\mathrm{mb}} = \eta_1\eta_2 \mathcal{A}_{21} + \eta_2^2 \mathcal{A}_{22} + \mathcal{O}(\eta_{\max}^3 b^3), \tag{9}$$

where $\eta_{\max} = \max\{\eta_1, \eta_2\}$. The diagonal terms are within-layer interactions, and the off-diagonal terms are directed cross-layer couplings. The expansion is proved in Appendix A.2, and closed forms for all four interactions are given in Appendix C. We focus on the two that enter the output memory $\boldsymbol{W}_2$.

Memory interaction $\mathcal{A}_{22}$. Earlier writes to $\boldsymbol{W}_2$ change the residuals seen by later tokens. With the features h_t held fixed, the sequential $\boldsymbol{W}_2$ updates are exactly the delta-rule recurrence. They can therefore be computed exactly, capturing $\mathcal{A}_{22}$ and all its higher-order terms, via the WY representation of Section 4.

Feature-transport interaction $\mathcal{A}_{21}$. Earlier updates to $\boldsymbol{W}_1$ change the features on which later $\boldsymbol{W}_2$ writes are computed. To first order, the prefix update $-\eta_1 G_{1,<t}$ shifts the feature of token t by $\delta h_t = -\eta_1 p_t$, where

$$p_t \coloneqq \phi'(u_t) \odot (G_{1,<t}\, k_t). \tag{10}$$

Since $g_{2,t} = r_t h_t^\top$ and $r_t = \boldsymbol{W}_2 h_t - v_t$, this shift changes $g_{2,t}$ by $(\boldsymbol{W}_2 \delta h_t) h_t^\top + r_t\, \delta h_t^\top$ to first order, so

$$\mathcal{A}_{21} = \sum_{t=1}^{b} \left[(\boldsymbol{W}_2 p_t)\, h_t^\top + r_t\, p_t^\top \right]. \tag{11}$$

This is the key structural observation behind our method: *recovering $\mathcal{A}_{21}$ does not require the sequential $\boldsymbol{W}_1$ trajectory.* It suffices to transport the effect of the $\boldsymbol{W}_1$ prefix into the features consumed by the $\boldsymbol{W}_2$ writes. $\boldsymbol{W}_1$ can thus remain a block-level state while its causal influence on token-level $\boldsymbol{W}_2$ writes is retained. We call $\mathcal{A}_{21}$ the *feature-transport interaction*.

4 Fast Weights in Fast Weights

4.1 Memory interaction

Following the design of Section 3.2, $\boldsymbol{W}_1$ acts as a block-level feature map and $\boldsymbol{W}_2$ acts as a token-level memory. We therefore freeze $\boldsymbol{W}_1$ within each block, update it once per block with the mini-batch step and update $\boldsymbol{W}_2$ sequentially on the fixed features $h_t = \phi(\boldsymbol{W}_1 k_t)$. With token-wise step sizes $\eta_{2,t}$, the $\boldsymbol{W}_2$ update is exactly the delta rule:

$$\boldsymbol{W}_{2,t} = \boldsymbol{W}_{2,t-1}\big(I - \eta_{2,t}\, h_t h_t^\top\big) + \eta_{2,t}\, v_t h_t^\top. \tag{12}$$

The result is a token-scale fast weight $\boldsymbol{W}_2$ nested inside the slower TTT-MLP.

Exact block solve. Each step writes a rank-one update $\boldsymbol{W}_{2,t} = \boldsymbol{W}_{2,t-1} + z_t h_t^\top$ with $z_t = \eta_{2,t}(v_t - \boldsymbol{W}_{2,t-1} h_t)$. Each z_t depends linearly on the earlier writes $z_{i<t}$. Stacking $Z = [z_1, \ldots, z_b]$, $\Phi = [h_1, \ldots, h_b]$, $V = [v_1, \ldots, v_b]$, and $\Lambda = \mathrm{diag}(\eta_{2,1}, \ldots, \eta_{2,b})$, the sequential endpoint is given by a triangular system:

$$Z\, T = (V - \boldsymbol{W}_2 \Phi)\, \Lambda, \qquad T = I + \mathcal{U}(\Phi^\top \Phi\, \Lambda), \qquad \boldsymbol{W}_2^{\mathrm{WY}} = \boldsymbol{W}_2 + Z \Phi^\top, \tag{13}$$

where $\boldsymbol{W}_2$ is the block-entrance state and $\mathcal{U}(\cdot)$ keeps the strictly upper-triangular part. Z is obtained by a single triangular solve, and neither T^{-1} nor the $d \times d$ product of delta-rule factors is ever formed. A proof that this solve reproduces the sequential recurrence exactly, together with the equivalent compact WY form, is given in Appendix A.3.

Connection to BEA. For a constant step $\eta_{2,t} = \eta_2$, expanding $T^{-1} = I - \eta_2 \mathcal{U}(\Phi^\top \Phi) + \cdots$ gives

$$\boldsymbol{W}_2^{\mathrm{WY}} = \boldsymbol{W}_2 - \eta_2 \sum_{t=1}^{b} g_{2,t} + \eta_2^2\, \mathcal{A}_{22} + \mathcal{O}(\eta_2^3 b^3). \tag{14}$$

The solve therefore recovers the memory interaction $\mathcal{A}_{22}$ (Lemma A.5). Unlike a second-order additive correction, it also retains every higher-order term generated by repeated $\boldsymbol{W}_2$ updates on fixed features.

Relation to existing parallel forms. The solve itself is not new. Representing products of rank-one updates with a triangular factor is the classical WY representation (Bischof & Van Loan, 1987). The delta rule as a fast-weight memory is due to Schlag et al. (2021). And equation 13 is the chunkwise delta-rule kernel of Yang et al. (2024c), with the features h_t playing the role of keys. Our contribution is identifying where this kernel applies. The missing $\boldsymbol{W}_2$ dynamics split into a fixed-feature memory interaction $\mathcal{A}_{22}$, which the kernel captures exactly, and a feature-to-memory interaction $\mathcal{A}_{21}$, which it misses. Section 4.2 recovers the latter via feature transport.

4.2 FEATURE TRANSPORT INTERACTION

The fixed-feature WY solve recovers $\mathcal{A}_{22}$ but misses $\mathcal{A}_{21}$: the effect of earlier first-layer updates on the features that later $\boldsymbol{W}_2$ writes consume. In this section, we present how to recover this interaction.

Recurrent form. To first order, the sequential first-layer state seen by token t is $\boldsymbol{W}_1 + \Delta\boldsymbol{W}_{1,t-1}$, where the *virtual prefix* $\Delta\boldsymbol{W}_{1,t-1} = -\eta_1 G_{1,<t}$ accumulates the entrance gradients of earlier tokens. Since $g_{1,t} = e_t k_t^\top$, it obeys a rank-one recurrence, and it shifts the preactivation of token t by δu_t:

$$\Delta\boldsymbol{W}_{1,t} = \Delta\boldsymbol{W}_{1,t-1} + d_t k_t^\top, \qquad \delta u_t = \Delta\boldsymbol{W}_{1,t-1}\, k_t = \sum_{i<t} d_i\,(k_i^\top k_t), \tag{15}$$

with $\Delta\boldsymbol{W}_{1,0} = 0$ and $d_t = -\eta_1 e_t$. The transported feature is $\widetilde{h}_t = h_t + \phi'(u_t) \odot \delta u_t$. Equation 15 is causal linear attention with d_t as the value and k_t as the key. For the key features that $\boldsymbol{W}_2$ writes with, k_t is also the query. Feature transport therefore adds a linear-attention fast weight that corrects the features of the TTT-MLP.

Parallel form. Stacking $K = [k_1, \ldots, k_b]$, $D = [d_1, \ldots, d_b]$, and $J = [\phi'(u_1), \ldots, \phi'(u_b)]$, all shifts in a block are computed at once:

$$\Delta U = D\,\mathcal{U}(K^\top K), \qquad \widetilde{\Phi} = \Phi + J \odot \Delta U, \tag{16}$$

where the strictly upper-triangular mask $\mathcal{U}$ ensures that token t sees only tokens $i < t$. This costs one extra $\mathcal{O}(b^2 d)$ matrix product per block, the same order as the WY solve. When a TTT block is split into several kernel chunks, $\Delta\boldsymbol{W}_1$ is carried across chunk boundaries as in chunkwise linear attention. For a chunk with incoming state $\Delta\boldsymbol{W}_1^{\text{in}}$,

$$\Delta U_c = \Delta\boldsymbol{W}_1^{\text{in}} K_c + D_c\,\mathcal{U}(K_c^\top K_c), \qquad \Delta\boldsymbol{W}_1^{\text{out}} = \Delta\boldsymbol{W}_1^{\text{in}} + D_c K_c^\top. \tag{17}$$

The state resets to zero at each TTT block boundary, where $\boldsymbol{W}_1$ takes its mini-batch step. Our update, denoted `FF-TTT`, then runs the solve equation 13 on $\widetilde{\Phi}$ in place of Φ.

Proposition 4.1 (Feature transport recovers $\mathcal{A}_{21}$)**.** *Let $\boldsymbol{W}_2^{\mathrm{FW}^2}$ be the endpoint of equation 13 with features $\widetilde{\Phi}$ and constant step η_2. Under Assumption 3.1,*

$$\boldsymbol{W}_2^{\mathrm{FW}^2} = \boldsymbol{W}_2 - \eta_2 \sum_{t=1}^{b} g_{2,t} + \eta_1\eta_2\,\mathcal{A}_{21} + \eta_2^2\,\mathcal{A}_{22} + \mathcal{O}(\eta_{\max}^3 b^3). \tag{18}$$

Comparing with equation 9, `FF-TTT`matches the sequential $\boldsymbol{W}_2$ endpoint through second order. The proof is in Appendix A.4. Adding $\eta_1\eta_2\mathcal{A}_{21}$ to $\boldsymbol{W}_2^{\mathrm{WY}}$ directly would agree to the same order, but transporting the features propagates the correction through every order of the delta recurrence.

Query transport. Since reads are post-update, the query feature also drifts with $\boldsymbol{W}_1$ within the block. We therefore transport queries with the same virtual prefix:

$$\widetilde{\Phi}^q = \hat{\Phi} + \hat{J} \odot D\,\mathcal{U}(K^\top Q), \tag{19}$$

which reuses D and costs one extra $\mathcal{O}(b^2 d)$ product per block. This removes the $\mathcal{O}(\eta_1 t)$ intra-block read error (Corollary A.10).

Remark 4.2 (Why $\boldsymbol{W}_1$ is not resummed). The WY form exists because equation 12 right-multiplies the state by rank-one perturbations of the identity. The $\boldsymbol{W}_1$ recurrence lacks this structure. Even for $\phi = \mathrm{id}$ with $\boldsymbol{W}_2$ fixed, $g_{1,t} = \boldsymbol{W}_2^\top(\boldsymbol{W}_2\boldsymbol{W}_1 k_t - v_t)k_t^\top$ acts on $\boldsymbol{W}_1$ from both sides, and the joint $(\boldsymbol{W}_1, \boldsymbol{W}_2)$ recurrence is bilinear. We know of no compact triangular factorization for either case. We therefore treat the $\boldsymbol{W}_1$ side perturbatively, and $\mathcal{A}_{22}$ is the only interaction that admits exact resummation.

4.3 FF-TTT ARCHITECTURE

We use a SwiGLU MLP (Shazeer, 2020) as the fast-weight network, following `LaCT` (Zhang et al., 2026a):

$$f_{\boldsymbol{W}}(x) = \boldsymbol{W}_2\big[\mathrm{SiLU}(\boldsymbol{W}_1 x) \odot (\boldsymbol{W}_3 x)\big], \qquad \boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3). \tag{20}$$

Unlike LaCT, we train it with the reconstruction loss $\ell_t = \frac{1}{2}\|f_{\boldsymbol{W}}(k_t) - v_t\|^2$. This fits the analysis of Section 3.2, with the gated map $h_t = \mathrm{SiLU}(\boldsymbol{W}_1 k_t) \odot (\boldsymbol{W}_3 k_t)$ as the feature layer. $\boldsymbol{W}_2$ is the token-level memory updated by the WY solve, while $\boldsymbol{W}_1$ and $\boldsymbol{W}_3$ are block-level fast weights updated by the mini-batch step, and feature transport accounts for the prefixes of both. The analysis of Sections 3.2 and 4.2 carries over unchanged with the gated map $h_t = \mathrm{SiLU}(\boldsymbol{W}_1 k_t) \odot (\boldsymbol{W}_3 k_t)$ as the feature layer and $(\boldsymbol{W}_1, \boldsymbol{W}_3)$ jointly playing the role of the first layer. The only change is that the transported feature combines both prefixes, $p_t = \left[\mathrm{SiLU}'(\boldsymbol{W}_1 k_t) \odot (G_{1,<t} k_t)\right] \odot (\boldsymbol{W}_3 k_t) + \mathrm{SiLU}(\boldsymbol{W}_1 k_t) \odot (G_{3,<t} k_t)$, where $G_{3,<t}$ is the prefix gradient of $\boldsymbol{W}_3$. Query features are transported with the same prefixes $G_{1,<t}$ and $G_{3,<t}$, with q_t in place of every k_t in p_t. Both have parallel forms (see Section D for the derivation).

Initialization. We take $\boldsymbol{W}_1$ and $\boldsymbol{W}_3$ to be square and initialize each as a structured base map plus a low-rank residual:

$$\boldsymbol{W}_{1,0} = cI + \Delta\boldsymbol{W}_1, \qquad \boldsymbol{W}_{3,0} = c\,\Pi + \Delta\boldsymbol{W}_3, \tag{21}$$

with $c = 0.5$ and Π the cyclic shift permutation. The permutation matters. With $\boldsymbol{W}_{3,0} = cI$ and the residuals neglected, each feature coordinate is $\mathrm{SiLU}(cx_i)\,cx_i = c^2 x_i^2\,\sigma(cx_i) \geq 0$. The initial features therefore lose all sign information and collapse into the nonnegative orthant, where they are poorly separated in angle. With Π, the coordinates become $\mathrm{SiLU}(cx_i)\,c\,x_{\pi(i)}$, which mix pairs of coordinates and retain both signs. Because Π is orthogonal and parameter-free, this improves the initial feature geometry at no parameter cost and negligible compute.

5 Experiments and Results

5.1 Experimental Setup

Baselines. We compare FF-TTT with strong baselines in real-world language modeling settings. Our main baselines include: Transformer (Vaswani et al., 2017), GLA (Yang et al., 2024b), DeltaNet (Yang et al., 2024c), Gated DeltaNet (Yang et al., 2024a), Mamba (Gu & Dao, 2024), a selective state-space model with data-dependent decay and LaCT (Zhang et al., 2026a).

Training Setup. We train models at three scales—340M, 760M, and 1B parameters—using Chinchilla-style token budgets (Hoffmann et al., 2022). The models are trained on FineWeb-Edu (Penedo et al., 2024) for 7.52B, 16.11B, and approximately 20B tokens, respectively; Training is implemented with Flame (Zhang & Yang, 2025) and TorchTitan (Liang et al., 2025), with model operators and Triton kernels built upon FLA (Yang & Zhang, 2024).

Evaluation. We evaluate language modeling perplexity on WikiText (Merity et al., 2016) and LAMBADA (Paperno et al., 2016), together with a suite of zero-shot commonsense reasoning tasks, including PiQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), Arc-challenge (ARC-c) (Clark et al., 2018), BoolQ (Clark et al., 2019), OpenBookQA (OBQA) (Mihaylov et al., 2018) and SciQ (Johannes Welbl, 2017). We reported perplexity for Wikitext and LAMBADA and accuracy for the downstream reasoning tasks.

Table 1 reports zero-shot results at three model scales (340M, 760M, and 1B parameters). We compare FF-TTT against a Transformer baseline and several efficient token-mixing models: DeltaNet, Gated DeltaNet, GLA, Mamba, and LaCT. Among these, LaCT is our most directly relevant baseline, as it is the closest prior method in design and the strongest competing efficient model overall. We report perplexity on WikiText and LAMBADA, and accuracy on eight commonsense-reasoning and QA benchmarks. FF-TTT achieves the highest average accuracy at every scale. Relative to LaCT specifically, it improves the average by 0.51 points at 340M (48.20 vs. 47.69), 0.49 at 760M (52.95 vs. 52.46), and 0.60 at 1B (55.25 vs. 54.65). It also attains lower LAMBADA perplexity than LaCT at all three scales and has lower WikiText perplexity. No single model wins every task; for instance, LaCT leads on SciQ at 760M and 1B. Nevertheless, FF-TTT is the only model that ranks first or second on most columns at every scale, indicating that its gains are not concentrated in any one benchmark.

Long Memory Tasks. We also evaluate long-context performance, comprising the code-completion tasks LCC (Guo et al., 2023) and RepoBench-P (Liu et al., 2024b), together with the few-shot learning evaluation on TREC (Li & Roth, 2002), TriviaQA (Joshi et al., 2017), and

Table 1: Zero-shot language-model evaluation. WikiText and LAMBADA perplexity (PPL; lower is better) are excluded from Mean. All other columns report accuracy in percent (higher is better).

		Perplexity (↓)		Accuracy (↑)								
Scale	**Model**	**Wiki.**	**LMB.**	**ARC-c**	**Hella.**	**PIQA**	**Wino.**	**LMB.**	**BoolQ**	**OBQA**	**SciQ**	**Avg.**
340M	Transformer	27.66	40.89	27.56	38.48	64.85	51.78	31.98	60.67	**33.80**	74.20	47.92
	DeltaNet	29.09	38.25	27.39	37.81	64.96	**51.85**	30.91	60.06	32.40	71.90	47.16
	Gated DeltaNet	28.14	36.21	26.79	38.77	64.58	51.07	31.96	58.84	33.00	74.10	47.39
	GLA	31.13	40.88	28.67	37.51	64.36	49.88	31.22	**61.31**	31.40	72.40	47.09
	Mamba	29.43	38.73	27.90	**39.67**	65.29	48.15	30.23	56.76	32.80	73.80	46.83
	LaCT	27.41	40.40	**29.69**	38.89	64.96	51.70	31.92	58.53	31.20	74.60	47.69
	FF-TTT	**27.28**	**35.41**	27.73	39.14	**65.94**	50.59	**33.59**	60.40	32.60	**75.60**	**48.20**
760M	Transformer	21.09	19.02	30.72	47.40	68.99	54.38	**41.92**	58.75	35.80	81.10	52.38
	DeltaNet	21.68	19.43	30.46	46.48	68.61	51.54	40.31	**60.52**	36.40	80.30	51.83
	Gated DeltaNet	23.89	27.63	29.86	47.43	69.26	54.93	33.61	60.15	**37.20**	78.40	51.36
	GLA	23.41	21.34	30.46	45.51	67.95	54.14	38.70	60.06	37.00	78.40	51.53
	Mamba	22.04	21.37	**33.02**	**48.53**	**69.37**	53.51	38.85	60.24	35.60	79.50	52.33
	LaCT	21.01	18.04	31.31	47.63	68.72	54.22	41.24	59.97	34.40	**82.20**	52.46
	FF-TTT	**20.77**	**17.77**	32.51	48.19	68.82	**56.12**	41.79	60.06	36.00	80.10	**52.95**
1B	Transformer	**18.89**	17.21	34.73	52.40	70.99	56.35	43.82	61.75	**39.70**	81.20	55.12
	DeltaNet	19.90	17.52	33.96	49.59	70.67	55.56	41.14	60.95	39.40	81.60	54.11
	Mamba	20.36	20.79	**34.90**	51.25	70.35	52.64	38.48	56.39	36.80	78.90	52.46
	LaCT	19.58	17.54	34.73	51.48	70.73	55.49	43.70	59.36	39.40	**82.30**	54.65
	FF-TTT	19.09	**16.85**	34.87	**52.68**	**71.95**	**56.67**	**44.26**	**61.99**	38.60	81.00	**55.25**

SAMSum (Gliwa et al., 2019). We report edit similarity for LCC and RepoBench-P, accuracy for TREC, F1 for TriviaQA, and ROUGE-L for SAMSum. We evaluate each task over the 0–4K, 4–8K, and 8K+ context-length buckets and report the unweighted average across the three buckets. We compare against LaCT, DeltaNet, and LaCT-DeltaNet, which updates $\boldsymbol{W}_2$ with the exact WY solve of Section 4 but omits feature transport; it is therefore close to FF-TTT without transport and serves as an ablation.

Table 2: Comparison on long memory tasks. Each score is the macro-average over the 0–4K, 4–8K, and 8K+ context-length buckets.

Model	**LCC**	**RepoBench-P**	**TREC**	**TriviaQA**	**SAMSum**	**Average**
FF-TTT	17.78	**21.30**	**39.67**	**27.10**	**18.95**	**24.96**
LaCT-DeltaNet	17.37	20.12	27.33	23.61	17.50	21.19
DeltaNet	**18.34**	15.97	32.67	18.59	16.59	20.43
LaCT	13.86	12.23	27.00	18.66	10.76	16.50

FF-TTT achieves the best average score of 24.96 and ranks first on four of the five selected tasks, outperforming DeltaNet by 4.53 points and LaCT by 8.46 points on average. DeltaNet performs best on LCC, exceeding FF-TTT by 0.56 points. LaCT-DeltaNet improves over LaCT on all five tasks. LaCT-DeltaNet trails FF-TTT by 3.77 points, which reflects the contribution of feature transport and is far larger than the 0.5–0.6-point zero-shot gains over LaCT in Table 1, so the advantage of FF-TTT shows mainly on long-range tasks.

6 Discussion

This section examines two practical questions raised by the main results: where first-order feature transport remains accurate, and what it costs during training.

6.1 Operating Regime of Feature Transport

The three left panels examine approximation accuracy at a trained post-norm checkpoint, using fully token-sequential execution as the reference. Lact-DeltaNet retains the exact within-chunk Delta recurrence, while FF-TTT additionally applies first-order query and key feature transport. When both the Delta write rate and slow-update rate are scaled by s, the FF-TTT output gap generally increases with s and chunk size b. Its advantage over Lact-DeltaNet is largest at intermediate within-chunk

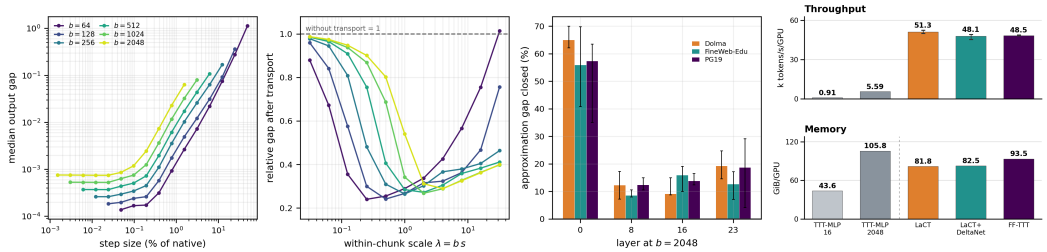


Figure 1: Feature transport reduces the sequential approximation gap across block sizes, transport strengths, layers, and datasets, suggesting that it captures a stable first-order interaction absent from the fixed-feature solve. At the 340M scale, Our implementation of `FF-TTT` consumes a smaller memory footprint than the `TTT-MLP` baseline with minibatch size 2048 while maintaining training throughput competitive as `LaCT`.

scale $\lambda = bs$: the ratio of their median output errors falls to approximately 0.25–0.4, then rises as the scale increases. This nonmonotonicity suggests a finite trust region for first-order transport. This matches remark A.6: once the within-chunk $\boldsymbol{W}_2$ writes are large, the excluded interaction $\mathcal{A}_{12}$ reaches the $\boldsymbol{W}_2$ endpoint through the features at first order in η_1. At native rates and $b = 2048$, transport still closes part of the approximation gap across Dolma (Olmo et al., 2026), FineWeb-Edu Penedo et al. (2024), and PG19 (Rae et al., 2019), with particularly strong effects in layer 0 (56–65%) and smaller positive effects in the later layers shown (9–19%). These checkpoint-level diagnostics support the proposed mechanism, but do not by themselves establish an improvement in downstream training loss.

6.2 Training Throughput and Memory

The rightmost column reports a separate 340M-class efficiency comparison on eight H200 GPUs. `FF-TTT` reaches 48.5k tokens/s/GPU, 5.5% below the matched `LaCT` configuration (51.3k) and close to `LaCT-DeltaNet` (48.1k). Its measured peak reserved memory is 93.5 GiB/GPU, compared with 81.8 GiB for `LaCT` and 82.5 GiB for `LaCT-DeltaNet`. Increasing the TTT-MLP inner mini-batch from 16 to 2048 raises its measured throughput from 0.91k to 5.59k tokens/s/GPU, while its XLA planned allocation rises from 43.6 to 105.8 GiB/GPU. The TTT-MLP results come from separate JAX synthetic-step benchmarks, whereas the `LaCT`-family throughput values come from completed training runs; moreover, XLA planned allocation and PyTorch peak reserved memory are different metrics. Thus, the cross-framework bars provide implementation context rather than a strict architecture-only ranking. Within the matched `LaCT` comparison, `FF-TTT` retains similar throughput at the cost of a measurable memory premium.

7 Conclusion

Using block-level backward error analysis, we identified the causal interaction $\mathcal{A} = \sum_t H_t G_{<t}$ as the leading change that mini-batching introduces into the TTT inner-loop dynamics. For a two-layer fast-weight MLP, $\mathcal{A}$ splits into four layer-wise interactions. The two acting on the output layer are structured: $\mathcal{A}_{22}$ is resummed exactly by a WY solve, and $\mathcal{A}_{21}$ reduces to feature transport, a causal linear-attention shift of the keys and queries. `FF-TTT` keeps $\boldsymbol{W}_1$ at chunk level and makes $\boldsymbol{W}_2$ a token-level memory. The resulting `FF-TTT` update matches the sequential output-layer endpoint through second order, and the excluded first-layer terms $\eta_1^2\mathcal{A}_{11} + \eta_1\eta_2\mathcal{A}_{12}$ form the explicit residual. From 340M to 1B parameters, `FF-TTT` improves on `LaCT` in perplexity and average zero-shot accuracy at every scale, with modest throughput overhead. Extending exact resummation beyond the squared loss and going beyond first-order transport are natural next steps.

REFERENCES

David Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3q5IqUrkcF.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

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, 2025a. URL https://arxiv.org/abs/2505.23735.

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, 2025b. URL https://arxiv.org/abs/2504.13173.

Christian Bischof and Charles Van Loan. The wy representation for products of householder matrices. *SIAM Journal on Scientific and Statistical Computing*, 8(1):s2–s13, 1987.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Guhao Feng, Shengjie Luo, Kai Hua, Ge Zhang, Di He, Wenhao Huang, and Tianle Cai. In-place test-time training, 2026. URL https://arxiv.org/abs/2604.06169.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (eds.), *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409/.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian Mcauley. LongCoder: A long-range pre-trained language model for code completion. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12098–12107. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/guo23j.html.

Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. ISBN 3-540-30663-3; 978-3-540-30663-4. Structure-preserving algorithms for ordinary differential equations.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Matt Gardner Johannes Welbl, Nelson F. Liu. Crowdsourcing multiple choice science questions. 2017.

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

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1150/.

Zeman Li, Ali Behrouz, Yuan Deng, Peilin Zhong, Praneeth Kacham, Mahdi Karami, Meisam Razaviyayn, and Vahab Mirrokni. TNT: Improving chunkwise training for test-time memorization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rajioNWfRs.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. Torchtitan: One-stop pytorch native solution for production ready LLM pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SFN6Wm7YBI.

Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. Longhorn: State space models are amortized online learners. *arXiv preprint arXiv:2407.14207*, 2024a.

Junchen Liu, Sven Elflein, Or Litany, Zan Gojcic, and Ruilong Li. Test-time training with KV binding is secretly linear attention. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=MpBeXMurIb.

Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 47832–47850, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d191ba4c8923ed8fd8935b7c98658b5f-Paper-Conference.pdf.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1911.05507.

Mihaela Claudia Rosca. On discretisation drift and smoothness regularisation in neural network training. *CoRR*, abs/2310.14036, 2023. doi: 10.48550/ARXIV.2310.14036. URL https://doi.org/10.48550/arXiv.2310.14036.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL http://proceedings.mlr.press/v139/schlag21a.html.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. Deltaproduct: Increasing the expressivity of deltanet through products of householders. *arXiv preprint arXiv:2502.10297*, 2025.

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL http://proceedings.mlr.press/v119/sun20b.html.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, Carlos Guestrin, Jed McCaleb, Yejin Choi, and Yu Sun. End-to-end test-time training for long context, 2025. URL https://arxiv.org/abs/2512.23675.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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, 2026. URL https://arxiv.org/abs/2506.05233.

Ke Alexander Wang, Jiaxin Shi, and Emily B. Fox. Test-time regression: a unifying framework for designing sequence models with associative memory, 2025. URL https://arxiv.org/abs/2501.12352.

Songlin Yang and Yu Zhang. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism, January 2024. URL https://github.com/fla-org/flash-linear-attention.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024a.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=ia5XvxFUJT.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in Neural Information Processing Systems*, 37: 115491–115522, 2024c.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=Tb9qAxT3xv.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=Tb9qAxT3xv.

Yu Zhang and Songlin Yang. Flame: Flash language modeling made easy, January 2025. URL https://github.com/fla-org/flame.

A Theoretical Results and Proofs

Conventions. $\|\cdot\|$ denotes the Frobenius norm for matrices and the Euclidean norm for vectors. Unless stated otherwise, C denotes a constant that may change from line to line. It depends only on the derivative bounds in Assumption 3.1 and on uniform bounds for the block-entrance quantities k_t, q_t, h_t, r_t, e_t, $\phi'(u_t)$, $\phi''(u_t)$, $\phi'(\hat{u}_t)$, and $\phi''(\hat{u}_t)$ over the block, where $\hat{u}_t = \boldsymbol{W}_1 q_t$, and it never depends on b, η_1, or η_2. Big-$\mathcal{O}$ bounds hold uniformly whenever $b\,\eta_{\max} \le c$ for a fixed constant c small enough that Assumption 3.1 applies. All quantities without an explicit argument are evaluated at the block entrance.

A.1 Proof of Proposition 3.2

Proof. Let L_1, L_2, and L_3 bound the first, second, and third derivatives of every ℓ_t on the neighborhood in Assumption 3.1. Every sequential step has length at most ηL_1, so $\|\boldsymbol{W}_t^{\mathrm{seq}} - \boldsymbol{W}_0\| \le t\eta L_1$.

Prefix displacement. Since $\left\|g_i(\boldsymbol{W}_{i-1}^{\mathrm{seq}}) - g_i\right\| \le L_2 \left\|\boldsymbol{W}_{i-1}^{\mathrm{seq}} - \boldsymbol{W}_0\right\| \le (i-1)\eta L_1 L_2$,

$$\left\|\boldsymbol{W}_{t-1}^{\mathrm{seq}} - \boldsymbol{W}_0 + \eta\, G_{<t}\right\| = \eta \left\| \sum_{i<t} \big(g_i(\boldsymbol{W}_{i-1}^{\mathrm{seq}}) - g_i\big)\right\| \le \tfrac{1}{2} L_1 L_2\, t^2 \eta^2. \tag{22}$$

Gradient of token t. A second-order Taylor expansion of g_t at $\boldsymbol{W}_0$ gives $g_t(\boldsymbol{W}) = g_t + H_t(\boldsymbol{W} - \boldsymbol{W}_0) + \rho_t(\boldsymbol{W})$ with $\|\rho_t(\boldsymbol{W})\| \le \frac{1}{2} L_3 \|\boldsymbol{W} - \boldsymbol{W}_0\|^2$. Evaluating at $\boldsymbol{W} = \boldsymbol{W}_{t-1}^{\mathrm{seq}}$ and using equation 22,

$$g_t(\boldsymbol{W}_{t-1}^{\mathrm{seq}}) = g_t - \eta\, H_t G_{<t} + \varepsilon_t, \qquad \|\varepsilon_t\| \le \tfrac{1}{2}\big(L_1 L_2^2 + L_1^2 L_3\big)\, t^2 \eta^2. \tag{23}$$

Summation. Hence $\boldsymbol{W}_t^{\mathrm{seq}} - \boldsymbol{W}_{t-1}^{\mathrm{seq}} = -\eta\, g_t + \eta^2 H_t G_{<t} - \eta\, \varepsilon_t$. Summing over $t = 1, \dots, b$ yields

$$\boldsymbol{W}_b^{\mathrm{seq}} = \boldsymbol{W}_0 - \eta \sum_{t=1}^{b} g_t + \eta^2 \mathcal{A} - \eta \sum_{t=1}^{b} \varepsilon_t, \qquad \left\|\eta \sum_{t=1}^{b} \varepsilon_t\right\| \le C\, b^3 \eta^3. \tag{24}$$

Subtracting the mini-batch map in equation 4 gives equation 5. □

A.2 Layer-wise step sizes

Corollary A.1 (Layer-wise expansion)**.** *Partition the parameters into blocks $(\boldsymbol{W}_1, \boldsymbol{W}_2)$ and let $\Omega = \mathrm{diag}(\eta_1 I, \eta_2 I)$ act blockwise. Consider the sequential map $\boldsymbol{W}_t^{\mathrm{seq}} = \boldsymbol{W}_{t-1}^{\mathrm{seq}} - \Omega\, g_t(\boldsymbol{W}_{t-1}^{\mathrm{seq}})$ and the mini-batch map $\boldsymbol{W}_b^{\mathrm{mb}} = \boldsymbol{W}_0 - \Omega \sum_t g_t$. Under Assumption 3.1 with $b\,\eta_{\max}$ small,*

$$\boldsymbol{W}_b^{\mathrm{seq}} - \boldsymbol{W}_b^{\mathrm{mb}} = \sum_{t=1}^{b} \Omega H_t \Omega\, G_{<t} + \mathcal{O}(\eta_{\max}^3 b^3). \tag{25}$$

Its $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ blocks are equation 8 and equation 9.

Proof. Repeat the proof of Proposition 3.2 with η replaced by Ω and use $\|\Omega\|_{\mathrm{op}} = \eta_{\max}$. The prefix displacement becomes $-\Omega\, G_{<t} + \mathcal{O}(t^2 \eta_{\max}^2)$, the gradient of token t becomes $g_t - H_t \Omega\, G_{<t} + \mathcal{O}(t^2 \eta_{\max}^2)$, and the step becomes $-\Omega g_t + \Omega H_t \Omega\, G_{<t} + \mathcal{O}(t^2 \eta_{\max}^3)$. Summing over t gives equation 25. For the blocks, $(\Omega H_t \Omega\, G_{<t})_\alpha = \eta_\alpha \sum_\beta H_{\alpha\beta,t}\, \eta_\beta\, G_{\beta,<t}$, and summing over t gives $\sum_\beta \eta_\alpha \eta_\beta\, \mathcal{A}_{\alpha\beta}$. □

A.3 Exact block solve and compact WY form

The results in this subsection hold for arbitrary features $h_1, \dots, h_b$. They are applied below both to the entrance features Φ and to the transported features $\widetilde{\Phi}$.

Lemma A.2 (Exact block solve)**.** *Fix $h_1, \dots, h_b$ and step sizes $\eta_{2,1}, \dots, \eta_{2,b}$. Let $\boldsymbol{W}_{2,t}$ follow the delta rule equation 12 from $\boldsymbol{W}_{2,0} = \boldsymbol{W}_2$, and set $z_t = \eta_{2,t}(v_t - \boldsymbol{W}_{2,t-1} h_t)$. Then*

(i) $\boldsymbol{W}_{2,t} = \boldsymbol{W}_2 + \sum_{i \le t} z_i h_i^\top$ for every $t = 0, \dots, b$;

(ii) $Z = [z_1, \dots, z_b]$ *is the unique solution of* $ZT = (V - \boldsymbol{W}_2\Phi)\Lambda$ *with* $T = I + \mathcal{U}(\Phi^\top\Phi\Lambda)$*, so that* $\boldsymbol{W}_{2,b} = \boldsymbol{W}_2^{\mathrm{WY}}$ *in equation 13.*

Proof. (i) By equation 12, $\boldsymbol{W}_{2,t} - \boldsymbol{W}_{2,t-1} = \eta_{2,t}(v_t - \boldsymbol{W}_{2,t-1}h_t)h_t^\top = z_t h_t^\top$. Now telescope.

(ii) Substituting (i) at time $t-1$ into the definition of z_t gives

$$z_t + \eta_{2,t}\sum_{i<t} z_i\,(h_i^\top h_t) = \eta_{2,t}\,(v_t - \boldsymbol{W}_2 h_t). \tag{26}$$

Right multiplication by the diagonal Λ scales columns, so $\mathcal{U}(\Phi^\top\Phi)\Lambda = \mathcal{U}(\Phi^\top\Phi\Lambda)$ and its (i,t) entry is $\eta_{2,t}\,h_i^\top h_t$ for $i < t$ and 0 otherwise. The left side of equation 26 is therefore the t-th column of $Z\big(I + \mathcal{U}(\Phi^\top\Phi\Lambda)\big) = ZT$, and the right side is the t-th column of $(V - \boldsymbol{W}_2\Phi)\Lambda$. Since T is unit upper triangular, it is invertible and Z is unique. Finally, $\boldsymbol{W}_{2,b} = \boldsymbol{W}_2 + Z\Phi^\top$ by (i). □

Corollary A.3 (Compact WY form)**.** *With* T *as in Lemma A.2,*

$$\prod_{t=1}^{b}\big(I - \eta_{2,t}\,h_t h_t^\top\big) = I - \Phi\,\Lambda T^{-1}\,\Phi^\top, \tag{27}$$

$$\boldsymbol{W}_2^{\mathrm{WY}} = \boldsymbol{W}_2\big(I - \Phi\,\Lambda T^{-1}\Phi^\top\big) + V\Lambda T^{-1}\Phi^\top, \tag{28}$$

where the product is ordered with t *increasing from left to right. The matrix* ΛT^{-1} *is upper triangular, so equation 27–equation 28 is the compact form of the WY representation (Bischof & Van Loan, 1987) of the* b *rank-one factors, with* Φ *serving as both the* W *and* Y *factors.*

Proof. Unrolling equation 12 gives $\boldsymbol{W}_{2,b} = \boldsymbol{W}_2\prod_{t=1}^{b}(I - \eta_{2,t}h_t h_t^\top) + \sum_t \eta_{2,t}v_t h_t^\top\prod_{s>t}(I - \eta_{2,s}h_s h_s^\top)$. By Lemma A.2, $\boldsymbol{W}_{2,b} = \boldsymbol{W}_2 + (V - \boldsymbol{W}_2\Phi)\Lambda T^{-1}\Phi^\top$. Neither T nor the product depends on $\boldsymbol{W}_2$ or V. Setting $V = 0$ and comparing the two expressions as linear maps of $\boldsymbol{W}_2$ gives the product identity, and the second identity follows by rearranging. □

Remark A.4 (Intra-block reads). By Lemma A.2(i), the post-update read of token t with query feature $\hat{h}_t = \phi(\boldsymbol{W}_1 q_t)$ is $o_t = \boldsymbol{W}_{2,t}\hat{h}_t = \boldsymbol{W}_2\hat{h}_t + \sum_{i\le t} z_i\,(h_i^\top\hat{h}_t)$. Stacking $\hat{\Phi} = [\hat{h}_1, \dots, \hat{h}_b]$ gives $O = \boldsymbol{W}_2\hat{\Phi} + Z\,\mathcal{U}_{\le}(\Phi^\top\hat{\Phi})$, where $\mathcal{U}_{\le}$ keeps the upper triangle including the diagonal. All intra-block reads therefore come from the same Z at the cost of one extra $b \times b$ masked product. In `FF-TTT`, both sides are transported: the keys use $\widetilde{\Phi}$ from equation 16 and the queries use $\widetilde{\Phi}^q$, the parallel form of equation 19, both formed from the same prefix state $\Delta\boldsymbol{W}_1$. Lemma A.2(i) applied to $\widetilde{\Phi}$ then gives

$$O = \boldsymbol{W}_2\widetilde{\Phi}^q + Z\,\mathcal{U}_{\le}\big(\widetilde{\Phi}^\top\widetilde{\Phi}^q\big).$$

Lemma A.5 (The WY solve resums the memory interaction)**.** *Let* $\eta_{2,t} \equiv \eta_2$*,* $N = \mathcal{U}(\Phi^\top\Phi)$*,* $\kappa = \max_t\|h_t\|$*, and* $\varrho = \max_t\|r_t\|$*, where* $r_t = \boldsymbol{W}_2 h_t - v_t$*. Then*

$$\boldsymbol{W}_2^{\mathrm{WY}} = \boldsymbol{W}_2 - \eta_2\sum_{t=1}^{b} g_{2,t} + \eta_2^2\,\mathcal{A}_{22} + \mathcal{R}_{\mathrm{WY}}, \qquad \mathcal{R}_{\mathrm{WY}} = \sum_{k=2}^{b-1}(-1)^{k+1}\eta_2^{k+1}\,RN^k\Phi^\top, \tag{29}$$

where $g_{2,t} = r_t h_t^\top$*,* $\mathcal{A}_{22} = \sum_t H_{22,t}[G_{2,<t}]$*, and*

$$\|\mathcal{R}_{\mathrm{WY}}\| \le \tfrac{1}{6}\,\varrho\,\kappa^5\,(b\eta_2)^3\,e^{\,b\eta_2\kappa^2}. \tag{30}$$

In particular, $\mathcal{R}_{\mathrm{WY}} = \mathcal{O}(\eta_2^3 b^3)$ *whenever* $b\eta_2\kappa^2$ *is bounded, which proves equation 14.*

Proof. With $\Lambda = \eta_2 I$ we have $T = I + \eta_2 N$. Since N is strictly upper triangular, $N^b = 0$ and $T^{-1} = \sum_{k=0}^{b-1}(-\eta_2 N)^k$ exactly. Since $V - \boldsymbol{W}_2\Phi = -R$, Lemma A.2 gives

$$\boldsymbol{W}_2^{\mathrm{WY}} = \boldsymbol{W}_2 - \eta_2 R\,T^{-1}\Phi^\top = \boldsymbol{W}_2 + \sum_{k=0}^{b-1}(-1)^{k+1}\eta_2^{k+1}RN^k\Phi^\top. \tag{31}$$

The $k = 0$ term is $-\eta_2 R\Phi^\top = -\eta_2 \sum_t r_t h_t^\top = -\eta_2 \sum_t g_{2,t}$. For fixed features, $g_{2,t} = (\boldsymbol{W}_2 h_t - v_t)h_t^\top$ is affine in $\boldsymbol{W}_2$ with derivative $H_{22,t}[V_2] = V_2 h_t h_t^\top$. The $k = 1$ term is therefore

$$\eta_2^2 RN\Phi^\top = \eta_2^2 \sum_{i<t} r_i \,(h_i^\top h_t)\, h_t^\top = \eta_2^2 \sum_t G_{2,<t}\, h_t h_t^\top = \eta_2^2 \mathcal{A}_{22}, \tag{32}$$

where the last step uses $G_{2,<t} h_t h_t^\top = H_{22,t}[G_{2,<t}]$. For $k \geq 2$, $(N^k)_{i,t}$ is a sum over chains $i < j_1 < \cdots < j_{k-1} < t$ of products of k entries of N, each bounded by κ^2 in absolute value. Hence $\sum_{i,t} |(N^k)_{i,t}| \leq \kappa^{2k} \binom{b}{k+1}$, since $\binom{b}{k+1}$ counts the strictly increasing index sequences of length $k+1$. Because $RN^k\Phi^\top = \sum_{i,t} (N^k)_{i,t}\, r_i h_t^\top$ and $\left\| r_i h_t^\top \right\| \leq \varrho\kappa$, we get $\left\| RN^k\Phi^\top \right\| \leq \varrho\, \kappa^{2k+1} \binom{b}{k+1}$. With $j = k+1$, $\binom{b}{j} \leq b^j/j!$, and $\sum_{j\geq 3} y^j/j! \leq \frac{1}{6} y^3 e^y$,

$$\|\mathcal{R}_{\mathrm{WY}}\| \leq \frac{\varrho}{\kappa} \sum_{j\geq 3} \binom{b}{j} (\eta_2\kappa^2)^j \leq \frac{\varrho}{\kappa} \cdot \frac{(b\eta_2\kappa^2)^3}{6}\, e^{\,b\eta_2\kappa^2}, \tag{33}$$

which is equation 30. The case $\kappa = 0$ is trivial. □

Because $g_{2,t}$ is affine in $\boldsymbol{W}_2$ for fixed features, the sequential $\boldsymbol{W}_2$ map has no Taylor remainder beyond this finite Neumann series, and the WY solve sums the series exactly. This is the precise sense in which the solve recovers $\mathcal{A}_{22}$ "to all orders."

Remark A.6 (Beyond the small-step regime). The factor $e^{b\eta_2\kappa^2}$ in equation 30 comes from bounding the Neumann series term by term. The solve itself is exact for any step size. With ℓ_2-normalized features, $b\eta_2\kappa^2 = b\eta_2$, which is large at practical chunk sizes. The second-order statements built on equation 29, namely Proposition 4.1 and Corollary A.9, are therefore small-step statements. The following lemma bounds the effect of feature perturbations on the $\boldsymbol{W}_2$ endpoint for any $b\eta_2$.

Lemma A.7 (Stability of the delta rule)**.** *Let $\boldsymbol{W}_{2,t}$ and $\boldsymbol{W}'_{2,t}$ follow equation 12 from the same $\boldsymbol{W}_{2,0}$, with the same v_t and $\eta_{2,t}$ but with features h_t and h'_t, respectively. Let $\delta_t = h_t - h'_t$ and $r'_t = \boldsymbol{W}'_{2,t-1} h'_t - v_t$. If $0 \leq \eta_{2,t} \|h_t\|^2 \leq 2$ for all t, then*

$$\left\| \boldsymbol{W}_{2,b} - \boldsymbol{W}'_{2,b} \right\| \leq \sum_{t=1}^{b} \eta_{2,t} \left(\|r'_t\| + \left\| \boldsymbol{W}'_{2,t-1} \right\|_{\mathrm{op}} \|h_t\| \right) \|\delta_t\| . \tag{34}$$

Proof. Let $\Delta_t = \boldsymbol{W}_{2,t} - \boldsymbol{W}'_{2,t}$. Using $h_t h_t^\top - h'_t h_t'^\top = h'_t \delta_t^\top + \delta_t h_t^\top$, subtracting the two recurrences gives $\Delta_t = \Delta_{t-1}(I - \eta_{2,t} h_t h_t^\top) - \eta_{2,t}\left(r'_t \delta_t^\top + \boldsymbol{W}'_{2,t-1} \delta_t h_t^\top \right)$. The eigenvalues of $I - \eta_{2,t} h_t h_t^\top$ lie in $[-1, 1]$, so the first term has norm at most $\|\Delta_{t-1}\|$. Bounding the second term by the triangle inequality and unrolling from $\Delta_0 = 0$ gives equation 34. □

Consequences for `FF-TTT`. Sequential TTT and `FF-TTT` both run the delta rule on $\boldsymbol{W}_2$. The first uses the features $h_t^{\mathrm{seq}} = \phi(\boldsymbol{W}_{1,t-1}^{\mathrm{seq}} k_t)$ and the second uses $\widetilde{h}_t$. Lemma A.7 with $h_t = h_t^{\mathrm{seq}}$ and $h'_t = \widetilde{h}_t$ therefore bounds the $\boldsymbol{W}_2$ endpoint error by the feature error $\left\| h_t^{\mathrm{seq}} - \widetilde{h}_t \right\|$. In particular, the pure-η_2 part of the remainder in Corollary A.9 vanishes.

By Lemma A.8,

$$h_t^{\mathrm{seq}} - \widetilde{h}_t = -\eta_1 \phi'(u_t) \odot \textstyle\sum_{i<t} (e_i^{\mathrm{seq}} - e_i)(k_i^\top k_t) + \mathcal{O}(\eta_1^2 t^2),$$

where e_i^{seq} is e_i evaluated at the sequential state.

- *Small steps.* For small $b\eta_{\max}$, $e_i^{\mathrm{seq}} - e_i = \mathcal{O}(\eta_{\max} i)$, and the endpoint error is $\mathcal{O}(\eta_1\eta_2\eta_{\max} b^3)$ with no exponential factor.
- *Large steps.* When $\sum_{j<i} \eta_{2,j} \|h_j\|^2 \gtrsim 1$, $\boldsymbol{W}_2$ moves substantially within the block and $e_i^{\mathrm{seq}} - e_i$ is no longer small. The endpoint error is then of order $\eta_1 \sum_t \eta_{2,t} t$, the same order as $\eta_1\eta_2\mathcal{A}_{21}$. In this regime, the excluded interaction $\mathcal{A}_{12}$ reaches the $\boldsymbol{W}_2$ endpoint through the features at first order in η_1.

The large-step behavior is consistent with the finite trust region of transport in Figure 1.

A.4 Proof of Proposition 4.1

Lemma A.8 (Transported features)**.** *With $D = -\eta_1 E$ and $E = [e_1, \dots, e_b]$, the columns of $\widetilde{\Phi}$ in equation 16 are exactly $\widetilde{h}_t = h_t - \eta_1 p_t$, with p_t as in equation 10. Moreover, $\|p_t\| \le Ct$. Under the sequential two-layer trajectory, the feature of token t is $\phi(\boldsymbol{W}_{1,t-1}^{\mathrm{seq}} k_t) = \widetilde{h}_t + \mathcal{O}(\eta_{\max}^2 t^2)$.*

Proof. Since $G_{1,<t} = \sum_{i<t} e_i k_i^\top$, the t-th column of $\Delta U = D\,\mathcal{U}(K^\top K)$ is

$$-\eta_1 \sum_{i<t} e_i \,(k_i^\top k_t) = -\eta_1 G_{1,<t} k_t.$$

Multiplying elementwise by $\phi'(u_t)$ gives $-\eta_1 p_t$. The bound $\|p_t\| \le \|\phi'(u_t)\|_\infty \, \|G_{1,<t}\| \, \|k_t\| \le Ct$ follows because $G_{1,<t}$ has $t-1$ bounded summands. For the last claim, the $\boldsymbol{W}_1$ block of the prefix displacement in the proof of Corollary A.1 gives $\boldsymbol{W}_{1,t-1}^{\mathrm{seq}} = \boldsymbol{W}_1 - \eta_1 G_{1,<t} + \mathcal{O}(\eta_{\max}^2 t^2)$. A first-order Taylor expansion of ϕ around u_t then gives the claim. □

Proof of Proposition 4.1. Write $\widetilde{r}_t = \boldsymbol{W}_2 \widetilde{h}_t - v_t = r_t - \eta_1 \boldsymbol{W}_2 p_t$ and $\widetilde{R} = [\widetilde{r}_1, \dots, \widetilde{r}_b]$. By Lemma A.8, $\left\|\widetilde{h}_t - h_t\right\| \le C\eta_1 b$ and $\|\widetilde{r}_t - r_t\| \le C\eta_1 b$, so $\widetilde{h}_t$ and $\widetilde{r}_t$ are uniformly bounded when $b\eta_{\max} \le c$. The series equation 29 and the bound equation 30 hold for arbitrary features, with (κ, ϱ) computed from the features and residuals actually used. Applying them to $\widetilde{\Phi}$ and $\widetilde{R}$ gives

$$\boldsymbol{W}_2^{\mathrm{FW}^2} = \boldsymbol{W}_2 - \eta_2 \sum_t \widetilde{r}_t \widetilde{h}_t^\top + \eta_2^2 \sum_{i<t} \widetilde{r}_i \,(\widetilde{h}_i^\top \widetilde{h}_t)\, \widetilde{h}_t^\top + \mathcal{O}(\eta_2^3 b^3). \tag{35}$$

First-order term. Expanding the product, which is exactly quadratic in η_1, and using equation 54,

$$\begin{aligned} \widetilde{r}_t \widetilde{h}_t^\top &= r_t h_t^\top - \eta_1 \left[(\boldsymbol{W}_2 p_t) h_t^\top + r_t p_t^\top \right] + \eta_1^2 (\boldsymbol{W}_2 p_t) p_t^\top \\ &= g_{2,t} - \eta_1 H_{21,t}[G_{1,<t}] + \eta_1^2 (\boldsymbol{W}_2 p_t) p_t^\top. \end{aligned} \tag{36}$$

Summing over t and using $\left\|\sum_t (\boldsymbol{W}_2 p_t) p_t^\top\right\| \le C \sum_t t^2 \le C b^3$,

$$-\eta_2 \sum_t \widetilde{r}_t \widetilde{h}_t^\top = -\eta_2 \sum_t g_{2,t} + \eta_1 \eta_2 \, \mathcal{A}_{21} + \mathcal{O}(\eta_1^2 \eta_2 \, b^3). \tag{37}$$

Second-order term. Every factor in $\widetilde{r}_i (\widetilde{h}_i^\top \widetilde{h}_t) \widetilde{h}_t^\top$ differs from its untransported counterpart by at most $C\eta_1 b$, and all factors are bounded. Each summand therefore differs from $r_i (h_i^\top h_t) h_t^\top$ by at most $C\eta_1 b$. There are at most $b^2/2$ summands, and $\sum_{i<t} r_i (h_i^\top h_t) h_t^\top = \mathcal{A}_{22}$ by the proof of Lemma A.5, so

$$\eta_2^2 \sum_{i<t} \widetilde{r}_i \,(\widetilde{h}_i^\top \widetilde{h}_t)\, \widetilde{h}_t^\top = \eta_2^2 \mathcal{A}_{22} + \mathcal{O}(\eta_1 \eta_2^2 \, b^3). \tag{38}$$

Substituting both terms into equation 35 and absorbing $\mathcal{O}(\eta_1^2 \eta_2 b^3)$, $\mathcal{O}(\eta_1 \eta_2^2 b^3)$, and $\mathcal{O}(\eta_2^3 b^3)$ into $\mathcal{O}(\eta_{\max}^3 b^3)$ gives equation 18. □

Corollary A.9 (Residual of `FF-TTT`)**.** *Let $\boldsymbol{W}_1^{\mathrm{FW}^2} = \boldsymbol{W}_1 - \eta_1 \sum_t g_{1,t}$ be the mini-batch first-layer update used by* `FF-TTT`*, and compare against sequential two-layer TTT with constant steps (η_1, η_2). Under the assumptions of Proposition 4.1,*

$$\boldsymbol{W}_{2,b}^{\mathrm{seq}} - \boldsymbol{W}_2^{\mathrm{FW}^2} = \mathcal{O}(\eta_{\max}^3 b^3), \qquad \boldsymbol{W}_{1,b}^{\mathrm{seq}} - \boldsymbol{W}_1^{\mathrm{FW}^2} = \eta_1^2 \mathcal{A}_{11} + \eta_1 \eta_2 \mathcal{A}_{12} + \mathcal{O}(\eta_{\max}^3 b^3). \tag{39}$$

Proof. The mini-batch $\boldsymbol{W}_2$ endpoint is $\boldsymbol{W}_2 - \eta_2 \sum_t g_{2,t}$. Adding equation 9 and subtracting equation 18 gives the first identity. The first-layer update of `FF-TTT` equals the mini-batch update, so the second identity is equation 8. □

Corollary A.9 states the scope of the method precisely. The output-layer endpoint is second-order accurate, while the first layer carries the second-order residual $\eta_1^2 \mathcal{A}_{11} + \eta_1 \eta_2 \mathcal{A}_{12}$, which the design excludes by keeping $\boldsymbol{W}_1$ at chunk granularity.

Corollary A.10 (Intra-block reads)**.** *Let $o_t^{\mathrm{seq}} = \boldsymbol{W}_{2,t}^{\mathrm{seq}}\,\phi(\boldsymbol{W}_{1,t}^{\mathrm{seq}} q_t)$ be the post-update read of sequential two-layer TTT with constant steps (η_1, η_2), and let $o_t = \boldsymbol{W}_{2,t}^{\mathrm{FW}^2}\widetilde{h}_t^q$ be the read of* FF-TTT*, with $\widetilde{h}_t^q$ from equation 19. Under the assumptions of Proposition 4.1, for every $t \le b$,*

$$o_t - o_t^{\mathrm{seq}} = \eta_1\, \boldsymbol{W}_2\big[\phi'(\hat{u}_t) \odot e_t\big]\,(k_t^\top q_t) + \mathcal{O}(\eta_{\max}^2 t^2). \tag{40}$$

With the untransported query feature $\hat{h}_t$ in place of $\widetilde{h}_t^q$, the leading term is instead $\eta_1 \boldsymbol{W}_2\big[\phi'(\hat{u}_t) \odot (G_{1,\le t}\, q_t)\big] = \mathcal{O}(\eta_1 t)$, where $G_{1,\le t} = G_{1,<t} + g_{1,t}$.

Proof. Query feature. The $\boldsymbol{W}_1$ block of the prefix displacement in the proof of Corollary A.1, taken after token t, gives $\boldsymbol{W}_{1,t}^{\mathrm{seq}} = \boldsymbol{W}_1 - \eta_1 G_{1,\le t} + \mathcal{O}(\eta_{\max}^2 t^2)$. A first-order Taylor expansion of ϕ around $\hat{u}_t$ gives $\phi(\boldsymbol{W}_{1,t}^{\mathrm{seq}} q_t) = \hat{h}_t - \eta_1 \phi'(\hat{u}_t) \odot (G_{1,\le t}\, q_t) + \mathcal{O}(\eta_{\max}^2 t^2)$. Since $\delta\hat{u}_t = -\eta_1 G_{1,<t}\, q_t$ and $G_{1,\le t} - G_{1,<t} = e_t k_t^\top$,

$$\widetilde{h}_t^q - \phi(\boldsymbol{W}_{1,t}^{\mathrm{seq}} q_t) = \eta_1\, \phi'(\hat{u}_t) \odot e_t\, (k_t^\top q_t) + \mathcal{O}(\eta_{\max}^2 t^2).$$

Memory. Transport is causal, so FF-TTT restricted to the first t tokens of the block is FF-TTT on a block of length t. Proposition 4.1 and equation 9, applied with b replaced by t, give $\boldsymbol{W}_{2,t}^{\mathrm{FW}^2} - \boldsymbol{W}_{2,t}^{\mathrm{seq}} = \mathcal{O}(\eta_{\max}^3 t^3)$. Also $\boldsymbol{W}_{2,t}^{\mathrm{FW}^2} = \boldsymbol{W}_2 + \mathcal{O}(\eta_{\max} t)$.

Combination. Write $o_t - o_t^{\mathrm{seq}} = \boldsymbol{W}_{2,t}^{\mathrm{FW}^2}\big(\widetilde{h}_t^q - \phi(\boldsymbol{W}_{1,t}^{\mathrm{seq}} q_t)\big) + \big(\boldsymbol{W}_{2,t}^{\mathrm{FW}^2} - \boldsymbol{W}_{2,t}^{\mathrm{seq}}\big)\phi(\boldsymbol{W}_{1,t}^{\mathrm{seq}} q_t)$. Substitute the two bounds and replace $\boldsymbol{W}_{2,t}^{\mathrm{FW}^2}$ by $\boldsymbol{W}_2$ in the first-order term at cost $\mathcal{O}(\eta_{\max}^2 t)$. Since $t\eta_{\max} \le b\eta_{\max} \le c$, the $\mathcal{O}(\eta_{\max}^3 t^3)$ term is absorbed into $\mathcal{O}(\eta_{\max}^2 t^2)$, which gives equation 40. The untransported case follows from the same expansion with $\delta\hat{u}_t = 0$. □

The $\mathcal{O}(\eta_{\max}^2 t^2)$ remainder in equation 40 contains the within-block second-order $\boldsymbol{W}_1$ terms, the prefix analogues of $\eta_1^2\mathcal{A}_{11}$ and $\eta_1\eta_2\mathcal{A}_{12}$, acting through the query feature, together with the curvature of ϕ. It is excluded by the same design choice as in Corollary A.9.

Remark A.11 (Token-wise step sizes). FF-TTT uses token-wise steps $\eta_{2,t} \le \eta_{\max}$. The arguments above carry over once $\eta_2 g_{2,t}$ is replaced by $\eta_{2,t} g_{2,t}$ throughout. With $\Lambda = \mathrm{diag}(\eta_{2,t})$, the WY series uses $T^{-1} = \sum_k (-N\Lambda)^k$, and its $k = 1$ term becomes $\sum_t \eta_{2,t} H_{22,t}\big[\sum_{i<t} \eta_{2,i} g_{2,i}\big]$. This is exactly the weighted memory interaction produced by the sequential map with the same steps. The feature-transport term likewise becomes $\eta_1 \sum_t \eta_{2,t} H_{21,t}[G_{1,<t}]$, and the remainder bounds hold with η_2 replaced by $\max_t \eta_{2,t}$.

B Additional BEA Interpretation

The main text uses BEA only to show that $\mathcal{A}$ is the leading change in the effective block dynamics. For completeness, this appendix derives the modified vector fields and decomposes their difference into conservative and order-dependent components.

B.1 Modified vector fields

Let $S = \sum_{t=1}^b g_t$ and $H_S = \sum_{t=1}^b H_t$. Consider a generic block map

$$\boldsymbol{W}^+ = \boldsymbol{W}_0 - \eta S + \eta^2 M + \mathcal{O}(b^3\eta^3) \tag{41}$$

and a modified vector field $\widetilde{f} = -\frac{1}{b}S + \eta f_1 + \mathcal{O}(\eta^2)$. Its exact flow over block time $b\eta$ is

$$\boldsymbol{W}(b\eta) = \boldsymbol{W}_0 - \eta S + \eta^2\Big(b f_1 + \tfrac{1}{2} H_S S\Big) + \mathcal{O}(\eta^3), \tag{42}$$

so matching coefficients gives $f_1 = \frac{1}{b}\big(M - \frac{1}{2}H_S S\big)$. The mini-batch map has $M = 0$, and Proposition 3.2 gives $M = \mathcal{A}$ for the sequential map. Hence

$$\widetilde{f}_{\mathrm{mb}} = -\frac{1}{b}S - \frac{\eta}{2b} H_S S + \mathcal{O}(b^2\eta^2), \tag{43}$$

$$\widetilde{f}_{\mathrm{seq}} = -\frac{1}{b}S + \frac{\eta}{b}\Big(\mathcal{A} - \frac{1}{2}H_S S\Big) + \mathcal{O}(b^2\eta^2), \tag{44}$$

and subtracting recovers equation 6. The remainder is $\mathcal{O}(b^2\eta^2)$ because the map remainder is $\mathcal{O}(b^3\eta^3)$ over block time $b\eta$. Since $\mathcal{A} = \mathcal{O}(b^2)$, the leading gap $\frac{\eta}{b}\mathcal{A} = \mathcal{O}(b\eta)$ dominates this remainder whenever $b\eta$ is small.

B.2 Conservative and order-dependent components

The mini-batch field is conservative through this order:

$$\widetilde{f}_{\mathrm{mb}} = -\nabla\widetilde{L}_{\mathrm{mb}} + \mathcal{O}(b^2\eta^2), \qquad \widetilde{L}_{\mathrm{mb}} = \frac{1}{b}\Big[\sum_{t=1}^{b}\ell_t + \frac{\eta}{4}\Big\|\sum_{t=1}^{b} g_t\Big\|^2\Big]. \tag{45}$$

For the sequential field, splitting $H_S S = \sum_{i,t} H_t g_i$ into diagonal, $i < t$, and $i > t$ terms gives

$$\mathcal{A} - \frac{1}{2}H_S S = -\frac{1}{2}\sum_t H_t g_t + \mathcal{B}, \qquad \mathcal{B} \coloneqq \frac{1}{2}\sum_{i<t}\big(H_t g_i - H_i g_t\big). \tag{46}$$

Since $H_t g_t = \nabla(\frac{1}{2}\|g_t\|^2)$,

$$\widetilde{f}_{\mathrm{seq}} = -\nabla\widetilde{L}^{\mathrm{cons}}_{\mathrm{seq}} + \frac{\eta}{b}\mathcal{B} + \mathcal{O}(b^2\eta^2), \qquad \widetilde{L}^{\mathrm{cons}}_{\mathrm{seq}} = \frac{1}{b}\Big[\sum_{t=1}^{b}\ell_t + \frac{\eta}{4}\sum_{t=1}^{b}\|g_t\|^2\Big]. \tag{47}$$

The two conservative parts differ by $\widetilde{L}_{\mathrm{mb}} - \widetilde{L}^{\mathrm{cons}}_{\mathrm{seq}} = \frac{\eta}{2b}\sum_{i<t}\langle g_i, g_t\rangle$. Equivalently, $\mathcal{A}$ admits the exact decomposition

$$\mathcal{A} = \nabla\Big(\frac{1}{2}\sum_{i<t}\langle g_i, g_t\rangle\Big) + \mathcal{B}, \tag{48}$$

where the first term is conservative and $\mathcal{B}$ is the antisymmetric, order-dependent part, which changes sign when the token order is reversed. Mini-batching therefore alters both the finite-step regularization and the directed, order-sensitive component of the dynamics. We keep this result in the appendix because our algorithm does not approximate the two pieces separately. It acts on the layer-wise decomposition of $\mathcal{A}$ instead.

C Exact Layer-wise Interaction Forms

This appendix gives the exact Hessian blocks and token-space factorizations underlying Section 3.2. They are needed to measure all four interactions but not to understand feature transport.

C.1 Hessian blocks

Write $s_t = \phi'(u_t)$ and $c_t = \phi''(u_t)$, and let $V = (V_1, V_2)$ be a parameter-space direction. Define

$$a_t = V_1 k_t, \qquad p_t = s_t \odot a_t, \qquad z_t = V_2 h_t + \boldsymbol{W}_2 p_t, \tag{49}$$

which are the first-order changes in u_t, h_t, and r_t along V. Differentiating $g_{1,t} = e_t k_t^\top$ and $g_{2,t} = r_t h_t^\top$ along V gives

$$(H_t V)_2 = z_t h_t^\top + r_t p_t^\top, \tag{50}$$

$$(H_t V)_1 = \Big[c_t \odot a_t \odot \boldsymbol{W}_2^\top r_t + s_t \odot \big(V_2^\top r_t + \boldsymbol{W}_2^\top z_t\big)\Big] k_t^\top. \tag{51}$$

Restricting V to a single block yields

$$H_{11,t}[V_1] = \big[c_t \odot a_t \odot \boldsymbol{W}_2^\top r_t + s_t \odot \boldsymbol{W}_2^\top \boldsymbol{W}_2 p_t\big] k_t^\top, \tag{52}$$

$$H_{12,t}[V_2] = \big[s_t \odot \big(V_2^\top r_t + \boldsymbol{W}_2^\top V_2 h_t\big)\big] k_t^\top, \tag{53}$$

$$H_{21,t}[V_1] = (\boldsymbol{W}_2 p_t)\, h_t^\top + r_t\, p_t^\top, \tag{54}$$

$$H_{22,t}[V_2] = V_2\, h_t h_t^\top. \tag{55}$$

The interactions follow as $\mathcal{A}_{\alpha\beta} = \sum_{t=1}^{b} H_{\alpha\beta,t}[G_{\beta,<t}]$. Setting $V_1 = G_{1,<t}$ for token t gives $a_t = G_{1,<t}k_t$, so p_t coincides with equation 10, and equation 54 recovers equation 11. $H_{22,t}$ does not depend on the residual or on activation derivatives, and it coincides with the corresponding generalized Gauss–Newton block. The other three blocks depend on feature learning through s_t or c_t.

C.2 TOKEN-SPACE FACTORIZATIONS

Stack the token quantities as columns,

$$K = [k_1, \dots, k_b], \quad \Phi = [h_1, \dots, h_b], \quad R = [r_1, \dots, r_b],$$
$$E = [e_1, \dots, e_b], \quad J = [s_1, \dots, s_b], \quad J^{(2)} = [c_1, \dots, c_b], \tag{56}$$

so that $E = J \odot (\boldsymbol{W}_2^\top R)$. Introduce the strictly causal Gram matrices

$$\Gamma_K = \mathcal{U}(K^\top K), \qquad \Gamma_\Phi = \mathcal{U}(\Phi^\top \Phi), \qquad \Gamma_R = \mathcal{U}(R^\top R), \tag{57}$$

and the prefix quantities

$$A = [a_1, \dots, a_b] = E\,\Gamma_K, \qquad P = [p_1, \dots, p_b] = J \odot A, \tag{58}$$

whose t-th columns are $a_t = G_{1,<t} k_t$ and $p_t = s_t \odot a_t$. The virtual first-layer prefix used by feature transport can therefore be formed entirely in token space. In the notation of equation 16, $\Delta U = -\eta_1 A$ and $\widetilde{\Phi} = \Phi - \eta_1 P$.

Proposition C.1 (Matrix forms of the layer-wise interactions)**.** *The four second-order interactions admit the factorizations*

$$\mathcal{A}_{22} = R\,\Gamma_\Phi\,\Phi^\top, \tag{59}$$
$$\mathcal{A}_{21} = (\boldsymbol{W}_2 P)\,\Phi^\top + R\,P^\top, \tag{60}$$
$$\mathcal{A}_{12} = \Big\{ J \odot \big[\Phi\,\Gamma_R + \boldsymbol{W}_2^\top R\,\Gamma_\Phi\big] \Big\} K^\top, \tag{61}$$
$$\mathcal{A}_{11} = \Big\{ J^{(2)} \odot A \odot (\boldsymbol{W}_2^\top R) + J \odot \big(\boldsymbol{W}_2^\top \boldsymbol{W}_2 P\big) \Big\} K^\top. \tag{62}$$

C.3 PROOF OF PROPOSITION C.1

Proof. For $\mathcal{A}_{22}$, the t-th prefix action is $G_{2,<t} h_t = \sum_{i<t} r_i (h_i^\top h_t)$, which is the t-th column of $R\Gamma_\Phi$. Multiplying by $\Phi^\top$ sums $(G_{2,<t} h_t) h_t^\top = H_{22,t}[G_{2,<t}]$ over t, which proves equation 59.

For $\mathcal{A}_{21}$, the t-th columns of A and P are $a_t = G_{1,<t} k_t$ and $p_t = s_t \odot a_t$. By equation 54, $H_{21,t}[G_{1,<t}] = (\boldsymbol{W}_2 p_t) h_t^\top + r_t p_t^\top$, and summing over t gives equation 60.

For $\mathcal{A}_{12}$, the two prefix products in equation 53 are

$$G_{2,<t}^\top r_t = \sum_{i<t} h_i (r_i^\top r_t), \qquad \boldsymbol{W}_2^\top G_{2,<t} h_t = \boldsymbol{W}_2^\top \sum_{i<t} r_i (h_i^\top h_t). \tag{63}$$

Their stacked forms are $\Phi\Gamma_R$ and $\boldsymbol{W}_2^\top R\Gamma_\Phi$. Elementwise multiplication by J and right multiplication by $K^\top$ give equation 61.

For $\mathcal{A}_{11}$, substitute a_t and p_t into equation 52, stack the hidden-space coefficients, and multiply by $K^\top$. This gives equation 62. □

D FEATURE TRANSPORT FOR THE SWIGLU FAST-WEIGHT NETWORK

Recurrent form. Let $u_{1,t} = \boldsymbol{W}_1 k_t$, $u_{3,t} = \boldsymbol{W}_3 k_t$, and $h_t = \phi(u_{1,t}) \odot u_{3,t}$, where $\phi = \mathrm{SiLU}$. With $\delta_t = \boldsymbol{W}_2^\top r_t$, the entrance gradients are $g_{\alpha,t} = e_{\alpha,t} k_t^\top$ for $\alpha \in \{1, 3\}$, where $e_{1,t} = \phi'(u_{1,t}) \odot u_{3,t} \odot \delta_t$ and $e_{3,t} = \phi(u_{1,t}) \odot \delta_t$. Set $d_{\alpha,t} = -\eta_1 e_{\alpha,t}$, using the shared first-layer step size η_1. The virtual prefixes of the two branches obey

$$\Delta \boldsymbol{W}_{\alpha,t} = \Delta \boldsymbol{W}_{\alpha,t-1} + d_{\alpha,t} k_t^\top, \qquad \Delta \boldsymbol{W}_{\alpha,0} = 0, \tag{64}$$

so token t receives the strictly causal preactivation shifts

$$\delta u_{\alpha,t} = \Delta \boldsymbol{W}_{\alpha,t-1} k_t = \sum_{i<t} d_{\alpha,i}\,(k_i^\top k_t), \qquad \alpha \in \{1, 3\}. \tag{65}$$

By the product rule, the first-order transported feature is

$$\widetilde{h}_t = h_t + \big(\phi'(u_{1,t}) \odot u_{3,t}\big) \odot \delta u_{1,t} + \phi(u_{1,t}) \odot \delta u_{3,t}. \tag{66}$$

Queries are transported with the same prefixes. With $\hat{u}_{\alpha,t} = \boldsymbol{W}_\alpha q_t$ and $\hat{h}_t = \phi(\hat{u}_{1,t}) \odot \hat{u}_{3,t}$, token t receives $\delta \hat{u}_{\alpha,t} = \Delta \boldsymbol{W}_{\alpha,t-1} q_t = \sum_{i<t} d_{\alpha,i}\,(k_i^\top q_t)$, and

$$\widetilde{h}_t^q = \hat{h}_t + \big(\phi'(\hat{u}_{1,t}) \odot \hat{u}_{3,t}\big) \odot \delta \hat{u}_{1,t} + \phi(\hat{u}_{1,t}) \odot \delta \hat{u}_{3,t}. \tag{67}$$

Parallel form. Stack $K = [k_1, \dots, k_b]$, $U_\alpha = [u_{\alpha,1}, \dots, u_{\alpha,b}]$, and $D_\alpha = [d_{\alpha,1}, \dots, d_{\alpha,b}]$, so that $\Phi = \phi(U_1) \odot U_3$. Then

$$\Delta U_\alpha = D_\alpha \mathcal{U}(K^\top K), \qquad \widetilde{\Phi} = \Phi + \big(\phi'(U_1) \odot U_3\big) \odot \Delta U_1 + \phi(U_1) \odot \Delta U_3, \tag{68}$$

where $\mathcal{U}$ keeps entries (i, t) with $i < t$. For the queries, with $Q = [q_1, \dots, q_b]$ and $\hat{U}_\alpha = [\hat{u}_{\alpha,1}, \dots, \hat{u}_{\alpha,b}]$,

$$\Delta \hat{U}_\alpha = D_\alpha \mathcal{U}(K^\top Q), \qquad \widetilde{\Phi}^q = \hat{\Phi} + \big(\phi'(\hat{U}_1) \odot \hat{U}_3\big) \odot \Delta \hat{U}_1 + \phi(\hat{U}_1) \odot \Delta \hat{U}_3. \tag{69}$$

Both branches share the masked Gram matrices $\mathcal{U}(K^\top K)$ and $\mathcal{U}(K^\top Q)$, so stacking $[D_1; D_3]$ computes the key shifts and the query shifts of both branches in one matrix product each.

Chunkwise form. For a kernel chunk c with incoming prefixes $\Delta \boldsymbol{W}_\alpha^{\text{in}}$,

$$\Delta U_{\alpha,c} = \Delta \boldsymbol{W}_\alpha^{\text{in}} K_c + D_{\alpha,c} \mathcal{U}(K_c^\top K_c), \qquad \Delta \boldsymbol{W}_\alpha^{\text{out}} = \Delta \boldsymbol{W}_\alpha^{\text{in}} + D_{\alpha,c} K_c^\top. \tag{70}$$

Query shifts use the same incoming state, $\Delta \hat{U}_{\alpha,c} = \Delta \boldsymbol{W}_\alpha^{\text{in}} Q_c + D_{\alpha,c} \mathcal{U}(K_c^\top Q_c)$. The prefixes reset at each TTT block boundary, where $\boldsymbol{W}_1$ and $\boldsymbol{W}_3$ take their mini-batch steps. We apply stop-gradient to ΔU_α and $\Delta \hat{U}_\alpha$, so the outer-loop gradient does not flow through either transport correction. `FF-TTT` then uses $\widetilde{\Phi}$ in place of Φ in equation 13 and reads with $\widetilde{\Phi}^q$ (Remark A.4). This is lines 7–9 of Algorithm 1 with $r \in \{q, k\}$.

Second-order guarantee. Let $\theta = (\boldsymbol{W}_1, \boldsymbol{W}_3)$ denote the joint first layer, $G_{\theta,<t} = (G_{1,<t}, G_{3,<t})$ its prefix gradient, and $H_{2\theta,t}$ the Hessian block mapping a perturbation of θ to the induced change in $g_{2,t}$. Define $\mathcal{A}_{2\theta} = \sum_t H_{2\theta,t}[G_{\theta,<t}]$.

Corollary D.1 (Feature transport for SwiGLU)**.** *With shared first-layer step η_1, constant η_2, and $\widetilde{\Phi}$ from equation 68,*

$$\boldsymbol{W}_2^{\mathrm{FW}^2} = \boldsymbol{W}_2 - \eta_2 \sum_t g_{2,t} + \eta_1 \eta_2 \, \mathcal{A}_{2\theta} + \eta_2^2 \, \mathcal{A}_{22} + \mathcal{O}(\eta_{\max}^3 b^3), \tag{71}$$

which matches the sequential three-matrix TTT endpoint $\boldsymbol{W}_{2,b}^{\text{seq}}$ through second order.

Proof. The proof of Proposition 4.1 uses only three facts: (a) $g_{2,t} = (\boldsymbol{W}_2 h_t - v_t) h_t^\top$ depends on the first layer only through h_t; (b) $\widetilde{h}_t = h_t - \eta_1 p_t$ exactly, with $p_t = D_\theta h_t[G_{\theta,<t}]$; and (c) $\|p_t\| \le Ct$. Fact (a) holds by the form of equation 20. For (b), the t-th column of ΔU_α is $-\eta_1 \sum_{i<t} e_{\alpha,i} (k_i^\top k_t) = -\eta_1 G_{\alpha,<t} k_t$, and $D_\theta h_t[(V_1, V_3)] = \big(\phi'(u_{1,t}) \odot u_{3,t}\big) \odot V_1 k_t + \phi(u_{1,t}) \odot V_3 k_t$. So equation 68 gives $\widetilde{h}_t = h_t - \eta_1 p_t$, with p_t as in the main text. Fact (c) follows as in Lemma A.8. By (a) and the chain rule, $H_{2\theta,t}[G_{\theta,<t}] = (\boldsymbol{W}_2 p_t) h_t^\top + r_t p_t^\top$, so the proof of Proposition 4.1 applies verbatim with $\mathcal{A}_{21}$ replaced by $\mathcal{A}_{2\theta}$. For the matching claim, apply Corollary A.1 with parameter blocks $(\theta, \boldsymbol{W}_2)$ and $\Omega = \mathrm{diag}(\eta_1 I, \eta_2 I)$. This gives $\boldsymbol{W}_{2,b}^{\text{seq}} - \boldsymbol{W}_{2,b}^{\text{mb}} = \eta_1 \eta_2 \mathcal{A}_{2\theta} + \eta_2^2 \mathcal{A}_{22} + \mathcal{O}(\eta_{\max}^3 b^3)$. □

Corollary A.10 carries over in the same way. With the shared step η_1 and $\widetilde{\Phi}^q$ from equation 69, the leading read error becomes $\eta_1 \boldsymbol{W}_2 \big[(\phi'(\hat{u}_{1,t}) \odot \hat{u}_{3,t}) \odot e_{1,t} + \phi(\hat{u}_{1,t}) \odot e_{3,t}\big](k_t^\top q_t)$, and the untransported read again has an $\mathcal{O}(\eta_1 t)$ leading term.

E THE PSEUDO-ALGORITHM OF FF-TTT

Algorithm 1 FF-TTT (one head). Blocks are processed in order. Block m runs FW^2 from its entrance state $(P^{(m)}, \boldsymbol{W}_2^{(m)})$, and all tokens within a block are processed in parallel.

Require: $X^q, X^k, V \in \mathbb{R}^{C\times b\times d}$ (C blocks of b tokens; rows are tokens); step sizes $\eta^{(1)}, \eta^{(3)}, \eta^{(2)} \in \mathbb{R}^{C\times b}$; initial fast weights $P^{(0)} = [\boldsymbol{W}_1^{(0)}; \boldsymbol{W}_3^{(0)}]$ and $\boldsymbol{W}_2^{(0)}$

Ensure: Outputs $O = [O_0; \ldots; O_{C-1}]$ and final states $P^{(C)}$, $\boldsymbol{W}_2^{(C)}$

for $m = 0, \ldots, C-1$ **do** ▷ entrance state $(P^{(m)}, \boldsymbol{W}_2^{(m)})$

$[A_m^r, B_m^r] \leftarrow X_m^r P^{(m)\top},\ r \in \{q, k\}; \qquad H_m \leftarrow \mathrm{SiLU}(A_m^k) \odot B_m^k$

$F_m \leftarrow \big(V_m - H_m \boldsymbol{W}_2^{(m)\top}\big)\, \boldsymbol{W}_2^{(m)}$ ▷ rows $-(\boldsymbol{W}_2^{(m)\top} r_t)^\top$

$D_m \leftarrow \big[\,\mathrm{diag}(\eta_m^{(1)})\big(F_m \odot \mathrm{SiLU}'(A_m^k) \odot B_m^k\big),\ \mathrm{diag}(\eta_m^{(3)})\big(F_m \odot \mathrm{SiLU}(A_m^k)\big)\big]$

$[\Delta A_m^r, \Delta B_m^r] \leftarrow \mathrm{tril}_{-1}\left(X_m^r (X_m^k)^\top\right) D_m,\ r \in \{q, k\}$

$\widetilde{H}_m^r \leftarrow \mathrm{SiLU}(A_m^r) \odot B_m^r + \mathrm{sg}\left(\mathrm{SiLU}'(A_m^r) \odot B_m^r \odot \Delta A_m^r + \mathrm{SiLU}(A_m^r) \odot \Delta B_m^r\right)$

$(O_m, \boldsymbol{W}_2^{(m+1)}) \leftarrow \mathrm{DeltaWY}\left(\widetilde{H}_m^q, \widetilde{H}_m^k, V_m, \eta_m^{(2)}; \boldsymbol{W}_2^{(m)}\right)$

$P^{(m+1)} \leftarrow P^{(m)} + D_m^\top X_m^k$

end for

F TRAINING DETAILS

Training Setup. We pretrain models from scratch at three scales—340M, 760M, and 1B parameters—using Chinchilla-style compute-optimal budgets of approximately 20 training tokens per parameter (Hoffmann et al., 2022). All models are trained on FineWeb-Edu (Penedo et al., 2024). We construct a non-overlapping split from `sample-100BT` and use a nominal training budget of 20B tokens, for models require less than 20B tokens, we sampled subsets from it. We use the 32K-vocabulary tokenizer released with `fla-hub/transformer-1.3B-100B`[1]. Individual sequences are limited to 4,096 tokens and packed into 65,536-token training blocks, yielding a global batch size of 524,288 tokens across eight GPUs. We train in `bfloat16` with FSDP using fused AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-15}$, and weight decay 0.1), a peak learning rate of 10^{-3}, 1,024 warmup steps followed by cosine decay, and gradient clipping at 1.0. Our training pipeline is implemented using Flame (Zhang & Yang, 2025), which builds on TorchTitan (Liang et al., 2025), while the model operators and hardware-efficient Triton kernels are implemented with Flash Linear Attention (FLA) (Yang & Zhang, 2024).

G MODEL ARCHITECTURE AND IMPLEMENTATION

Architecture. For a fair comparison with `LaCT`, we follow the official `LaCT` language-model architecture[2] and keep all components outside the fast-weight sequence mixer unchanged. Specifically, our models use the same vocabulary, token embeddings, model width, number of layers, SwiGLU feed-forward networks, RMSNorm layers, rotary positional embeddings, and causal sliding-window attention as their corresponding `LaCT` baselines. The sliding-window attention and the fast-weight branch operate in parallel and share the same QKV projection. Their outputs are summed before a common output projection and residual connection.

The only architectural change is that we replace the original `LaCT` fast-weight module with our `FF-TTT` module. `LaCT` uses four fast-weight heads, whose dimensions increase with model size, whereas `FF-TTT` fixes the fast-weight head dimension to 256 and scales the number of heads with the model width. Both methods use a large-chunk size of 2,048 tokens. Because the two fast-weight modules use different state parameterizations, their exact parameter counts are close but not necessarily identical. All reported comparisons therefore use matched outer architectures and training settings, while reporting the exact parameter counts of both models.

All models use a vocabulary size of 32,000, untied input and output embeddings, SwiGLU feed-forward layers, and a causal sliding-window size of 2,048. The attention head dimension is fixed to

[1] `https://huggingface.co/fla-hub/transformer-1.3B-100B`

[2] `https://github.com/a1600012888/LaCT/tree/main/lact_llm`

Table 3: Model configurations used for the comparison with LaCT. "FW heads" denotes the number and dimension of fast-weight heads. Parameter counts are the exact trainable parameter counts rather than the nominal model-size labels.

Model scale	FF-TTT params	LaCT params	Layers	d_{model}	d_{FFN}	FF-TTT FW heads	LaCT FW heads	SWA heads / window
340M	383.9M	383.9M	24	1024	2816	4×256	4×256	16×64 / 2048
760M	793.3M	797.6M	24	1536	4096	6×256	4×384	24×64 / 2048
1B	1068.8M	1076.4M	24	1792	4864	7×256	4×448	28×64 / 2048

64, while the FF-TTT fast-weight head dimension is fixed to 256. The fast-weight macro-chunk size is 2,048 for all three model scales.

Kernel implementation. We implement the FF-TTT forward and backward kernels ourselves. Our implementation builds on the open-source infrastructure and optimized primitives provided by Flash Linear Attention (FLA)[3] and the official LaCT implementation. In particular, we reuse and adapt the chunked Delta-rule primitives from FLA and the large-chunk SwiGLU fast-weight design from LaCT, while implementing the FF-TTT-specific feature transport, fast-weight state transitions, two-pass execution, and corresponding backward kernels. Thus, our experiments retain the same surrounding LLM implementation as LaCT while replacing only the fast-weight operator and its optimized kernel implementation.

Ethics Statement and Limitations

Ethics. This work studies the optimization dynamics of test-time training layers and proposes a correction to their mini-batch update. It involves no human subjects or personal data, and all experiments use publicly available datasets and models. Because our method targets the efficiency and fidelity of sequence models rather than any specific application, we do not foresee risks beyond those common to language models in general, such as misuse for generating misleading content. Making long-context models cheaper to train may lower barriers to both beneficial and harmful uses.

Limitations. First, our analysis and method cover two-layer MLP fast weights with plain gradient descent in the inner loop. Deeper fast-weight networks and inner optimizers with momentum, weight decay, or orthogonalized updates (e.g., Muon) would need their own transport rules. Second, feature transport is a first-order correction: it removes the $\mathcal{O}(\eta t)$ intra-block error but leaves an $\mathcal{O}(\eta_{\max}^2 t^2)$ residual, so its guarantees weaken when ηb is large, which is precisely the large-chunk regime favored for hardware efficiency. Third, the method adds $\mathcal{O}(b^2 d)$ computation per block, which becomes non-negligible for very large blocks.

AI Use Statement

The use of AI assistants was limited to formatting and grammar checking. Research design, execution, results collection and presentation were done by authors themselves. All technical content and claims are the responsibility of the authors.

[3] https://github.com/fla-org/flash-linear-attention