Error-Free Linear Attention is a Free Lunch: Exact Solution from Continuous-Time Dynamics

Jingdi Lei[1], **Di Zhang**[2], **Soujanya Poria**[1]

[1]Nanyang Technological University, [2]Fudan University

Linear-time attention and State Space Models (SSMs) promise to solve the quadratic cost bottleneck in long-context language models employing softmax attention. We introduce **Error-Free Linear Attention (EFLA)**, a numerically stable, fully parallelism and generalized formulation of the delta rule. Specifically, we formulate the online learning update as a continuous-time dynamical system and prove that its exact solution is not only attainable but also computable in linear time with full parallelism. By leveraging the *rank-1* structure of the dynamics matrix, we directly derive the exact closed-form solution effectively corresponding to the infinite-order Runge–Kutta method. This attention mechanism is theoretically free from error accumulation, perfectly capturing the continuous dynamics while preserving the linear-time complexity. Through an extensive suite of experiments, we show that EFLA enables robust performance in noisy environments, achieving lower language modeling perplexity and superior downstream benchmark performance than DeltaNet without introducing additional parameters. Our work provides a new theoretical foundation for building high-fidelity, scalable linear-time attention models.

Github: `https://github.com/declare-lab/EFLA`
Correspondance: Di Zhang (`di.zhang@ustc.edu`)
Date: December 16, 2025

1 Introduction

As large language models (LLMs) evolve into increasingly capable agents (Yao et al., 2022; Team et al., 2025b; Google, 2025; OpenAI, 2025), the efficiency of inference computation has emerged as a critical bottleneck (Dao et al., 2022; Kwon et al., 2023; Kim et al., 2024). This challenge becomes particularly acute in demanding scenarios such as long-context processing and reinforcement learning (RL) environments (Guo et al., 2025; Lai et al., 2025) where models are required to handle extended reasoning trajectories or engage in complex tool-use interactions (Lightman et al., 2023; Yao et al., 2023), the quadratic time complexity (Vaswani et al., 2017) inherent in standard attention mechanisms leads to severe inefficiencies. These inefficiencies introduce substantial computational overhead, significantly constraining model throughput, scalability to long contexts, and real-time interactivity (Liu et al., 2023; Jiang et al., 2024; Katharopoulos et al., 2020).

This has led to a surge of research in linear-time attention methods, aiming to approximate or reformulate the attention operation in sub-quadratic time. Prior works like Mamba-2 (Dao and Gu, 2024), DeltaNet (Schlag et al., 2021; Yang et al., 2024b) connect attention mechanisms with continuous-time systems. They bridge modern sequence modeling with the mathematical foundations of control theory and signal processing. We observe that this type of linear attention is not merely an approximation, but also a suboptimal modeling of continuous-time dynamics, a discretization of an ODE system, analogous to a physical system with exponential decay and input injection. Under this interpretation, we formalize the classical linear attention method as a continuous-time dynamical system. Thus, attention can be understood as solving this ODE via numerical integration methods such as the Euler scheme (Euler, 1792). From an analytical perspective, linear-attention formulations implicitly reduce to a first-order Euler discretization, which limits their precision. While computationally simple, Euler discretization introduces truncation errors and suffers from stability issues. This explains why linear attention often exhibits instability under long sequences or large decay rates, it is numerically integrating a stiff ODE with an insufficient integration scheme. Several models have tried to improve the Euler update and mitigate the error accumulation by introducing decay factors or gating functions (Dao and Gu, 2024; Ma et al., 2022; Sun et al., 2023a), or adaptive forgetting coefficients (Yang et al., 2023, 2024a; Team et al., 2025a). While these methods stabilize training and improve long-term retention, they remain heuristic corrections to an inherently low-order numerical approximation. However, these low-order methods cannot eliminate the discretization error itself, they merely rescale or damp its effect.

Unlike prior works that rely on approximations, we propose **EFLA**, a principled approach to eliminate discretization errors by solving the underlying ODE exactly. This results in a solution that is both analytically tractable and computationally efficient. This exact closed-form solution can be mathematically interpreted as the infinite-order Runge–Kutta (RK-∞) (Runge, 1895; Kutta, 1901) limit or the general solution of a first-order ODE. In other words, it pushes the approximation order to infinity, yielding a continuous-time, error-free formulation of linear attention. This exact integration not only ensures numerical stability but also establishes a theoretical bridge between linear attention and modern state-space models like Mamba. By bypassing the limitations of Euler-based approximations, our computable analytic solution offers a fundamental path toward high-fidelity attention mechanisms. Empirically, we demonstrate that **EFLA** exhibits superior robustness in noisy environments and achieves significantly accelerated convergence compared to baseline methods. Furthermore, it consistently outperforms DeltaNet across a diverse set of downstream benchmarks, validating the practical efficacy and scalability of our error-free formulation.

Our contributions are summarized as follows:

- **Precise Identification of Error Sources in Existing Linear Attention:** We analyze the numerical error in mainstream linear attention methods and point out that the core limitation lies in the low-order discretization of an underlying continuous-time process. These approximations introduce significant truncation errors and instability, especially in long-context scenarios.
- **Reformulating Linear Attention as a Continuous-Time Dynamical System:** By treating the online-learning update as a first-order ordinary differential equation (ODE), we reconstruct it from the perspective of continuous-time dynamics.
- **Deriving an Exact Closed-Form Solution with Linear-Time Complexity:** Leveraging the *rank-1* property of the dynamics matrix, we theoretically derive an exact, closed-form solution to the continuous-time ODE governing linear attention. Importantly, our formulation maintains the desirable linear time complexity, while eliminating numerical integration errors.

2 Background

2.1 Scaled Dot-Product Attention

Given queries $\mathbf{Q} \in \mathbb{R}^{n\times d}$, keys $\mathbf{K} \in \mathbb{R}^{n\times d}$, and values $\mathbf{V} \in \mathbb{R}^{n\times d}$, the scaled dot-product attention (Vaswani et al., 2017) is defined as

$$\mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}\right)\mathbf{V}, \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the additive causal mask (zeros on and below the diagonal, and $-\infty$ above).

2.2 Linear Attention

Linear Attention as Online Learning. Linear attention (Katharopoulos et al., 2020) maintains a matrix-valued recurrent state that accumulates key–value associations:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{k}_t\mathbf{v}_t^\top, \quad \mathbf{o}_t = \mathbf{S}_t^\top \mathbf{q}_t. \tag{2}$$

From the fast-weight perspective (Schlag et al., 2021), $\mathbf{S}_t$ serves as an associative memory storing transient mappings from keys to values. This update can be viewed as performing gradient descent on an unbounded correlation objective:

$$\mathcal{L}_t(\mathbf{S}) = -\langle \mathbf{S}^\top \mathbf{k}_t, \mathbf{v}_t\rangle, \tag{3}$$

which continually reinforces recent key–value pairs without any forgetting mechanism. However, such an objective lacks a criterion for erasing old memories; consequently, the accumulated state grows unbounded, leading to interference over long contexts.

DeltaNet: A Reconstruction Loss Perspective. DeltaNet reinterprets this recurrence as online gradient descent on a reconstruction loss objective:

$$\mathcal{L}_t(\mathbf{S}) = \tfrac{1}{2}\|\mathbf{S}^\top \mathbf{k}_t - \mathbf{v}_t\|^2. \tag{4}$$

Taking a gradient step with learning rate β_t yields the update rule:

$$\mathbf{S}_t = \mathbf{S}_{t-1} - \beta_t \nabla_{\mathbf{S}} \mathcal{L}_t(\mathbf{S}_{t-1}) = (\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top)\mathbf{S}_{t-1} + \beta_t \mathbf{k}_t \mathbf{v}_t^\top. \tag{5}$$

This classical *delta rule* treats $\mathbf{S}$ as a learnable associative memory that continually corrects itself toward the mapping $\mathbf{k}_t \mapsto \mathbf{v}_t$. The *rank-1* update structure, equivalent to a generalized Householder transformation, facilitates hardware-efficient chunkwise parallelization (Bischof and Van Loan, 1987; Yang et al., 2024b).

2.3 Euler and Runge-Kutta Methods in ODE

Numerical Solutions to ODEs. Given a first-order ordinary differential equation (ODE) form $\frac{d\mathbf{S}(t)}{dt} = f(t, \mathbf{S}(t))$, numerical methods aim to approximate the solution $\mathbf{S}(t)$ at discrete time point t.

Euler Method. The Explicit Euler method (Euler, 1792) is the most fundamental numerical integration scheme. It approximates the state at the next time step using the derivative at the current position:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \beta_t \cdot f(t-1, \mathbf{S}_{t-1}), \tag{6}$$

where β_t represents the integration step size. While computationally efficient, Euler discretization is a first-order method with a local truncation error of $\mathcal{O}(\beta_t^2)$. Existing linear attention models, such as DeltaNet (Schlag et al., 2021), implicitly adopt this formulation. However, due to its low-order nature, Euler integration often suffers from numerical instability and error accumulation, particularly when integrating stiff dynamics over long sequences.

Runge-Kutta Methods. To achieve higher precision, the Runge-Kutta (RK) family (Runge, 1895; Kutta, 1901; Butcher, 1996, 2016) of methods estimates the future state by aggregating multiple slope estimates (stages) within a single step. For a general N-th order RK method, the update is given by a weighted sum of intermediate derivatives:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \beta_t \sum_{i=1}^{N} c_i k_i, \tag{7}$$

where k_i represents the slope at the i-th stage.

3 Error-Free Linear Attention

3.1 Numerical Approximations.

We begin by revisiting DeltaNet (Schlag et al., 2021), which formulates linear attention as online gradient descent on a reconstruction objective:

$$\mathcal{L}_t(\mathbf{S}) = \frac{1}{2}\|\mathbf{S}^\top \mathbf{k}_t - \mathbf{v}_t\|^2. \tag{8}$$

Applying a single gradient descent step with learning rate β_t yields:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \beta_t \left(-\mathbf{k}_t \mathbf{k}_t^\top \mathbf{S}_{t-1} + \mathbf{k}_t \mathbf{v}_t^\top\right). \tag{9}$$

To formalize the underlying dynamics, we define the dynamics matrix $\mathbf{A}_t = \mathbf{k}_t\mathbf{k}_t^\top$ and the input forcing term $\mathbf{b}_t = \mathbf{k}_t\mathbf{v}_t^\top$. Since the input data arrives as a discrete sequence, we model the continuous signal using the Zero-Order Hold (ZOH) (Iserles, 2009) assumption. Under this physically grounded assumption for digital systems, the time-varying matrices $\mathbf{A}(t)$ and $\mathbf{b}(t)$ become piecewise constant. The system evolves according to a first-order ODE:

$$\frac{d\mathbf{S}(t)}{dt} = -\mathbf{A}_t\mathbf{S}(t) + \mathbf{b}_t. \tag{10}$$

In this framework, standard DeltaNet corresponds to the first-order Explicit Euler discretization. To mitigate the discretization errors inherent in this low-order scheme, one might resort to higher-order solvers. For instance, the second-order (**RK-2**) and fourth-order (**RK-4**) Runge-Kutta updates are given by (see Appendix E for details):

RK-2:

$$\mathbf{S}_t = \left(\mathbf{I} - \beta_t\mathbf{A}_t + \frac{1}{2}\beta_t^2\mathbf{A}_t^2\right)\mathbf{S}_{t-1} + \beta_t\left(\mathbf{I} - \frac{1}{2}\beta_t\mathbf{A}_t\right)\mathbf{b}_t. \tag{11}$$

RK-4:

$$\mathbf{S}_t = \left(\sum_{n=0}^{4}\frac{(-\beta_t\mathbf{A}_t)^n}{n!}\right)\mathbf{S}_{t-1} + \beta_t\left(\sum_{n=0}^{3}\frac{(-\beta_t\mathbf{A}_t)^n}{(n+1)!}\right)\mathbf{b}_t. \tag{12}$$

Based on the same principle, by truncating the Runge–Kutta expansion, we obtain an N-th order form (see Appendix E for details):

$$\mathbf{S}_t = \left[\sum_{n=0}^{N}\frac{(-\beta_t\mathbf{A}_t)^n}{n!}\right]\mathbf{S}_{t-1} + \beta_t\left[\sum_{n=0}^{N-1}\frac{(-\beta_t\mathbf{A}_t)^n}{(n+1)!}\right]\mathbf{b}_t. \tag{13}$$

As the order N increases, the numerical approximation becomes progressively more accurate. Taking the limit as $N \to \infty$, the truncated series converges to their analytical limits:

$$\lim_{N\to\infty}\sum_{n=0}^{N}\frac{(-\beta_t\mathbf{A}_t)^n}{n!} = e^{-\beta_t\mathbf{A}_t}, \tag{14}$$

$$\lim_{N\to\infty}\sum_{n=0}^{N}\beta_t\frac{(-\beta_t\mathbf{A}_t)^n}{(n+1)!}\mathbf{b}_t = \int_0^{\beta_t} e^{-(\beta_t-\tau)\mathbf{A}_t}\mathbf{b}_t d\tau. \tag{15}$$

Thus, the truncated polynomial expansions converge to the exact analytic form of a matrix exponential and its associated integral, yielding:

$$\mathbf{S}_t = e^{-\beta_t\mathbf{A}_t}\mathbf{S}_{t-1} + \int_0^{\beta_t} e^{-(\beta_t-\tau)\mathbf{A}_t}\,\mathbf{b}_t\,d\tau. \tag{16}$$

This expression represents the general solution of the first-order ODE as shown in Eq. 10, and corresponds to the infinite-order (error-free) limit of the Runge–Kutta family[1].

[1]This analytical solution can also be directly obtained by applying the general solution method for ordinary differential equations. The detailed derivation of this closed-form expression is provided in Appendix D.

3.2 The "Aha!" Moment: Efficient Computation via rank-1 Property

While the infinite-order solution eliminates discretization error, naively evaluating the matrix exponential $e^{-\beta_t \mathbf{A}_t}$ for a general matrix typically requires $\mathcal{O}(d^3)$ complexity (Gu et al., 2020). We bypass this computational bottleneck by leveraging the *rank-1* structure of the dynamics matrix $\mathbf{A}_t = \mathbf{k}_t\mathbf{k}_t^\top$, which allows the exponential to be computed in linear time.

We observe that $\mathbf{A}_t$ satisfies the idempotence-like property $\mathbf{A}_t^n = \lambda_t^{n-1}\mathbf{A}_t$ for $n \geq 1$, where $\lambda_t = \mathbf{k}_t^\top \mathbf{k}_t$ (see Appendix C for the proof). This property allows us to collapse the Taylor series of the matrix exponential into a computable closed form:

$$e^{-\beta_t \mathbf{A}_t} = \mathbf{I} + \sum_{n=1}^{\infty} \frac{(-\beta_t)^n}{n!}\mathbf{A}_t^n = \mathbf{I} - \frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t}\mathbf{A}_t. \tag{17}$$

Substituting this transition operator into the integral term $\int_0^{\beta_t} e^{-(\beta_t - \tau)\mathbf{A}_t}\mathbf{b}_t \, d\tau$ yields the exact input injection:

$$\begin{aligned} \mathbf{I}_t &= \int_0^{\beta_t} \left(\mathbf{I} - \frac{1 - e^{-\lambda_t(\beta_t - \tau)}}{\lambda_t}\mathbf{A}_t\right)\mathbf{b}_t \, d\tau \\ &= \beta_t \mathbf{b}_t - \frac{\mathbf{A}_t\mathbf{b}_t}{\lambda_t}\left(\beta_t - \frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t}\right). \end{aligned} \tag{18}$$

Crucially, since $\mathbf{b}_t = \mathbf{k}_t\mathbf{v}_t^\top$ and $\mathbf{A}_t = \mathbf{k}_t\mathbf{k}_t^\top$, we have $\mathbf{A}_t\mathbf{b}_t = \lambda_t\mathbf{b}_t$. This algebraic relationship allows for significant simplification of the integral term:

$$\mathbf{I}_t = \beta_t\mathbf{b}_t - \beta_t\mathbf{b}_t + \frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t}\mathbf{b}_t = \frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t}\mathbf{b}_t. \tag{19}$$

Combining these results, the final Error-Free Linear Attention update rule is given by:

$$\mathbf{S}_t = \left(\mathbf{I} - \frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t}\mathbf{k}_t\mathbf{k}_t^\top\right)\mathbf{S}_{t-1} + \frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t}\mathbf{k}_t\mathbf{v}_t^\top. \tag{20}$$

This update maintains linear time complexity with respect to sequence length, i.e., $\mathcal{O}(Ld^2)$, while capturing the exact continuous-time dynamics.

4 Chunkwise Parallelism Form

We observe that the EFLA update rule shares an identical algebraic structure with DeltaNet. Given this structural equivalence, we can seamlessly adapt the hardware-efficient parallelization strategies originally developed for DeltaNet (Yang et al., 2024b). In this section, we derive the chunkwise parallel formulation specifically for EFLA.

To derive the chunkwise parallel form, we first unroll the recurrence relation. Denoting $\frac{1 - e^{-\beta_t\lambda_t}}{\lambda_t} = \alpha_t$, the state update becomes:

$$\mathbf{S}_t = (\mathbf{I} - \alpha_t\mathbf{k}_t\mathbf{k}_t^\top)\mathbf{S}_{t-1} + \alpha_t\mathbf{k}_t\mathbf{v}_t^\top = \sum_{i=1}^{t}\left(\prod_{j=i+1}^{t}(\mathbf{I} - \alpha_j\mathbf{k}_j\mathbf{k}_j^\top)\right)\alpha_i(\mathbf{k}_i\mathbf{v}_i^\top). \tag{21}$$

Then we can define the following variables:

$$\mathbf{P}_i^j = \prod_{t=i}^{j}(I - \alpha_t\mathbf{k}_t\mathbf{k}_t^\top), \qquad \mathbf{H}_i^j = \sum_{t=i}^{j}\mathbf{P}_{t+1}^j\alpha_t\mathbf{k}_t\mathbf{v}_t^\top \tag{22}$$

where $\mathbf{P}_i^j = \mathbf{I}$ when $i > j$. $\mathbf{P}_i^j$ can be considered as decay factor applied to $\mathbf{S}_i$ to obtain $\mathbf{S}_j$ and $\mathbf{H}_i^j$ is an accumulation term to $\mathbf{S}_j$ from token i. The Chunkwise can be written as follows:

$$\mathbf{S}_{[t]}^r = \mathbf{P}_{[t]}^r \mathbf{S}_{[t]}^0 + \mathbf{H}_{[t]}^r \tag{23}$$

where we define the chunkwise variables $\mathbf{S}_{[t]}^r = \mathbf{S}_{tC+r}$, $\mathbf{P}_{[t]}^r = \mathbf{P}_{tC+1}^{tC+r}$, $\mathbf{H}_{[t]}^r = \mathbf{H}_{tC+1}^{tC+r}$. Here we have $\frac{L}{C}$ chunks of size C. We can use induction to derive the WY representations of $\mathbf{P}_{[t]}^r$ and $\mathbf{H}_{[t]}^r$:

$$\mathbf{P}_{[t]}^r = \mathbf{I} - \sum_{i=1}^{r} \mathbf{k}_{[t]}^i \mathbf{w}_{[t]}^{i\top}, \qquad \mathbf{H}_{[t]}^r = \sum_{i=1}^{r} \mathbf{k}_{[t]}^i \mathbf{u}_{[t]}^{i\top} \tag{24}$$

$$\mathbf{w}_{[t]}^{r\top} = \alpha_{[t]}^r \left(\mathbf{k}_{[t]}^{r\top} - \sum_{i=1}^{r-1} (\mathbf{k}_{[t]}^{r\top} \mathbf{k}_{[t]}^i) \mathbf{w}_{[t]}^{i\top} \right), \qquad \mathbf{u}_{[t]}^{r\top} = \alpha_{[t]}^r \left(\mathbf{v}_{[t]}^{r\top} - \sum_{i=1}^{r-1} (\mathbf{k}_{[t]}^{r\top} \mathbf{k}_{[t]}^i) \mathbf{u}_{[t]}^{i\top} \right) \tag{25}$$

subsequently, we can obtain the chunk-level recurrence for states and outputs:

$$\mathbf{S}_{[t]}^r = \mathbf{S}_{[t]}^0 - \left((\sum_{i=1}^{r} \mathbf{k}_{[t]}^i \mathbf{w}_{[t]}^{i\top}) \mathbf{S}_{[t]}^0 \right) + \sum_{i=1}^{r} \mathbf{k}_{[t]}^i \mathbf{u}_{[t]}^{i\top} = \mathbf{S}_{[t]}^0 + \sum_{i=1}^{r} \mathbf{k}_{[t]}^i (\mathbf{u}_{[t]}^{i\top} - \mathbf{w}_{[t]}^{i\top} \mathbf{S}_{[t]}^0) \tag{26}$$

$$\mathbf{o}_{[t]}^r = \mathbf{S}_{[t]}^{r\top} \mathbf{q}_{[t]}^r = \mathbf{S}_{[t]}^{0\top} \mathbf{q}_{[t]}^r + \sum_{i=1}^{r} (\mathbf{u}_{[t]}^i - \mathbf{S}_{[t]}^{0\top} \mathbf{w}_{[t]}^i)(\mathbf{k}_{[t]}^{i\top} \mathbf{q}_{[t]}^i) \tag{27}$$

letting $\mathbf{S}_{[t]} = \mathbf{S}_{[t]}^0$, the above can be simplified to matrix notations:

$$\mathbf{S}_{[t]} = \mathbf{S}_{[t]} + \mathbf{K}_{[t]}^\top \left(\mathbf{U}_{[t]} - \mathbf{W}_{[t]} \mathbf{S}_{[t]} \right) \tag{28}$$

$$\mathbf{O}_{[t]} = \mathbf{Q}_{[t]} \mathbf{S}_{[t]} + \left(\mathbf{Q}_{[t]} \mathbf{K}_{[t]}^\top \odot \mathbf{M} \right) \left(\mathbf{U}_{[t]} - \mathbf{W}_{[t]} \mathbf{S}_{[t]} \right), \tag{29}$$

where $\square_{[t]} = \square_{[t]}^{1:C} \in \mathbb{R}^{C \times d}$ for $\square \in \{\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{O}, \mathbf{U}, \mathbf{W}\}$ defines the chunkwise matrices that are formed from stacking the $\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t, \mathbf{o}_t, \mathbf{u}_t, \mathbf{w}_t$ vectors and $\mathbf{M}$ is the lower triangular causal mask.

Finally, we can apply the UT transform (Joffrain et al., 2006) to simplify the recurrence calculations of $\mathbf{u}_{[t]}^r$ and $\mathbf{w}_{[t]}^r$.

$$\mathbf{T}_{[i]} = \left(I + \mathrm{StrictTril}(\mathrm{diag}(\alpha_t) \mathbf{K}_{[i]} \mathbf{K}_{[i]}^\top) \right)^{-1} \mathrm{diag}(\alpha_t) \tag{30}$$

$$\mathbf{W}_{[t]} = \mathbf{T}_{[t]} \mathbf{K}_{[t]}, \qquad \mathbf{U}_{[t]} = \mathbf{T}_{[t]} \mathbf{V}_{[t]} \tag{31}$$

5 Empirical Study

5.1 Numerical Stability and Robustness Verification

As mentioned in Section 3, linear attention mechanisms like DeltaNet employ a first-order approximation. While computationally efficient, they still suffer from error accumulation, particularly when dealing with noisy environments or high-energy inputs. In this section, we conduct the following three tests to demonstrate this limitation:

- **Image Pixel Dropout**. We apply Bernoulli dropout to input tokens with probability p to simulate data corruption. This setting evaluates the model's robustness against information loss and its ability to preserve long-range dependencies despite corrupted input signals.

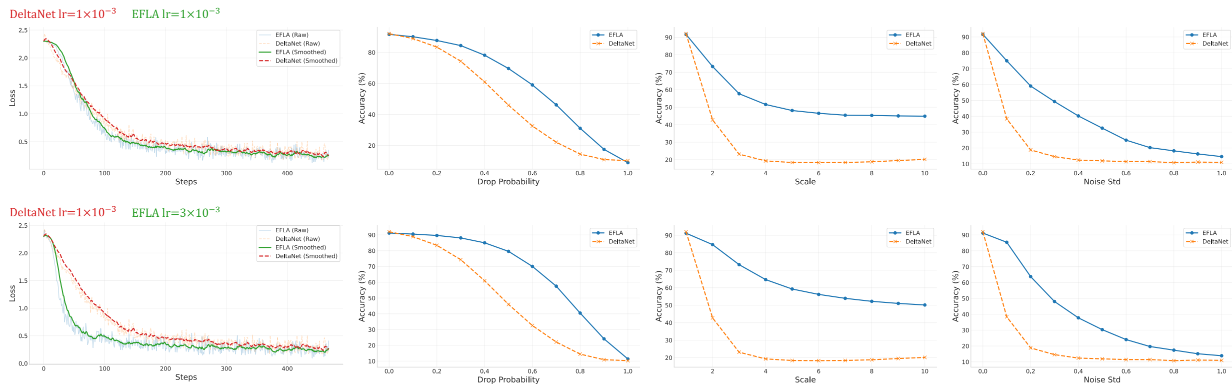

Figure 1 EFLA outperforms DeltaNet in both convergence speed and robustness on sMNIST. The plots illustrate training dynamics and robustness against dropout, scale intensity, and additive noise. Notably, EFLA maintains significantly higher accuracy as interference intensity increases, particularly when trained with a larger learning rate (bottom row, lr$=3 \times 10^{-3}$; see Appendix B for a detailed discussion on learning rate).

- **OOD Intensity Scaling**. We amplify input signals by a factor to conduct a rigorous stress test on numerical stability. The primary objective of this scaling is to simulate deployment scenarios involving unnormalized or high-variance inputs.
- **Additive Gaussian Noise**. We inject Gaussian noise with varying standard deviations. This setting aims to assess the model's robustness against signal corruption and its ability to filter out random perturbations

We conduct experiments on the pixel-level Sequential MNIST (LeCun et al., 2010) (sMNIST) task, flattening 28×28 images into sequences of length $L = 784$. The model is a Linear Attention Classifier with a hidden dimension of $d = 64$. We compare EFLA against the DeltaNet baseline. DeltaNet employs L_2-normalized queries and keys ($||\mathbf{k}_t|| = 1$), whereas EFLA utilizes unnormalized keys. This allows EFLA to leverage the key norm ($||\mathbf{k}_t||^2$) as a dynamic gate for the exact decay factor. Both models are trained using AdamW with a batch size of 128.

The performance comparisons are illustrated in Figure 1. As the input scale factor increases, DeltaNet exhibits a rapid performance collapse, confirming the vulnerability of Euler approximations to high-energy inputs. In stark contrast, EFLA maintains high accuracy even at large scales, empirically confirming that its exact saturation mechanism effectively mitigates error accumulation and state explosion. Across both the additive noise and dropout benchmarks, EFLA consistently outperforms DeltaNet. Under severe interference conditions, EFLA demonstrates a significantly slower rate of degradation. This indicates that by eliminating discretization errors, EFLA constructs a higher-fidelity memory representation that is intrinsically more resilient to data corruption than the approximation-based methods.

5.2 Language Modeling

Experimental setup. We adapt the same model architecture with Yang et al. (2024b), see Appendix A for hyperparamter settings. We evaluate on Wikitext (Merity et al., 2016) perplexity and a comprehensive suite of zero-shot common sense reasoning tasks, including LAMBADA (Paperno et al., 2016), PiQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-easy (ARC-e), 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 report the perplexity (lower is better) for Wikitext and LAMBADA, and accuracy (higher is better) for other downstream tasks.

Main Results. Our main language modeling results are shown in Table 1. With an identical training budget of 8B tokens, EFLA consistently outperforms the DeltaNet baseline across the majority of tasks. For instance, on LAMBADA, which evaluates the prediction of the final word based on broad context, EFLA

Model	Perplexity (↓)		Accuracy (↑)									
	Wiki. ppl	LMB. ppl	LMB. acc	PIQA acc	Hella. acc_n	Wino. acc	ARC-e acc	ARC-c acc_n	BoolQ acc	OBQA acc	SciQ acc	Avg.
340M Parameters												
DeltaNet	38.09	96.26	22.5	60.7	30.1	**51.9**	40.4	22.1	53.0	16.2	71.9	40.9
EFLA	**37.01**	**81.28**	**23.9**	**61.9**	**31.1**	51.3	**41.5**	**22.5**	**60.4**	**17.2**	**73.3**	**42.6**
1.3B Parameters												
DeltaNet	31.82	69.27	25.9	62.3	32.1	**53.3**	42.0	**23.1**	58.9	17.0	73.0	43.0
EFLA	**31.48**	**61.83**	**26.1**	**63.8**	**32.4**	52.3	**42.7**	22.9	**60.4**	**17.8**	**77.0**	**43.9**

Table 1 Main language modeling results compared with DeltaNet. **Perplexity:** Lower (↓) is better. **Accuracy:** Higher (↑) is better. Best results are bolded.

achieves a significantly lower perplexity of 81.28 (vs. 96.26 for DeltaNet) alongside a higher accuracy of 23.9%. Notably, on the BoolQ, EFLA improves accuracy by a substantial margin (+7.4% absolute). We further assessed the scalability of our approach using 1.3B parameter models. As detailed in the bottom of Table 1, even at an intermediate checkpoint of 16B tokens, where models have not yet reached full convergence[2], EFLA still establishes a distinct performance lead. We anticipate this performance gap to widen as training proceeds toward the full token budget. These results empirically validate that by eliminating the discretization error inherent in Euler-based methods, EFLA maintains higher fidelity of historical information over long sequences, a capability critical for complex reasoning.

6 Analysis of Memory Dominance

To understand the mechanism govering memory retention in EFLA, we analyze the spectral properties of the *rank-1* dynamics matrix $\mathbf{A}_t = \mathbf{k}_t\mathbf{k}_t^\top$. We show that the key norm $||\mathbf{k}_t||^2$ act as a dynamic gate, regulating the trade-off between forgetting and retention.

Spectral Decomposition and Exact Decay. Since $\mathbf{A}_t$ is symmetric and *rank-1*, it possesses a single non-zero eigenvalue $\lambda_t = ||\mathbf{k}_t||^2$ while remaining eigenvalues are zero. Leveraging this property, the matrix exponential in Eq. 14 admits a simplified closed-form:

$$e^{-\beta_t \mathbf{A}_t} = \mathbf{I} - \frac{1 - e^{-\beta_t \lambda_t}}{\lambda_t}\mathbf{k}_t\mathbf{k}_t^\top. \tag{32}$$

This operator induces a directional decay, it contracts the component of memory state S_{t-1} aligned with $\mathbf{k}_t$ by a factor of $e^{-\beta_t\lambda_t}$. λ_t serves as a spectral gate: strong input signals (large key norms) cause rapid exponential decay along $\mathbf{k}_t$, effectively clearing the memory slot for new information, while weak signals result in a slower, linear decay governed by β_t, thereby prioritizing the retention of historical context.

Asymptotic Connection to Delta Rule. In the regime of vanishing key norms, i.e. $\lambda_t \to 0$, the exponential term can be linearized via a first-order Taylor expansion:

$$\lim_{\lambda_t \to 0} \left(\mathbf{I} - \beta_t\mathbf{k}_t\mathbf{k}_t^\top\right)\mathbf{S}_{t-1} + \beta_t\mathbf{k}_t\mathbf{v}_t^\top, \tag{33}$$

[2]The results for the 1.3B model are preliminary due to limited computational resources. We are currently training the model and will update the paper with final results soon.

In this case, the update rule asymptotically recovers to the delta rule, indicating that delta-rule linear attention serves as a first-order approximation of EFLA, valid only when the dynamics are non-stiff (i.e., small λ_t). In contrast, EFLA remains robust regardless of signal magnitude due to its exact integration.

7 Related Works

7.1 Linear Attention as Low-Order Numerical Integrators

Early linear-time attention mechanisms approximate softmax attention by replacing the normalized kernel with a feature map that enables associative accumulation of key–value statistics in a recurrent state. Linear Transformers (Katharopoulos et al., 2020) and Performer (Choromanski et al., 2020) rewrite causal attention as a running sum of outer products,

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \phi(\mathbf{k}_t)\phi(\mathbf{v}_t)^\top, \quad \mathbf{o}_t = \mathbf{S}_t^\top \phi(\mathbf{q}_t),$$

yielding an RNN-like formulation of attention that scales linearly in sequence length. Schlag et al. (Schlag et al., 2021) interpret such mechanisms as fast-weight programmers, where the matrix state $\mathbf{S}_t$ implements a dynamic associative memory updated via low-rank modifications.

DeltaNet (Yang et al., 2024b) makes this perspective explicit by viewing $\mathbf{S}_t$ as the parameter of an online regression problem. At each time step, it minimizes a reconstruction loss

$$\mathcal{L}_t(\mathbf{S}) = \tfrac{1}{2}\|\mathbf{S}^\top \mathbf{k}_t - \mathbf{v}_t\|^2$$

via a single gradient step with learning rate β_t, leading to the delta-rule update

$$\mathbf{S}_t = (\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top)\mathbf{S}_{t-1} + \beta_t \mathbf{k}_t \mathbf{v}_t^\top.$$

This recurrence coincides with an explicit Euler discretization of a linear continuous-time dynamical system

$$\frac{d\mathbf{S}}{dt} = -\mathbf{k}_t \mathbf{k}_t^\top \mathbf{S} + \mathbf{k}_t \mathbf{v}_t^\top$$

with step size β_t. From a numerical-analysis viewpoint, this corresponds to a first-order (RK-1) integrator with local truncation error $\mathcal{O}(\beta_t^2)$.

Subsequent work such as Gated DeltaNet (Yang et al., 2024a) and Kimi Delta Attention (KDA) (Team et al., 2025a) enriches the delta rule with channel-wise gates, data-dependent learning rates, and more expressive value mixing, leading to strong empirical performance on long-context language modeling. However, these methods still rely on first-order explicit Euler integration of the underlying linear dynamics: the continuous-time model is fixed, and improvements come from better parameterizations of the right-hand side rather than from a more accurate time integrator.

In contrast, we adopt the continuous-time viewpoint as the primary design principle and ask a different question: for the same linear attention dynamics, is it possible to eliminate discretization error altogether? We show that, by exploiting the rank-1 structure of the dynamics matrix, the exact closed-form solution corresponding to the infinite-order Runge–Kutta limit (RK-∞) can be computed in linear time, yielding an error-free linear attention mechanism.

7.2 State Space Models and Discretization of Continuous Dynamics

Structured State Space Models (SSMs) provide a principled framework for modeling long-range sequence dependencies via linear time-invariant (LTI) dynamics. The S4 family (Gu et al., 2022a,b, 2020) and related models specify a continuous-time state equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{y} = \mathbf{C}\mathbf{x},$$

and then derive efficient discrete-time implementations by discretizing $(\mathbf{A}, \mathbf{B}, \mathbf{C})$. Common choices include the bilinear (Tustin) transform and zero-order hold (ZOH) (Gu et al., 2022a, 2020). The bilinear transform can be interpreted as a trapezoidal rule or implicit second-order Runge–Kutta scheme applied to the underlying ODE, while ZOH yields an exact discretization for piecewise-constant inputs in time. In practice, these methods enable stable and efficient convolutional realizations, but they still approximate or indirectly parameterize the matrix exponential $e^{\Delta t \mathbf{A}}$ for general full-rank $\mathbf{A}$.

Mamba (Gu and Dao, 2024) and its successors extend this line of work by introducing selective state spaces, where the effective continuous-time parameters $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ depend on the input sequence. The resulting models retain the continuous-time flavor but still require a discretization step to obtain a practical recurrent or convolutional update. In particular, Mamba employs a bilinear transform (or closely related variants), which is formally equivalent to an implicit second-order Runge–Kutta method applied to the linear SSM. Thus, despite their continuous-time interpretation, modern SSM architectures ultimately rely on finite-order approximations of the exact propagator $e^{\Delta t \mathbf{A}}$ for tractability in the general case.

Classical exponential moving average (EMA) filters can also be written as simple SSMs with scalar $\mathbf{A}$ and $\mathbf{B}$, further highlighting the tight connection between memory mechanisms and linear dynamical systems. Recent EMA-based sequence models (Fu et al., 2022; Ma et al., 2022; Sun et al., 2023b) typically design forgetting factors directly in discrete time, without explicitly deriving them from an underlying continuous-time system.

Our formulation is conceptually close to these SSM approaches: we also start from a continuous-time linear ODE and ask how to implement it efficiently in discrete time. The key difference lies in the structure of the dynamics. Whereas S4, Mamba and related SSMs must handle general (potentially full-rank) matrices $\mathbf{A}$ and therefore resort to finite-order approximations of $e^{\Delta t \mathbf{A}}$, the attention dynamics studied in this work have a rank-1 transition matrix $\mathbf{A}_t = \mathbf{k}_t \mathbf{k}_t^\top$. We show that this special structure makes both the matrix exponential and the associated input integral analytically tractable, enabling an exact RK-∞ update without incurring the cubic cost typically associated with matrix exponentials.

7.3 Continuous-Time Transformers and ODE-Based Attention

A parallel line of work applies ideas from neural ordinary differential equations (Neural ODEs) (Chen et al., 2019) to Transformer-like architectures. ODE-based Transformers reinterpret residual blocks as discrete steps of an ODE solver in depth, and explore higher-order Runge–Kutta schemes or adaptive step sizes to improve stability and expressivity along the layer dimension. Similarly, continuous-time Transformers for irregular time series treat token positions as continuous timestamps and parameterize the hidden dynamics by ODEs while retaining attention-like interaction structure.

More recently, PDE- or ODE-guided attention mechanisms have been proposed to study the evolution of attention weights in a pseudo-time dimension, with the goal of obtaining smoother or more interpretable attention maps for very long sequences. These methods typically introduce auxiliary continuous variables or diffusion-like processes on top of standard attention, and then perform numerical integration over the additional dimension using standard finite-step solvers.

Conceptually, these works share with us the high-level idea of viewing Transformers and attention through the lens of continuous-time dynamics. However, there are two crucial differences. First, most existing ODE-based Transformer variants operate at the level of residual blocks, hidden states, or attention weights, rather than on the fast-weight memory state of linear attention. Second, they generally depend on off-the-shelf ODE solvers (e.g., Euler, RK2, RK4, or adaptive-step methods) with a finite number of stages, and do not attempt to derive an exact closed-form solution that remains efficient in sequence length. In contrast, we focus directly on the continuous-time dynamics of the linear attention state and show that, thanks to its rank-1 structure, the exact infinite-order Runge–Kutta limit is both analytically available and computationally efficient.

7.4 Other Linear-Time Long-Context Models

Beyond linear attention and SSMs, several architectures achieve linear or near-linear time complexity for long-context modeling without relying on attention in its classical form. RetNet (Sun et al., 2023b) introduces a retention mechanism that admits parallel, recurrent, and chunkwise implementations, providing a unified framework that interpolates between convolutional and recurrent behaviors. Hyena (Poli et al., 2023) and related long-convolution models exploit implicit filters and hierarchical convolutional structures to capture long-range dependencies with sub-quadratic complexity, while eschewing explicit pairwise token interactions.

These models demonstrate that long-context sequence modeling does not necessarily require attention or SSMs, and that alternative primitives such as long convolutions and retention can also be highly effective. Our work is complementary to these approaches. Rather than proposing a new structural building block, we revisit the numerical foundations of existing and widely used primitive linear attention, shows that its continuous-time dynamics admit an exact, error-free discretization in the rank-1 setting. This provides a new theoretical lens for understanding and improving attention-like mechanisms, and may inform the design of future linear-time architectures.

8 Conclusion

We presented **Error-Free Linear Attention (EFLA)**, a new attention paradigm that bridges the gap between computational efficiency and mathematical precision. We identified that the performance bottleneck in classical linear attention stems from the accumulation of errors in low-order discretization schemes. Instead of refining these approximations, we bypassed them entirely by deriving the exact analytical solution to the attention dynamics. This results in an update mechanism that is theoretically free from error accumulation yet remains fully parallelizable and computable in linear time. Empirically, we demonstrated that this theoretical precision translates into tangible gains: **EFLA** exhibits superior robustness in noisy environments, accelerated convergence, and consistent performance improvements over DeltaNet across diverse benchmarks. By proving that exact integration is attainable without sacrificing scalability, our work lays a solid foundation for the next generation of stable, high-fidelity sequence models. We hope this work inspires future exploration into exact solvers for complex continuous-time attention architectures.

References

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, pages 7432–7439, 2020.

John Charles Butcher. A history of runge-kutta methods. *Applied numerical mathematics*, 20(3):247–260, 1996.

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019. https://arxiv.org/abs/1806.07366.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 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.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Leonhard Euler. *Institutiones calculi integralis*, volume 1. impensis Academiae imperialis scientiarum, 1792.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Google. Gemini deep research, 2025. https://gemini.google/overview/deep-research/. Accessed: 2025-12-11.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Re. Hippo: Recurrent memory with optimal polynomial projections, 2020. https://arxiv.org/abs/2008.07669.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022a. https://arxiv.org/abs/2111.00396.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models, 2022b. https://arxiv.org/abs/2206.11893.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081): 633–638, 2025.

Arieh Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, 2024.

Thierry Joffrain, Tze Meng Low, Enrique S Quintana-Ortí, Robert van de Geijn, and Field G Van Zee. Accumulating householder transformations, revisited. *ACM Transactions on Mathematical Software (TOMS)*, 32(2):169–179, 2006.

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

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. An llm compiler for parallel function calling. In *Forty-first International Conference on Machine Learning*, 2024.

W. Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*. Teubner, 1901. https://books.google.com.hk/books?id=Zc4TAQAAIAAJ.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

Hanyu Lai, Xiao Liu, Junjie Gao, Jiale Cheng, Zehan Qi, Yifan Xu, Shuntian Yao, Dan Zhang, Jinhua Du, Zhenyu Hou, et al. A survey of post-training scaling in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2771–2791, 2025.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: Moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*, 2022.

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.

OpenAI. Introducing deep research, 2025. https://openai.com/index/introducing-deep-research/. Accessed: 2025-12-11.

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)*, pages 1525–1534, 2016.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models, 2023. https://arxiv.org/abs/2302.10866.

Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

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 *International conference on machine learning*, pages 9355–9366. PMLR, 2021.

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

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

Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, et al. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025a.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

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, 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, 2024b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

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.

Appendix

A Experimental Setting

We used 8 A100 GPUs for 340M and 1.3B language modeling experiments. The random seed it set to 42. Each model uses AdamW for optimization, with a peak learning rate of 3×10^{-4}. The 340M models are trained for 8 billion tokens with a global batch size of 1M tokens, while the 1.3B models are trained for 16 billion tokens with a global batch size of 4M tokens. We use a cosine learning rate schedule, starting with a warm-up phase of 1 billion tokens for the 340M models and 4 billion tokens for the 1.3B models (corresponding to 1024 warm-up steps). Both have configurations that initial and final learning rates set at 3×10^{-5}. We apply a weight decay of 0.1 and use gradient clipping at a maximum of 1.0. The head dimension is set to 128, and the kernel size for convolution layers is set at 4.

B Discussion: Saturation Effects and Learning Rate Scaling

While EFLA eliminates the discretization error inherent in Euler-based methods like DeltaNet, our empirical observations reveal a distinct optimization behavior, EFLA demonstrates superior semantic capture in the early training stages but exhibits a slower convergence rate in the final asymptotic regime. We attribute this phenomenon to the *saturation property* of the exact decay factor.

The Stability-Responsiveness Trade-off. Analytically, the Euler update employed by DeltaNet implies a linear response to the input magnitude, where the update step $\Delta S \propto \beta_t$. In contrast, the EFLA update is governed by the soft-gating term $\alpha_t = \frac{(1-e^{-\beta_t \lambda_t})}{\lambda_t}$. Considering the inequality $\frac{1-e^{-x}}{x} < 1$ for all $x > 0$, the effective update magnitude of EFLA is strictly sub-linear with respect to the energy of the key $\lambda_t = \|\mathbf{k}_t\|^2$. In the early training phase, this saturation acts as a robust filter against high-variance gradients and outliers (large λ_t), preventing the catastrophic divergence often seen in unnormalized Euler updates. This allows EFLA to establish stable semantic representations rapidly. However, as the model approaches convergence, this same mechanism dampens the magnitude of parameter updates. Specifically, for high-confidence features where λ_t is large, the gradient signal is exponentially suppressed, leading to a "vanishing update" problem that slows down fine-grained optimization.

Implication for Hyperparameters. In this case, EFLA naturally necessitates a larger global learning rate to compensate for this exponential saturation, allowing the model to maintain responsiveness in the saturation regime without sacrificing its theoretical error-free guarantees. To validate this, we performed a robustness ablation study across varying learning rates, as illustrated in Figure 2. The results reveal a critical sensitivity: when trained with a conservative learning rate (e.g., $\text{lr} = 1 \times 10^{-4}$, purple curve), the model fails to learn robust features, resulting in performance degradation. This empirical evidence confirms that a relatively larger learning rate is a structural necessity for EFLA to counteract the dampening effect of the exponential gate and achieve its full potential.

C Construction and Properties of rank-1 Matrices

$\mathbf{A}_t$ is a *rank-1* matrix, and it satisfies:

$$\mathbf{A}_t^2 = \mathbf{k}_t \mathbf{k}_t^\top \mathbf{k}_t \mathbf{k}_t^\top = \mathbf{k}_t (\mathbf{k}_t^\top \mathbf{k}_t) \mathbf{k}_t^\top = \lambda_t \mathbf{A}_t \tag{34}$$

Where $\lambda_t = \mathbf{k}_t^\top \mathbf{k}_t$ is scalar value.

Then it gives us a key property: $\mathbf{A}_t$ is a scaled projection matrix (i.e., $\mathbf{A}_t^2 = \lambda_t \mathbf{A}_t$).

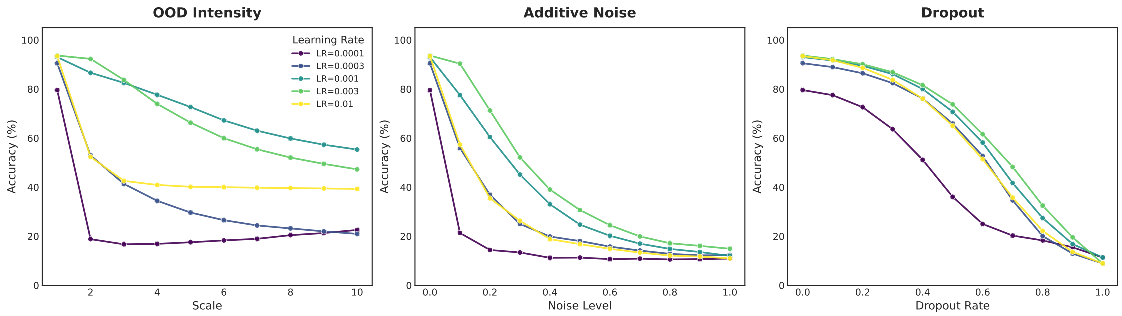

Figure 2 Impact of learning rate scaling on EFLA robustness. We evaluate the test accuracy of EFLA on sMNIST under three interference settings: OOD Intensity scaling (left), Additive Gaussian Noise (middle), and Dropout (right). The curves demonstrate a clear correlation between learning rate magnitude and model robustness. Notably, increasing the learning rate from 1×10^{-4} to 3×10^{-3} significantly mitigates performance degradation under high interference, empirically validating the necessity of larger step sizes to counteract the saturation effect.

D General Solutions of Ordinary Differential Equations

We start with a first-order linear matrix ODE:

$$\frac{d\mathbf{S}}{dt} = -A\mathbf{S} + \mathbf{b}, \tag{35}$$

Which can be rewrite as:

$$\frac{d\mathbf{S}}{dt} + A\mathbf{S} = \mathbf{b}, \tag{36}$$

For this type of differential equation, the integrating factor is:

$$e^{\int \mathbf{A}\, dt}, \tag{37}$$

Since $\mathbf{A}$ is constant, the integrating factor is simply:

$$e^{\mathbf{A}t} \tag{38}$$

Multiply the entire equation by $e^{\mathbf{A}t}$:

$$e^{\mathbf{A}t}\left(\frac{d\mathbf{S}}{dt} + A\mathbf{S}\right) = e^{\mathbf{A}t}\mathbf{b}, \tag{39}$$

Expanding the left-hand side:

$$e^{\mathbf{A}t}\frac{d\mathbf{S}}{dt} + e^{\mathbf{A}t}\mathbf{A}\mathbf{S} = e^{\mathbf{A}t}\mathbf{b}, \tag{40}$$

By the product rule for matrix-vector multiplication:

$$\frac{d}{dt}(e^{\mathbf{A}t}\mathbf{S}) = \left(\frac{d}{dt}e^{\mathbf{A}t}\right)\mathbf{S} + e^{\mathbf{A}t}\frac{d\mathbf{S}}{dt}, \tag{41}$$

and since $\frac{d}{dt}e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t}$, we have:

$$\frac{d}{dt}(e^{\mathbf{A}t}\mathbf{S}) = (\mathbf{A}e^{\mathbf{A}t})\mathbf{S} + e^{\mathbf{A}t}\frac{d\mathbf{S}}{dt}, \tag{42}$$

Notice this matches exactly the left-hand side of Eq. 40 (since $\mathbf{A}$ and $e^{\mathbf{A}t}$ commute).

The equation becomes:

$$\frac{d}{dt}(e^{\mathbf{A}t}\mathbf{S}) = e^{\mathbf{A}t}\mathbf{b}, \tag{43}$$

Integrate both sides from t (initial time) to $t+\beta_t$ (final time). To avoid confusion, we use τ as the integration variable:

$$\int_t^{t+\beta_t} \frac{d}{d\tau}(e^{A\tau}\mathbf{S}(\tau))\,d\tau = \int_t^{t+\beta_t} e^{A\tau}\mathbf{b}\,d\tau, \tag{44}$$

$$\left[e^{\mathbf{A}\tau}\mathbf{S}(\tau)\right]_t^{t+\beta_t} = \int_t^{t+\beta_t} e^{\mathbf{A}\tau}\mathbf{b}\,d\tau, \tag{45}$$

Thus:

$$e^{\mathbf{A}(t+\beta_t)}\mathbf{S}(t+\beta_t) - e^{\mathbf{A}t}\mathbf{S}(t) = \int_t^{t+\beta_t} e^{\mathbf{A}\tau}\mathbf{b}\,d\tau, \tag{46}$$

Multiply both sides by $e^{-\mathbf{A}(t+\beta_t)}$:

$$\mathbf{S}(t+\beta_t) - e^{-\mathbf{A}(t+\beta_t)}e^{\mathbf{A}t}\mathbf{S}(t) = e^{-\mathbf{A}(t+\beta_t)}\int_t^{t+\beta_t} e^{\mathbf{A}\tau}\mathbf{b}\,d\tau, \tag{47}$$

Simplify using exponential properties:

$$\mathbf{S}(t+\beta_t) - e^{-\mathbf{A}\beta_t}\mathbf{S}(t) = \int_t^{t+\beta_t} e^{-\mathbf{A}(t+\beta_t-\tau)}\mathbf{b}\,d\tau, \tag{48}$$

Since $e^{-\mathbf{A}(t+\beta_t)}$ is constant, it can be moved inside the integral.

Let $s = \tau - t$. Then:

$$\begin{cases} s = 0, & \tau = t \\ s = \beta_t, & \tau = t+\beta_t \end{cases} \quad \text{and} \quad d\tau = ds, \tag{49}$$

Substitute:

$$\int_0^{\beta_t} e^{-\mathbf{A}[t+\beta_t-(s+t)]}\mathbf{b}\,ds = \int_0^{\beta_t} e^{-\mathbf{A}(\beta_t-s)}\mathbf{b}\,ds, \tag{50}$$

Rename s back to τ (dummy variable) and replace constants with $\mathbf{A}_t, \mathbf{b}_t$:

$$\int_0^{\beta_t} e^{-(\beta_t-\tau)\mathbf{A}_t}\mathbf{b}_t\,d\tau, \tag{51}$$

Combining everything, the full solution is:

$$\mathbf{S}(t+\beta_t) = e^{-\beta_t\mathbf{A}_t}\mathbf{S}(t) + \int_0^{\beta_t} e^{-(\beta_t-\tau)\mathbf{A}_t}\mathbf{b}_t\,d\tau, \tag{52}$$

E Detailed Derivation of Runge-Kutta Methods

Given the ordinary differential equation (ODE):

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \tag{53}$$

We advance from t_n to $t_{n+1} = t_n + h$.

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f\left(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1\right), \\ y_{n+1} &= y_n + hk_2 \end{aligned}$$

Continuous system:

$$\frac{d\mathbf{S}}{dt} = -\mathbf{k}_t\mathbf{k}_t^\top \mathbf{S}(t) + \mathbf{k}_t\mathbf{v}_t^\top \tag{54}$$

Let:

$$\mathbf{A}_t = \mathbf{k}_t\mathbf{k}_t^\top, \quad \mathbf{b}_t = \mathbf{k}_t\mathbf{v}_t^\top \tag{55}$$

Then:

$$\begin{aligned}
k_1 &= -\mathbf{A}_t\mathbf{S}_{t-1} + \mathbf{b}_t \\
k_2 &= -\mathbf{A}_t\left(\mathbf{S}_{t-1} + \tfrac{\beta_t}{2}k_1\right) + \mathbf{b}_t \\
\mathbf{S}_t &= \mathbf{S}_{t-1} + \beta_t k_2
\end{aligned}$$

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \beta_t\Big[-\mathbf{k}_t\mathbf{k}_t^\top\left(\mathbf{S}_{t-1} + \tfrac{\beta_t}{2}(-\mathbf{k}_t\mathbf{k}_t^\top\mathbf{S}_{t-1} + \mathbf{k}_t\mathbf{v}_t^\top)\right) + \mathbf{k}_t\mathbf{v}_t^\top\Big]$$

$$\mathbf{S}_t = \left(\mathbf{I} - \beta_t\mathbf{k}_t\mathbf{k}_t^\top + \tfrac{1}{2}\beta_t^2(\mathbf{k}_t\mathbf{k}_t^\top)^2\right)\mathbf{S}_{t-1} + \beta_t\left(\mathbf{I} - \tfrac{\beta_t}{2}\mathbf{k}_t\mathbf{k}_t^\top\right)\mathbf{k}_t\mathbf{v}_t^\top \tag{56}$$

General RK-4:

$$\begin{aligned}
k_1 &= f(t_n, y_n), \\
k_2 &= f\left(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1\right), \\
k_3 &= f\left(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_2\right), \\
k_4 &= f(t_n + h, y_n + hk_3), \\
y_{n+1} &= y_n + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}$$

$$\mathbf{S}_t = \left(\mathbf{I} - \beta_t\mathbf{A}_t + \tfrac{1}{2}\beta_t^2\mathbf{A}_t^2 - \tfrac{1}{6}\beta_t^3\mathbf{A}_t^3 + \tfrac{1}{24}\beta_t^4\mathbf{A}_t^4\right)\mathbf{S}_{t-1} + \left(\beta_t\mathbf{I} - \tfrac{1}{2}\beta_t^2\mathbf{A}_t + \tfrac{1}{6}\beta_t^3\mathbf{A}_t^2 - \tfrac{1}{24}\beta_t^4\mathbf{A}_t^3\right)\mathbf{b}_t \tag{57}$$

Then the general **Runge-Kutta-(N)** method is defined by:

$$\begin{aligned}
k_1 &= f(t_n, y_n), \\
k_2 &= f\left(t_n + a_2h, ; y_n + hb_{21}k_1\right), \\
k_3 &= f\left(t_n + a_3h, ; y_n + h(b_{31}k_1 + b_{32}k_2)\right), \\
&\vdots \\
k_N &= f\left(t_n + a_Nh, ; y_n + h\sum_{j=1}^{N-1} b_{Nj}k_j\right), \\
y_{n+1} &= y_n + h\sum_{i=1}^{N} c_ik_i,
\end{aligned}$$

where the coefficients a_i, b_{ij}, c_i are specified by the Butcher tableau of the corresponding RK scheme. Then, for a linear dynamical system $f(t, \mathbf{S}) = -\mathbf{A}_t\mathbf{S} + \mathbf{b}_t$, the *RK–N* update can be expressed as a truncated matrix power series:

$$\mathbf{S}_t = \left[\sum_{n=0}^{N}\frac{(-\beta_t\mathbf{A}_t)^n}{n!}\right]\mathbf{S}_{t-1} + \beta_t\left[\sum_{n=0}^{N-1}\frac{(-\beta_t\mathbf{A}_t)^n}{(n+1)!}\right]\mathbf{b}_t. \tag{58}$$