LEARNING TO ADAPT: IN-CONTEXT LEARNING BEYOND STATIONARITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer models have become foundational across a wide range of scientific and engineering domains due to their strong empirical performance. A key capability underlying their success is in-context learning (ICL): when presented with a short prompt from an unseen task, transformers can perform per-token and next-token predictions without any parameter updates. Recent theoretical efforts have begun to uncover the mechanisms behind this phenomenon, particularly in supervised regression settings. However, these analyses predominantly assume stationary task distributions, which overlook a broad class of real-world scenarios where the target function varies over time. In this work, we bridge this gap by providing a theoretical analysis of ICL under non-stationary regression problems. We study how the gated linear attention (GLA) mechanism adapts to evolving input-output relationships and rigorously characterize its advantages over standard linear attention in this dynamic setting. To model non-stationarity, we adopt a first-order autoregressive process and show that GLA achieves lower training and testing errors by adaptively modulating the influence of past inputs–effectively implementing a learnable recency bias. Our theoretical findings are further supported by empirical results, which validate the benefits of gating mechanisms in non-stationary ICL tasks.

1 INTRODUCTION

Transformer-based architectures (Vaswani et al., 2017) have emerged as a powerful and versatile modeling framework, achieving state-of-the-art results across a wide spectrum of scientific and engineering domains. Their remarkable effectiveness has been demonstrated in natural language processing (Radford et al., 2019; Brown et al., 2020), recommendation systems (Zhou et al., 2018; Chen et al., 2019), reinforcement learning (Chen et al., 2021; Janner et al., 2021), computer vision (Dosovitskiy et al., 2020), and multi-modal signal processing (Tsai et al., 2019), as well as in more specialized areas such as quantum information (Ma et al., 2025) and wireless communication systems (Kim et al., 2023). A particularly notable instance is their pivotal role in the development of large language models like GPT-4 (Achiam et al., 2023), where the Transformer backbone enables highly advanced generative capabilities.

A distinctive and increasingly studied feature of Transformer models is in-context learning (ICL) (Min et al., 2021), which allows the model to perform previously unseen tasks at inference time by conditioning on sequences of input-output examples, without requiring any explicit parameter updates. This emergent capability has spurred a growing body of research aiming to understand the underlying mechanisms that enable such behavior (Brown et al., 2020; Min et al., 2021; Dong et al., 2022; Wies et al., 2023; Zhang et al., 2023; Bai et al., 2023; Li et al., 2024a; Bertsch et al., 2024; Akyürek et al., 2024; Jiang et al., 2025a; Song et al., 2024; Wu et al., 2023; Qin et al., 2025). In particular, recent theoretical works have investigated the realization of ICL in supervised regression settings, showing that certain architectural components–such as linear attention mechanisms–can effectively emulate simple learning algorithms, e.g., a single step of gradient descent, when the input data distribution is stationary (Garg et al., 2022; Akyürek et al., 2022; Von Oswald et al., 2023; Zhang et al., 2024; Huang et al., 2023; Chen et al., 2024; Yang et al., 2024; Zhang et al., 2025; Mahankali et al., 2023; Ahn et al., 2023; Li et al., 2024b; 2025; Fu et al., 2024; Ding et al.). These findings offer valuable insights into the algorithmic behaviors implicitly encoded by architectural

design, shedding light on the interplay between representation, memory, and adaptation in modern Transformer models.

However, much of the existing theoretical understanding is limited to stationary data settings, where the input-output relationships remain consistent across in-context examples and the query point. In contrast, many practical scenarios–including time-series forecasting, streaming data, and natural language–exhibit non-stationarity, where the underlying target function evolves over time. In such settings, recency bias, or the increased predictive relevance of more recent examples, plays a crucial role in accurate prediction. Empirically, linear attention mechanisms are often insufficient for these non-stationary tasks, motivating the introduction of architectural variants that incorporate inductive biases better suited for adaptation, such as gated linear attention (GLA) (Yang et al., 2023; Jiang et al., 2025b), RetNet (Sun et al., 2023), Gateloop (Katsch, 2023), RWKV-6 (Peng et al., 2024), as well as state-space models like Mamba-2 (Gu & Dao, 2023). These methods have achieved strong performance in non-stationary sequence modeling, yet there remains a lack of formal theoretical understanding of their behavior in ICL settings.

Contribution: In this paper, we aim to bridge this gap by presenting a theoretical analysis of ICL in non-stationary or time-varying regression problems. We investigate how the GLA mechanism adapts to evolving input-output relationships and provide a rigorous characterization of its advantages over standard linear attention in this setting. To model non-stationarity, we adopt a first-order autoregressive process, which allows us to analytically capture temporal variations in the regression targets. Within this framework, we show that standard linear attention incurs higher training and testing errors due to its limited capacity to adapt to distributional shifts over time. In contrast, GLA exhibits inherent adaptability by dynamically modulating the contributions of past inputs, effectively inducing a learnable recency bias. This gating mechanism enables the model to better accommodate time-varying input-output mappings, thereby achieving more robust in-context generalization. Our analysis underscores the importance of architectural components–particularly gating–in equipping transformer models with the ability to implement adaptive learning algorithms in non-stationary environments. Experimental results further corroborate our theoretical findings. Collectively, our work contributes a theoretical perspective that clarifies the design choices behind transformer variants and offers a conceptual framework for understanding and developing architectures suited for adaptive ICL.

Notation We use bold capital letters (e.g., $\boldsymbol{Y}$) to denote matrices, bold lowercase letters (e.g., $\boldsymbol{y}$) to denote vectors, and italic letters (e.g., y) to denote scalar quantities. Elements of matrices are denoted in parentheses, as in Matlab notation. For example, $\boldsymbol{Y}(s_1, s_2)$ denotes the element in position (s_1, s_2) of the matrix $\boldsymbol{Y}$. The inner product of $\boldsymbol{A} \in \mathbb{R}^{d_1 \times d_2}$ and $\boldsymbol{B} \in \mathbb{R}^{d_1 \times d_2}$ can be denoted as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{s_1=1}^{d_1} \sum_{s_2=1}^{d_2} \boldsymbol{A}(s_1, s_2)\boldsymbol{B}(s_1, s_2)$. $\|\boldsymbol{X}\|_F$ represents the Frobenius norm of $\boldsymbol{X}$. $\mathbf{0}_d$ and $\mathbf{0}_{d \times d}$ denote the zero vector in $\mathbb{R}^d$ and the zero matrix in $\mathbb{R}^{d \times d}$, respectively. For a positive integer K, $[K]$ denotes the set $\{1, \ldots, K\}$.

1.1 Related Works

A growing body of work has investigated the emergent phenomenon of ICL, with a focus on understanding its behavior in stationary regression tasks. For example, (Garg et al., 2022) empirically demonstrated the ICL capabilities of transformers by analyzing prompts where each input is labeled by a task-specific function drawn from a predefined function class, such as linear models. Along similar lines, (Akyürek et al., 2022) investigated linear regression and introduced a transformer construction capable of performing a single gradient descent (GD) step using in-context examples. Building upon this, (Von Oswald et al., 2023) designed weight matrices for linear attention-only transformers that replicate GD updates in linear regression tasks, and notably, they observed that the learned weights resemble those obtained through end-to-end training on ICL prompts.

Further progress has been made by studying the convergence behavior of transformer architectures. In particular, (Zhang et al., 2024) showed that, for a single-layer linear self-attention model, gradient flow with carefully chosen random initialization converges to a global minimum, yielding low prediction error on anisotropic Gaussian data. Complementary work by (Huang et al., 2023) initiated the theoretical study of softmax attention, analyzing the training dynamics of one-layer, single-head transformers and providing convergence guarantees for linear regression. This line of research was

subsequently extended by (Chen et al., 2024; Yang et al., 2024; Zhang et al., 2025), who provided sufficient conditions for the convergence of multi-head softmax transformers trained with GD in ICL scenarios. Alternative theoretical perspectives have also been explored: for instance, (Mahankali et al., 2023) demonstrated that a transformer performing a single GD step on a least-squares objective can serve as a global minimizer of the pre-training loss, offering a different interpretation of training objectives in ICL. Similarly, (Ahn et al., 2023) showed that a single-layer model, when trained on random linear regression tasks, implicitly learns to perform a preconditioned GD step at test time, further reinforcing the connection between ICL and optimization-based learning rules. Meanwhile, (Li et al., 2024b; 2025) offered a theoretical interpretation of GLA through the lens of weighted preconditioned GD, although their analysis remains limited to stationary regression settings. Beyond first-order methods, more advanced optimization techniques have also been considered; for example, (Fu et al., 2024) analyzed the convergence behavior of second-order methods in ICL, highlighting their potential for accelerated adaptation relative to first-order approaches.

2 In-context Learning Time-varying Functions

This work builds upon the well-established in-context learning (ICL) framework introduced in (Garg et al., 2022), which aims to train models capable of performing ICL within a specified function class. As discussed in prior work, significant efforts have been devoted to elucidating the mechanisms underlying ICL. In particular, a number of studies (Garg et al., 2022; Akyürek et al., 2022; Mahankali et al.; Ahn et al., 2023; Huang et al., 2024; Zhang et al., 2024; Li et al., 2024b; 2025; Zhang et al., 2025) have investigated the dynamics of ICL in transformer architectures through the lens of linear regression tasks, where the target function is typically assumed to take the form $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$. However, these studies commonly rely on the simplifying assumption that the regression weight vector $\boldsymbol{w}$ remains fixed throughout the task. This stationarity assumption creates a theoretical-practical gap, as it does not faithfully reflect real-world scenarios in which data distributions are often non-stationary and the underlying regression weights may vary across different input samples.

In-context Learning Time-varying Functions To bridge this gap and advance the theoretical understanding of ICL in non-stationary settings, we introduce a more realistic framework in which the labels in the training prompt are generated by time-varying functions. Formally, let $\mathcal{D}_{\mathcal{X}}$ denote a distribution over inputs and $\mathcal{D}_{\mathcal{F}_i}$ a time-varying distribution over functions in $\mathcal{F}_i$. A prompt P is defined as a sequence $(\boldsymbol{x}_1, f_1(\boldsymbol{x}_1), \ldots, \boldsymbol{x}_n, f_n(\boldsymbol{x}_n), \boldsymbol{x}_{\text{query}})$, where the inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ and query $\boldsymbol{x}_{\text{query}} = \boldsymbol{x}_{n+1} \in \mathbb{R}^d$ are drawn from $\mathcal{D}_{\mathcal{X}}$, and each f_i is drawn from $\mathcal{D}_{\mathcal{F}_i}$. One may consider two canonical types of time-varying functions inspired by the literature: (i) Deterministic time-varying functions: Here, $f_i = f(\cdot, i/(n+1))$, where f is assumed to vary smoothly over rescaled time. This setting captures gradual and predictable evolution in the underlying mapping, as extensively studied in time-varying nonlinear regression models (Zhang & Wu, 2012; 2015). (ii) Stochastic time-varying functions: In this case, the evolution of f_i is modeled as a stochastic process, allowing for random fluctuations in the function mapping. A representative model is $f_i(x) = \gamma f_{i-1}(x) + e_i(x)$, where $0 < \gamma < 1$ is a forgetting factor modeling gradual drift in task mappings and $e_i(x)$ is a zero-mean stochastic perturbation.

We say that a model $\mathcal{M}$ can *in-context learn* the time-varying function class $\mathcal{F}_i$ up to accuracy ϵ, with respect to $(\mathcal{F}_i, \mathcal{D}_{\mathcal{X}})$, if it can predict $f_{n+1}(\boldsymbol{x}_{\text{query}})$ based on the prompt P with average error

$$\mathbb{E}_P \left[\ell(\mathcal{M}(P), f_{n+1}(\boldsymbol{x}_{\text{query}})) \right] \leq \epsilon, \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes an appropriate loss function, such as squared error. Within this framework, we can then pose the following central question:

Question: Can we train a model to in-context learn a given time-varying function class?

In this work, to facilitate theoretical analysis while preserving non-stationarity, we consider a simple yet expressive instantiation of the function class:

$$y_i = f_i(\boldsymbol{x}_i) = \langle \boldsymbol{w}_i, \boldsymbol{x}_i \rangle \in \mathbb{R}, i \in [n+1], \tag{2}$$

where each weight vector $\boldsymbol{w}_i$ evolves according to a first-order autoregressive process given by

$$\boldsymbol{w}_i = \gamma \boldsymbol{w}_{i-1} + \boldsymbol{e}_i, i \in [n+1]. \tag{3}$$

Here, $\gamma \geq 0$ is the autoregressive coefficient that controls the temporal correlation of the weight vectors, the sequence $\boldsymbol{w}_i$ follows a random walk model, which is a widely adopted generative model in signal processing and adaptive filtering literature (Sayed, 2011). To facilitate tractable analysis, we further assume that the initial weight vector is drawn i.i.d. as $\boldsymbol{w}_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$, the noisy terms are i.i.d. Gaussian with $\boldsymbol{e}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I})$, and the input vectors are i.i.d. samples from a zero-mean Gaussian distribution with covariance matrix $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$. Moreover, we assume that the random variables $\boldsymbol{w}_{i-1}$, $\boldsymbol{e}_i$, and $\boldsymbol{x}_i$ are mutually independent. Following a long line of theoretical work on in-context learning (Mahankali et al.; Ahn et al., 2023; Zhang et al., 2024; Chen et al., 2024; Yang et al., 2024; Li et al., 2024b; 2025; Zhang et al., 2025), we adopt Gaussian assumptions in our analysis. This modeling choice enables sharp and explicit characterizations of both the training and test errors–rather than only providing loose upper bounds–and is therefore essential for isolating how key quantities such as γ govern the behavior of the learned in-context learner.

Gated Linear Attention In the non-stationary regression setting introduced above, where the underlying task weights evolve gradually over time, it is crucial for the model to effectively capture pairwise correlations while adapting to the dynamics of changing tasks. Although standard linear attention mechanisms offer computational efficiency and scalability, they lack the flexibility to modulate the influence of prior context based on its relevance to the current input–an ability that is particularly important in nonstationary environments.

To address this limitation, we employ Gated Linear Attention (GLA) (Yang et al., 2023; Li et al., 2024b; 2025), which enhances linear attention by introducing a gating mechanism that controls the flow of past information. This structure enables the model to selectively integrate relevant historical patterns while suppressing outdated ones, thereby offering a better inductive bias for capturing evolving structures in non-stationary tasks.

Formally, we consider the following implementation of GLA. Let $\boldsymbol{W}_Q \in \mathbb{R}^{(d+1)\times(d+1)}$, $\boldsymbol{W}_K \in \mathbb{R}^{(d+1)\times(d+1)}$, and $\boldsymbol{W}_V \in \mathbb{R}^{(d+1)\times(d+1)}$ denote the query, key, and value weight matrices, respectively. To streamline the subsequent analysis, we follow prior works (Ahn et al., 2023; Huang et al., 2024; Zhang et al., 2024; Li et al., 2024b; 2025; Zhang et al., 2025) and construct the prompt by evaluating each function f_i on the sampled inputs and pairing each input with its corresponding output:.

$$\boldsymbol{Z} = [\boldsymbol{z}_1 \quad \cdots \quad \boldsymbol{z}_n \quad \boldsymbol{z}_{n+1}] = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n & \boldsymbol{x}_{n+1} \\ y_1 & \cdots & y_n & 0 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(n+1)}. \tag{4}$$

For each input $\boldsymbol{z}_i$, we define the corresponding query, key, and value vectors as $\boldsymbol{q}_i = \boldsymbol{W}_Q \boldsymbol{z}_i$, $\boldsymbol{k}_i = \boldsymbol{W}_K \boldsymbol{z}_i$ and $\boldsymbol{v}_i = \boldsymbol{W}_V \boldsymbol{z}_i$. The output of GLA at position i is given by:

$$\boldsymbol{o}_i = \boldsymbol{S}_i \boldsymbol{q}_i \;\text{ and }\; \boldsymbol{S}_i = \lambda \boldsymbol{S}_{i-1} + \boldsymbol{v}_i \boldsymbol{k}_i^\top, \tag{5}$$

where $\lambda \in (0, 1]$ is a forgetting factor that determines how quickly the attention mechanism discounts earlier information. For ease of theoretical analysis, we adopt a simplified formulation where a single global forgetting factor λ is used, rather than assigning a separate, data-dependent gating coefficient to each token as done in the original GLA model. By unrolling the recursive update in (5), we obtain:

$$\boldsymbol{S}_{n+1} = \lambda \boldsymbol{S}_n + \boldsymbol{v}_{n+1} \boldsymbol{k}_{n+1}^\top = \sum_{i=1}^{n+1} \lambda^{n+1-i} \boldsymbol{v}_i \boldsymbol{k}_i^\top = \boldsymbol{W}_V \left(\sum_{i=1}^{n+1} \lambda^{n+1-i} \boldsymbol{z}_i \boldsymbol{z}_i^\top \right) \boldsymbol{W}_K^\top, \tag{6}$$

which leads to the following expression for the output vector:

$$\boldsymbol{o}_{n+1} = \boldsymbol{S}_{n+1} \boldsymbol{q}_{n+1} = \boldsymbol{W}_V \left(\sum_{i=1}^{n+1} \lambda^{n+1-i} \boldsymbol{z}_i \boldsymbol{z}_i^\top \right) \boldsymbol{W}_K^\top \boldsymbol{W}_Q \boldsymbol{z}_{n+1}. \tag{7}$$

It is worth noting that when $\lambda = 1$, the weighted sum degenerates into an unweighted accumulation, i.e., $\sum_{i=1}^{n+1} \boldsymbol{z}_i \boldsymbol{z}_i^\top = \boldsymbol{Z}\boldsymbol{Z}^\top$, under which the GLA formulation reduces to the standard linear attention model. This highlights that GLA generalizes linear attention by introducing a learnable memory decay.

Since the final prediction is taken as the last entry of the token vector output by the GLA layer, only a subset of the entries in the weight matrices $\boldsymbol{W}_V$ and $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$ influence the output. To simplify the notation and subsequent analysis, we merge the query and key matrices into a single matrix and define

$$\boldsymbol{W}_V = \begin{bmatrix} \boldsymbol{W}_{11}^V & \boldsymbol{w}_{12}^V \\ \boldsymbol{w}_{21}^{V\top} & w_{-1}^V \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \text{ and } \boldsymbol{W}_{KQ} = \begin{bmatrix} \boldsymbol{W}_{11}^{KQ} & \boldsymbol{w}_{12}^{KQ} \\ \boldsymbol{w}_{21}^{KQ\top} & w_{-1}^{KQ} \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \quad (8)$$

where $\boldsymbol{W}_{11}^V, \boldsymbol{W}_{11}^{KQ} \in \mathbb{R}^{d\times d}$, $\boldsymbol{w}_{12}^V, \boldsymbol{w}_{21}^V, \boldsymbol{w}_{12}^{KQ}, \boldsymbol{w}_{21}^{KQ} \in \mathbb{R}^{d\times 1}$ and $w_{-1}^V, w_{-1}^{KQ} \in \mathbb{R}$. Using this decomposition, we express the predicted output as

$$\widehat{y}_{n+1} = \boldsymbol{o}_{n+1}(d+1) = \begin{bmatrix} \boldsymbol{w}_{21}^{V\top} & w_{-1}^V \end{bmatrix} \left(\sum_{i=1}^{n+1} \lambda^{n+1-i} \boldsymbol{z}_i \boldsymbol{z}_i^\top \right) \begin{bmatrix} \boldsymbol{W}_{11}^{KQ} \\ \boldsymbol{w}_{21}^{KQ\top} \end{bmatrix} \boldsymbol{x}_{n+1}. \quad (9)$$

Note that only the last row of $\boldsymbol{W}_V$ and the first d columns of $\boldsymbol{W}_{KQ}$ contribute to the final prediction. Therefore, without loss of generality, we may set the remaining entries in $\boldsymbol{W}_V$ and $\boldsymbol{W}_{KQ}$ to zero in the subsequent analysis.

3 Theoretical Analysis of GLA for Time-varying Regression

In this work, we investigate the convergence behavior, training error, and testing error of ICL linear predictors based on the GLA model for time-varying functions. Each task prompt corresponds to an embedding matrix $\boldsymbol{Z}_\tau$, for $\tau = 1, \ldots, B$, constructed according to the transformation defined in (4):

$$\boldsymbol{Z}_\tau = [\boldsymbol{z}_{\tau,1} \quad \cdots \quad \boldsymbol{z}_{\tau,n} \quad \boldsymbol{z}_{\tau,n+1}] = \begin{bmatrix} \boldsymbol{x}_{\tau,1} & \cdots & \boldsymbol{x}_{\tau,n} & \boldsymbol{x}_{\tau,n+1} \\ \langle \boldsymbol{w}_{\tau,1}, \boldsymbol{x}_{\tau,1} \rangle & \cdots & \langle \boldsymbol{w}_{\tau,n}, \boldsymbol{x}_{\tau,n} \rangle & 0 \end{bmatrix}. \quad (10)$$

We denote the prediction produced by the GLA model on the query input of task τ as $\widehat{y}_{\tau,n+1}$, whose exact form is given in (9). The empirical risk over B independent task prompts is then defined as:

$$l(\boldsymbol{\theta}) = \frac{1}{2B} \sum_{\tau=1}^{B} \left(\widehat{y}_{\tau,n+1} - \langle \boldsymbol{w}_{\tau,n+1}, \boldsymbol{x}_{\tau,n+1} \rangle \right)^2, \quad (11)$$

where the model parameters are denoted by $\boldsymbol{\theta} = \{\boldsymbol{W}_{KQ}, \boldsymbol{W}_V\}$. To analyze the learning dynamics, we consider the population risk induced in the limit as the number of training prompts tends to infinity, i.e., $B \to \infty$:

$$L(\boldsymbol{\theta}) = \lim_{B\to\infty} l(\boldsymbol{\theta}) = \frac{1}{2} \mathbb{E}_{\boldsymbol{w}_{n+1}, \boldsymbol{x}_{n+1}} [(\widehat{y}_{n+1} - \langle \boldsymbol{w}_{n+1}, \boldsymbol{x}_{n+1} \rangle)^2], \quad (12)$$

where we omit the task index τ for notational simplicity.

We study the evolution of the model parameters under gradient flow, which characterizes the continuous-time limit of gradient descent with infinitesimal step sizes. The parameter dynamics are governed by the ordinary differential equation $\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta})$.

In the following, we analyze the gradient flow dynamics under an initialization that satisfies the following assumption.

Assumption 1. *(Initialization) Let $\sigma > 0$ be a parameter and $\boldsymbol{\Theta} \in \mathbb{R}^{d\times d}$ be any matrix satisfying $\|\boldsymbol{\Theta}\boldsymbol{\Theta}^\top\|_F = 1$ and $\boldsymbol{\Lambda}\boldsymbol{\Theta} \neq \mathbf{0}_{d\times d} \in \mathbb{R}^{d\times d}$. We assume*

$$\boldsymbol{W}_V(0) = \sigma \begin{bmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 1 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \text{ and } \boldsymbol{W}_{KQ}(0) = \sigma \begin{bmatrix} \boldsymbol{\Theta}\boldsymbol{\Theta}^\top & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}. \quad (13)$$

This initialization follows the scheme proposed in (Zhang et al., 2024). Under this setup, we next show that the gradient flow dynamics with respect to the population loss converge to a specific global optimum. Specifically, we establish the following result.

Theorem 1. *(Convergence of gradient flow) Consider gradient flow over the population loss in (12). Assume that the initial task weight $\boldsymbol{w}_0 \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$, noises $\boldsymbol{e}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I})$ and inputs*

$\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$. *Suppose the initialization satisfies Assumption 1 with initialization scale* $\sigma > 0$ *satisfying* $\sigma < \sqrt{\frac{2D_1}{\sqrt{d}\|\widetilde{\mathbf{\Lambda}}\|}}$ *where*

$$D_1 = \begin{cases} \frac{\gamma^{n+2}-\gamma^{2n+2}}{1-\gamma}\sigma_w^2 + \frac{\gamma-\gamma^{n+1}-\gamma^{n+2}+\gamma^{2n+2}}{(1-\gamma)^2(1+\gamma)}\sigma_e^2, & \lambda = 1, \gamma \neq 1, \\ \lambda^{2n+2} n\sigma_w^2 + \left(\frac{\lambda^2(1-\lambda^{2n})}{(1-\lambda^2)^2} - \frac{\lambda^{2n+2}}{1-\lambda^2} n\right)\sigma_e^2, & \lambda \neq 1, \gamma \neq 1, \lambda = \gamma, \\ \frac{\lambda^{n+1}\gamma^{n+2}-\lambda\gamma^{2n+2}}{\lambda-\gamma}\sigma_w^2 + \left(\frac{\lambda\gamma(1-\lambda^n\gamma^n)}{(1-\gamma^2)(1-\lambda\gamma)} - \frac{\lambda^{n+1}\gamma^{n+2}-\lambda\gamma^{2n+2}}{(\lambda-\gamma)(1-\gamma^2)}\right)\sigma_e^2, & \lambda \neq 1, \gamma \neq 1, \lambda \neq \gamma, \end{cases}$$

and $\widetilde{\mathbf{\Lambda}} = D_2(2\mathbf{\Lambda} + \operatorname{trace}(\mathbf{\Lambda})\mathbf{I}) + D_3\mathbf{\Lambda}$ *with*

$$D_2 = \begin{cases} \frac{\gamma^2-\gamma^{2n+2}}{1-\gamma^2}\sigma_w^2 + \left(\frac{n}{1-\gamma^2} - \frac{\gamma^2-\gamma^{2n+2}}{(1-\gamma^2)^2}\right)\sigma_e^2, & \lambda = 1, \gamma \neq 1, \\ \lambda^{2n+2} n\sigma_w^2 - \left(\frac{n\lambda^{2n+2}}{1-\lambda^2} - \frac{\lambda^4-\lambda^{2n+2}}{(1-\lambda^2)^2}\right)\sigma_e^2, & \lambda \neq 1, \gamma \neq 1, \lambda = \gamma, \\ \frac{\gamma^2\lambda^{2n+2}-\lambda^2\gamma^{2n+2}}{\lambda^2-\gamma^2}\sigma_w^2 - \left(\frac{\gamma^2\lambda^{2n+2}-\lambda^2\gamma^{2n+2}}{(\lambda^2-\gamma^2)(1-\gamma^2)} - \frac{\lambda^2-\lambda^{2n+2}}{(1-\gamma^2)(1-\lambda^2)}\right)\sigma_e^2, & \lambda \neq 1, \gamma \neq 1, \lambda \neq \gamma, \end{cases}$$

and

$$D_3 = \begin{cases} \left(2\frac{\gamma^3-\gamma^{2n+1}}{(1-\gamma)^2(1+\gamma)} - 2\frac{\gamma^{n+2}-\gamma^{2n+1}}{(1-\gamma)^2}\right)\sigma_w^2 + \left(\frac{2}{\gamma^2-1}\left(\frac{\gamma^3-\gamma^{2n+1}}{(1-\gamma)^2(1+\gamma)} - \frac{\gamma^{n+2}-\gamma^{2n+1}}{(1-\gamma)^2}\right)\right. \\ \left. - \frac{2\gamma}{(\gamma^2-1)(1-\gamma)}\left(n-1-\frac{\gamma^n-\gamma}{\gamma-1}\right)\right)\sigma_e^2, & \lambda = 1, \gamma \neq 1, \\ \lambda^{2n+2} n(n-1)\sigma_w^2 + \left(\frac{2n(\lambda^4-\lambda^{2n+2})}{(1-\lambda^2)^2} - \frac{2(\lambda^{2n+4}-n\lambda^6+(n-1)\lambda^4)}{(1-\lambda^2)^3}\right. \\ \left. - \frac{\lambda^{2n+2}n(n-1)}{1-\lambda^2}\right)\sigma_e^2, & \lambda \neq 1, \gamma \neq 1, \lambda = \gamma, \\ \left(\frac{2\gamma^3\lambda^{2n+3}-2\lambda^5\gamma^{2n+1}}{\lambda(\lambda-\gamma)^2(\lambda+\gamma)} - \frac{2\gamma^{n+2}\lambda^{n+2}-2\gamma^{2n+1}\lambda^3}{\lambda-\gamma}\right)\sigma_w^2 + \left(\frac{2\gamma^{-1}(\lambda^4-\lambda^{2n+2})}{(1-\gamma^2)(\lambda-\gamma)(1-\lambda^2)}\right. \\ \left. - \frac{2(\lambda^3-\lambda^{n+2}\gamma^{n-1})}{(1-\gamma^2)(\lambda-\gamma)(1-\lambda\gamma)} - \frac{2\gamma^3\lambda^{2n+3}-2\lambda^5\gamma^{2n+1}}{\lambda(\lambda-\gamma)^2(\lambda+\gamma)(1-\gamma^2)} + \frac{2\gamma^{n+2}\lambda^{n+2}-2\gamma^{2n+1}\lambda^3}{(\lambda-\gamma)(1-\gamma^2)}\right)\sigma_e^2, & \lambda \neq 1, \gamma \neq 1, \lambda \neq \gamma. \end{cases}$$

Then gradient flow converges to a global minimum of the population loss (12). *Moreover,* $\boldsymbol{W}_{KQ}(0)$ *and* $\boldsymbol{W}_V(0)$ *respectively converge to*

$$\lim_{t\to\infty} \boldsymbol{W}_V(t) = \sqrt{D_1\|\widetilde{\mathbf{\Lambda}}^{-1}\|_F}\begin{bmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 1 \end{bmatrix} \quad \textit{and} \quad \lim_{t\to\infty} \boldsymbol{W}_{KQ}(t) = \sqrt{D_1\|\widetilde{\mathbf{\Lambda}}^{-1}\|_F^{-1}}\begin{bmatrix} \widetilde{\mathbf{\Lambda}}^{-1} & \mathbf{0}_d \\ \mathbf{0}_d^\top & 0 \end{bmatrix}. \tag{14}$$

The proof is deferred to Appendix B. Despite the non-stationary nature of the regression model considered in this work, we establish that gradient flow converges to a global minimum even under random initialization. The closed-form solution in (14) reveals that the location of the global optimum is explicitly determined by λ and γ, highlighting their structural influence on the solution. While the main theorem focuses on the regime $0 < \lambda \leq 1$ and $0 < \gamma < 1$, a more general result accommodating arbitrary $\lambda > 0$ and $\gamma > 0$ is established in Theorem 4 of Appendix B. Moreover, in the limiting case where $\lambda = \gamma = 1$ and $\sigma_e^2 = 0$, the expression reduces precisely to that in (Zhang et al., 2024, Theorem 4), thereby recovering the stationary setting as a special case of our more general formulation.

Training error We now analyze the training error of the learned network. At the global optimum– i.e., when the parameters converge to $\lim_{t\to\infty} \boldsymbol{W}_V(t)$ and $\lim_{t\to\infty} \boldsymbol{W}_{KQ}(t)$ in (14), a straightforward calculation yields the prediction $\widehat{y}_{n+1}$ as follows:

$$\widehat{y}_{n+1} = D_1\begin{bmatrix}\mathbf{0}_d^\top & 1\end{bmatrix}\left(\sum_{i=1}^{n+1}\lambda^{n+1-i}\boldsymbol{z}_i\boldsymbol{z}_i^\top\right)\begin{bmatrix}\widetilde{\mathbf{\Lambda}}^{-1} \\ \mathbf{0}_d^\top\end{bmatrix}\boldsymbol{x}_{n+1} = D_1\left(\sum_{i=1}^{n}\lambda^{n+1-i}\boldsymbol{w}_i^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top\right)\widetilde{\mathbf{\Lambda}}^{-1}\boldsymbol{x}_{n+1}. \tag{15}$$

This expression confirms that, for sufficiently long prompts, the trained model successfully in-context learns the family of linear predictors. We emphasize that both λ and γ jointly influence the degree of time variation in the underlying model. We next quantify the training error at the global optimum.

Theorem 2. *(Training error) Assuming the conditions in Theorem 1 hold, the recovery error between* (15) *and* (2) *is*

$$\begin{aligned}\mathbb{E}[(\widehat{y}_{n+1} - y_{n+1})^2] = D_1^2 \operatorname{trace}\big(&D_2(\mathbf{\Lambda}\operatorname{trace}(\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}) + 2\mathbf{\Lambda}\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}) \\ &+ D_3\mathbf{\Lambda}\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}\big) + D_4\operatorname{trace}(\mathbf{\Lambda}) - 2D_1^2\operatorname{trace}(\mathbf{\Lambda}\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{\Lambda}),\end{aligned} \tag{16}$$

where $D_4 = \gamma^{2n+2}\sigma_w^2 + \frac{1-\gamma^{2n+2}}{1-\gamma^2}\sigma_e^2$.

The proof is provided in Appendix C. Equation (16) illustrates that the training error depends jointly on the parameters λ and γ. Consequently, for fixed λ (or γ), there exists an optimal value of γ (or λ) that minimizes the error. Although the expressions of D_i suggest a symmetric structure in λ and γ, it does not necessarily imply that choosing $\lambda = \gamma$ minimizes the recovery error. In fact, the error involves a subtle balance between the σ_w^2- and σ_e^2-dependent terms as well as the trace terms with $\widetilde{\Lambda}^{-1}$. When $\lambda = \gamma$, the simplification of D_i may amplify certain noise-dependent factors and deteriorate the overall error. This observation highlights that the optimal choice of λ depends not only on the apparent algebraic symmetry but also on the interplay between noise statistics, system dimension, and the spectral structure of $\mathbf{\Lambda}$.

We next consider a special case with $\mathbf{\Lambda} = \mathbf{I}$, in which (16) reduces to $\mathbb{E}[(\widehat{y}_{n+1} - y_{n+1})^2] = \frac{D_1^2(d^2D_2+2dD_2+dD_3)+dD_4a^2-2aD_1^2}{a^2}$ with $a = (2+d)D_2 + D_3$. Note that, when γ is fixed, D_1, D_2, and D_3 are monotonically increasing functions of λ. Accordingly, in this expression, the numerator comprises positive terms that grow with D_1, D_2, and D_3, while the negative terms and the division by a^2 partially counterbalance this growth. As a result, the function is generally non-monotonic. Nevertheless, under certain parameter configurations, it may exhibit convexity with respect to λ over $(0, 1]$. The subsequent experiments provide direct validation of these theoretical observations.

Testing error In this part, we characterize the prediction performance of the trained transformer when evaluated on a test prompt drawn from a potentially different task distribution. Notably, the model parameters are fixed at their global optimum obtained from training, and the test prompt may differ in its length, data distribution, and underlying dynamics. We consider test prompts of the form

$$\begin{aligned}\overline{\boldsymbol{Z}} = [\overline{\boldsymbol{z}}_1 \quad \cdots \quad \overline{\boldsymbol{z}}_m \quad \overline{\boldsymbol{z}}_{m+1}] &= \begin{bmatrix} \overline{\boldsymbol{x}}_1 & \cdots & \overline{\boldsymbol{x}}_m & \overline{\boldsymbol{x}}_{m+1} \\ \overline{y}_1 & \cdots & \overline{y}_m & 0 \end{bmatrix} \\ &= \begin{bmatrix} \overline{\boldsymbol{x}}_1 & \cdots & \overline{\boldsymbol{x}}_m & \overline{\boldsymbol{x}}_{m+1} \\ \langle \overline{\boldsymbol{w}}_1, \overline{\boldsymbol{x}}_1 \rangle & \cdots & \langle \overline{\boldsymbol{w}}_m, \overline{\boldsymbol{x}}_m \rangle & 0 \end{bmatrix}, \end{aligned} \tag{17}$$

where the latent task weights $\{\overline{\boldsymbol{w}}_i\}_{i=1}^{m+1}$ evolve according to the first-order autoregressive model $\overline{\boldsymbol{w}}_i = \overline{\gamma} \cdot \overline{\boldsymbol{w}}_{i-1} + \overline{\boldsymbol{e}}_i, i = 1, \ldots, m+1$. To distinguish between training and testing distributions, we assume that the initial weight vector satisfies $\overline{\boldsymbol{w}}_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \overline{\sigma}_w^2 \mathbf{I})$, and the driving noise $\overline{\boldsymbol{e}}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \overline{\sigma}_e^2 \mathbf{I})$. The inputs are drawn independently as $\overline{\boldsymbol{x}}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \overline{\mathbf{\Lambda}})$, and we assume mutual independence among random variables $\overline{\boldsymbol{w}}_{i-1}$, $\overline{\boldsymbol{e}}_i$, and $\overline{\boldsymbol{x}}_i$.

Given a forgetting factor $\overline{\lambda}$, the prediction $\widetilde{y}_{m+1}$ produced by the model at test time (evaluated at the training global optimum) is

$$\widetilde{y}_{m+1} = D_1 \big(\sum_{i=1}^{m} \overline{\lambda}^{m+1-i} \overline{\boldsymbol{w}}_i^\top \overline{\boldsymbol{x}}_i \overline{\boldsymbol{x}}_i^\top \big) \widetilde{\mathbf{\Lambda}}^{-1} \overline{\boldsymbol{x}}_{m+1}. \tag{18}$$

We now characterize the mean squared prediction error on the test prompt:

Theorem 3. *(Testing error) Under the assumptions in Theorem 4, the expected prediction error of the model on the test prompt is given by*

$$\begin{aligned}\mathbb{E}[(\widetilde{y}_{m+1} - \overline{y}_{m+1})^2] = D_1^2 \operatorname{trace} \big(\overline{D}_2 (\overline{\mathbf{\Lambda}} \operatorname{trace}(\widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}} \widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}}) + 2 \overline{\mathbf{\Lambda}} \widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}} \widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}} \\ + \overline{D}_3 \overline{\mathbf{\Lambda}} \widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}} \widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}} \big) + \overline{D}_4 \operatorname{trace}(\overline{\mathbf{\Lambda}}) - 2 D_1 \cdot \overline{D}_1 \operatorname{trace}(\overline{\mathbf{\Lambda}} \widetilde{\mathbf{\Lambda}}^{-1} \overline{\mathbf{\Lambda}}), \end{aligned} \tag{19}$$

where $\overline{D}_i$ for $i = 1, \ldots, 4$ are defined analogously to the D_i constants from training, with the substitution $\lambda \to \overline{\lambda}$, $\gamma \to \overline{\gamma}$, $\sigma_w^2 \to \overline{\sigma}_w^2$, $\sigma_e^2 \to \overline{\sigma}_e^2$, and $n \to m$.

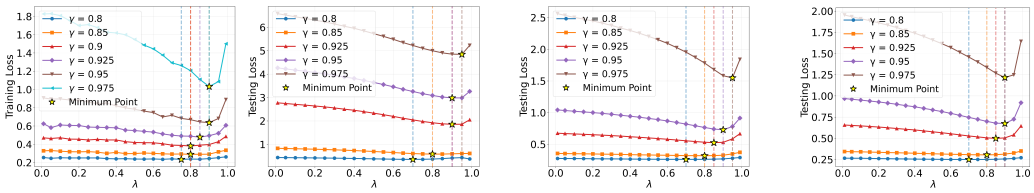
The proof has been provided in Appendix D. This result quantifies the generalization behavior of the trained model when applied to unseen prompts sampled from a potentially different distribution. Notably, the prediction error depends jointly on the training and testing task statistics through the interaction between $\widetilde{\mathbf{\Lambda}}$ and $\overline{\mathbf{\Lambda}}$. Moreover, the expected error $\mathbb{E}[(\widetilde{y}_{m+1} - \overline{y}_{m+1})^2]$ is inherently nonzero due to the stochastic nature of the task evolution–specifically, the noise in the dynamics of $\overline{\boldsymbol{w}}_i$ introduces irreducible uncertainty in the test labels $\overline{y}_i$. This highlights the importance of employing GLA, which adaptively modulates the influence of past observations and better accommodates temporal variations in the underlying regression weights. In the subsequent experimental section, we empirically demonstrate the effectiveness of the GLA mechanism in handling non-stationary tasks.

Comparison with Adaptive Signal Processing The non-stationary regression setting considered in this paper is closely related to classical problems in adaptive signal processing, where the underlying model parameters evolve gradually over time (Sayed, 2011; Das et al., 2015; Abdolee et al., 2016; Qin et al., 2020; Claser & Nascimento, 2021; Yu et al., 2021; Wang et al., 2022). To track such non-stationary dynamics, a wide range of online algorithms have been developed, including the least mean squares (LMS) algorithm, the affine projection algorithm (APA), and the recursive least squares (RLS) algorithm. These methods are designed to update model parameters iteratively in response to streaming data, with the goal of minimizing instantaneous or long-term prediction error. Under non-stationary models such as the first-order autoregressive process described in (3), the corresponding theoretical error analyses for these methods also indicate that, for a fixed γ, there exists an optimal choice of step size (in LMS/APA) or forgetting factor (in RLS) that minimizes the tracking error.

While classical adaptive signal processing methods explicitly update model parameters over time based on streaming observations, the paradigm studied in this paper–in-context learning with the GLA model–adopts a fundamentally different approach. Instead of relying on explicit parameter updates, as in LMS, APA, or RLS, the GLA implicitly adapts to task dynamics via internal representations conditioned on the prompt. In particular, the gating mechanism in GLA enables the model to selectively integrate past information in a soft and differentiable manner, thereby tracking non-stationary structures without modifying its parameters. This architectural distinction offers a new perspective on learning in non-stationary environments, where adaptation arises not from external optimization procedures, but from the model's forward computation itself.

4 Experimental Results

In this section, we present experiments to validate the theoretical analysis and demonstrate the advantages of GLA in non-stationary models. The experiments are conducted under the following settings. The training and testing losses are defined as $\frac{1}{B}\sum_{\tau=1}^{B}(\widehat{y}_{\tau,n+1} - y_{\tau,n+1})^2$ and $(\widetilde{y}_{m+1} - \overline{y}_{m+1})^2$, respectively. Unless otherwise specified, we set $d = 10$, $n = 100$, $\sigma_w^2 = 1$, $\sigma_e^2 = 0.01$, and $B = 10^7$. The AdamW optimizer is adopted with learning rate 10^{-2}, weight decay 0.05, and momentum parameter 0.9. Each model is trained for 2000 epochs with a batch size of 5000 samples. The loss associated with the optimal λ is highlighted by a star. Although the theoretical analysis imposes a constraint on the initialization matrix, our experiments use a random Gaussian initialization and still observe the predicted behavior, indicating that the constraint is not necessary in practice.

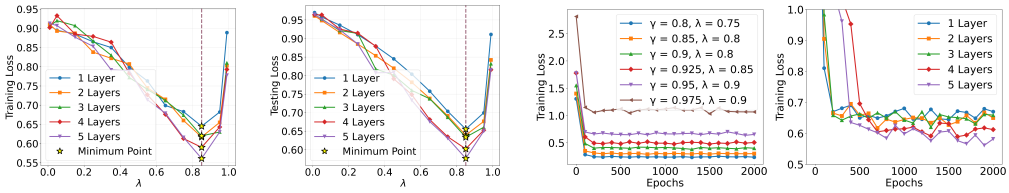


(a) Training loss (b) Testing loss ($m = 10$) (c) Testing loss ($m = 50$) (d) Testing loss ($m = 100$)

Figure 1: Training and testing performance of the one-layer GLA model with different λ and γ.

The first experiment compares the training and testing performance of the one-layer GLA model under varying choices of γ and λ. As shown in Figure 1a, when the autoregressive coefficient γ decreases and the impact of noise becomes more pronounced, an appropriate choice of λ is required to attain the lowest training loss. During testing, we evaluate the GLA model trained with $\lambda = 0.9$ under different sequence lengths $m \in \{10, 50, 100\}$. The results in Figures 1b to 1d show that, across different values of γ, selecting an appropriate λ remains crucial for minimizing the test loss. These results highlight the role of GLA in stabilizing learning under non-stationary conditions. By introducing a gating mechanism into linear attention, GLA effectively regulates the influence of past inputs, thereby mitigating error accumulation and enhancing the models adaptability to distributional shifts. Consequently, GLA achieves longer effective memory and improved generalization, underscoring its advantage in handling time-varying data. As mentioned previously,

a one-layer GLA model applied to a first-order autoregressive process functions analogously to an adaptive filter. To illustrate this, we compare its performance with LMS and RLS algorithms. We set the LMS step size to 0.01 and the RLS forgetting factor to 0.98, train on sequences of length 1000, and perform 10,000 Monte Carlo trials, averaging the results. The training errors for LMS and RLS are respectively $[0.2639\ 0.3168\ 0.6058\ 1.0072\ 1.4758]$ and $[0.2555\ 0.3746\ 0.6658\ 0.8881\ 1.2916]$ for $\gamma = [0.8\ 0.85\ 0.925\ 0.95\ 0.975]$. Compared to LMS and RLS, which require fixed or slowly adapting parameters, a one-layer GLA model achieves lower training errors (see Figure 1a) because it possesses higher representational flexibility. Furthermore, LMS and RLS adapt only to a single sequence at a time, requiring retraining for each new input, and therefore cannot leverage cross-sequence information. In contrast, GLA's learnable weights are shared across sequences, allowing the model to generalize and adapt efficiently to new inputs without retraining.



(a) Training loss (b) Testing loss ($m = 100$) (c) One Layer (d) Multiple Layers

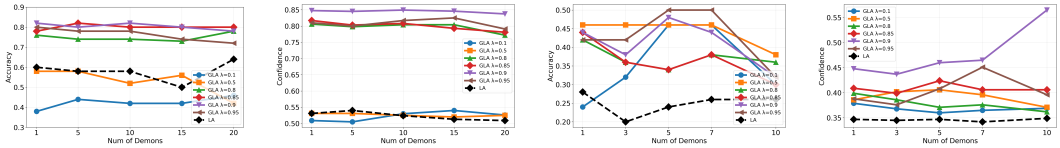
Figure 2: (a-b) Training and testing performance of the multi-layer GLA model with $\gamma = 0.95$ and different λ; (c) convergence performance of the one-layer GLA model; (d) convergence performance for GLA models with different layers.

In the second experiment, we investigate the impact of network depth on the performance of the GLA model. As illustrated in Figures 2a and 2b, increasing the number of layers consistently enhances both training and testing performance, suggesting that deeper architectures can more effectively capture long-range dependencies in non-stationary sequences. Conceptually, each GLA layer implements a linear adaptive filter whose effective behavior is determined by its gating weights. When multiple layers are stacked, these adaptive filters operate at different timescales, enabling the network to simultaneously capture short-term fluctuations and longer-term trends in the evolving regression weights. This multi-timescale structure explains why deeper GLA models achieve better performance under non-stationary regression: a single layer can track only one effective timescale of drift, while multiple layers collectively approximate a richer family of dynamic predictors. While formal theoretical analysis for multi-layer GLA models is not yet established, the empirical results underscore the critical role of the adaptive gating mechanism in regulating information flow across layers, thereby mitigating error accumulation and improving generalization.

Under the same experimental settings as the first two experiments, we examine the training convergence of the one-layer GLA with the optimal λ corresponding to the minimum loss, and of the multi-layer GLA with $\lambda = 0.85$. With random Gaussian initialization and a sufficiently large number of training samples, Figure 2c shows that the one-layer GLA achieves linear convergence, in agreement with our previous analysis. Figure 2d further demonstrates that the multi-layer GLA maintains linear convergence, indicating that the adaptive gating mechanism effectively stabilizes gradient propagation across layers. A rigorous theoretical characterization of convergence for multi-layer GLA is left for future work.

In the third experiment, we assess the ICL capability of GLA and Linear Attention (LA) models on a real-world language task. We focus on sentiment classification using the SST-2 dataset (Socher et al., 2013), which contains 67,349 training samples and 872 validation samples with binary labels (positive/negative). To initialize the models, we employ GPT-2 (small) (Radford et al., 2019), which consists of 12 layers, a hidden size of 768, 12 attention heads, and approximately 117M parameters. We then replace the original softmax attention with (i) linear attention, resulting in LinearGPT2, and (ii) gated linear attention, resulting in GatedLinearGPT2. Both models are optimized using AdamW with a learning rate of 5×10^{-5}, weight decay of 0.05, and momentum parameter of 0.9 for 1,000 iterations. For ICL fine-tuning, we provide 20 in-context demonstrations per instance, computing the loss only on label tokens. During evaluation on the SST-2 validation set, we vary the number of demonstrations $K \in \{1, 5, 10, 15, 20\}$. Performance is assessed using two metrics: (1) Accuracy, defined as the standard prediction accuracy; and (2) Confidence, calculated for each

correctly classified example by converting the models logits over positive, negative to probabilities $(p_{\text{pos}}, p_{\text{neg}})$ and taking $\max(p_{\text{pos}}, p_{\text{neg}})$, with the reported value being the average over all correctly classified examples. As shown in Figures 3a and 3b, when $\lambda = 0.9$, GLA achieves the highest accuracy and confidence, outperforming LA by a clear margin. This empirical advantage can be attributed to its gating mechanism: unlike LA, which implicitly assumes a stationary linear regression structure, GLA is able to adapt to the non-stationarity of real-world data by selectively integrating or discarding historical information–an ability that proves critical for reliable prediction.



(a) SST-2 Accuracy (b) SST-2 Confidence (c) MNLI Accuracy (d) MNLI Confidence

Figure 3: Accuracy and confidence of GatedLinearGPT2 vs. LinearGPT2 on SST-2 sentiment classification (left two) and MNLI natural language inference (right two) across different numbers of demonstrations.

In the final experiment, we evaluate the ICL capabilities of GLA and Linear Attention (LA) models on a more challenging natural language inference task, which requires determining the logical relationship between a premisehypothesis pair (entailment, contradiction, or neutral) across a broad range of text genres. We use the Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018), which spans multiple genres and contains approximately 393k training examples with three class labels. Following the same setup as in the third experiment, we provide 10 in-context demonstrations per instance for ICL fine-tuningconstrained by context lengthand compute the loss only on the label tokens. For evaluation on the MNLI validation set, we vary the number of demonstrations $K \in \{1, 3, 5, 7, 10\}$. As shown in Figures 3c and 3d, GLA consistently achieves higher accuracy and confidence than LA, highlighting the benefit of the gating mechanism.

5 Conclusion

This work presents a theoretical investigation of in-context learning in non-stationary regression problems, addressing an important gap in the current understanding of transformer models. Under a first-order autoregressive model of non-stationarity, we show that GLA outperforms standard linear attention by dynamically reweighting past inputs, enabling more accurate prediction in time-varying settings. Our analysis provides rigorous justification for the advantage of gating in capturing distributional shifts and highlights its role as an architectural inductive bias in adaptive learning. These findings not only deepen the theoretical foundations of ICL in dynamic environments but also suggest broader implications for the design of transformer variants in real-world applications characterized by non-stationarity.

A natural direction for future work is to generalize the first-order autoregressive assumption to a broader class of dynamic-weight models. In particular, allowing more flexible temporal evolutions–such as higher-order dynamics, stochastic drift, or slowly varying adversarial changes–would further illuminate how in-context learning behaves in general non-stationary settings. A second direction for future work is to develop a rigorous theoretical characterization of how gating mechanisms interact across multiple GLA layers. While our experiments show that stacking layers consistently improves performance, a principled analysis of how multi-layer structures capture multiple timescales of drift remains an important open problem. The third future direction is to analyze the global optimization landscape of the GLA model studied in this paper. Our numerical experiments suggest that random Gaussian initialization consistently converges to a global minimum under gradient flow, even when the theoretical initialization conditions are violated. This indicates the existence of a benign global optimum and motivates a deeper theoretical study of the model's optimization landscape.

REFERENCES

Reza Abdolee, Vida Vakilian, and Benoit Champagne. Tracking performance and optimal adaptation step-sizes of diffusion-lms networks. *IEEE Transactions on Control of Network Systems*, 5(1): 67–78, 2016.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data*, pp. 1–4, 2019.

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

Raffaello Claser and Vitor H Nascimento. On the tracking performance of adaptive filters and their combinations. *IEEE Transactions on Signal Processing*, 69:3104–3116, 2021.

Bijit Kumar Das, Luis A Azpicueta Ruiz, Mrityunjoy Chakraborty, and Jerónimo Arenas-García. On steady state tracking performance of adaptive networks. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 843–847. IEEE, 2015.

Nan Ding, Tomer Levinboim, Jialin Wu, Sebastian Goodman, and Radu Soricut. Causallm is not optimal for in-context learning. In *The Twelfth International Conference on Learning Representations*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Deqing Fu, Tian-qi Chen, Robin Jia, and Vatsal Sharan. Transformers learn to achieve second-order convergence rates for in-context linear regression. *Advances in Neural Information Processing Systems*, 37:98675–98716, 2024.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 19660–19722, 2024.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Jiachen Jiang, Yuxin Dong, Jinxin Zhou, and Zhihui Zhu. From compression to expansion: A layerwise analysis of in-context learning. *arXiv preprint arXiv:2505.17322*, 2025a.

Jiachen Jiang, Zhen Qin, and Zhihui Zhu. In-context learning for non-stationary mimo equalization. *arXiv preprint arXiv:2510.08711*, 2025b.

Tobias Katsch. Gateloop: Fully data-controlled linear recurrence for sequence modeling. *arXiv preprint arXiv:2311.01927*, 2023.

Seungnyun Kim, Anho Lee, Hyungyu Ju, Khoa Anh Ngo, Jihoon Moon, and Byonghyo Shim. Transformer-based channel parameter acquisition for terahertz ultra-massive mimo systems. *IEEE Transactions on Vehicular Technology*, 72(11):15127–15132, 2023.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024a.

Yingcong Li, Ankit S Rawat, and Samet Oymak. Fine-grained analysis of in-context linear estimation: Data, architecture, and beyond. *Advances in Neural Information Processing Systems*, 37: 138324–138364, 2024b.

Yingcong Li, Davoud Ataee Tarzanagh, Ankit Singh Rawat, Maryam Fazel, and Samet Oymak. Gating is weighting: Understanding gated linear attention through in-context learning. *arXiv preprint arXiv:2504.04308*, 2025.

Hailan Ma, Zhenhong Sun, Daoyi Dong, Chunlin Chen, and Herschel Rabitz. Tomography of quantum states from structured measurements via quantum-aware transformer. *IEEE Transactions on Cybernetics*, 2025.

Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

Arvind V Mahankali, Tatsunori Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Xingjian Du, Teddy Ferdinan, Haowen Hou, et al. Eagle and finch: Rwkv with matrix-valued states and dynamic recurrence. *arXiv preprint arXiv:2404.05892*, 2024.

Zhen Qin, Jun Tao, and Yili Xia. A proportionate recursive least squares algorithm and its performance analysis. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(1):506–510, 2020.

Zhen Qin, Jinxin Zhou, and Zhihui Zhu. On the convergence of gradient descent on learning transformers with residual connections. *arXiv preprint arXiv:2506.05249*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ali H Sayed. *Adaptive filters*. John Wiley & Sons, 2011.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Bingqing Song, Boran Han, Shuai Zhang, Jie Ding, and Mingyi Hong. Unraveling the gradient descent dynamics of transformers. *Advances in Neural Information Processing Systems*, 37: 92317–92351, 2024.

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

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for computational linguistics. Meeting*, volume 2019, pp. 6558, 2019.

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.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Yu Wang, Zhen Qin, Jun Tao, and Le Yang. Performance analysis of prls-based time-varying sparse system identification. In *2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 251–255. IEEE, 2022.

Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36:36637–36651, 2023.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pp. 1112–1122, 2018.

Yongtao Wu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. On the convergence of encoder-only shallow transformers. *Advances in Neural Information Processing Systems*, 36:52197–52237, 2023.

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.

Tong Yang, Yu Huang, Yingbin Liang, and Yuejie Chi. In-context learning with representations: Contextual generalization of trained transformers. *arXiv preprint arXiv:2408.10147*, 2024.

Yi Yu, Rodrigo C de Lamare, Tao Yang, and Qiangming Cai. Tracking analyses of m-estimate based lms and nlms algorithms. In *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 1–5. IEEE, 2021.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Ting Zhang and Wei Biao Wu. Inference of time-varying regression models. 2012.

Ting Zhang and Wei Biao Wu. Time-varying nonlinear regression models: nonparametric estimation and model selection. 2015.

Yedi Zhang, Aaditya K Singh, Peter E Latham, and Andrew Saxe. Training dynamics of in-context learning in linear attention. *arXiv preprint arXiv:2501.16265*, 2025.

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36:17773–17794, 2023.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1059–1068, 2018.