ON LEARNING LINEAR DYNAMICAL SYSTEMS IN CONTEXT WITH ATTENTION LAYERS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies the expressive power of linear attention layers for in-context learning (ICL) of linear dynamical systems (LDS). We consider training on sequences of inexact observations produced by noise-corrupted LDSs, with all perturbations being Gaussian; importantly, we study the non-i.i.d. setting as it is closer to real-world scenarios. We provide the optimal weight construction for a single linear-attention layer and show its equivalence to one step of Gradient Descent relative to an autoregression objective of window size one. Guided by experiments, we posit an extension to larger window sizes. We back our findings with numerical evidence. These results add to the existing understanding of transformers' expressivity as in-context learners, and offer plausible hypotheses for experimental observations whereby they compete with Kalman filters — the optimal model-dependent learners for this setting.

1 INTRODUCTION

We contribute towards understanding transformers' expressive power when learning from *non-i.i.d.* data produced by linear dynamical systems (LDSs). The starting point of our work is the well-known ability of transformers to perform in-context learning (ICL) (Brown et al., 2020).

Specifically, this boils down to accurately answering a query based on a set of examples given as a textual prefix ("in context") (Brown et al., 2020). This behaviour is desirable, as it loosens the requirement for expensive data collection and fine-tuning stages (Liu et al., 2023). Current research efforts are split between enhancing ICL through specialized training and prompt engineering, and building a mechanistic understanding of it — see the comprehensive review of Dong et al. (2022).

Currently there exist two perspectives on ICL mechanics: a Bayesian view, whereby transformers recover latent concepts from prompts, thus performing implicit Bayesian inference (Wang et al., 2023; Jiang, 2023; Wies et al., 2023; Xie et al., 2021), and a view of transformers as implementers of implicitly learned algorithms (Von Oswald et al., 2023a; Giannou et al., 2023; Akyürek et al., 2022; Garg et al., 2022; Ahn et al., 2023; Mahankali et al., 2023; Sander & Peyré, 2024; Von Oswald et al., 2023b; Sander et al., 2024). Within the latter works, investigations center around whether transformers can perform linear regression (and variants thereof) in context, and how. They give weight to this hypothesis by proving that, for certain token formats, data distributions, and architecture, the transformers' optimal weights effectively execute an optimization algorithm iteration in the forward pass, relative to a context-dependent loss (Von Oswald et al., 2023a; Mahankali et al., 2023; Ahn et al., 2023; Von Oswald et al., 2023b; Sander et al., 2024). Though this algorithmic view does not account for the "emergent" aspect of "in-the-wild" ICL (Shen et al., 2023), it provides concrete expressions for transformers' modelling power and identifies the minimal functional unit that instantiates it — a single, causally-masked, linear attention layer, without positional encoding. Despite this rich progress in understanding ICL for i.i.d. data settings, our grasp of the non-i.i.d. case is missing. A significant hurdle in analyzing this scenario is handling a token's statistical dependence on the entire context preceding it. This work takes the first steps towards unraveling this difficulty.

Specifically, we study the ability of a single linear attention layer to learn in context from sequences of observations $\{y_t\}_t$ generated by a time-invariant LDS doubly-corrupted by Gaussian noise

$$\begin{cases} \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{w}_{t+1}, \\ y_t = \boldsymbol{c}^\top \boldsymbol{x}_t + v_t, \end{cases} \tag{1}$$

where $\boldsymbol{w}_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{w}})$ and $v_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_v^2)$ with mutually independent $\boldsymbol{w}_t$ and v_t. Studying this setting has a threefold motivation. Firstly, the sequence $\{y_t\}_t$ is built on a temporal scaffold closer in nature to that of language-induced tokens, in stark contrast to the i.i.d. setup predominantly addressed by prior works (with few exceptions discussed in detail later). Secondly, this setting moves closer to the works taking a Bayesian view on ICL, where the data follows a Hidden Markov Model (HMM) (Xie et al., 2021) of which LDSs are a subclass (Minka, 1999). With HMMs being a mainstay in language modelling, setting (1) is particularly relevant. Finally, prior empirical observations emphasize the close performance of transformers relative to the Kalman Filter (KF) (Kalman, 1960), with the former matching the latter in settings where KF is the optimal predictor (Du et al., 2023). To our knowledge, the underlying mechanism is yet to be understood formally.

The goal of this paper is to characterize the structure of a single linear self-attention layer trained to optimality for predicting y_T in-context, when presented with sequences $\{y_t\}_{t=1}^{T-1}$. We proceed in two steps: first, we define an appropriate context-dependent loss for dealing with the time-series data. To this end, we rely on the improper learning approach of the system identification literature, whereby sequence generating processes of type (1) are well approximated by autoregressive ones. Second, we link the structure of optimally trained linear attention layers with algorithmic steps on the context-dependent loss. In doing so, we rely on a token augmentation scheme akin to prior works (Von Oswald et al., 2023a; Ahn et al., 2023; Mahankali et al., 2023). Our contributions are the following.

- **C1.** In Theorem 4.1, we prove that for an order-one autoregressive approximation of (1), the optimal linear attention layer implements a step of Gradient Descent on the associated least-squares loss. To our knowledge, this is the first optimality result for LDS data.
- **C2.** In Lemma 4.1, we identify a salient banded pattern of the matrices involved in the stationarity condition for generic order-s approximations of (1). We further define a class of parameters that satisfy this structural constraint and empirically observe that minimizers obey it.
- **C3.** In Section 5, we provide numerical experiments verifying our theory for order-one autoregressive approximations, and identify a recursive pattern within the empirically-determined optimal weights for order-s, $s \geq 2$ autoregressive approximations. Together with the point above, this narrows down the path to finding provably-optimal parameters in the latter case.
- **C3.** Conceptually, we make the case for the view of ICL as implicit optimization having a viable extension to LDS-produced data. We do so by bridging works from the system identification literature with empirical observations of transformers' in-context performance rivaling that of Kalman Filters.

2 Related literature

We review the niche of studies viewing ICL as in-context optimization, together with relevant works on filtering and system identification. Further comparisons are discussed in Section 4.1.

ICL for linear regression with i.i.d data. This line of work studies whether transformers trained on a few-shot learning objective can perform linear regression in-context, and how. Garg et al. (2022); Akyürek et al. (2022); Von Oswald et al. (2023a) provide empirical results in the affirmative, along with possible architecture constructions implementing Gradient Descent (GD) steps relative to a context-induced least squares loss. Through this lens, ICL reduces to on-the-fly optimization executed in the transformer's forward pass. Mahankali et al. (2023); Zhang et al. (2024); Ahn et al. (2023) complement these findings by proving that one-layer linear self-attention implementing such a GD step (possibly preconditioned) is a global minimizer of the pretraining loss when covariates are i.i.d. and Gaussian drawn. Finally, Zhang et al. (2024) complete the picture by proving that Gradient Flow converges to these global minimizers. Our results extend this line of work to non-i.i.d. setting.

ICL and system identification. This line of work asks whether transformers can perform autoregressive learning in context, and how. Different from the prior section, the following papers use the autoregressive pretraining loss and, unless stated otherwise, the results concern a single layer of linear self-attention. Von Oswald et al. (2023b) give a construction implementing a GD step on $\mathcal{L}(\boldsymbol{W}) \coloneqq \sum_{i=1}^{t-1} \| \boldsymbol{W}\boldsymbol{y}_i - \boldsymbol{y}_{i+1} \|^2$ in parallel for all positions t, under an appropriate token

configuration. Sander et al. (2024), further characterize the global minimizers of the autoregressive pretraining loss relative to the noiseless data $\boldsymbol{y}_{t+1} = \boldsymbol{A}\boldsymbol{y}_t$, with $\boldsymbol{A}$ uniformly sampled from the set of commuting orthogonal matrices. Notably, they recover Von Oswald et al. (2023b) construction when using the same token augmentation. Sander et al. (2024) further characterizes minimizers for the case of substituting token augmentation with positional encoding and a dimension-dependent number of attention heads — this setting's analysis, however, requires a diagonal weight structure. Zheng et al. (2024) complement these results by showing that, with a diagonal weight initialization and a controlled distribution of $\boldsymbol{y}_0$, pretraining with Gradient Flow (GF) recovers the previously identified GD-implementing optimum. Finally, Sander & Peyré (2024) extend these results to arbitrary orthogonal $\boldsymbol{A}$s via an infinite-depth attention-only transformer that correctly predicts $\boldsymbol{y}_T$ in the limit $T \to \infty$. This result holds for softmax, exponential, and linear activations.

Moving away from the noiseless settings above, Cole et al. (2025) establish approximation theoretic results for deep attention-only transformers predicting the sequence $\boldsymbol{y}_{t+1} = \boldsymbol{A}\boldsymbol{y}_t + \boldsymbol{w}_t$, with $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\boldsymbol{w}}^2 \boldsymbol{I})$ and $\boldsymbol{A} \in \mathbb{S}_{++}^d$. They prove by construction that there exists a $\log(T)$-depth transformer attaining a uniform-over-$\boldsymbol{A}$ $\frac{\log(T)}{T}$ error for predicting $\mathbb{E}[\boldsymbol{x}_{T+1}|\boldsymbol{x}_t, \boldsymbol{A}]$, and give a lower bound for the accuracy with which a single linear attention layer can make predictions. Related to the problem of capacity, Ziemann et al. (2024) establish a learner predicting the next observation with a uniform-in-time error bound requires a number of parameters at least quadratic in the algebraic multiplicities of $\boldsymbol{A}$'s unstable eigenvalues, and must operate on a context length at least logarithmic in the length of $\{y_t\}_{t=1}^T$.

In summary, these works either study transformers' ICL ability with respect to simplified LDSs or do not address the question of weight structure optimality. In contrast, we study fully-fledged systems (1) with the aim of characterizing the pretraining loss minimizers in the few-shot training setting.

Transformers and linear filtering. The classical model-based prediction tool for systems of type (1) is the Kalman Filter (KF) (Kalman, 1960). Using knowledge of system parameters, the KF gives the minimum expected squared error estimates $\hat{\boldsymbol{x}}_i$ of the hidden states $\boldsymbol{x}_i$ as linear combinations of the past y_is. Transformers as potential implementers of KF were studied by Goel & Bartlett (2024), who prove that a softmax causal attention layer is an arbitrarily good approximator. Akram & Vikalo (2024) further construct a transformer emulating the KF. Finally, Du et al. (2023) provide empirical evidence that a GPT-2 architecture (Radford et al., 2019) competes in accuracy with the KF for predicting the next observation in a previously unseen sequence, though the mechanism remains unstudied. We partially fill this gap with our present work.

3 Preliminaries, problem formulation & assumptions

Notation. Vectors and matrices are denoted by bold, lowercase and uppercase letters, respectively, with regular lowercase letters reserved for scalars. We denote by $\mathbf{1}_d$ and $\mathbf{0}_d$ the all-ones and all-zeros vectors of dimension d, and by $\mathbf{1}_{d\times m}$ and $\mathbf{0}_{d\times m}$ the analogous matrices. Unless stated otherwise, we use $\|\cdot\|$ for the Euclidean norm of vectors and the spectral norm of matrices. We denote by $\mathrm{Tr}\,(\cdot)$ the trace of a matrix, $\langle\cdot,\cdot\rangle$ the inner product, by $\|\cdot\|_F$ its Frobenius norm, and by $\rho(\cdot)$ its spectral radius. We use $\boldsymbol{e}_i$ for the i^{th} vector of the canonical basis in the appropriate dimension and $\boldsymbol{I}$ to denote the identity matrix of appropriate dimensions. The notations $\mathbb{S}_+^d$ and $\mathbb{S}_{++}^d$ define the cones of symmetric positive-semidefinite and positive-definite matrices in $\mathbb{R}^{d\times d}$, respectively. We use $\mathbb{S}^{d-1}$ to denote the unit sphere in $\mathbb{R}^d$. We use $\odot$ to denote the Hadamard product. Finally, we use $[n]$ when referencing the set of integers $\{1, 2, \dots n\}$.

The big picture: filtering, system identification, and linear regression. The KF (Kalman, 1960) computes the optimal estimates $\hat{\boldsymbol{x}}_i$ of $\boldsymbol{x}_i$ through the system of recursions

$$\begin{cases} \textbf{Predict: } \hat{\boldsymbol{x}}_{t+1|t} \coloneqq \boldsymbol{A}\hat{\boldsymbol{x}}_t, \quad \boldsymbol{P}_{t+1|t} = \boldsymbol{A}\boldsymbol{P}_t\boldsymbol{A}^\top + \Sigma_{\boldsymbol{w}} \\ \textbf{Gain: } \quad \boldsymbol{k}_{t+1} = \boldsymbol{P}_{t+1|t}\boldsymbol{c}\left(\boldsymbol{c}^\top\boldsymbol{P}_{t+1|t}\boldsymbol{c} + \sigma_v\right)^{-1} \\ \textbf{Update: } \hat{\boldsymbol{x}}_{t+1} = \hat{\boldsymbol{x}}_{t+1|t} + \boldsymbol{k}_{t+1}(\boldsymbol{y}_{t+1} - \boldsymbol{c}^\top\hat{\boldsymbol{x}}_{t+1|t}), \quad \boldsymbol{P}_{t+1} \coloneqq (\boldsymbol{I}_d - \boldsymbol{k}_{t+1}\boldsymbol{c}^\top)\boldsymbol{P}_{t+1|t}, \end{cases} \tag{2}$$

where $\hat{\boldsymbol{x}}_0$ and error covariance estimate $\boldsymbol{P}_0$ are given as input. Under the Gaussian errors assumption, the state prediction satisfies $\hat{\boldsymbol{x}}_t = \mathbb{E}[\boldsymbol{x}_t \,|\, y_t, \dots y_1]$ and, consequently, the forward observation

prediction follows $\hat{y}_{t+1} \coloneqq \boldsymbol{c}^\top \boldsymbol{A}\hat{\boldsymbol{x}}_t = \mathbb{E}[y_{t+1} \mid y_t, \dots y_1]$. The fast, constant-time KF predictions, however, require knowing the LDS parameters — a condition generally not satisfied in practice.

Consequently, "proper learning" approaches seek to reconstruct the underlying model, by first estimating $\boldsymbol{A}$, $\boldsymbol{c}$, Σ_w, σ_v through costly parameter identification techniques and then producing forward observation predictions using the KF (Hamilton, 1995). In contrast, "improper learning" methods eschew structural constraints and solely seek to reliably achieve low error with respect to the underlying data distribution and the learning objective (Kozdoba et al., 2019, and references therein). For LDSs, this boils down to expressing the next observation as a linear function of the recent past. Not only does the latter approach have the computational advantage of foregoing parameter estimation, but it also benefits from convex formulations, thus being amenable to classical optimization techniques. Most importantly, for certain LDS classes, improper learning methods can closely track $\mathbb{E}[y_{t+1} \mid y_t, \dots y_1]$, as follows.

Tsiamis & Pappas (2019) highlight the following rephrasing of the data-generating process via the KF and for some fixed window size s of past observations,

$$\begin{aligned}[y_{s+1}, \dots y_{T-1}] \;=\; \boldsymbol{c}^\top[(\boldsymbol{A} - \boldsymbol{k}\boldsymbol{c}^\top)^{s-1}\boldsymbol{k}, \dots (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{c}^\top)\boldsymbol{k}, \boldsymbol{k}] \;\; [\bar{\boldsymbol{y}}_1, \dots \bar{\boldsymbol{y}}_{T-s-1}] \\ +\; \boldsymbol{c}^\top(\boldsymbol{A} - \boldsymbol{k}\boldsymbol{c}^\top)^s[\hat{\boldsymbol{x}}_1, \dots \hat{\boldsymbol{x}}_{T-s+1}] \;+\; [\varepsilon_{s+1}, \dots \varepsilon_{T-1}], \end{aligned} \tag{3}$$

where $\bar{\boldsymbol{y}}_t \;\coloneqq\; [y_t, y_{t+1}, \dots y_{t+s-1}]^\top$, $\boldsymbol{k}$ is the steady-state gain, and $e_i \in \mathbb{R}$ are i.i.d, zero-mean Gaussian errors. Under KF convergence conditions, quantity $\rho(\boldsymbol{A} - \boldsymbol{k}\boldsymbol{c}^\top) < 1$ makes the second term vanish exponentially in s and thus renders it negligible. We are now in the familiar setting of noisy linear regression, albeit with non-i.i.d. data. The resulting order-s autoregressive process (AR(s)) is associated with the optimization objective

$$\min_{\boldsymbol{w} \in \mathbb{R}^s} \; \mathcal{L}_{AR(s)}(\boldsymbol{w}) \;\coloneqq\; \frac{1}{2(T-s-1)} \sum_{t=1}^{T-s-1} (y_{t+s} - \boldsymbol{w}^\top \bar{\boldsymbol{y}}_t)^2. \tag{4}$$

This simplification is the crux of improper learning approaches to system identification (Kozdoba et al., 2019) and becomes of note in conjunction with the idea that transformers perform on-the-fly optimization on the context-induced least squares objective. Should this latter view hold up to scrutiny under the new data distribution, it would imply that transformers could learn LDS-based time series in context arbitrarily well as a function of the available s. This is our incentive for seeking characterizations of the few-shot pretraining loss minimizers.

Technical assumptions. We make two technical assumptions that hold throughout the paper.

Assumption 3.1 (System assumptions)**.** *LDS* (1) *has isotropic and strictly positive definite noise covariances $\Sigma_{\boldsymbol{w}} = \sigma_{\boldsymbol{w}}\boldsymbol{I}$, with $\sigma_{\boldsymbol{w}} > 0$, and $\sigma_v > 0$. The system transition matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is marginally stable, with $\rho(\boldsymbol{A}) \leq 1$, and the pair $(\boldsymbol{A}, \boldsymbol{c})$ is observable, meaning that*

$$\boldsymbol{O} = \begin{bmatrix} \boldsymbol{c}^\top \\ \boldsymbol{c}^\top \boldsymbol{A} \\ \dots \\ \boldsymbol{c}^\top \boldsymbol{A}^{d-1} \end{bmatrix} \tag{5}$$

has a column rank of d.

Assumption 3.2 (LDS simplicity)**.** *We consider a family of LDSs* (1) *with fixed measurement vector $\boldsymbol{c} = \boldsymbol{1}_d$ and diagonal matrices $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, with distinct, non-zero diagonal elements.*

Assumption 3.1 is standard in the literature, and ensures KF convergence (Harrison, 1997) along with the exponential vanishing of the bias term in (3). Furthermore, it ensures the closeness of forward observation predictions given by the KF with those produced by a linear autoregressive predictor determined by expression (4) (Kozdoba et al., 2019). Assumption (3.2) helps with ease of exposition and ensures that observability holds, since the associated $\boldsymbol{O}$ is a Vandermonde matrix with distinct column-defining elements. This assumption can be relaxed to symmetry at the expense of added complexity in the later data sampling step.

Transformer architecture. Transformers (Vaswani et al., 2017) are neural architectures performing sequence-to-sequence mapping. For a set of input tokens $\boldsymbol{S}_T = [\boldsymbol{s}_1, \dots \boldsymbol{s}_T]^\top \in \mathbb{R}^{T\times p}$, the transformer produces a corresponding $\hat{\boldsymbol{S}}_T = [\hat{\boldsymbol{s}}_1, \dots \hat{\boldsymbol{s}}_T]^\top \in \mathbb{R}^{T\times p}$ by dynamically mixing tokens via its attention mechanism. An L-layer transformer $\mathcal{T}_\theta : \mathbb{R}^{T\times p} \to \mathbb{R}^{T\times p}$ parametrized by $\theta = [\theta_i]_{i=1}^L$ is a composition of blocks $\mathcal{T}_L = \mathcal{T}_{\theta_1} \circ \dots \mathcal{T}_{\theta_L}$. Each $\mathcal{T}_{\theta_i}$ is a sequence-to-sequence function given by

$$\mathcal{T}_{\theta_i}(\boldsymbol{S}) \;\coloneqq\; (\mathrm{MLP}_{\theta_i^{\mathrm{MLP}}} \circ \mathcal{A}_{\theta_i^{\mathrm{att}}})(\boldsymbol{S}),$$

where $\mathrm{MLP}_{\theta_i^{\mathrm{MLP}}}$ is a multilayer perceptron and $\mathcal{A}_{\theta_i^{\mathrm{att}}}$ is the attention mapping. This paper studies the simplified block $\mathcal{T}_\theta(\boldsymbol{S}) \;\coloneqq\; \mathcal{A}_\theta(\boldsymbol{S})$, thus setting $L = 1$ and $\mathrm{MLP}_{\theta_1^{\mathrm{MLP}}}$ to identity.

The causal h-headed attention block with residual connections is given by

$$\mathcal{A}_\theta(\boldsymbol{S}) \;\coloneqq\; \boldsymbol{S} + \sum_{h=1}^{H} \sigma\left(\boldsymbol{M} \odot \frac{1}{\tau}\boldsymbol{S}\boldsymbol{W}_Q^h(\boldsymbol{W}_K^h)^\top \boldsymbol{S}^\top\right)\boldsymbol{S}\boldsymbol{W}_V^h\boldsymbol{W}_O^h,$$

where the parameters $\theta = [\boldsymbol{W}_Q^h, \boldsymbol{W}_K^h, \boldsymbol{W}_V^h, \boldsymbol{W}_O^h]_{h=1}^H$ represent the query, key, value, and projection matrices, respectively; $\tau > 0$ is a scaling constant; σ is the softmax normalizing function applied row-wise; and $\boldsymbol{M} \in \mathbb{R}^{T\times T}$, with $\boldsymbol{M}_{i,j} = 1$ if $i \geq j$ and $-\infty$ otherwise is a mask enforcing causality.

Similar to prior works (Von Oswald et al., 2023a; Ahn et al., 2023; Mahankali et al., 2023), we restrict our study to the analytically tractable setting of single-headed linear attention (Katharopoulos et al., 2020). Without loss of expressivity, we drop the projection matrix $\boldsymbol{W}_O$ and consider the $\boldsymbol{W}_Q\boldsymbol{W}_K^\top$ as a single matrix $\boldsymbol{W}_{QK} \in \mathbb{R}^{p\times p}$. Since we're working in the few-shot scenario, we're concerned solely with predicting the final position as

$$\hat{\boldsymbol{s}}_T \;\coloneqq\; \mathcal{T}_\theta(\boldsymbol{S})_t = \boldsymbol{s}_T + \frac{1}{T-1}\boldsymbol{W}_V^\top \sum_{i=1}^{T-1} \boldsymbol{s}_i\boldsymbol{s}_i^\top \boldsymbol{W}_{QK}^\top \boldsymbol{s}_T, \tag{6}$$

where we set $\tau = T - 1$ and omit the last sum element due to a token asymmetry discussed next.

Token construction. We construct the tokens following the same scheme of Von Oswald et al. (2023a); Ahn et al. (2023); Mahankali et al. (2023). The input matrix $\boldsymbol{Y}_0$ constructed using AR(s) data (4) is

$$\boldsymbol{Y}_0 = \begin{bmatrix} \bar{\boldsymbol{y}}_1 & \bar{\boldsymbol{y}}_2 & \cdots & \bar{\boldsymbol{y}}_{T-s-1} & \bar{\boldsymbol{y}}_{T-s} \\ y_{s+1} & y_{s+2} & \cdots & y_{T-1} & 0 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{T-s-1} & \cdots & y_{T-s} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_s & y_{s+1} & \cdots & y_{T-2} & \cdots & y_{T-1} \\ y_{s+1} & y_{s+2} & \cdots & y_{T-1} & \cdots & 0 \end{bmatrix}, \tag{7}$$

where $s >= 1$ is the window size of the AR process. The last column represents the "test" token, whose final position is filled in the transformer's forward pass by y_T's estimate $\hat{y}_T$. This asymmetry motivates the last term's removal in (6).

Lemma 3.1 ensures, by construction, the existence of a linear attention layer producing $\boldsymbol{Y}_0$ from the raw sequence $\{y_t\}_t$. Its proof is deferred to Appendix C due to space constraints.

Lemma 3.1. *For a given $s >= 1$, there exists an $s+1$-headed linear attention layer with positional encoding which transforms input sequences $[y_1, y_2, \dots, y_T]^\top$ into*

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_s & y_{s+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{T-s-1} & y_{T-s} & \cdots & y_{T-2} & y_{T-1} \\ y_{T-s} & y_{T-s+1} & \cdots & y_{T-1} & 0 \\ & & \mathbf{0}_{T-s-1\,\times\, s} & & \end{bmatrix}.$$

The latter are essentially equivalent to tokens (7)*.*

Data generation, loss function, and training paradigm. We consider trajectories $\{y_i\}_{i=1}^T$ sampled from systems of type (1), where each trajectory corresponds to different, fixed parameter $\boldsymbol{A}$ and $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}_d, \sigma_{\boldsymbol{x}_0}^2 \boldsymbol{I})$. We let $\mathcal{D}_{\boldsymbol{A}}$ denote a symmetric, continuous distribution over $\boldsymbol{A}$'s diagonal

with marginals supported on $[-1, 1]$. For example, we can choose $\mathcal{D}_{\boldsymbol{A}} = \mathrm{Unif}(\mathbb{S}^{d-1})$ — uniform on the unit sphere; $\mathcal{D}_{\boldsymbol{A}} = \mathrm{Unif}([-1, 1]^d)$ — uniform inside the symmetric hypercube; or $\mathcal{D}_{\boldsymbol{A}} = \mathrm{Unif}(\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\})$ — uniform inside the unit ball. Any of these choices ensures that Assumptions 3.1 and 3.2 are satisfied with probability one.

Data generation proceeds in two steps: we sample $\boldsymbol{A}$ and $\boldsymbol{x}_0$ independently and observe the evolution of system (1) for T steps. We then construct $\boldsymbol{Y}_0$ (7) for a fixed s, and train our model to minimize

$$\mathcal{L}(\theta) \;\coloneqq\; \mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{\boldsymbol{v}_t\}_t} \left[\frac{1}{2} \left(\mathcal{T}_\theta(\boldsymbol{Y}_0)_{s+1,T-s} - y_T \right)^2 \right], \tag{8}$$

where the subscript marks that we solely consider the last position of the last output token.

4 Optimal parameter configurations

This section presents our theoretical results and discusses their implications relative to prior literature.

Our theoretical contribution is two-fold. First, in Lemma 4.1 we reveal a salient structure within the first-order optimality condition, which plays an important role in finding optimum configurations for the in-context loss of AR(s). Second, in Theorem 4.1 we prove that the transformer configuration implementing one-step GD is a global minimizer for AR(1) using this salient structure.

Unlike the i.i.d. case, each token generated by the LDS depends on the entire history. This results in high-order data moments populating the in-context loss, which can only be dealt with by unrolling to the initial state. A general approach to compute and match them is presented in Appendix D. We now describe the structure emerging within the first-order optimality condition.

Following (Ahn et al., 2023), we use basic algebraic manipulations (Appendix E) to rewrite loss (8) as

$$\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\},\{v_t\}} \left[\left(\frac{1}{T-s-1} \sum_{k=1}^{s} \langle \boldsymbol{Y}_0 \boldsymbol{Y}_0^\top, \boldsymbol{b}\boldsymbol{a}_k^\top \rangle y_{T-s-1+k} - y_T \right)^2 \right], \tag{9}$$

where $\boldsymbol{W}_V^\top = [\boldsymbol{0}_{(s+1)\times s},\ \boldsymbol{b}]^\top$ and $\boldsymbol{W}_{QK}^\top = [\boldsymbol{a}_1, \dots \boldsymbol{a}_s, \boldsymbol{0}_{s+1}]$. The zero-padding of both matrices comes from predicting solely the last position of the final token. Consequently, parameters ensuring

$$\begin{aligned} \mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\},\{v_t\}} \left[\frac{1}{T-s-1} \sum_{k=1}^{s} \langle \boldsymbol{Y}_0 \boldsymbol{Y}_0^\top, \boldsymbol{b}\boldsymbol{a}_k^\top \rangle y_{T-s-1+k} y_{T-s-1+j} \boldsymbol{Y}_0 \boldsymbol{Y}_0^\top \right] \\ = \;\; \mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\},\{v_t\}} \left[\, y_T y_{T-s-1+j} \boldsymbol{Y}_0 \boldsymbol{Y}_0^\top \, \right], \;\; \forall\, j \in [s] \end{aligned} \tag{10}$$

are critical points of the loss.

Notably, the right-hand side of (10) obeys a banded structure, as follows

$$\begin{bmatrix} \star & 0 & \star & \cdots & \cdots & \\ 0 & \star & 0 & \star & & \vdots \\ \star & 0 & \star & \ddots & \ddots & \vdots \\ \vdots & \star & \ddots & \ddots & \ddots & \star \\ \vdots & & \ddots & \ddots & \star & 0 \\ & \cdots & \cdots & \star & 0 & \star \end{bmatrix} \text{ for odd } s+j; \quad \text{or} \quad \begin{bmatrix} 0 & \star & 0 & \cdots & \cdots & \\ \star & 0 & \star & 0 & & \vdots \\ 0 & \star & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 & \star \\ & \cdots & \cdots & 0 & \star & 0 \end{bmatrix} \text{ for even } s+j; \tag{11}$$

where $\star$ is a placeholder for arbitrary reals (the proof is deferred to Appendix D). We formalize a class of parameters ensuring matching structures between the left and right-hand sides of (11) for arbitrary s in Lemma 4.1.

Lemma 4.1. *For an arbitrary s, the following parameters induce a banded structure in the left-hand side of* (10) *matching that of the right-hand side.*

$$\boldsymbol{W}_{QK} = \begin{bmatrix} \star & 0 & \star & \cdots & \\ 0 & \star & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \star \\ 0 & \cdots & 0 & \star & 0 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad \boldsymbol{W}_{V} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \vdots \\ \vdots & & & \vdots & 0 \\ \vdots & & & \vdots & \star \\ \vdots & & & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & \star \end{bmatrix}. \tag{12}$$

Lemma 4.1 can be understood as a narrowing-down based on structure of the parameter class likely to hold minimizers of (8).

Our second step is to use structure (12) to identify a global minimizer of loss (8) in the AR(1) case, yielding Theorem 4.1 with proof deferred to Appendix F.

Theorem 4.1. *Let $\mathbf{Y}_0$ encode the input tokens according to construction* (7) *for $s = 1$. Then, the optimal parameters $\theta^\star = \left(\boldsymbol{W}_{QK}^\star, \boldsymbol{W}_V^\star\right)$ of a single linear self-attention layer with respect to loss $\mathcal{L}(\theta)$ are*

$$\boldsymbol{W}_{QK}^\star = \begin{bmatrix} \frac{(T-2)\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} y_{T-1} y_T\right]}{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1} y_T\right]} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{W}_V^\star = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{13}$$

up to rescaling with $\gamma \neq 0$.

Broadly, the proof of Theorem 4.1 encounters two difficulties compared to the i.i.d. case: the number of terms that need to be matched in satisfying the first-order optimality condition, and the full-history dependence of the data. We address the first obstacle using the result of Lemma 4.1, and we sift through the second by relying on Isserlis' theorem (Isserlis, 1918) to handle higher-order moments of $\bar{\boldsymbol{y}}_t$ that would have factored out of expectations in the i.i.d. case. Details can be found in Appendix D.

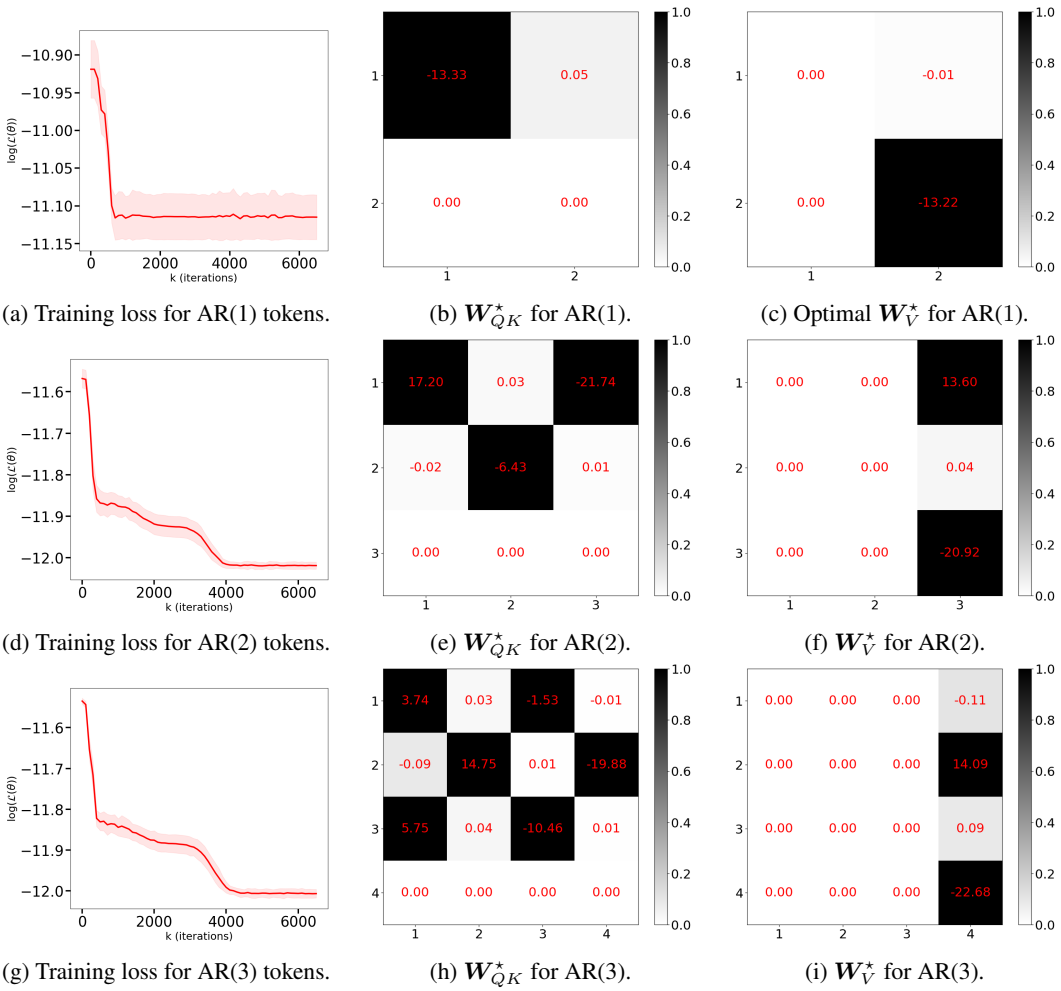
Notably, a forward pass using the optimal parameters (13) amounts to the prediction given after one GD step on $\mathcal{L}_{AR(1)}(w)$ starting from $w_0 = 0$. We thus recover the ICL-as-optimization view upheld by works in the i.i.d. setting (Ahn et al., 2023; Mahankali et al., 2023) but for LDS-produced data.

4.1 Discussion

To our knowledge, the only other architecture proposed for handling noisy observations y_t of type (1) is given by Cole et al. (2025). Theirs is part of a proof of existence by construction and, as such, is not accompanied by confirming experimental evidence. Different from us, they propose an attention-only transformer that unrolls a *modified Richardson iteration* meant to esitmate $\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_{i+1}\boldsymbol{x}_i^\top\right)\left(\frac{1}{T}\sum_{i=1}^{T} \boldsymbol{x}_i\boldsymbol{x}_i^\top\right)^{-1}$ for a simpler LDS with direct state access. Their construction extends to the setting of objective (4) via the work of Tsiamis & Pappas (2019), who give a high probability result for the existence of $\left(\sum_{t=1}^{T-s-1} \bar{\boldsymbol{y}}_t\bar{\boldsymbol{y}}_t^\top\right)^{-1}$ under our assumptions. However, their transformer has a minimum of two layers, of which the first is fixed, therefore providing no guarantee that training will recover it. Our results take a first step towards filling this gap.

Tangentially, Akram & Vikalo (2024) construct a transformer emulating the KF, contingent on knowledge of the system parameters and an elaborate token augmentation scheme. While this architecture is capable of computing the forward KF observation $\hat{y}_T$, it relies on ideal knowledge of LDS (1) which is rarely encountered in practice.

Theorem 4.1 sets forth a plausible hypothesis for prior experiments (Du et al., 2023, Fig. 2) using a GPT-2 architecture trained autoregressively with data (1) for stable $\boldsymbol{A} \in \mathbb{S}_{++}^d$. Their results highlight the transformer's competitive performance relative to the KF for predicting the next observation of a previously unseen sequence, in-context. These experiments suggest an implicit form of system identification might be executed in context, though the mechanism remains unstudied. Through the ICL-as-optimization lens, we can interpret the high accuracy of GPT-2's in-context predictions as a possible consequence of Theorem 2 of (Kozdoba et al., 2019). Importantly, the latter result implies



(a) Training loss for AR(1) tokens. (b) $\boldsymbol{W}_{QK}^{\star}$ for AR(1). (c) Optimal $\boldsymbol{W}_{V}^{\star}$ for AR(1).

(d) Training loss for AR(2) tokens. (e) $\boldsymbol{W}_{QK}^{\star}$ for AR(2). (f) $\boldsymbol{W}_{V}^{\star}$ for AR(2).

(g) Training loss for AR(3) tokens. (h) $\boldsymbol{W}_{QK}^{\star}$ for AR(3). (i) $\boldsymbol{W}_{V}^{\star}$ for AR(3).

Figure 1: Experimental results for AR(1–3) tokens showing the optimally-trained attention parameters.

that for an arbitrary, finite family S of LDSs (1) and an $\varepsilon > 0$, there exists a window-length $s(\varepsilon)$ such that the optimal AR($s(\varepsilon)$) predictor incurs an average error that is at least as good, up to ε, as that of the forward observation prediction $\hat{y}_{t+1}$ of the best KF in S. Our results take the first step in the exploration of this hypothesis.

5 Experiments

We now present numerical evidence supporting our theory. All experiments were implemented in Python 3.12 and run on a ThinkPad T14p with 32 GB RAM and a 22-core Intel Core™ Ultra 9 185H processor. The code is provided as part of the supplementary material.

We train architecture (6) on sequences $\{y_t\}_{t=1}^{T}$, $T = 30$, each sampled from a different LDS of type (1) with a hidden state dimension $d = 5$. The number of training iterations is 6500 for all cases with a doubling of the batchsize for every increase in order starting from 2000 for AR(1). A fresh batch of LDSs is sampled at every iteration (i.e., online setting). We sample $\boldsymbol{A}$'s diagonal entries uniformly at random in the interval $[-1, 1]$ and set $\boldsymbol{c} = \boldsymbol{1}_d$. The noise magnitudes are set to $\sigma_v^2 = \sigma_w^2 = 1\mathrm{e}{-2}$. We use window-sizes s ranging from 1 to 4, with results being averaged over 3 random seeds. The weights are learned using AdamW (Loshchilov & Hutter, 2017) with gradient clipping and a learning rate schedule consisting of a linear warm-up phase followed by cosine annealing (Loshchilov & Hutter, 2016). A full list of hyperparameters is provided in Tables 1 and 2 of Appendix B.

The results are depicted in Figure 1 for AR(1–3)-style tokens. Further experiments for AR(4) follow in the same vein, but are deferred to Appendix B due to lack of space. Subfigures (b,c) show an optimum conforming to Theorem 4.1 for AR(1) tokens. Moreover, subfigures (e,f) and (h,i) confirm experimentally the pattern uncovered by Lemma 4.1 for general $s > 0$. However, a quick calculation of the forward pass reveals that weights trained to optimality with AR(s) tokens (7) for $s \geq 2$ *do not* implement GD in the forward pass, or at least not in the formulaic manner of prior works. Instead, we notice a recursive pattern of the optimal weights that builds on top of the GD-inducing parameters recovered for AR(1). For ease of exposition, we illustrate the recursion for AR(2), but the pattern generalizes to higher orders according to our experiments.

To begin, we transpose expression (6) for generating the final token of the transformer's output as

$$[\bar{\boldsymbol{y}}_{T-2},\ \hat{y}_T] \ = \ [\bar{\boldsymbol{y}}_{T-2},\ 0] \ + \ \frac{1}{T-2}[\bar{\boldsymbol{y}}_{T-2}, 0]\ \boldsymbol{W}_{QK}\ \sum_{i=1}^{T-2} \boldsymbol{s}_i \boldsymbol{s}_i^\top\ \boldsymbol{W}_V,$$

and note that the linear predictor $\boldsymbol{W}_{QK}\ \sum_{i=1}^{T-2} \boldsymbol{s}_i \boldsymbol{s}_i^\top\ \boldsymbol{W}_V$ returned by our experiments amounts to

$$\left[\begin{array}{c|cc} c_1 & 0 & c_2 \\ \hline 0 & c_3 & 0 \\ 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c|cc} \sum_t y_t^2 & \sum_t y_t y_{t+1} & \sum_t y_t y_{t+2} \\ \hline \sum_t y_t y_{t+1} & \sum_t y_{t+1}^2 & \sum_t y_{t+1} y_{t+2} \\ \sum_t y_t y_{t+2} & \sum_t y_{t+1} y_{t+2} & \sum_t y_{t+2}^2 \end{array}\right] \left[\begin{array}{c|cc} 0 & 0 & c_4 \\ \hline 0 & 0 & 0 \\ 0 & 0 & c_5 \end{array}\right]$$

where $\{c_i\}_{i\in[5]}$ denote non-zero entries. We observe that the lower-right blocks of dimension 2×2 implement, up to a constant, the predictor corresponding to one GD step starting from $w_0 = 0$ on the forward-shifted AR(1) loss

$$\mathcal{L}_{AR(1)}(w) \ := \ \frac{1}{2(T-2)} \sum_{t=1}^{T-2} (y_{t+2} - w\, y_{t+1})^2.$$

This pattern holds recursively, with the empirically-determined optimal AR($s+1$) predictor building upon the structure of the empirically-determined optimal AR(s) one, but relative to a forward-shifted $\mathcal{L}_{AR(s)}$ (4). For example, in Figure 1 we observe the optimal AR(2) parameters embedded in the bottom right 3×3 blocks of their AR(3) counterparts. This observation hints at the possibility of arriving at AR(s)-optimal parameters by induction — an approach we leave for future exploration.

6 Conclusion, limitations, future directions

This paper presented the first steps towards characterizing the optimal configuration of a single self-attention layer trained with LDS-produced data and its ability to learn in context. We sketched a path forward by leveraging results from the literature on improper learning approaches to system identification, whereby autoregressive processes can well-approximate Kalman filters given a sufficient window size. Using this starting point for our study of ICL with non-i.i.d. data, we showed that for a length-one window, the optimal attention layer implements a step of GD on the context-induced autoregressive loss. Furthermore, we narrowed down the class of potential minimizers based on a structural property of the optimality condition, which we further confirmed empirically. Finally, our experiments uncovered a recursive pattern of the optimal weights, hinting at a structured family of global optima for this class of problems.

Due to the difficulties induced by correlated data, several limitations remain: establishing optimality for $s \geq 2$ by searching for optima within the structured class of parameters identified by Lemma 4.1; understanding what algorithmic primitive, if any, is implemented by configurations pertaining to AR(s), $s \geq 2$; and finally, extending this analysis to autoregressive pretraining objectives. Our present contributions provide the necessary building blocks for addressing these directions in future work.

REFERENCES

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.

Usman Akram and Haris Vikalo. Can transformers in-context learn behavior of a linear dynamical system?, 2024. URL https://arxiv.org/abs/2410.16546.

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.

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.

Frank Cole, Yulong Lu, Tianhao Zhang, and Yuxuan Zhao. In-context learning of linear dynamical systems with transformers: Error bounds and depth-separation. *arXiv preprint arXiv:2502.08136*, 2025.

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.

Zhe Du, Haldun Balim, Samet Oymak, and Necmiye Ozay. Can transformers learn optimal filtering for unknown systems? *IEEE Control Systems Letters*, 7:3525–3530, 2023.

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.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, pp. 11398–11442. PMLR, 2023.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Gautam Goel and Peter Bartlett. Can a transformer represent a kalman filter? In *6th Annual Learning for Dynamics & Control Conference*, pp. 1502–1512. PMLR, 2024.

James D Hamilton. Time series analysis, 1995.

P Jeff Harrison. Convergence and the constant dynamic linear model. *Journal of Forecasting*, 16(5): 287–292, 1997.

Leon Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12(1/2):134–139, 1918.

Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

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

Mark Kozdoba, Jakub Marecek, Tigran Tchrakian, and Shie Mannor. On-line learning of linear dynamical systems: Exponential forgetting in kalman filters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4098–4105, 2019.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Thomas P Minka. From hidden markov models to linear dynamical systems. Technical report, Citeseer, 1999.

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.

Michael E Sander and Gabriel Peyré. Towards understanding the universality of transformers for next-token prediction. *arXiv preprint arXiv:2410.03011*, 2024.

Michael E Sander, Raja Giryes, Taiji Suzuki, Mathieu Blondel, and Gabriel Peyré. How do transformers perform in-context autoregressive learning? *arXiv preprint arXiv:2402.05787*, 2024.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Do pretrained transformers learn in-context by gradient descent? *arXiv preprint arXiv:2310.08540*, 2023.

Anastasios Tsiamis and George J Pappas. Finite sample analysis of stochastic system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3648–3654. IEEE, 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, 2023a.

Johannes Von Oswald, Maximilian Schlegel, Alexander Meulemans, Seijin Kobayashi, Eyvind Niklasson, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Max Vladymyrov, et al. Uncovering mesa-optimization algorithms in transformers. *arXiv preprint arXiv:2309.05858*, 2023b.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36:15614–15638, 2023.

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

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 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.

Chenyu Zheng, Wei Huang, Rongzhen Wang, Guoqiang Wu, Jun Zhu, and Chongxuan Li. On mesa-optimization in autoregressively trained transformers: Emergence and capability. *Advances in Neural Information Processing Systems*, 37:49081–49129, 2024.

Ingvar Ziemann, Nikolai Matni, and George J Pappas. State space models, emergence, and ergodicity: How many parameters are needed for stable predictions? *arXiv preprint arXiv:2409.13421*, 2024.

A LLM USAGE DISCLOSURE

LLMs were used in elaborating this paper as follows:

- Finding related work.
- Computing the result of polynomial multiplications.
- Generating LaTeX tables and tikz figures.
- Transferring proofs from pen-and-paper format into LaTeX automatically using the online tool Manus `https://manus.im/`.

B EXPERIMENTS — FURTHER DETAILS

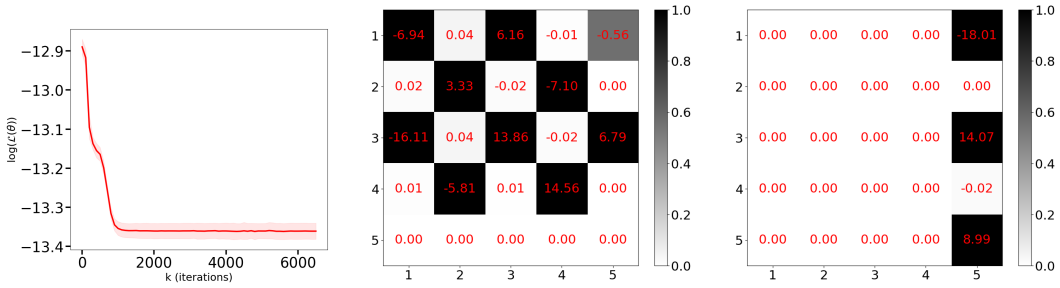
B.1 HYPERPARAMETERS

Below are the full details of the training procedure described in Section 5.

Table 1: LDS hyperparameters

Hyperparameter	**Value**
Hidden state size	$\boldsymbol{x} \in \mathbb{R}^5$
Observation size	$\boldsymbol{y} \in \mathbb{R}$
State transition	$\boldsymbol{A} = \mathrm{diag}(\boldsymbol{a}), \boldsymbol{a} \sim \mathcal{U}([-1,1]^5)$
Observation matrix	$\boldsymbol{c} = \mathbf{1}_5$
Process noise magnitude	$\sigma_w^2 = \texttt{1e-2}$
Observation noise magnitude	$\sigma_v^2 = \texttt{1e-2}$
Sequence length	30
$\boldsymbol{x}_0$	$\boldsymbol{x}_0 \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{I}), \sigma_0^2 = \texttt{1e-3}$

B.2 ADDITIONAL EXPERIMENTS



(a) Training loss for AR(4) tokens. (b) $\boldsymbol{W}_{QK}^{\star}$ for AR(4). (c) $\boldsymbol{W}_{V}^{\star}$ for AR(4).

Figure 2: Experimental results for various token configurations AR(4) showing the optimal attention parameters.

Table 2: Training hyperparameters

Hyperparameter	**Value**
Weight initialization	Xavier normal distribution (Glorot & Bengio, 2010) with gain $= 1e{-}5$
Optimizer	AdamW (Loshchilov & Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e{-}9$
Weight decay	$5e{-}3$
Learning rate (i.e., max. val.)	$5e{-}2$
Min. learning rate	$1e{-}3$
Linear warmup	500 iter.
Decay schedule	Cosine annealing (Loshchilov & Hutter, 2016)
Max. decay steps	2000 iter.
Max. grad norm (clipping)	300
Random seeds	$\{666013, 1, 0\}$
Batch size / iter.	2000 for AR(1), 4000 for AR(2), 8000 for AR(3), 16000 for AR(4)
Total iter.	6501

C Proof of token construction Lemma

Lemma 3.1. *For a given $s >= 1$, there exists an $s+1$-headed linear attention layer with positional encoding which transforms input sequences $[y_1, y_2, \ldots, y_T]^\top$ into*

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_s & y_{s+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{T-s-1} & y_{T-s} & \cdots & y_{T-2} & y_{T-1} \\ y_{T-s} & y_{T-s+1} & \cdots & y_{T-1} & 0 \\ & & \mathbf{0}_{T-s-1 \times s} & & \end{bmatrix}.$$

The latter are essentially equivalent to tokens (7)*.*

Proof. We first define a matrix right-shift operator, which shifts each row one position to the right, padding the first column with zeros. Let $\gg$: $R^{m\times n} \to R^{m\times n}$ be $\gg(\boldsymbol{M}) = \boldsymbol{M}\boldsymbol{R}$, where

$$\boldsymbol{R} = \begin{bmatrix} 0 & \mathbf{0}_{n-1}^\top \\ \mathbf{0}_{n-1} & \boldsymbol{I}_{n-1} \end{bmatrix}. \tag{14}$$

We follow Von Oswald et al. (2023a) in using the one-hot positional encodings, concatenated to the input sequence to obtain tokens $\{[y_t, \boldsymbol{e}_t]\}_{t=1}^T$. We define $s+1$ attention heads given by

Define $\boldsymbol{W}_Q \in \mathbb{R}^{T+1\times T}$, $\boldsymbol{W}_K \in \mathbb{R}^{T+1\times T}$ and $\boldsymbol{W}_V \in \mathbb{R}^{T+1\times s}$ as follows:

$$\boldsymbol{W}_Q^h = \begin{bmatrix} \mathbf{0}_T^\top \\ \boldsymbol{I}_T \end{bmatrix},\ \forall h \in [s+1]$$

$$(\boldsymbol{W}_K^h)^\top = \begin{bmatrix} \mathbf{0}_T, & \underbrace{\gg(\ldots \gg}_{h-1 \text{ times}}(\boldsymbol{I}_T)\ldots) \end{bmatrix}$$

$$\boldsymbol{W}_V^h = \begin{matrix} 1 & \ldots & h & \ldots & s+1 \end{matrix} \\ \begin{bmatrix} \mathbf{0}_{T+1} & \ldots & \boldsymbol{e}_1 & \ldots & \mathbf{0}_{T+1} \end{bmatrix},\ \forall h \in [s+1] \tag{15}$$

Each head then computes the following

$$\underbrace{\begin{bmatrix} y_1 & 1 & 0 & \dots & 0 \\ y_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_T & 0 & 0 & \dots & 1 \end{bmatrix} \boldsymbol{W}_Q^k}_{=\boldsymbol{I}_T} \underbrace{(\boldsymbol{W}_K^h)^\top \begin{bmatrix} y_1 & y_2 & y_3 & \dots & y_T \\ 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}}_{=\begin{bmatrix} \mathbf{0}_{T-h+1\times h-1} & \boldsymbol{I}_{T-h+1} \\ \mathbf{0}_{h-1\times h-1} & \mathbf{0}_{h-1\times T-h+1} \end{bmatrix}} \underbrace{\boldsymbol{W}_V \begin{bmatrix} y_1 & 1 & 0 & \dots & 0 \\ y_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_T & 0 & 0 & \dots & 1 \end{bmatrix}}_{= \begin{bmatrix} 1 & \dots & h & \dots & s+1 \\ 0 & \dots & y_1 & \dots & 0 \\ 0 & \dots & y_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & \dots & y_T & \dots & 0 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & \dots & h & \dots & s+1 \\ 0 & \dots & y_h & \dots & 0 \\ 0 & \dots & y_{h+1} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & \dots & y_T & \dots & 0 \\ & & \mathbf{0}_{h\times s+1} & & \end{bmatrix}$$

Summing over the outputs of all heads, we get an equivalent representation to (7). □

D COMPUTATION OF TOKENS' HIGH-ORDER MOMENTS

For brevity, in Appendix D, Appendix E and Appendix F use $\tilde{\boldsymbol{D}}$, with $\tilde{\boldsymbol{D}} \coloneqq \boldsymbol{A}, \boldsymbol{x}_0, \{\boldsymbol{w}_t\}, \{v_t\}$.

Lemma D.1. *If y_i and y_j are generated by (1),* $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_i y_j\right] =$

$$\begin{cases} \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^j)^\top\right] \boldsymbol{c}^\top + \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\right] \boldsymbol{c}^\top, \textit{if } i \neq j, \\ \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^i)^\top\right] \boldsymbol{c}^\top + \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{i-1-k}\right] \boldsymbol{c}^\top + \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[v_i^2\right], \textit{if } i = j. \end{cases} . \tag{16}$$

Proof. Assume $i \leq j$ without loss of generality, unroll y_i and y_j to $\boldsymbol{x}_i$

$$\begin{aligned} \mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j] &= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[(\boldsymbol{c}\boldsymbol{x}_i + v_i)\left(\boldsymbol{c}\boldsymbol{A}^{j-i}\boldsymbol{x}_i + \boldsymbol{c}\sum_{k=i}^{j-1}\boldsymbol{A}^{j-1-k}\boldsymbol{w}_k + v_j\right)^\top\right] \\ &= \mathbb{E}_{\tilde{\boldsymbol{D}}}[\boldsymbol{c}\boldsymbol{x}_i\boldsymbol{x}_i^\top(\boldsymbol{A}^\top)^{j-i}\boldsymbol{c}^\top] + \underbrace{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{c}\boldsymbol{x}_i\left(\boldsymbol{c}\sum_{k=i}^{j-1}\boldsymbol{A}^{j-1-k}\boldsymbol{w}_k\right)^\top\right]}_{=0,\text{ since } \boldsymbol{x}_i \text{ is independent of } \boldsymbol{w}_k \text{ for } k=i,\ldots,j-1} + \underbrace{\mathbb{E}_{\tilde{\boldsymbol{D}}}[\boldsymbol{c}\boldsymbol{x}_i v_j^\top]}_{=0,\text{ since } \boldsymbol{x}_i \text{ is independent of } v_j} \\ &+ \underbrace{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[v_i(\boldsymbol{c}\boldsymbol{A}^{j-i}\boldsymbol{x}_i)^\top\right]}_{=0,\text{ since } v_i \text{ is independent of } \boldsymbol{x}_i} + \underbrace{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[v_i\left(\boldsymbol{c}\sum_{k=i}^{j-1}\boldsymbol{A}^{j-1-k}\boldsymbol{w}_k\right)^\top\right]}_{=0,\text{ since } v_i \text{ is independent of } \boldsymbol{w}_{k=i\ldots j-1}} + \underbrace{\mathbb{E}_{\tilde{\boldsymbol{D}}}[v_i v_j^\top]}_{\substack{\neq 0,\text{ if } i=j,\\ 0,\text{ if } i\neq j}}. \end{aligned} \tag{17}$$

Unroll $\boldsymbol{x}_i$ to $\boldsymbol{x}_0$, the remaining non-zero term

$$\begin{aligned} &\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{c}\boldsymbol{x}_i\boldsymbol{x}_i^\top(\boldsymbol{A}^{j-i})^\top\boldsymbol{c}^\top\right] \\ &= \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\boldsymbol{A}^i\boldsymbol{x}_0 + \sum_{k=0}^{i-1}\boldsymbol{A}^{i-1-k}\boldsymbol{w}_k\right)\left(\boldsymbol{A}^j\boldsymbol{x}_0 + \sum_{l=0}^{i-1}\boldsymbol{A}^{i-1-l}\boldsymbol{w}_l\right)^\top\right]\boldsymbol{c}^\top \\ &= \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i\boldsymbol{x}_0\boldsymbol{x}_0^\top(\boldsymbol{A}^j)^\top\right] + \underbrace{\boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i\boldsymbol{x}_0\left(\sum_{l=0}^{j-1}\boldsymbol{A}^{j-1-l}\boldsymbol{w}_l\right)^\top\right]\boldsymbol{c}^\top}_{=0,\text{ since } \boldsymbol{x}_0 \text{ is independent of } \boldsymbol{w}_l} \\ &+ \underbrace{\boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\sum_{k=0}^{i-1}\boldsymbol{A}^{i-1-k}\boldsymbol{w}_k\right)\boldsymbol{x}_0^\top(\boldsymbol{A}^j)^\top\right]\boldsymbol{c}^\top}_{=0,\text{ since } \boldsymbol{x}_0 \text{ is independent of } \boldsymbol{w}_k} \\ &+ \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\sum_{k=0}^{i-1}\boldsymbol{A}^{i-1-k}\boldsymbol{w}_k\right)\left(\sum_{l=0}^{j-1}\boldsymbol{A}^{j-1-l}\boldsymbol{w}_l\right)^\top\right]\boldsymbol{c}^\top. \end{aligned} \tag{18}$$

Since $\boldsymbol{w}_k$ is zero-mean and temporally independent, $\mathbb{E}[\boldsymbol{w}_k\boldsymbol{w}_l^\top] = \boldsymbol{0}$, if $k \neq l$. The remaining non-zero part of $\boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\sum_{k=0}^{i-1}\boldsymbol{A}^{i-1-k}\boldsymbol{w}_k\right)\left(\sum_{l=0}^{j-1}\boldsymbol{A}^{j-1-l}\boldsymbol{w}_l\right)^\top\right]\boldsymbol{c}^\top$ is

$$\boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1}\boldsymbol{A}^{i-1-k}\boldsymbol{w}_k\boldsymbol{w}_k^\top(\boldsymbol{A}^\top)^{j-1-k}\right]\boldsymbol{c}^\top. \tag{19}$$

Based on the computation above,

$$\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\, y_i y_j \,\right] =$$
$$\begin{cases} \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^j)^\top\right] \boldsymbol{c}^\top + \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\right] \boldsymbol{c}^\top, \text{if } i \neq j, \\ \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^i)^\top\right] \boldsymbol{c}^\top + \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{i-1-k}\right] \boldsymbol{c}^\top + \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[v_i^2\right], \text{if } i = j. \end{cases} \tag{20}$$

□

Lemma D.2. *If $i + j$ is odd, $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j] = 0$; if $i + j$ is even, $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j] \geq 0$. This extends to the 4^{th} and the 6^{th} moments of y, which means if the sum of y indices is odd, $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j y_k y_l] = 0$ and $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j y_k y_l y_m y_n] = 0$; if the sum of y indices is even, $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j y_k y_l] \geq 0$ and $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j y_k y_l y_m y_n] \geq 0$.*

For brevity, only the proof for $\mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j] = 0$ is given.

Proof. $\mathbb{E}_{\tilde{\boldsymbol{D}}}$ is the sum of 2 or 3 terms. Compute each term respectively.

$$\begin{aligned} \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^j)^\top\right] \boldsymbol{c}^\top &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^j)^\top\right] \boldsymbol{c}^\top \\ &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[\boldsymbol{A}^i \mathbb{E}_{\boldsymbol{x}_0}[x_0 x_0^\top | \boldsymbol{A}](\boldsymbol{A}^j)^\top] \boldsymbol{c}^\top \\ &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[\boldsymbol{A}^i \Sigma_0 (\boldsymbol{A}^j)^\top] \boldsymbol{c}^\top \\ &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[\boldsymbol{A}^i (\boldsymbol{A}^j)^\top] \Sigma_0 \boldsymbol{c}^\top \\ &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[\boldsymbol{A}^{i+j}] \Sigma_0 \boldsymbol{c}^\top. \end{aligned} \tag{21}$$

$$\begin{aligned} \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\right] \boldsymbol{c}^\top &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A},\boldsymbol{w}_k}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\right] \boldsymbol{c}^\top \\ &= \sum_{k=0}^{i-1} \boldsymbol{c}\mathbb{E}_{A,\boldsymbol{w}_k}[A^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (A^\top)^{j-1-k}] \boldsymbol{c}^\top \\ &= \sum_{k=0}^{i-1} \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[A^{i-1-k} E_{\boldsymbol{w}_k}[\boldsymbol{w}_k \boldsymbol{w}_k^\top | A](A^\top)^{j-1-k}] \boldsymbol{c}^\top \\ &= \sum_{k=0}^{i-1} \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[A^{i-1-k} \sigma_w^2 \boldsymbol{I} (A^\top)^{j-1-k}] \boldsymbol{c}^\top \\ &= \sum_{k=0}^{i-1} \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[A^{i-1-k} A^{j-1-k}] \sigma_w^2 \boldsymbol{I} \boldsymbol{c}^\top \\ &= \sum_{k=0}^{i-1} \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[A^{i+j-2-2k}] \sigma_w^2 \boldsymbol{I} \boldsymbol{c}^\top. \end{aligned} \tag{22}$$

Based on the computation above,

$$\begin{aligned} \mathbb{E}_{\tilde{\boldsymbol{D}}}[y_i y_j] &= \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^j)^\top\right] \boldsymbol{c}^\top + \boldsymbol{c}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k} \boldsymbol{w}_k \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\right] \boldsymbol{c}^\top (+\mathbb{E}_{\tilde{\boldsymbol{D}}}[v_i v_j]) \\ &= \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[\boldsymbol{A}^{i+j}] \Sigma_0 \boldsymbol{c}^\top + \sum_{k=0}^{i-1} \boldsymbol{c}\mathbb{E}_{\boldsymbol{A}}[A^{i+j-2-2k}] \sigma_w^2 \boldsymbol{I} \boldsymbol{c}^\top + (+\mathbb{E}_{\tilde{\boldsymbol{D}}}[v_i v_j]). \end{aligned} \tag{23}$$

□

E PROOFS OF THE IN-CONTEXT LOSS' GRADIENT LEMMA

Lemma 4.1. *For an arbitrary s, the following parameters induce a banded structure in the left-hand side of* (10) *matching that of the right-hand side.*

$$\boldsymbol{W}_{QK} = \begin{bmatrix} \star & 0 & \star & \cdots & \\ 0 & \star & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \star \\ 0 & \cdots & 0 & \star & 0 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad \boldsymbol{W}_{V} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \vdots \\ \vdots & & & \vdots & 0 \\ \vdots & & & \vdots & \star \\ \vdots & & & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & \star \end{bmatrix}. \tag{12}$$

Proof. Recall the in-context loss in (8) with a general AR(s)-constructed input token matrix $\boldsymbol{Y}_0 = \begin{bmatrix} \bar{\boldsymbol{y}}_1 & \bar{\boldsymbol{y}}_2 & \cdots & \bar{\boldsymbol{y}}_{T-s-1} & \bar{\boldsymbol{y}}_{T-s} \\ y_{s+1} & y_{s+2} & \cdots & y_{T-1} & 0 \end{bmatrix}$ is defined as

$$\mathcal{L}(\theta) \coloneqq \mathbb{E}_{\tilde{\boldsymbol{D}}} \left[\left(\mathcal{T}_\theta \left(\boldsymbol{Y}_0 \right)_{s+1,T-s} - y_T \right)^2 \right]. \tag{24}$$

From (25) to (29), we use the same reformulations in (Ahn et al., 2023). The last column of the transformer's output above can be written as

$$\begin{bmatrix} \bar{\boldsymbol{y}}_{T-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{y}}_{T-1} \\ 0 \end{bmatrix} + \frac{1}{T-s-1} \boldsymbol{W}_V^\top \left(\sum_{i=1}^{T-s-1} \begin{bmatrix} \bar{\boldsymbol{y}}_i \bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s} \\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2 \end{bmatrix} \right) \boldsymbol{W}_{QK}^\top \begin{bmatrix} \bar{\boldsymbol{y}}_{T-s} \\ 0 \end{bmatrix}, \tag{25}$$

where the summation is for $i = 1, 2, ..., n$ due to the causal mask. The transformer's prediction of y_T, $\mathcal{T}_\theta \left(\boldsymbol{Y}_0 \right)_{s+1,T-s}$ can be written as

$$\frac{1}{T-s-1} \boldsymbol{b}^\top \left(\underbrace{\sum_{i=1}^{T-s-1} \begin{bmatrix} \bar{\boldsymbol{y}}_i \bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s} \\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2 \end{bmatrix}}_{\coloneqq \bar{\boldsymbol{Y}} \in \mathbb{R}^{(s+1)\times(s+1)}} \right) \left[\boldsymbol{a}_1 \boldsymbol{a}_2 \cdots \boldsymbol{a}_s \right] \bar{\boldsymbol{y}}_{T-s}, \tag{26}$$

where $\boldsymbol{b}^\top \in \mathbb{R}^{1\times(s+1)}$ is the last row of $\boldsymbol{W}_V^\top$ and $\boldsymbol{a}_j \in \mathbb{R}^{(s+1)}$ is the j^{th} column of $\boldsymbol{W}_{QK}^\top$. So the in-context loss $\mathcal{L}(\boldsymbol{W}_V, \boldsymbol{W}_{QK})$ can be rewritten as a function of $\boldsymbol{b}^\top$ and $\boldsymbol{a}_j$

$$\mathcal{L}(\boldsymbol{b}^\top, \boldsymbol{a}_j) \coloneqq \mathbb{E}_{\tilde{\boldsymbol{D}}} \left[\left(\frac{1}{T-s-1} \boldsymbol{b}^\top \bar{\boldsymbol{Y}} \boldsymbol{A} \bar{\boldsymbol{y}}_{T-s} - y_T \right)^2 \right]. \tag{27}$$

Plugging in the expression of $\boldsymbol{y}_{\bar{T-s}}$, the in-context loss is

$$\begin{aligned} \mathcal{L}(\boldsymbol{b}^\top, \boldsymbol{a}_j) &= \mathbb{E}_{\tilde{\boldsymbol{D}}} \left[\left(\frac{1}{T-s-1} \boldsymbol{b}^\top \bar{\boldsymbol{Y}} \left[\boldsymbol{a}_1 \boldsymbol{a}_2 \cdots \boldsymbol{a}_s \right] \begin{bmatrix} y_{T-s} \\ y_{T-s+1} \\ \vdots \\ y_{T-1} \end{bmatrix} - y_T \right)^2 \right] \\ &= \mathbb{E}_{\tilde{\boldsymbol{D}}} \left[\left(\frac{1}{T-s-1} \sum_{k=1}^{s} \boldsymbol{b}^\top \bar{\boldsymbol{Y}} \boldsymbol{a}_k y_{T-s-1+k} - y_T \right)^2 \right] \end{aligned}$$

$$= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\mathrm{Tr}(\bar{\boldsymbol{Y}}\boldsymbol{a}_k\boldsymbol{b}^\top)y_{T-s-1+k} - y_T\right)^2\right]$$

$$= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{b}\boldsymbol{a}_k^\top\rangle y_{T-s-1+k} - y_T\right)^2\right]. \tag{28}$$

Write the in-context loss as a function of $\boldsymbol{X}_k \coloneqq \boldsymbol{b}\boldsymbol{a}_k^\top$, which represent the transformer parameters

$$\mathcal{L}(\boldsymbol{X}_{k=1\cdots s}) = \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{X}_k\rangle y_{T-s-1+k} - y_T\right)^2\right]. \tag{29}$$

Now we compute the gradient of the in-context loss with respect to each $\boldsymbol{X}_k$, which will be used for showing the optimality

$$\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{k=1\cdots s}) = 2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{X}_k\rangle y_{T-s-1+k} - y_T\right) y_{T-s-1+j}\bar{\boldsymbol{Y}}\right]. \tag{30}$$

The gradient $\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{k=1\cdots s})$ contains 2 terms. $\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{k=1\cdots s}) = \mathbf{T}^1_{\boldsymbol{X}_j} + \mathbf{T}^2_{\boldsymbol{X}_j}$, with

$$\mathbf{T}^1_{\boldsymbol{X}_j} \coloneqq \frac{2}{T-s-1}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\underbrace{\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{X}_k\rangle y_{T-s-1+k}y_{T-s-1+j}}_{\coloneqq C^1}\bar{\boldsymbol{Y}}\right] \tag{31}$$

$$= \frac{2}{T-s-1}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[C^1\sum_{i=1}^{T-s-1}\begin{bmatrix}\bar{\boldsymbol{y}}_i\bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s}\\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2\end{bmatrix}\right], \tag{32}$$

$$\mathbf{T}^2_{\boldsymbol{X}_j} \coloneqq -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\underbrace{\sum_{i=1}^{s} y_T y_{T-s-1+j}}_{\coloneqq C^2}\bar{\boldsymbol{Y}}\right] \tag{33}$$

$$= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[C^2\sum_{i=1}^{T-s-1}\begin{bmatrix}\bar{\boldsymbol{y}}_i\bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s}\\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2\end{bmatrix}\right]. \tag{34}$$

Each matrix element of $\mathbf{T}^1_{\boldsymbol{X}_j}$ can be written as

$$\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[C^1\sum_{i=1}^{T-s-1} y_{i+m}y_{i+n}\right] \tag{35}$$

$$= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=1}^{s} C_k^{pq} y_{T-s-1+k}\left(\sum_{r=1}^{T-s-1} y_{r+p}y_{r+q}\right) y_{T-s-1+j}\sum_{i=1}^{T-s-1} y_{i+m}y_{i+n}\right], \tag{36}$$

with C_k^{pq} represents the matrix elements of $\boldsymbol{X}_k$, $j \in [1, s]$, $p \in [0, s]$, $q \in [0, s]$, $m \in [0, s]$ and $n \in [0, s]$.

A general term in the sum of $\mathbf{T}^1_{\boldsymbol{X}_j}$ is

$$C_k^{pq}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\, y_{T-s-1+k}y_{r+p}y_{r+q}y_{T-s-1+j}y_{i+m}y_{i+n}\,\right]. \tag{37}$$

The sum of y's indices in (37) is $2T + 2i + 2r - 2s - 2 + (m + n + p + q + j + k)$. The parity depends only on $(m + n + p + q + j + k)$ and is the same for all the terms in the sum (no matter which i and which r).

According to D.2, (37) is 0, if $(m+n-s-1+j)$ is odd; is non-negative, if $(m+n+p+q+j+k)$ is even. So a general matrix element of $\mathbf{T}^1_{\boldsymbol{X}_j}$ is 0, if $(m + n + p + q + j + k)$ is odd; is non-negative, if $(m + n + p + q + j + k)$ is even.

Each matrix element of $\mathbf{T}^2_{\boldsymbol{X}_j}$ can be written as

$$\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[C^2 \sum_{i=1}^{T-s-1} y_{i+m} y_{i+n} \right] = \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-s-1+j} \sum_{i=1}^{T-s-1} y_{i+m} y_{i+n} \right], \tag{38}$$

with $j \in [1, s]$, $m \in [0, s]$ and $n \in [0, s]$.

A general term in the sum of $\mathbf{T}^2_{\boldsymbol{X}_j}$ is

$$\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\, y_T y_{T-s-1+j} y_{i+m} y_{i+n} \,\right]. \tag{39}$$

The sum of y's indices in (39) is $2T + 2i + (m + n - s - 1 + j)$. The parity depends only on $(m + n - s - 1 + j)$ and is the same for all the terms in the sum (no matter which i).

According to D.2, (39) is 0, if $(m + n - s - 1 + j)$ is odd; is non-negative, if $(m + n - s - 1 + j)$ is even. So a general matrix element of $\mathbf{T}^2_{\boldsymbol{X}_j}$ is 0, if $(m + n - s - 1 + j)$ is odd; is non-negative, if $(m + n - s - 1 + j)$ is even.

For a given AR(s)-constructed token (s is fixed) and a specific j, if a matrix element of $\mathbf{T}^2_{\boldsymbol{X}_j}$ is 0 only depends on $m + n$ (its position in the matrix). So

$$\mathbf{T}^2_{\boldsymbol{X}_j} = \begin{bmatrix} * & 0 & * & \cdots & \cdots & \\ 0 & * & 0 & * & & \vdots \\ * & 0 & * & \ddots & \ddots & \vdots \\ \vdots & * & \ddots & \ddots & \ddots & * \\ \vdots & & \ddots & \ddots & * & 0 \\ & \cdots & \cdots & * & 0 & * \end{bmatrix}, if(-s-1-j) \text{ is even;}$$

$$= \begin{bmatrix} 0 & * & 0 & \cdots & \cdots & \\ * & 0 & * & 0 & & \vdots \\ 0 & * & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 & * \\ & \cdots & \cdots & 0 & * & 0 \end{bmatrix}, if(-s-1-j) \text{ is odd.} \tag{40}$$

To make $\mathbf{T}^1_{\boldsymbol{X}_j}$ have the same 0 and non-zero elements at the same positions as $\mathbf{T}^2_{\boldsymbol{X}_j}$, $(p + q + k)$ should have the same parity as $(-s - 1 - j)$. It can be easily proved that a specific set of $\boldsymbol{X}_k$ can achieve this. So $\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{i=1\cdots s}) = \mathbf{T}^1_{\boldsymbol{X}_j} + \mathbf{T}^2_{\boldsymbol{X}_j}$ will have the pattern of (40).

□

F Proofs of the Optimality Theorem with AR(1)-Constructed Input Token

Theorem 4.1. *Let $\boldsymbol{Y}_0$ encode the input tokens according to construction (7) for $s = 1$. Then, the optimal parameters $\theta^\star = \left(\boldsymbol{W}_{QK}^\star, \boldsymbol{W}_V^\star\right)$ of a single linear self-attention layer with respect to loss $\mathcal{L}(\theta)$ are*

$$\boldsymbol{W}_{QK}^\star = \begin{bmatrix} \frac{(T-2)\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} y_{T-1} y_T\right]}{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1} y_T\right]} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{W}_V^\star = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{13}$$

up to rescaling with $\gamma \neq 0$.

Proof. For the transformer parameters in (13), the corresponding $\boldsymbol{b}^\top = [0 \quad 1]$ and the corresponding $\boldsymbol{A} = [c \quad 0]$, where $c \coloneqq \frac{(T-2)\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} y_{T-1} y_T\right]}{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1} y_T\right]}$.

So $\boldsymbol{X} = \boldsymbol{X}_1 = \boldsymbol{b}\boldsymbol{a}_1^\top = \boldsymbol{b}\boldsymbol{A}^\top = \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix}$. Use the result of $\boldsymbol{X}$ to compute the terms of the gradient of the in-context loss $\nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X})$

$$\begin{aligned}
\mathbf{T}^1_{\boldsymbol{X}_j} &= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\langle \bar{\boldsymbol{Y}}, \boldsymbol{X}\rangle y_{T-1}^2 \bar{\boldsymbol{Y}}\right] \\
&= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\langle \sum_{r=1}^{T-2} \begin{bmatrix} y_r y_r^\top & y_r y_{r+1} \\ y_{r+1} y_r & y_{r+1}^2 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix} \rangle y_{T-1}^2 \sum_{i=1}^{T-2} \begin{bmatrix} y_1^2 & y_i y_{i+1} \\ y_{i+1} y_i & y_{i+1}^2 \end{bmatrix}\right] \\
&= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} \begin{bmatrix} y_1^2 & y_i y_{i+1} \\ y_{i+1} y_i & y_{i+1}^2 \end{bmatrix}\right] \\
&= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} \begin{bmatrix} 0 & y_i y_{i+1} \\ y_{i+1} y_i & 0 \end{bmatrix}\right].
\end{aligned} \tag{41}$$

According to D.2, the 2 diagonal elements in (41) $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} y_i^2\right]$ and $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} y_{i+1}^2\right]$ are 0, since their sum of y indices are both odd.

$$\begin{aligned}
\mathbf{T}^2_{\boldsymbol{X}_j} &= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} \bar{\boldsymbol{Y}}\right] \\
&= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} \begin{bmatrix} y_1^2 & y_i y_{i+1} \\ y_{i+1} y_i & y_{i+1}^2 \end{bmatrix}\right] \\
&= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} \begin{bmatrix} 0 & y_i y_{i+1} \\ y_{i+1} y_i & 0 \end{bmatrix}\right].
\end{aligned} \tag{42}$$

According to D.2, the 2 diagonal elements in (42) $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} y_i^2\right]$ and $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} y_{i+1}^2\right]$ are 0, since their sum of y indices are both odd.

Plug in the expression of c, it can be easily found that

$$\nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}) = \mathbf{T}^1_{\boldsymbol{X}_j} + \mathbf{T}^2_{\boldsymbol{X}_j} = 0. \tag{43}$$

Since the in-context loss is convex in $\boldsymbol{X}$ and the $\boldsymbol{X}$ resulting from the $\boldsymbol{W}_V^\star$ and $\boldsymbol{W}_{QK}^\star$ above makes $\nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}) = 0$, the $\boldsymbol{W}_V^\star$ and $\boldsymbol{W}_{QK}^\star$ above is a global minimizer for the in-context loss. □