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 uncover a relation to the Preconditioned Conjugate Gradient method for 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). Furthermore, dynamical systems have been directly studied as potentially more flexible models for grammatical sentence formation, both empirically (Elman, 1995; Tabor et al., 1996) and more formally (Beim Graben et al., 2004; Belanger & Kakade, 2015), thus making setting (1) particularly relevant. 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, thus narrowing down the search for the provably-optimal linear attention layer when $s \geq 2$.
- **C3.** In Section 5, we provide numerical experiments verifying our theory for order-one autoregressive approximations. Furthermore, we connect the tiling pattern of empirically determined minimizers of order-s approximations, $s \geq 2$, with the Preconditioned Conjugate Gradient method iteration, thus further highlighting the view of ICL as on-the-fly optimization. To our knowledge, this is the first interpretation of the in-context algorithm for general order-s autoregression.
- **C4.** 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 $\boldsymbol{1}_d$ and $\boldsymbol{0}_d$ the all-ones and all-zeros vectors of dimension d, and by $\boldsymbol{1}_{d\times m}$ and $\boldsymbol{0}_{d\times m}$ 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 $\{1, 2, \dots n\}$. We write w.p. as an abbreviation of "with probability".

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

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

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 $\hat{y}_{t+1} \coloneqq \boldsymbol{c}^\top\boldsymbol{A}\hat{\boldsymbol{x}}_t = \mathbb{E}[y_{t+1} \,|\, y_t, \dots y_1]$. 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 $\mathbb{E}[y_{t+1} \,|\, y_t, \dots y_1]$, 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.

To ensure the above approximation is valid, we introduce the following LDS assumption.

Assumption 3.1 (System assumptions)**.** *LDS* (1) *has strictly positive definite noise covariances* $\Sigma_{\boldsymbol{w}}$ *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*

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

has a column rank of d*.*

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).

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.1 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 distribution, 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 parameters $\boldsymbol{A}$ and $\boldsymbol{c}$ sampled from appropriate distributions, and $\boldsymbol{x}_0$ sampled from $\mathcal{N}(\mathbf{0}_d, \Sigma_{\boldsymbol{x}_0})$. Our assumptions on the distributions of $\boldsymbol{A}$ and $\boldsymbol{c}$ are

Assumption 3.2 (LDS family). *The system matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is sampled from a centrally symmetric distribution supported on $\left\{\boldsymbol{M} \in \mathbb{R}^{d\times d} \;\middle|\; \rho(\boldsymbol{M}) \leq 1\right\}$, for which it holds that*

$$\mathbb{P}\left(\{\boldsymbol{A} \mid \exists i, j \in [d],\ s.t.\ \lambda_i(\boldsymbol{A}) = \lambda_j(\boldsymbol{A})\}\right) = 0. \tag{8}$$

In other words, $\boldsymbol{A}$ has a simple spectrum almost surely. The observation vector $\boldsymbol{c} \in \mathbb{R}^d$ is sampled independently, from a distribution that is absolutely continuous w.r.t. the Lebesgue measure over $\mathbb{R}^d$.

Except for the central symmetry assumption, the requirements of Assumption 3.2 ensure that Assumption 3.1 holds w.p. 1 for every sampled LDS. The proof can be found in Appendix C.2. The central symmetry of $\boldsymbol{A}$'s distribution, on the other hand, is a technical requirement for proving our main result.

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

$$\mathcal{L}(\theta) \;:=\; \mathbb{E}_{\boldsymbol{A},\boldsymbol{c},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{v_t\}_t}\left[\frac{1}{2}\left(\mathcal{T}_\theta(\boldsymbol{Y}_0)_{s+1,T-s} - y_T\right)^2\right], \tag{9}$$

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.3. We now describe the structure emerging within the first-order optimality condition.

Following (Ahn et al., 2023), we use basic algebraic manipulations (Appendix D.3) to rewrite loss (9) as

$$\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{v_t\}_t}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle \boldsymbol{Y}_0\boldsymbol{Y}_0^\top, \boldsymbol{b}\boldsymbol{a}_k^\top\rangle y_{T-s-1+k} - y_T\right)^2\right], \tag{10}$$

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

$$\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{v_t\}_t}\left[\frac{1}{T-s-1}\sum_{k=1}^{s}\langle \boldsymbol{Y}_0\boldsymbol{Y}_0^\top, \boldsymbol{b}\boldsymbol{a}_k^\top\rangle y_{T-s-1+k} y_{T-s-1+j} \boldsymbol{Y}_0\boldsymbol{Y}_0^\top\right]$$
$$= \mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\},\{v_t\}}\left[\, y_T y_{T-s-1+j} \boldsymbol{Y}_0\boldsymbol{Y}_0^\top \,\right],\quad \forall\, j \in [s] \tag{11}$$

are critical points of the loss.

Notably, the right-hand side of (11) obeys a banded structure, as follows

$$\begin{bmatrix} \star & 0 & \star & \cdots & \cdots & \\ 0 & \star & 0 & \star & & \vdots \\ \star & 0 & \star & \ddots & \ddots & \vdots \\ \vdots & \star & \ddots & \ddots & \ddots & \star \\ \vdots & & \ddots & \ddots & \star & 0 \\ & \cdots & \cdots & \star & 0 & \star \end{bmatrix} \text{ for odd } s+j; \quad \text{or} \quad \begin{bmatrix} 0 & \star & 0 & \cdots & \cdots & \\ \star & 0 & \star & 0 & & \vdots \\ 0 & \star & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 0 & \star \\ & \cdots & \cdots & 0 & \star & 0 \end{bmatrix} \text{ for even } s+j; \tag{12}$$

where $\star$ is a placeholder for arbitrary reals (the proof is deferred to Appendix D.3). We formalize a class of parameters ensuring matching structures between the left and right-hand sides of (12) 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* (11) *matching that of the right-hand side.*

$$\boldsymbol{W}_{QK} = \begin{bmatrix} \star & 0 & \star & \cdots & \\ 0 & \star & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \star \\ 0 & \cdots & 0 & \star & 0 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad \boldsymbol{W}_{V} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \vdots \\ \vdots & & & \vdots & 0 \\ \vdots & & & \vdots & \star \\ \vdots & & & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & \star \end{bmatrix}. \tag{13}$$

Lemma 4.1 can be understood as a narrowing-down based on structure of the parameter class likely to hold minimizers of (9).

Our second step is to use structure (13) to identify a global minimizer of loss (9) in the AR(1) case, yielding Theorem 4.1 with proof deferred to Appendix D.4.

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*

$$\boldsymbol{W}_{QK}^\star = \begin{bmatrix} \frac{(T-2)\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{\boldsymbol{v}_t\}_t}\left[\sum_{i=1}^{T-2} y_i y_{i+1} y_{T-1} y_T\right]}{\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{\boldsymbol{v}_t\}_t}\left[\sum_{i=1}^{T-2} y_i y_{i+1} \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2\right]} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{W}_V^\star = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{14}$$

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.3.

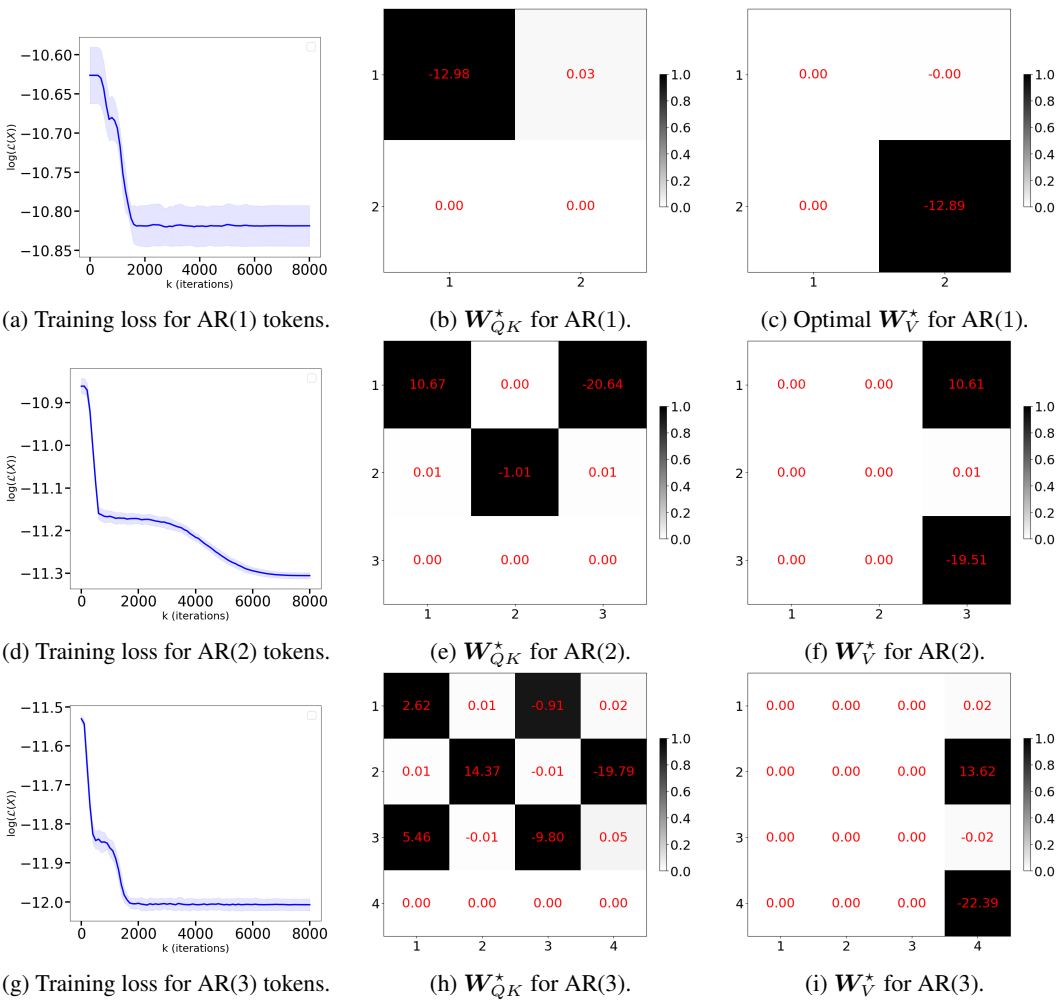
Notably, a forward pass using the optimal parameters (14) amounts to the prediction given after one GD step on $\mathcal{L}_{AR(1)}(w)$ starting from $w_0 = 0$. We thus recover the ICL-as-optimization view upheld by works in the i.i.d. setting (Ahn et al., 2023; Mahankali et al., 2023) but for LDS-produced 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



(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.

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 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 8000 for all cases with a increase of the batch size for every increase in order starting from 3000 for AR(1). A fresh batch of LDSs is sampled at every iteration (i.e., online setting). The experiments are done with the following 4 settings.

(a) For each sequence, sample $\boldsymbol{A}$'s diagonal entries uniformly at random in the interval $[-1, 1]$ and set $\boldsymbol{c} = \mathbf{1}_d$. The noise covariances are set to $\Sigma_w = \texttt{1e-2}\boldsymbol{I}$ and $\sigma_v^2 = \texttt{1e-2}$. The results are depicted in Figure 1 for AR(1–3) tokens, with experiments for AR(4) deferred to Figure 2 in Appendix B.

(b) For each sequence, sample $\boldsymbol{v} \sim \mathrm{Unif}([-1, 1]^d)$, independently sample $\boldsymbol{Q} \sim \mathrm{Haar}(O(d))$ and set $\boldsymbol{A} = \mathbf{Q}^\top \mathrm{diag}(\boldsymbol{v})\mathbf{Q}$. We also independently sample $\boldsymbol{c} \sim \mathrm{Unif}([-5, 5]^d)$. The noise covariances are set to $\Sigma_w = \texttt{1e-2}\boldsymbol{I}$ and $\sigma_v^2 = \texttt{1e-2}$. Experiments for AR(1–4)- tokens are provided in Figure 4 in Appendix B.4.

(c) For each sequence, sample $\boldsymbol{v} \sim \mathrm{Unif}([-1, 1]^d)$, sample $\mathbf{P} = [p_{i,j}]_{i,j=1}^d$ by sampling $p_{i,j}$ i.i.d. from $\mathcal{U}([-1, 1])$, and set $\boldsymbol{A} = \mathbf{P}^{-1}\mathrm{diag}(\boldsymbol{v})\mathbf{P}$. Sample $\boldsymbol{c} \sim \mathrm{Unif}([-5, 5]^d)$. The noise covariances are set to $\Sigma_w = \texttt{1e-2}\boldsymbol{I}$ and $\sigma_v^2 = \texttt{1e-2}$. Experiments for AR(1–4)- tokens are provided in Figure 6 in Appendix B.6.

(d) For each sequence, sample $\boldsymbol{v} \sim \mathrm{Unif}([-1, 1]^d)$, sample $\boldsymbol{Q} \sim \mathrm{Haar}(O(d))$ and set $\boldsymbol{A} = \mathbf{Q}^\top \mathrm{diag}(v)\mathbf{Q}$. Sample $\boldsymbol{c} \sim \mathrm{Unif}([-5, 5]^d)$. Fix the process noise covariance to $\Sigma_w = \mathbf{Q_w}^\top \mathrm{diag}(\texttt{1e-2} \cdot [0.8, 0.85, 0.9, 0.95, 1.0])\mathbf{Q_w}$, where $\mathbf{Q_w}$ is an orthogonal matrix. Set $\sigma_v^2 = \texttt{1e-2}$. Experiments AR(1–4)- tokens are provided in Figure 5 in Appendix B.5.

All the settings above have $\boldsymbol{x}_0 \sim \mathcal{N}(0, \sigma_0^2\boldsymbol{I})$, $\sigma_0^2 = \texttt{1e-2}$. Note that we could have used any other centrally symmetric distribution with marginals supported on $[-1, 1]$ for the sampling of the diagonal $\boldsymbol{v}$, e.g., $\mathrm{Unif}(\mathbb{S}^{d-1})$ — uniform on the unit sphere; $\mathrm{Unif}(\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\})$ — uniform inside the unit ball, etc. We prove these sampling schemes obey Assumption 3.2 in Appendix E.1. 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.

Figure 1 depicts the experiment results under setting (a), Figure 4 setting (b), Figure 5 setting (c) and Figure 6 setting (d). Figure 1 (b,c), Figure 4 (b,c), Figure 5 (b,c) and Figure 6 (b,c) show an optimum conforming to Theorem 4.1 for AR(1) tokens. Moreover, Figure 1 (e,f,h,i), Figure 2 (b,c), Figure 4 (e,f,h,i,k,l) and Figure 6 (e,f,h,i) confirm experimentally the pattern uncovered by Lemma 4.1 for general $s > 0$. Furthermore, we provide experiments in setting (a) showing that the weights converge to the sparsity pattern predicted by Lemma 4.1 in terms of the Jaccard distance between the non-zero supports — experimental details are given in Appendix B.3 and results are depicted in Figure 3 of the appendix.

Interpreting of the sparsity pattern for AR(s) $s \geq 2$. 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 standard GD in the forward pass, but an iteration resembling that of the Preconditioned Conjugate Gradient method (PCG) (Shewchuk et al., 1994), as follows.

Since our sampling scheme ensures $\rho(\boldsymbol{A}) < 1$ w.p. one, the stochastic process $\{\boldsymbol{y}_t\}_t$ approaches stationarity exponentially fast, meaning that autocorrelations become (almost) solely dependent on lag, i.e., $\mathbb{E}[\boldsymbol{y}_t \boldsymbol{y}_{t+k}] \approx \gamma(k)$, $\forall t \in \mathbb{N}_+$. In particular, the empirical counterparts become approximately equal $\frac{1}{T-s-1}\sum_{i=1}^{T-s-1} \boldsymbol{y}_i \boldsymbol{y}_{i+k} \approx \frac{1}{T-s-1}\sum_{i=1}^{T-s-1} \boldsymbol{y}_{i+p} \boldsymbol{y}_{i+p+k} \approx \hat{\gamma}(k)$. We can therefore approximate $\frac{1}{T-s-1}\mathbf{Y}_0\mathbf{Y}_0^\top$ with the symmetric Toeplitz matrix and remark it has a block structure involving $\nabla^2 \mathcal{L}_{AR(s)}$ (a constant matrix) and $\nabla \mathcal{L}_{AR(s)}(\mathbf{0})$ (the gradient at $\boldsymbol{w} = \mathbf{0}$)

$$\frac{1}{T-s-1}\mathbf{Y}_0\mathbf{Y}_0^\top \approx \begin{bmatrix} \hat{\gamma}(0) & \hat{\gamma}(1) & \cdots & \hat{\gamma}(s) \\ \hat{\gamma}(1) & \hat{\gamma}(0) & \cdots & \hat{\gamma}(s-1) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\gamma}(s) & \hat{\gamma}(s-1) & \cdots & \hat{\gamma}(0) \end{bmatrix} = \begin{bmatrix} \nabla^2 \mathcal{L}_{AR(s)} & \nabla \mathcal{L}_{AR(s)}(0) \\ \nabla \mathcal{L}_{AR(s)}(0)^\top & \hat{\gamma}(0) \end{bmatrix}. \quad (15)$$

Using expression (15) and the parameter structure from Lemma 4.1 and the experiments, we rewrite the transformer's forward pass in a manner that highlights the resemblance with two steps of the PCG method. We describe the case for even s, with identical reasoning applying for the odd case. Let

$s = 2k,\ k \in \mathbb{N}$ and $N = \frac{(s+1)^2+1}{2}$. Then, the weight matrices belonging to $\mathbb{R}^{s+1\times s+1}$ are

$$\mathbf{W_{QK}} = \left[\begin{array}{cccc|c} c_1 & 0 & c_2 & \cdots & c_{k+1} \\ 0 & c_{k+2} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & c_{N-2k} & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 \end{array}\right] \quad \mathbf{W_V} = \left[\begin{array}{ccc|c} 0 & \cdots & 0 & c_{N-k} \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & c_{N-k+1} \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ \hline 0 & \cdots & 0 & c_N \end{array}\right]. \tag{16}$$

Renaming the top left $s \times s$ block of $\mathbf{W_{QK}}$ as $\mathbf{P}$, the top-right $s \times 1$ block as $\mathbf{p}$, and the top right $s \times 1$ block of $\mathbf{W_V}$ as $\mathbf{q}$, the transformer-induced linear predictor $\frac{1}{T-s-1}\mathbf{W_{QK}}\mathbf{Y_0}\mathbf{Y_0}^\top\mathbf{W_V}_{:,s+1}$ is

$$\mathbf{P}\nabla^2\mathcal{L}_{AR(s)}\mathbf{q} + (\mathbf{p}\mathbf{q}^\top + c_N\mathbf{P})\nabla\mathcal{L}_{AR(s)}(0) + c_N\hat{\gamma}_0\mathbf{p}$$

Letting $\boldsymbol{P'} \coloneqq \Gamma\nabla^2\mathcal{L}_{AR(s)}$ with $\Gamma \coloneqq \frac{c_N\hat{\gamma}_0\mathbf{p}\mathbf{q}^\top}{\mathbf{q}^\top\nabla^2\mathcal{L}^2_{AR(s)}\mathbf{q}}$ and observing that $c_N\hat{\gamma}_0\mathbf{p} = \mathbf{P'}\nabla^2\mathcal{L}_{AR(s)}\mathbf{q}$ (see Appendix E.3), the transformer-induced predictor finally rewrites as

$$\left(\Gamma\nabla^2\mathcal{L}_{AR(s)} + \mathbf{P}\right)\ \nabla^2\mathcal{L}_{AR(s)}\mathbf{q}\ +\ (\mathbf{p}\mathbf{q}^\top + c_N\mathbf{P})\,\nabla\mathcal{L}_{AR(s)}(0). \tag{17}$$

Expression (17) resembles the predictor obtained after two PCG steps (Shewchuk et al., 1994, p. 51) on loss $\mathcal{L}_{AR(s)}$ with preconditioner $\mathbf{P}^{-1}$ starting from $\mathbf{w}_0 = \mathbf{0}$ and initial conjugate direction $\mathbf{d}_0 = \mathbf{q}$ (algorithm deferred to Appendix E.2). Note that $\boldsymbol{P}$'s invertibility is assumed. The resulting predictor is

$$\begin{aligned}\mathbf{w}_2 &= \left[\tau_1\nabla^2\mathcal{L}^{-1}_{AR(s)} - \tau_2\boldsymbol{P}\right]\nabla^2\mathcal{L}_{AR(s)}\boldsymbol{q} + \tau_3\boldsymbol{P}\nabla\mathcal{L}_{AR(s)}(0) \\ &\approx \left[2\tau_1\boldsymbol{I} - \tau_1\nabla^2\mathcal{L}_{AR(s)} - \tau_2\boldsymbol{P}\right]\nabla^2\mathcal{L}_{AR(s)}\boldsymbol{q} + \tau_3\boldsymbol{P}\nabla\mathcal{L}_{AR(s)}(0)\end{aligned}$$

where $\tau_1, \tau_2, \tau_3 \in \mathbb{R}$ are iteration-dependent constants (see Appendix E.2), and we used an order-one Neumann series approximation of the Hessian inverse. The latter was shown to exist with high probability for sufficiently large T by Tsiamis & Pappas (2019). Notably, this AR(s) analogy is in harmony with the plain GD step observed for AR(1), since PCG collapses to GD when covariates belong to $\mathbb{R}$.

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, 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 confirmed through experiments. We also reveal that for a length-s window, the trained attention layer approximates a step of PCG on the corresponding autoregressive loss.

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 of Lemma 4.1; explaining the non-standard initialization of the conjugate direction in the AR(s), $s \geq 2$ PCG approximation; 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.

Peter Beim Graben, Bryan Jurish, Douglas Saddy, and Stefan Frisch. Language processing by dynamical systems. *International Journal of Bifurcation and Chaos*, 14(02):599–621, 2004.

David Belanger and Sham Kakade. A linear dynamical system model for text. In *International Conference on Machine Learning*, pp. 833–842. PMLR, 2015.

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.

Jeffrey L Elman. Language as a dynamical system. 1995.

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.

Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

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.

H. L. Royden and P. M. Fitzpatrick. *Real Analysis*. Prentice Hall, 4th edition, 2010. ISBN 978-0131437470.

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.

Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain. 1994.

Whitney Tabor, Christopher Juliano, and Michael Tenenhaus. A dynamical system for language processing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 18, 1996. URL https://escholarship.org/uc/item/78r6h0cg.

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.

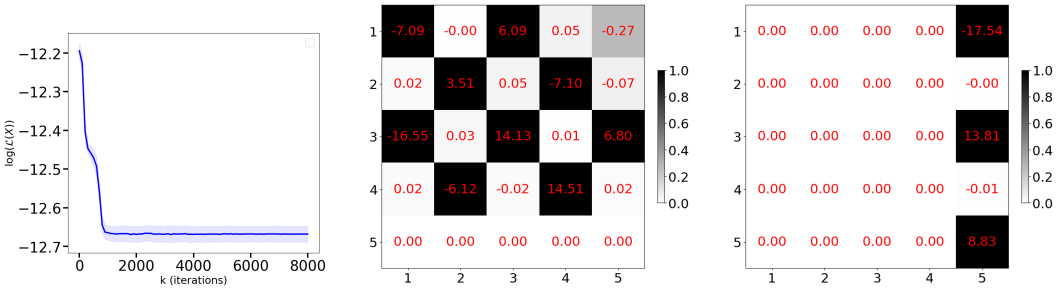
Table 1: Training hyperparameters of settings (a), (b) and (d) in Section 5

Hyperparameter	**Value**
Weight initialization	Xavier normal distribution (Glorot & Bengio, 2010) with gain $=$ `1e-5`
Optimizer	AdamW (Loshchilov & Hutter, 2017) with $\beta_1 = 0.98$ for AR(1), 0.92 for AR(2), 0.10 for AR(3), 0.76 for AR(4), $\beta_2 = 0.99$ for AR(1), 0.96 for AR(2), 0.55 for AR(3), 0.88 for AR(4), $\epsilon =$ `1e-9`
Weight decay	`5e-3` for AR(2), AR(4) and `1e-2` for AR(1), AR(3)
Learning rate (i.e., max. val.)	`2e-2` for AR(1), `3e-2` for AR(2), `9e-2` for AR(3), `9e-2` for AR(4)
Min. learning rate	`1e-4`
Linear warmup	800 iter.
Decay schedule	Cosine annealing (Loshchilov & Hutter, 2016)
Max. decay steps	7200 iter.
Max. grad norm (clipping)	300
Random seeds	$\{666013, 1, 0\}$
Batch size / iter.	3000 for AR(1), 4000 for AR(2), 8000 for AR(3), 16000 for AR(4)
Total iter.	8001

Table 2: Training hyperparameters of setting (c) in Section 5

Hyperparameter	**Value**
Weight initialization	Xavier normal distribution (Glorot & Bengio, 2010) with gain $=$ `1e-5`
Optimizer	AdamW (Loshchilov & Hutter, 2017) with $\beta_1 = 0.98$ for AR(1), 0.92 for AR(2), 0.92 for AR(3), $\beta_2 = 0.99$ for AR(1), 0.96 for AR(2), 0.96 for AR(3), $\epsilon =$ `1e-9`
Weight decay	`5e-3` for AR(2) and `1e-2` for AR(1), AR(3)
Learning rate (i.e., max. val.)	`3e-3` for AR(1), `5e-3` for AR(2), `7e-3` for AR(3)
Min. learning rate	`1e-5`
Linear warmup	800 iter.
Decay schedule	Cosine annealing (Loshchilov & Hutter, 2016)
Max. decay steps	7200 iter.
Max. grad norm (clipping)	300
Random seeds	$\{666013, 1, 0\}$
Batch size / iter.	3000 for AR(1), 4000 for AR(2), 8000 for AR(3)
Total iter.	8001

B.2 Experiments for larger window sizes



(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.

B.3 is NEW ADDED

B.3 Experiments showing convergence to the checkerboard pattern during training

This set of experiments serves to illustrate that parameters $\boldsymbol{W}_{QK}$ and $\boldsymbol{W}_{V}$ converge to the checkerboard pattern across iterations. Since the non-zero values of these parameters are of different magnitudes and we do not have their theoretical expressions for window-sizes greater than 1, we shall only consider their non-zero support, as follows.

Definition B.1. *For a matrix $\boldsymbol{M} \in \mathbb{R}^{d\times m}$, its support is defined as the collection of positions corresponding to non-zero values*

$$\operatorname{supp}(\boldsymbol{M}) \;\coloneqq\; \{(i,j) \in [d] \times [m] \mid a_{i,j} \neq 0\}. \tag{18}$$

Additionally, the support-induced mask is a binary matrix with unit entries on the support

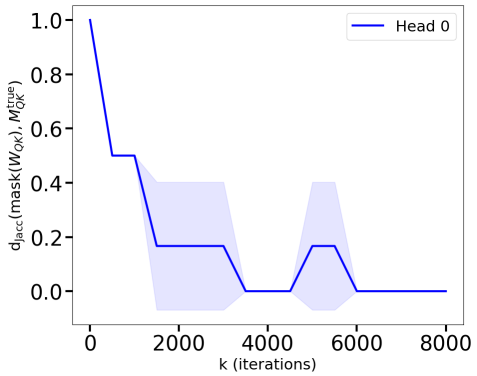
$$\operatorname{mask}(\boldsymbol{M}) \;\coloneqq\; \left[\mathbf{1}_{(i,j)\in\operatorname{supp}(\boldsymbol{M})}\right]_{i,j=1}^{i=d,j=m} \tag{19}$$

where $\mathbf{1}_C = 1$ if condition C is true and 0 otherwise, is the indicator function centered at z.

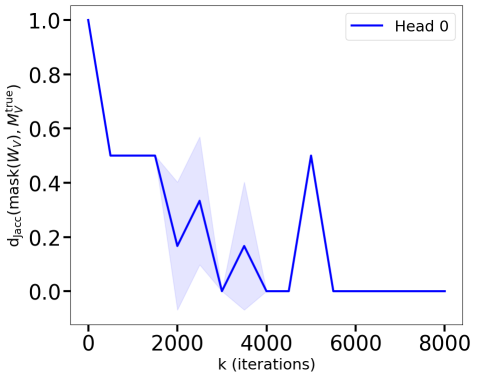
We rely on the Jaccard distance (Jaccard, 1901) adapted to binary matrices $\boldsymbol{A}$, $\boldsymbol{B}$

$$\mathrm{d}_{\mathrm{Jac}}(\boldsymbol{A},\boldsymbol{B}) \;\coloneqq\; 1 - \frac{\sum_{i,j} a_{i,j} b_{i,j}}{\sum_{i,j} \max\{a_{i,j}, b_{i,j}\}} \tag{20}$$

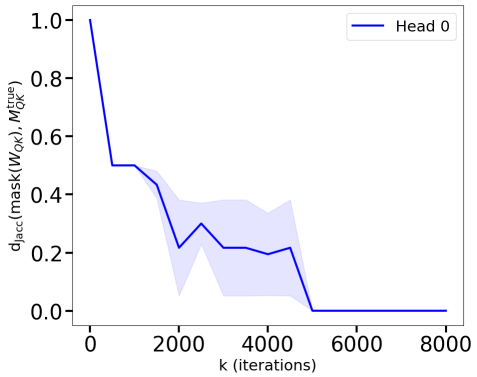
to track whether the support-induced masks of our parameters during training converge to the predicted (for AR(1)) or hypothesized (for AR(s) $s \geq 2$) sparsity patterns of Lemma 4.1. Our experiments employ a tolerance level of `1e-1` when computing the masks of $\boldsymbol{W}_{V}$ and $\boldsymbol{W}_{QK}$, meaning that any entry below this value is considered zero. The results are depicted in Figure 3 and its subplots for varying window sizes, where $\boldsymbol{M}_{QK}^{\text{true}}$ and $\boldsymbol{M}_{V}^{\text{true}}$ represent the masks posited in Lemma 4.1 for a null tolerance level. The illustrations empirically confirm that our parameters' supports converge to the ones identified in Lemma 4.1.



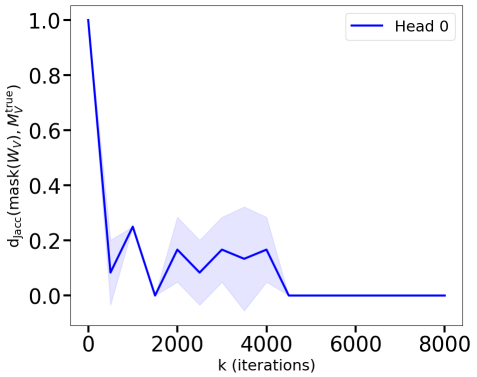
(a) The Jaccard distance of $\boldsymbol{W}_{QK}$ for AR(1).



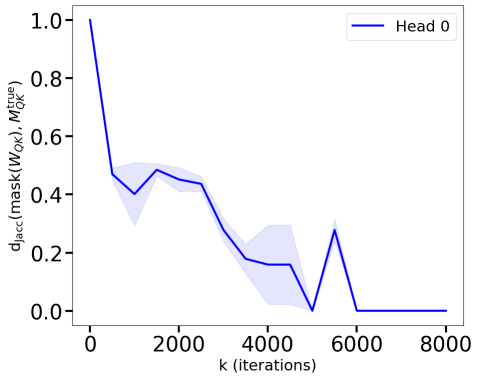
(b) The Jaccard distance of $\boldsymbol{W}_{V}$ for AR(1).



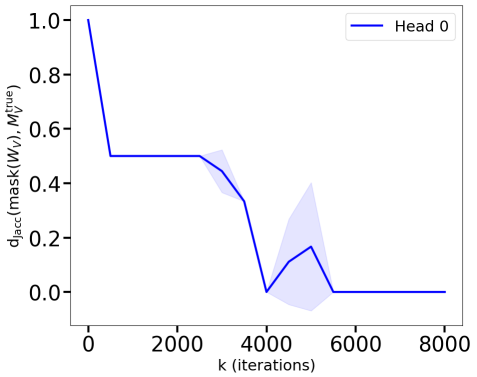
(c) The Jaccard distance of $\boldsymbol{W}_{QK}$ for AR(2).



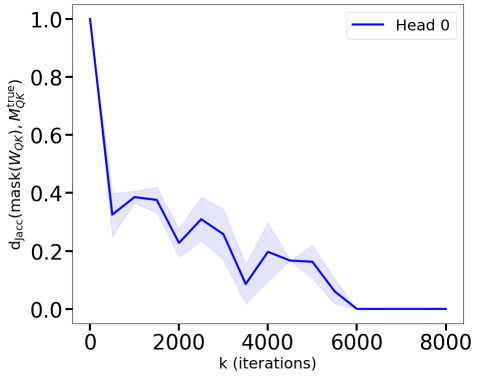
(d) The Jaccard distance of $\boldsymbol{W}_{V}$ for AR(2).



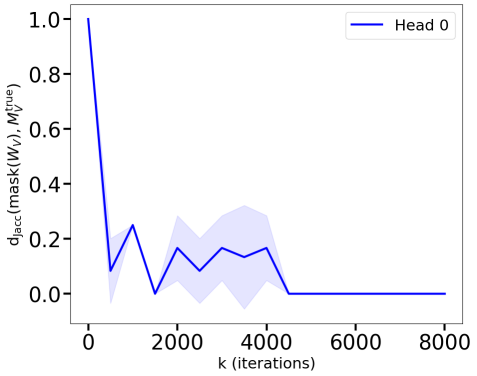
(e) The Jaccard distance of $\boldsymbol{W}_{QK}$ for AR(3).



(f) The Jaccard distance of $\boldsymbol{W}_{V}$ for AR(3).



(g) The Jaccard distance of $\boldsymbol{W}_{QK}$ for AR(4).



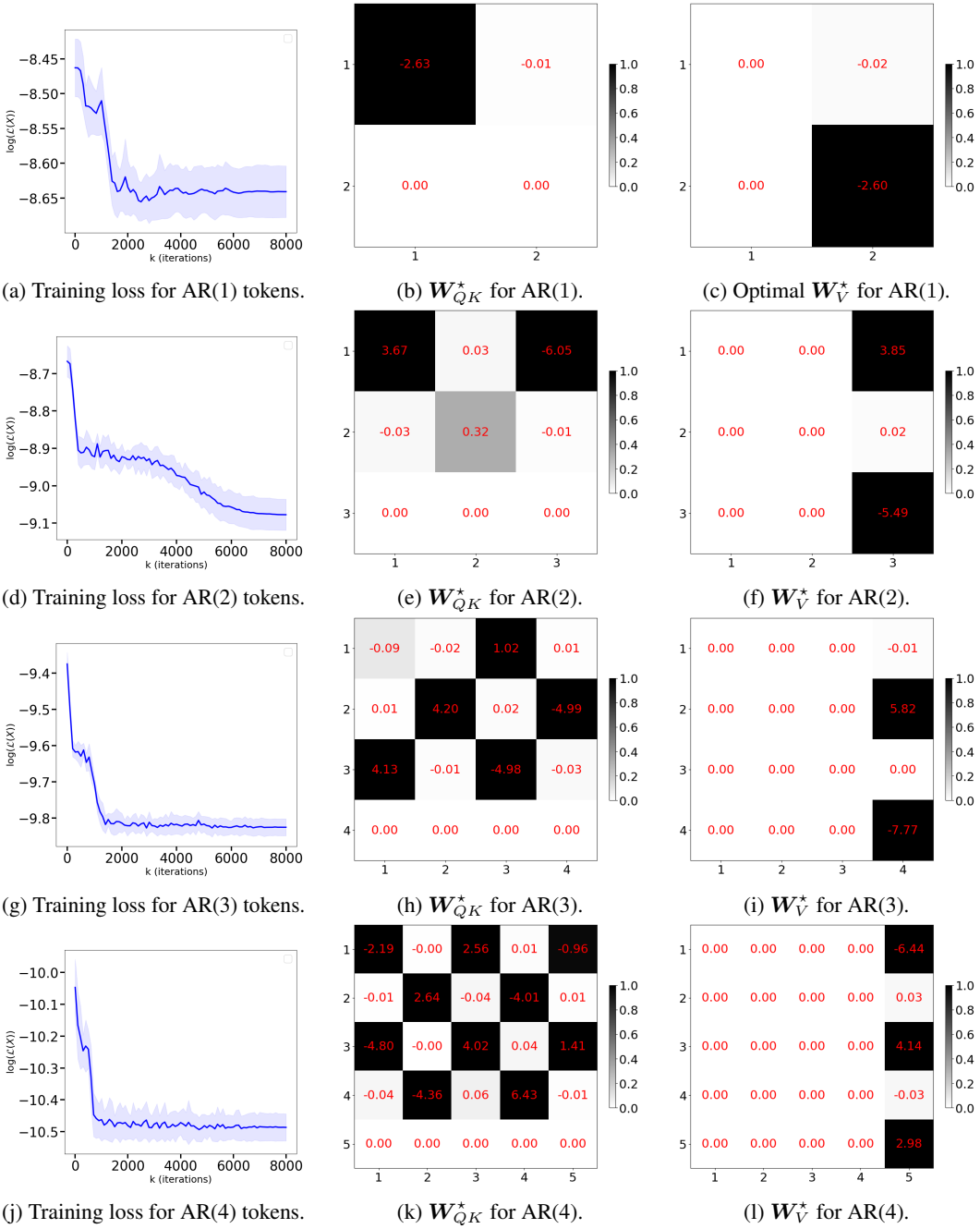
(h) The Jaccard distance of $\boldsymbol{W}_{V}$ for AR(4).

Figure 3: The experiment results of the Jaccard distance between the $\boldsymbol{M}_{QK}^{\text{true}}$ and $\boldsymbol{W}_{QK}$ and the Jaccard distance between the $\boldsymbol{M}_{V}^{\text{true}}$ and $\boldsymbol{W}_{V}$ for AR(1–4). Both converge to 0 at the end of the training.

B.4 is NEW ADDED

B.4 EXPERIMENTS WITH NON-DIAGONAL, SYMMETRIC $\boldsymbol{A}$, RANDOM $\boldsymbol{c}$ AND ISOTROPIC Σ_w

The LDS which generates the training data is as follows. For each sequence, sample $\boldsymbol{d} \sim \mathrm{Unif}([-1,1]^d)$, sample $\boldsymbol{Q} \sim \mathrm{Haar}(O(d))$ and set $\boldsymbol{A} = \mathbf{Q}^\top \mathrm{diag}(\boldsymbol{d})\mathbf{Q}$. Sample $\boldsymbol{c} \sim \mathrm{Unif}([-5,5]^d)$. The noise covariances are set to $\Sigma_w = \mathtt{1e{-}2}\boldsymbol{I}$ and $\sigma_v^2 = \mathtt{1e{-}2}$.



(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).

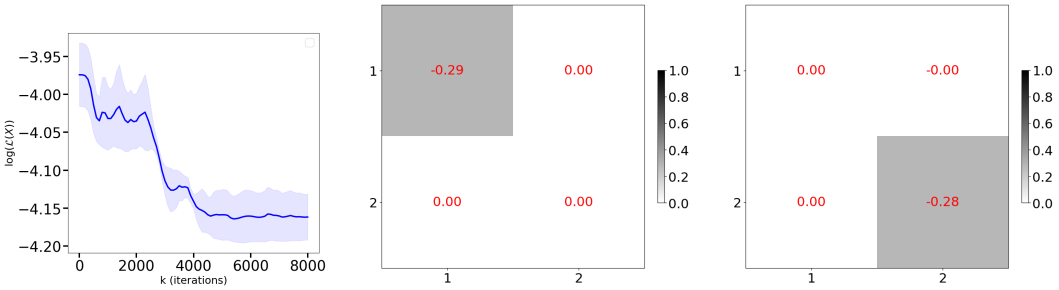
(j) Training loss for AR(4) tokens. (k) $\boldsymbol{W}_{QK}^\star$ for AR(4). (l) $\boldsymbol{W}_V^\star$ for AR(4).

Figure 4: Experimental results for AR(1–4) with non-diagonal, symmetric $\boldsymbol{A}$, random $\boldsymbol{c}$ and isotropic Σ_w, which align with the Lemma 4.1.

B.5 is NEW ADDED

B.5 EXPERIMENTS WITH NON-DIAGONAL, NON-SYMMETRIC $\boldsymbol{A}$, RANDOM $\boldsymbol{c}$ AND NON-DIAGONAL Σ_w

The LDS which generates the training data is as follows. For each sequence, sample $\boldsymbol{d} \sim \mathrm{Unif}([-1,1]^d)$; sample $\boldsymbol{Q} \sim \mathrm{Haar}(O(d))$ and set $\boldsymbol{A} = \mathbf{Q}^\top \mathrm{diag}(\boldsymbol{d})\mathbf{Q}$; sample $\boldsymbol{c} \sim \mathrm{Unif}([-5,5]^d)$. Set the process noise covariance $\Sigma_w = \mathbf{Q_w}^\top \mathrm{diag}(\texttt{1e-2} \cdot [0.8, 0.85, 0.9, 0.95, 1.0])\mathbf{Q_w}$, where $\mathbf{Q_w}$ is an orthogonal matrix. Set $\sigma_v^2 = \texttt{1e-2}$.



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

Figure 5: Experimental results for AR(1) with non-diagonal, non-symmetric $\boldsymbol{A}$, random $\boldsymbol{c}$ and non-diagonal Σ_w, which align with the Lemma 4.1.

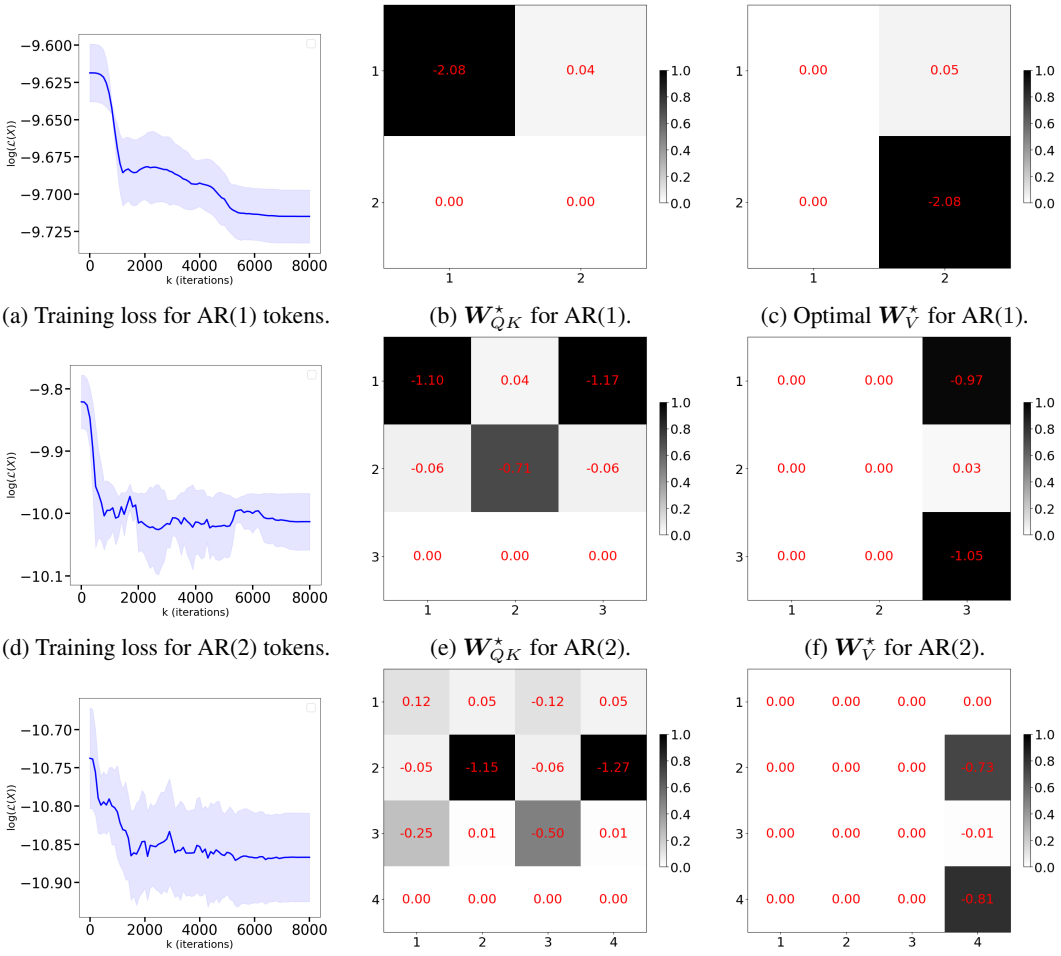
B.6 is NEW ADDED

B.6 EXPERIMENTS WITH NON-DIAGONAL, NON-SYMMETRIC $\boldsymbol{A}$, RANDOM $\boldsymbol{c}$ AND ISOTROPIC Σ_w

The LDS which generates the training data is as follows.

For each sequence, sample $\boldsymbol{d} \sim \mathrm{Unif}([-1,1]^d)$, sample $\mathbf{P} = [p_{i,j}]_{i,j=1}^d$ by sampling $p_{i,j}$ i.i.d. from $\mathcal{U}([-1,1])$, and set $\boldsymbol{A} = \mathbf{P}^{-1}\mathrm{diag}(\boldsymbol{d})\mathbf{P}$. Sample $\boldsymbol{c} \sim \mathrm{Unif}([-5,5]^d)$. The noise covariances are set to $\Sigma_w = \texttt{1e-2}\boldsymbol{I}$ and $\sigma_v^2 = \texttt{1e-2}$.

In practice, we need to guarantee $\boldsymbol{P}$ is well conditioned. After sampling $p_{i,j}$ i.i.d. from $\mathcal{U}([-1,1])$, we decompose $\boldsymbol{P} = \boldsymbol{QR}$, where $\boldsymbol{P}$ is an orthogonal matrix and $\boldsymbol{R}$ is an upper-triangle matrix. We modify the diagonals of $\boldsymbol{R}$ manually to make sure $\frac{\max_i \boldsymbol{R}_{ii}}{\min_i \boldsymbol{R}_{ii}} = 2$ and right multiply $\boldsymbol{Q}$ with the modified $\boldsymbol{R}$ to have the well conditioned $\boldsymbol{P}$.



(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 6: Experimental results for AR(1–3) with non-diagonal, non-symmetric $\boldsymbol{A}$, random $\boldsymbol{c}$ and isotropic Σ_w, which align with the Lemma 4.1.

C SECTION 3 PROOFS

C.1 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

$$\boldsymbol{R} = \begin{bmatrix} 0 & \mathbf{0}_{n-1}^\top \\ \mathbf{0}_{n-1} & \boldsymbol{I}_{n-1} \end{bmatrix}. \tag{21}$$

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:

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

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

$$\boldsymbol{W}_V^h = \begin{matrix} 1 & \ldots & h & \ldots & s+1 \end{matrix} \\ \begin{bmatrix} \mathbf{0}_{T+1} & \ldots & \boldsymbol{e}_1 & \ldots & \mathbf{0}_{T+1} \end{bmatrix},\ \forall h \in [s+1] \tag{22}$$

Each head then computes the following

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

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

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

NEW ADDED

C.2 Proof of the Almost Sure Observability of the LDS

We seek to show that Assumption 3.2 ensures LDS (1) observability w.p. 1. Note that the central symmetry of the distribution is irrelevant for this statement, and only relevant for the proofs in Section 4. We repeat Assumption 3.2 below for convenience.

Assumption 3.2 (LDS family)**.** *The system matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is sampled from a centrally symmetric distribution supported on $\left\{\boldsymbol{M} \in \mathbb{R}^{d\times d} \;\middle|\; \rho(\boldsymbol{M}) \leq 1\right\}$, for which it holds that*

$$\mathbb{P}\left(\{\boldsymbol{A} \mid \exists i, j \in [d],\ \textit{s.t. } \lambda_i(\boldsymbol{A}) = \lambda_j(\boldsymbol{A})\}\right) = 0. \tag{8}$$

In other words, $\boldsymbol{A}$ has a simple spectrum almost surely. The observation vector $\boldsymbol{c} \in \mathbb{R}^d$ is sampled independently, from a distribution that is absolutely continuous w.r.t. the Lebesgue measure over $\mathbb{R}^d$.

Lemma C.1. *Assumption 3.2 ensures the pair $(\boldsymbol{A}, \boldsymbol{c})$ is observable w.p. 1.*

Proof. Since $\boldsymbol{A}$ has distinct eigenvalues w.p. 1 (the simple spectrum condition), it is (block) diagonalizable almost surely, and its eigenvectors $\{\boldsymbol{v}_1, \dots \boldsymbol{v}_d\}$ are linearly independent. Therefore, observability is ensured if $\boldsymbol{c}^\top \boldsymbol{v}_i \neq 0$ almost surely for all $i \in [d]$.

Since $\boldsymbol{c}$ is sampled from a distribution that is absolutely continuous w.r.t. the Lebesgue measure in $\mathbb{R}^d$, we want to prove that the set

$$\mathcal{U} = \bigcup_{i=1}^{d} \left\{\boldsymbol{c} \in \boldsymbol{R}^d \;\middle|\; \boldsymbol{c}^\top \boldsymbol{v}_i = 0\right\}$$

has zero Lebesgue measure in the ambient $\mathbb{R}^d$. Each collection $\left\{\boldsymbol{c} \in \boldsymbol{R}^d \;\middle|\; \boldsymbol{c}^\top \boldsymbol{v}_i = 0\right\}$ forms a proper subspace of $\mathbb{R}^d$ with dimension at most $d-1$ (it can be less, for complex $\boldsymbol{v}_i$). Therefore, its Lebesgue measure is null (see, e.g., (Royden & Fitzpatrick, 2010, pg. 435)).

Since $\mathcal{U}$ is a finite union of measure zero sets, it is itself measure zero. Hence, observability holds w.p. 1. □

D SECTION 4 PROOFS

D.1 PRELIMINARIES

Since we're dealing with data generated from stochastic processes, our proofs will heavily rely on taking expectations conditioned on randomness up to a certain point in the process. In what follows, we formalize the natural filtrations with respect to process (1).

We denote the natural filtration associated with (1). as $\{\mathcal{F}_t\}_{t\geq 0}$, where

$$\mathcal{F}_t \;\coloneqq\; \sigma(\boldsymbol{A},\; \boldsymbol{c},\; \boldsymbol{x}_0,\; \boldsymbol{w}_0, \ldots, \boldsymbol{w}_{t-1},\; v_0, \ldots, v_{t-1})\,, \qquad t \geq 0. \tag{23}$$

By convention, when $t = 0$ the sets of noise variables are empty, and we define

$$\mathcal{F}_0 = \sigma(\boldsymbol{A},\, \boldsymbol{c},\, \boldsymbol{x}_0), \tag{24}$$

to illustrate that $\boldsymbol{A}$ and $\boldsymbol{c}$ are sampled once at time 0 and then remain fixed.

It follows that

(a) $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}, \forall t \geq 0$

(b) $\boldsymbol{x}_t$ is $\mathcal{F}_t$-measurable for all $t \geq 0$.

(c) y_t is $\mathcal{F}_{t+1}$-measurable (since y_t depends on v_t)

(d) The noise at time t is independent on the respective filtration: $\boldsymbol{w}_t \perp\!\!\!\perp \mathcal{F}_t,\; v_t \perp\!\!\!\perp \mathcal{F}_t$, for all $t \geq 0$.

D.2 AUXILIARY RESULTS AND TECHNICAL LEMMATA

Theorem D.1 (Isserlis (1918))**.** *Let $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^\top \sim \mathcal{N}_n(0, \Sigma)$ be an n-dimensional, mean-zero multivariate normal vector. Then, for any even integer n,*

$$\mathbb{E}[y_1 y_2 \cdots y_n] = \sum_{p \in \mathrm{PP}(n)} \prod_{(\ell,r)\in p} \mathbb{E}[\, y_\ell y_r\,],$$

where $\mathrm{PP}(n)$ denotes the set of all pairwise partitions of $[n]$ into disjoint pairs. If n is odd, then $\mathbb{E}[y_1 y_2 \cdots y_n] = 0$.

Lemma D.1. *Given random vectors $\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{q} \in \mathbb{R}^d$ and assuming that $\boldsymbol{w}$ is independent of $\boldsymbol{z}, \boldsymbol{q}$ and the relevant integrability conditions hold, then*

$$\mathbb{E}\left[\boldsymbol{z}^\top \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{q}\right] = \mathbb{E}\left[\boldsymbol{z}^\top \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top]\boldsymbol{q}\right] \tag{25}$$

Proof. We use the towering property of expectations,

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{z}^\top \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{z}\right] &= \mathbb{E}\left[\boldsymbol{z}^\top \mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^\top \middle| \boldsymbol{z}, \boldsymbol{q}\right] \boldsymbol{q}\right] \\ &= \mathbb{E}\left[\boldsymbol{z}^\top \mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^\top\right] \boldsymbol{q}\right],\end{aligned}$$

where the last line follows from the quantities' independence. □

Lemma D.2. *Let the sequence $\{y_i\}_{i\geq 0}$ be generated by an LDS* (1) *sampled according to Assumption 3.2. For time indices $0 \leq i \leq j$, it holds that*

$$\mathbb{E}\left[y_i y_j\right] = \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\right] + \sum_{k=0}^{i-1} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^{i-1-k} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-k} \boldsymbol{c}\right] + \mathbf{1}_{\{i=j\}} \sigma_v^2, \tag{26}$$

where $\mathbf{1}_{\{i=j\}}$ takes the value 1 if $i = j$ and 0 otherwise.

Proof. For process (1) it holds that

$$\boldsymbol{x}_j = \boldsymbol{A}^{j-i}\boldsymbol{x}_i + \sum_{k=i}^{j-1} \boldsymbol{A}^{j-1-k}\boldsymbol{w}_k$$

and therefore

$$y_j = \boldsymbol{c}^\top \boldsymbol{A}^{j-i}\boldsymbol{x}_i + \sum_{k=i}^{j-1} \boldsymbol{c}^\top \boldsymbol{A}^{j-1-k}\boldsymbol{w}_k + v_j.$$

The product of scalars $y_i y_j$ therefore takes the form

$$\begin{aligned} y_i y_j &= y_i y_j^\top \\ &= (\boldsymbol{c}^\top \boldsymbol{x}_i + v_i)\left(\boldsymbol{c}^\top \boldsymbol{A}^{j-i}\boldsymbol{x}_i + \sum_{k=i}^{j-1} \boldsymbol{c}^\top \boldsymbol{A}^{j-1-k}\boldsymbol{w}_k + v_j\right)^\top \\ &= \boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c} + \sum_{k=i}^{j-1} \boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{x}_i v_j \\ &\qquad + v_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c} + \sum_{k=i}^{j-1} v_i \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} + v_i v_j. \end{aligned} \tag{27}$$

Now, observing that $\mathbb{E}[y_i y_j] = \mathbb{E}\left[\mathbb{E}\left[y_i y_j \,|\, \mathcal{F}_i\right]\right]$ and remembering that $\boldsymbol{x}_i, \boldsymbol{A}, \boldsymbol{c}$ are $\mathcal{F}_i$-measurable, and that for all i and $p \geq i$, $\boldsymbol{w}_p \perp\!\!\!\perp \mathcal{F}_i$ and $v_p \perp\!\!\!\perp \mathcal{F}_i$, and $\boldsymbol{w}_p \perp\!\!\!\perp v_q, \forall p, q \geq 0$, we have

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c} \,\middle|\, \mathcal{F}_i\right] &= \boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c}, \\ \mathbb{E}\left[\sum_{k=i}^{j-1} \boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} \,\middle|\, \mathcal{F}_i\right] &= \sum_{k=i}^{j-1} \boldsymbol{c}^\top \boldsymbol{x}_i \mathbb{E}\left[\boldsymbol{w}_k^\top\right] (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} = 0, \\ \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{x}_i v_j \,\middle|\, \mathcal{F}_i\right] &= \boldsymbol{c}^\top \boldsymbol{x}_i \mathbb{E}[v_j] = 0, \\ \mathbb{E}\left[v_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c} \,\middle|\, \mathcal{F}_i\right] &= \mathbb{E}[v_i]\, \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c} = 0, \\ \mathbb{E}\left[\sum_{k=i}^{j-1} v_i \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} \,\middle|\, \mathcal{F}_i\right] &= \sum_{k=i}^{j-1} \mathbb{E}[v_i]\mathbb{E}[\boldsymbol{w}_k^\top](\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} = 0, \\ \mathbb{E}\left[v_i v_j \,|\, \mathcal{F}_i\right] &= \mathbb{E}[v_i v_j] = \mathbf{1}_{\{i=j\}}\sigma_v^2. \end{aligned}$$

Therefore,

$$\mathbb{E}[y_i y_j] = \mathbb{E}[\,\mathbb{E}[y_i y_j \,|\, \mathcal{F}_i]] = \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c}\right] + \mathbf{1}_{\{i=j\}}\sigma_v^2. \tag{28}$$

Noting that $\boldsymbol{x}_i = \boldsymbol{A}^i \boldsymbol{x}_0 + \sum_{k=0}^{i-1} \boldsymbol{A}^{i-1-k}\boldsymbol{w}_k$, we further unroll the first term inside the expectation in (28) and get

$$\begin{aligned} \boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i}\boldsymbol{c} &= \left[\boldsymbol{c}^\top \boldsymbol{A}^i \boldsymbol{x}_0 + \sum_{k=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-k}\boldsymbol{w}_k\right]\left[\boldsymbol{x}_0^\top (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{k=0}^{i-1} \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c}\right] \\ &= \boldsymbol{c}^\top \boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{k=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{w}_k^\top (\boldsymbol{A}^\top)^{j-1-k}\boldsymbol{c} \\ &\quad + \sum_{k=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-k}\boldsymbol{w}_k \boldsymbol{x}_0^\top (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{k,l=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-k}\boldsymbol{w}_k \boldsymbol{w}_l^\top (\boldsymbol{A}^\top)^{j-1-l}\boldsymbol{c}. \end{aligned} \tag{29}$$

Using $\boldsymbol{w}_p \perp\!\!\!\perp \mathcal{F}_0 \subset \mathcal{F}_i$, $\forall p \geq 0$ and $\boldsymbol{w}_p \perp\!\!\!\perp \boldsymbol{w}_q$, $\forall p \neq q$ in conjunction with (29) and Lemma D.1 we get

$$\mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i} \boldsymbol{c} \,\middle|\, \mathcal{F}_0\right] = \boldsymbol{c}^\top \boldsymbol{A}^i \boldsymbol{x}_0 \boldsymbol{x}_0^\top (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{k=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-k} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-k} \boldsymbol{c} \tag{30}$$

Furthermore, noting that $\sigma(\boldsymbol{A}, \boldsymbol{c}) \subset \mathcal{F}_0$, we have that

$$\mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top (\boldsymbol{A}^\top)^{j-i} \boldsymbol{c} \,\middle|\, \boldsymbol{A}, \boldsymbol{c}\right] = \boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{k=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-k} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-k} \boldsymbol{c}. \tag{31}$$

Taking full expectation in (31), and plugging everything back into (28), we get the stated result

$$\mathbb{E}\left[y_i y_j\right] = \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\right] + \sum_{k=0}^{i-1} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^{i-1-k} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-k} \boldsymbol{c}\right] + \mathbf{1}_{\{i=j\}} \sigma_v^2. \qquad \square$$

Lemma D.3. *Let $\{y_i\}_{i \geq 0}$ be a sequence of observations generated by an LDS* (1) *sampled according to Assumption 3.2. Then,*

(a) if $i + j = 2p + 1$ for some $p \in \mathbb{N}_+$, $\mathbb{E}\left[y_i y_j\right] = 0$;

(b) if $i + j + k + l = 2p + 1$ for some $p \in \mathbb{N}_+$, $\mathbb{E}\left[y_i y_j y_k y_l\right] = 0$;

(c) if $i + j + k + l + m + n = 2p + 1$ for some $p \in \mathbb{N}_+$, $\mathbb{E}\left[y_i y_j y_k y_l y_m y_n\right] = 0$.

Note that there is no condition on the indices being pairwise distinct.

Proof. To prove point (a), we start from the expression derived in Lemma D.2.

$$\mathbb{E}\left[y_i y_j\right] = \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\right] + \sum_{k=0}^{i-1} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^{i-1-k} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-k} \boldsymbol{c}\right] + \mathbf{1}_{\{i=j\}} \sigma_v^2$$

Clearly, since $i + j$ is odd, it holds that $i \neq j$ and hence the third term is zero. Furthermore, since $\boldsymbol{A}$ has a centrally symmetric distribution, we have that

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\right] &= \mathbb{E}\left[\boldsymbol{c}^\top (-\boldsymbol{A})^i \Sigma_{\boldsymbol{x}_0} (-\boldsymbol{A}^\top)^j \boldsymbol{c}\right] \\ &= (-1)^{i+j} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\right], \end{aligned} \tag{32}$$

implying that $\mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\right] = 0$. We apply a similar reasoning for the other term and obtain that

$$\mathbb{E}\left[y_i y_j\right] = 0,$$

thus proving the first point.

For both points (b) and (c), we will rely on Isserlis's theorem, which we replicate in Theorem D.1 for convenience. Note that conditioned, on $\boldsymbol{A}$ and $\boldsymbol{c}$, the vectors $[y_i y_j y_k y_l | \boldsymbol{A}, \boldsymbol{c}]$ and $[y_i y_j y_k y_l y_m y_n | \boldsymbol{A}, \boldsymbol{c}]$ are jointly Gaussian since they are linear transformations of the jointly Gaussian vectors $\boldsymbol{r}_1 = \left[\boldsymbol{x}_0^\top, \boldsymbol{w}_0^\top, \ldots \boldsymbol{w}_{\max\{i,j,k,l\}}^\top, v_0, \ldots v_{\max\{i,j,k,l\}}\right]^\top$ and $\boldsymbol{r}_2 = \left[\boldsymbol{x}_0^\top, \boldsymbol{w}_0^\top, \ldots \boldsymbol{w}_{\max\{i,j,k,l,m,n\}}^\top, v_0, \ldots v_{\max\{i,j,k,l,m,n\}}\right]^\top$, respectively. We can therefore apply the towering property along with Isserlis's result to get

$$\begin{aligned} \mathbb{E}\left[y_i y_j y_k y_l\right] &= \mathbb{E}\left[\mathbb{E}\left[y_i y_j y_k y_l \mid \boldsymbol{A}, \boldsymbol{c}\right]\right] \\ &= \mathbb{E}\Big[\mathbb{E}\left[y_i y_j \mid \boldsymbol{A}, \boldsymbol{c}\right] \mathbb{E}\left[y_k y_l \mid \boldsymbol{A}, \boldsymbol{c}\right] + \mathbb{E}\left[y_i y_k \mid \boldsymbol{A}, \boldsymbol{c}\right] \mathbb{E}\left[y_j y_l \mid \boldsymbol{A}, \boldsymbol{c}\right] \\ &\qquad + \mathbb{E}\left[y_i y_l \mid \boldsymbol{A}, \boldsymbol{c}\right] \mathbb{E}\left[y_j y_k \mid \boldsymbol{A}, \boldsymbol{c}\right] \Big], \end{aligned} \tag{33}$$

since $\mathrm{PP}(\{i,j,k,l\}) = \{\{(i,j),(k,l)\},\{(i,k),(j,l)\},\{(i,l),(j,k)\}\}$. Since $i+j+k+l$ is odd, the two pairs inside any $q \in \mathrm{PP}(\{i,j,k,l\})$ must have different parities (i.e., one even, one odd). W.l.o.g, we analyze the first term in (33), assuming $0 \le i \le j \le k \le l$. From (31), we know that

$$\mathbb{E}\left[y_i y_j \mid \boldsymbol{A}, \boldsymbol{c}\right]\mathbb{E}\left[y_k y_l \mid \boldsymbol{A}, \boldsymbol{c}\right] = \left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{t=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-t} \boldsymbol{c} + \mathbf{1}_{\{i=j\}}\sigma_v^2\right] \left[\boldsymbol{c}^\top \boldsymbol{A}^k \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^l \boldsymbol{c} + \sum_{t=0}^{k-1} \boldsymbol{c}^\top \boldsymbol{A}^{k-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{l-1-t} \boldsymbol{c} + \mathbf{1}_{\{k=l\}}\sigma_v^2\right] \tag{34}$$

Assume w.l.o.g that $i+j$ is even, and $k+l$ is odd. This implies that $\mathbf{1}_{\{k=l\}} = 0$. Taking full expectation on both sides and developing the product, we get

$$\begin{aligned}
&\mathbb{E}\left[\,\mathbb{E}\left[y_i y_j \mid \boldsymbol{A}, \boldsymbol{c}\right]\mathbb{E}\left[y_k y_l \mid \boldsymbol{A}, \boldsymbol{c}\right]\,\right] \\
&\quad= \mathbb{E}\left[\,\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\boldsymbol{c}^\top \boldsymbol{A}^k \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^l \boldsymbol{c}\,\right] \\
&\qquad+ \sum_{t=0}^{k-1} \mathbb{E}\left[\,\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c}\boldsymbol{c}^\top \boldsymbol{A}^{k-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{l-1-t} \boldsymbol{c}\,\right] \\
&\qquad+ \sum_{t=0}^{i-1} \mathbb{E}\left[\,\boldsymbol{c}^\top \boldsymbol{A}^{i-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-t} \boldsymbol{c}\boldsymbol{c}^\top \boldsymbol{A}^k \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^l \boldsymbol{c}\,\right] \\
&\qquad+ \sum_{t=0}^{i-1}\sum_{s=0}^{k-1} \mathbb{E}\left[\boldsymbol{c}^\top \boldsymbol{A}^{i-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-t} \boldsymbol{c}\, \boldsymbol{c}^\top \boldsymbol{A}^{k-1-s} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{l-1-s} \boldsymbol{c}\right] \\
&\qquad+ \mathbf{1}_{\{i=j\}}\sigma_v^2 \mathbb{E}\left[\,\boldsymbol{c}^\top \boldsymbol{A}^k \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^l \boldsymbol{c}\right] \\
&\qquad+ \mathbf{1}_{\{i=j\}}\sigma_v^2 \sum_{t=0}^{k-1} \mathbb{E}\left[\,\boldsymbol{c}^\top \boldsymbol{A}^{k-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{l-1-t} \boldsymbol{c}\,\right]
\end{aligned} \tag{35}$$

Using the index parity assumptions and the reasoning based on the central symmetry of $\boldsymbol{A}'s$ distribution from (32), we get that all the terms on the RHS of (35) are zero. We treat the remaining terms in (33) similarly to get the final result in (b).

Finally, point (c) follows a similar path. We have

$$\begin{aligned}
\mathrm{PP}(\{i,j,k,l,m,n\}) = \{&\{(i,j),(k,l),(m,n)\},\{(i,j),(k,m),(l,n)\},\{(i,j),(k,n),(l,m)\},\\
&\{(i,k),(j,l),(m,n)\},\{(i,k),(j,m),(l,n)\},\{(i,k),(j,n),(l,m)\},\\
&\{(i,l),(j,k),(m,n)\},\{(i,l),(j,m),(k,n)\},\{(i,l),(j,n),(k,m)\},\\
&\{(i,m),(j,k),(l,n)\},\{(i,m),(j,l),(k,n)\},\{(i,m),(j,n),(k,l)\},\\
&\{(i,n),(j,k),(l,m)\},\{(i,n),(j,l),(k,m)\},\{(i,n),(j,m),(k,l)\}\}.
\end{aligned}$$

For the parity hypothesis to be satisfied, not that inside a set $q \in \mathrm{PP}(\{i,j,k,l,m,n\})$, at least one pair must have an odd parity, while the other two must be of the same parity (either even or odd). W.o.l.g let $0 \le i \le j \le k \le l \le m \le n$, pick the first set in $\mathrm{PP}(\{i,j,k,l,m,n\})$ above (the rest follow the same logic) and assume that $m+n$ is odd. By the same logic as before, we have that $\mathbf{1}_{\{m=n\}} = 0$ and

$$\begin{aligned}
&\mathbb{E}\left[\,\mathbb{E}\left[y_i y_j \mid \boldsymbol{A}, \boldsymbol{c}\right]\mathbb{E}\left[y_k y_l \mid \boldsymbol{A}, \boldsymbol{c}\right]\mathbb{E}\left[y_m y_n \mid \boldsymbol{A}, \boldsymbol{c}\right]\,\right] \\
&\quad= \mathbb{E}\Bigg[\left[\boldsymbol{c}^\top \boldsymbol{A}^i \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^j \boldsymbol{c} + \sum_{t=0}^{i-1} \boldsymbol{c}^\top \boldsymbol{A}^{i-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{j-1-t} \boldsymbol{c} + \mathbf{1}_{\{i=j\}}\sigma_v^2\right] \\
&\qquad\left[\boldsymbol{c}^\top \boldsymbol{A}^k \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^l \boldsymbol{c} + \sum_{t=0}^{k-1} \boldsymbol{c}^\top \boldsymbol{A}^{k-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{l-1-t} \boldsymbol{c} + \mathbf{1}_{\{k=l\}}\sigma_v^2\right] \\
&\qquad\left[\boldsymbol{c}^\top \boldsymbol{A}^m \Sigma_{\boldsymbol{x}_0} (\boldsymbol{A}^\top)^n \boldsymbol{c} + \sum_{t=0}^{k-1} \boldsymbol{c}^\top \boldsymbol{A}^{m-1-t} \Sigma_{\boldsymbol{w}} (\boldsymbol{A}^\top)^{n-1-t} \boldsymbol{c}\right]\Bigg]
\end{aligned}$$

Without computing, one can see that every term in the expanded product will have powers of $\boldsymbol{A}$ whose sum is odd. Therefore, using the centrally symmetric property of $\boldsymbol{A}$'s distribution, all the terms evaluate to zero, and point (c) is proven. □

D.3 Proof of Lemma 4.1

Lemma 4.1. *For an arbitrary s, the following parameters induce a banded structure in the left-hand side of* (11) *matching that of the right-hand side.*

$$\boldsymbol{W}_{QK} = \begin{bmatrix} \star & 0 & \star & \cdots & \\ 0 & \star & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \star \\ 0 & \cdots & 0 & \star & 0 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad \boldsymbol{W}_{V} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & \vdots \\ \vdots & & & \vdots & 0 \\ \vdots & & & \vdots & \star \\ \vdots & & & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & \star \end{bmatrix}. \tag{13}$$

Proof. Recall the in-context loss in (9) 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

$$\mathcal{L}(\theta) \coloneqq \mathbb{E}_{\tilde{\boldsymbol{D}}} \left[\left(\mathcal{T}_\theta \left(\boldsymbol{Y}_0 \right)_{s+1,T-s} - y_T \right)^2 \right]. \tag{36}$$

For equations (37) to (41) below, we use the same reformulations as Ahn et al. (2023). The last column of the transformer's output above can be written as

$$\begin{bmatrix} \bar{\boldsymbol{y}}_{T-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{y}}_{T-1} \\ 0 \end{bmatrix} + \frac{1}{T-s-1} \boldsymbol{W}_V^\top \left(\sum_{i=1}^{T-s-1} \begin{bmatrix} \bar{\boldsymbol{y}}_i \bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s} \\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2 \end{bmatrix} \right) \boldsymbol{W}_{QK}^\top \begin{bmatrix} \bar{\boldsymbol{y}}_{T-s} \\ 0 \end{bmatrix}, \tag{37}$$

where the summation comes from the causal mask. Therefore, the transformer's prediction of y_T, $\mathcal{T}_\theta \left(\boldsymbol{Y}_0 \right)_{s+1,T-s}$ can be written as

$$\frac{1}{T-s-1} \boldsymbol{b}^\top \left(\underbrace{\sum_{i=1}^{T-s-1} \begin{bmatrix} \bar{\boldsymbol{y}}_i \bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s} \\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2 \end{bmatrix}}_{\coloneqq \bar{\boldsymbol{Y}} \in \mathbb{R}^{(s+1)\times(s+1)}} \right) \left[\boldsymbol{f}_1 \boldsymbol{f}_2 \cdots \boldsymbol{f}_s \right] \bar{\boldsymbol{y}}_{T-s}, \tag{38}$$

where $\boldsymbol{b}^\top \in \mathbb{R}^{1\times(s+1)}$ is the last row of $\boldsymbol{W}_V^\top$ and $\boldsymbol{f}_j \in \mathbb{R}^{(s+1)}$ 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 $\boldsymbol{F} = [\boldsymbol{f}_j]_{j=1}^s$

$$\mathcal{L}(\boldsymbol{b}, \boldsymbol{F}) \coloneqq \mathbb{E}_{\tilde{\boldsymbol{D}}} \left[\left(\frac{1}{T-s-1} \boldsymbol{b}^\top \bar{\boldsymbol{Y}} \boldsymbol{F} \bar{\boldsymbol{y}}_{T-s} - y_T \right)^2 \right]. \tag{39}$$

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

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

$$= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\boldsymbol{b}^\top\bar{\boldsymbol{Y}}\boldsymbol{f}_k y_{T-s-1+k} - y_T\right)^2\right]$$
$$= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\mathrm{Tr}(\bar{\boldsymbol{Y}}\boldsymbol{f}_k\boldsymbol{b}^\top) y_{T-s-1+k} - y_T\right)^2\right]$$
$$= \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{b}\boldsymbol{f}_k^\top\rangle y_{T-s-1+k} - y_T\right)^2\right]. \tag{40}$$

We reparametrize the in-context loss using $\boldsymbol{X}_k \coloneqq \boldsymbol{b}\boldsymbol{f}_k^\top$

$$\mathcal{L}(\boldsymbol{X}_{k\in[s]}) = \mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{X}_k\rangle y_{T-s-1+k} - y_T\right)^2\right]. \tag{41}$$

Note that the gradient of the in-context loss with respect to each $\boldsymbol{X}_j$ is

$$\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{k\in[s]}) = 2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\left(\frac{1}{T-s-1}\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{X}_k\rangle y_{T-s-1+k} - y_T\right) y_{T-s-1+j}\bar{\boldsymbol{Y}}\right]. \tag{42}$$

The gradient $\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{k\in[s]})$ is a sum of two terms, $\nabla_{\boldsymbol{X}_j}\mathcal{L}(\boldsymbol{X}_{k=1\cdots s}) = \mathbf{T}^{(1)}_{\boldsymbol{X}_j} + \mathbf{T}^{(2)}_{\boldsymbol{X}_j}$, where, replacing $\bar{\boldsymbol{Y}}$ we have

$$\mathbf{T}^{(1)}_{\boldsymbol{X}_j} \coloneqq \frac{2}{T-s-1}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{k=1}^{s}\langle\bar{\boldsymbol{Y}}, \boldsymbol{X}_k\rangle y_{T-s-1+k} y_{T-s-1+j}\sum_{i=1}^{T-s-1}\begin{bmatrix}\bar{\boldsymbol{y}}_i\bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s}\\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2\end{bmatrix}\right] \tag{43}$$

$$\mathbf{T}^{(2)}_{\boldsymbol{X}_j} \coloneqq -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-s-1+j}\sum_{i=1}^{T-s-1}\begin{bmatrix}\bar{\boldsymbol{y}}_i\bar{\boldsymbol{y}}_i^\top & \bar{\boldsymbol{y}}_i y_{i+s}\\ \bar{\boldsymbol{y}}_i^\top y_{i+s} & y_{i+s}^2\end{bmatrix}\right]. \tag{44}$$

Each matrix element of $\mathbf{T}^{(2)}_{\boldsymbol{X}_j}$ has the form

$$\sum_{i=1}^{T-s-1} 2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\, y_T y_{T-s-1+j} y_{i+m} y_{i+n}\,\right] \tag{45}$$

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

The sum of y's indices in (45) for each term in the above sum is $2T + 2i + (m + n - s - 1 + j)$. The parity is determined by that of $m + n - s - 1 + j$ and is independent of the sum counter i (i.e., the same for all terms). According to Lemma D.3, (45) is 0 if $(m + n - s - 1 + j)$ is odd, and of arbitrary value if it is even. So a general matrix element of $\mathbf{T}^{(2)}_{\boldsymbol{X}_j}$ is 0 if $(m + n - s - 1 + j)$ is odd and of arbitrary value if $(m + n - s - 1 + j)$ is even.

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

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

We now turn to $\mathbf{T}^{(1)}_{\boldsymbol{X}_j}$ with the end goal of finding a parameter configuration that matches the sparsity pattern of $\mathbf{T}^{(2)}_{\boldsymbol{X}_j}$. For this section, assume s is odd (the other case follows similarly). First, let $\boldsymbol{X}_k := \left[x^{(k)}_{i,j}\right]_{i,j=1}^{s+1}$ and unfold the expression of the matrix inner product

$$\langle \bar{\boldsymbol{Y}}, \boldsymbol{X}_k \rangle = \sum_{r=0}^{s} \sum_{l=0}^{s} \sum_{p=0}^{T-s-1} x^{(k)}_{l+1,r+1} y_{p+r} y_{p+l}, \tag{46}$$

where r, l are the indices traversing $\bar{\boldsymbol{Y}}$.

Furthermore, each matrix element of $\mathbf{T}^{(1)}_{\boldsymbol{X}_j}$ inside the expectation has the form

$$\frac{2}{T-s-1} \sum_{i=1}^{T-s-1} \sum_{k=1}^{s} \langle \bar{\boldsymbol{Y}}, \boldsymbol{X}_k \rangle \, y_{T-s-1+k} \, y_{T-s-1+j} \, y_{i+n} \, y_{i+m}, \tag{47}$$

where $n, m \in \{0, 1, \ldots s\}$ are the indices traversing $\bar{\boldsymbol{Y}}$.

Assume that j is fixed and odd (we discuss the even case afterwards). Note that the sparsity of each position in $\mathbf{T}^{(2)}_{\boldsymbol{X}_j}$ dictated by the parity of $(m+n-s-1+j)$ where, when s, j-odd, the respective element is zero whenever $m+n$ is even. Notice that except for the contribution of the matrix inner product, the sum of indices for the y-factors in (47) is $2(T-s-2+i)+k+j+n+m$ so the parity is determined by that of $k+j+n+m$. We distinguish two cases:

(a) when k is even, $k+j$ is odd, and we wish that the term zeroes out for even $m+n$. This means that $\boldsymbol{X}_k$ must select in (46) only pairs $y_{p+r}y_{p+l}$ for which $r+l$ is even and zero-out the others. Such an $\boldsymbol{X}_k$ may look like

$$\boldsymbol{X}_k = \begin{bmatrix} x^{(k)}_{11} & 0 & x^{(k)}_{13} & \cdots & x^{(k)}_{1,s} & 0 \\ 0 & x^{(k)}_{22} & 0 & \cdots & 0 & x^{(k)}_{2,s+1} \\ x^{(k)}_{31} & 0 & x^{(k)}_{33} & \cdots & x^{(k)}_{3,s} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^{(k)}_{s,1} & 0 & x^{(k)}_{s,3} & \cdots & x^{(k)}_{s,s} & 0 \\ 0 & x^{(k)}_{s+1,2} & 0 & \cdots & 0 & x^{(k)}_{s+1,s+1} \end{bmatrix}, \tag{48}$$

with arbitrary (possibly also zero) values for constants $x_{i,j}^{(k)}$.

(b) when k is odd, $k+j$ is even, and we wish that the term zeroes out for even $m+n$. This means that $\boldsymbol{X}_k$ must select in (46) only pairs $y_{p+r}y_{p+l}$ for which $r+l$ is odd and zero-out the others. Such an $\boldsymbol{X}_k$ may look like

$$\boldsymbol{X}_k = \begin{bmatrix} 0 & x_{12}^{(k)} & 0 & \cdots & 0 & x_{1,s+1}^{(k)} \\ x_{21}^{(k)} & 0 & x_{23}^{(k)} & \cdots & x_{2,s}^{(k)} & 0 \\ 0 & x_{32}^{(k)} & 0 & \cdots & 0 & x_{3,s+1}^{(k)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & x_{s,2}^{(k)} & 0 & \cdots & 0 & x_{s,s+1}^{(k)} \\ x_{s+1,1}^{(k)} & 0 & x_{s+1,3}^{(k)} & \cdots & x_{s+1,s}^{(k)} & 0 \end{bmatrix}, \tag{49}$$

with arbitrary (possibly also zero) values for constants $x_{i,j}^{(k)}$.

These patterns need to be coherent with the case of j-even. Note that in $\mathbf{T}_{\boldsymbol{X}_j}^{(2)}$, when s-odd, j-even, the respective element is zero whenever $m+n$ is odd. We again distinguish two cases:

(a) when k is even, $k+j$ is even, and we wish that the term zeroes out for odd $m+n$. This means that $\boldsymbol{X}_k$ must select in (46) only pairs $y_{p+r}y_{p+l}$ for which $r+l$ is even and zero-out the others. Notice that the pattern of $\boldsymbol{X}_k$ in (48) for even k satisfies this requirement and we have coherence.

(b) when k is odd, $k+j$ is odd, and we wish that the term zeroes out for odd $m+n$. This means that $\boldsymbol{X}_k$ must select in (46) only pairs $y_{p+r}y_{p+l}$ for which $r+l$ is odd and zero-out the others. Notice that the pattern of $\boldsymbol{X}_k$ in (49) for even k satisfies this requirement and we have coherence.

The same approach goes through for even window size s. Finally, recall that $\boldsymbol{X}_k \coloneqq \boldsymbol{b}\boldsymbol{f}_k^\top$. For our case of odd window sizes, the sparsity pattern of $\boldsymbol{b}$ and $\boldsymbol{f}_k^\top$ yielding the $\boldsymbol{X}_k$ is

$$\boldsymbol{b} = \begin{bmatrix} 0 \\ b_2 \\ \vdots \\ 0 \\ b_{s+1} \end{bmatrix} \qquad \boldsymbol{f}_k^\top = \begin{cases} \left[0, f_2^{(k)}, \ldots, 0, f_{s+1}^{(k)}\right], \text{if } k \text{ is even} \\ \left[f_1^{(k)}, 0, \ldots f_s^{(k)}, 0\right], \text{if } k \text{ is odd} \end{cases} \tag{50}$$

For even window size s, the patterns are

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ 0 \\ b_2 \\ \vdots \\ 0 \\ b_{s+1} \end{bmatrix} \qquad \boldsymbol{f}_k^\top = \begin{cases} \left[0, f_2^{(k)}, \ldots, 0, f_s^{(k)}, 0\right], \text{if } k \text{ is even} \\ \left[f_1^{(k)}, 0, \ldots f_{s-1}^{(k)}, 0, f_{s+1}^{(k)}\right], \text{if } k \text{ is odd} \end{cases} \tag{51}$$

Arranging these vectors inside $\boldsymbol{W}_{QK}$ and $\boldsymbol{W}_V$ gives the stated result. □

D.4 Proof of Theorem 4.1

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*

$$\boldsymbol{W}_{QK}^\star = \begin{bmatrix} \frac{(T-2)\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{\boldsymbol{v}_t\}_t}\left[\sum_{i=1}^{T-2} y_i y_{i+1} y_{T-1} y_T\right]}{\mathbb{E}_{\boldsymbol{A},\boldsymbol{x}_0,\{\boldsymbol{w}_t\}_t,\{\boldsymbol{v}_t\}_t}\left[\sum_{i=1}^{T-2} y_i y_{i+1} \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2\right]} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{W}_V^\star = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{14}$$

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

Proof. For the transformer parameters in (14), the corresponding $\boldsymbol{b}^\top = [0 \quad 1]$ and the corresponding $\boldsymbol{F} = [c \quad 0]$, where $c := \frac{(T-2)\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} y_{T-1} y_T\right]}{\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\sum_{i=1}^{T-2} y_i y_{i+1} \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1} y_T\right]}$.

So $\boldsymbol{X} = \boldsymbol{X}_1 = \boldsymbol{b}\boldsymbol{f}_1^\top = \boldsymbol{b}\boldsymbol{F}^\top = \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix}$. The gradient of the in-context loss $\nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X})$ is

$$\begin{aligned}
\mathbf{T}_{\boldsymbol{X}_j}^{(1)} &= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\langle \bar{\boldsymbol{Y}}, \boldsymbol{X}\rangle y_{T-1}^2 \bar{\boldsymbol{Y}}\right] \\
&= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[\langle \sum_{r=1}^{T-2} \begin{bmatrix} y_r^2 & y_r y_{r+1} \\ y_{r+1} y_r & y_{r+1}^2 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix} \rangle y_{T-1}^2 \sum_{i=1}^{T-2} \begin{bmatrix} y_i^2 & y_i y_{i+1} \\ y_{i+1} y_i & y_{i+1}^2 \end{bmatrix}\right] \\
&= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} \begin{bmatrix} y_i^2 & y_i y_{i+1} \\ y_{i+1} y_i & y_{i+1}^2 \end{bmatrix}\right] \\
&= \frac{2}{T-2}\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} \begin{bmatrix} 0 & y_i y_{i+1} \\ y_{i+1} y_i & 0 \end{bmatrix}\right].
\end{aligned} \tag{52}$$

According to Lemma D.3, the two diagonal elements in (52) $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[c \sum_{r=1}^{T-2} y_r y_{r+1} y_{T-1}^2 \sum_{i=1}^{T-2} y_i^2\right]$ 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 the sums of y indices are both odd.

$$\begin{aligned}
\mathbf{T}_{\boldsymbol{X}_j}^{(2)} &= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} \bar{\boldsymbol{Y}}\right] \\
&= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} \begin{bmatrix} y_i^2 & y_i y_{i+1} \\ y_{i+1} y_i & y_{i+1}^2 \end{bmatrix}\right] \\
&= -2\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} \begin{bmatrix} 0 & y_i y_{i+1} \\ y_{i+1} y_i & 0 \end{bmatrix}\right].
\end{aligned} \tag{53}$$

According to Lemma D.3, the two diagonal elements in (53) $\mathbb{E}_{\tilde{\boldsymbol{D}}}\left[y_T y_{T-1} \sum_{i=1}^{T-2} y_i^2\right]$ 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 the sums of y indices are both odd.

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

$$\nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}) = \mathbf{T}_{\boldsymbol{X}_j}^1 + \mathbf{T}_{\boldsymbol{X}_j}^2 = 0. \tag{54}$$

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. □

NEW ADDED

E PROOFS FOR SECTION 5

E.1 PROOF THAT OUR EXPERIMENTS' SAMPLING SCHEMES OBEY ASSUMPTION 3.2

All our experiments use a sampling schemes whose generalization is the following:

(a) $\boldsymbol{A}$ constructed by sampling $\boldsymbol{v} \sim \mathcal{P}$, where $\mathcal{P}$ is centrally symmetric and absolutely continuous w.r.t. the Lebesgue measure on $\mathbb{R}^d$ with marginals supported on $[-1, 1]$, and independently sampling $\mathbf{P}$, whose every entry is drawn i.i.d. from any absolutely continuous distribution w.r.t. Lebesgue measure in $\mathbb{R}$. Matrix $\boldsymbol{A}$ is then formed as $\mathbf{P}\mathrm{diag}(\boldsymbol{v})\mathbf{P}^{-1}$.

(b) $\boldsymbol{c}$ is sampled from an absolutely continuous distribution w.r.t. Lebesgue measure in $\mathbb{R}^d$, or otherwise fixed with $\boldsymbol{c} \neq \mathbf{0}_d$.

We need to show that

(a) $\boldsymbol{A}$'s distribution is centrally symmetric, i.e., that $\boldsymbol{A} \overset{\mathrm{d}}{=} -\boldsymbol{A}$;

(b) $\boldsymbol{A}$'s spectrum is simple w.p. 1;

(c) observability still holds when $\boldsymbol{c}$ is fixed according to the above condition.

The first point is achieved since, by the central symmetry of $\boldsymbol{v}$'s distribution,

$$-\boldsymbol{A} = -\boldsymbol{P}^{-1}\mathrm{diag}(\boldsymbol{v})\boldsymbol{P} = -\boldsymbol{P}^{-1}\mathrm{diag}(-\boldsymbol{v})\boldsymbol{P} \overset{\mathrm{d}}{=} \boldsymbol{P}^{-1}\mathrm{diag}(\boldsymbol{v})\boldsymbol{P} = \boldsymbol{A}. \tag{55}$$

The second point is ensured by $\boldsymbol{v}$'s distribution being absolutely continuous w.r.t. the Lebesgue measure in $\mathbb{R}^d$, and hence the probability of $\boldsymbol{v}$ belonging to $(d-1)$-dimensional subspaces (and lower) such as $\left\{\boldsymbol{x} \in \mathbb{R}^d \,\middle|\, \exists i, j \in [d] \text{ s.t. } x_i = x_j\right\}$ is null. In conjunction with the above, when we sample $\boldsymbol{c}$ from a continuous distribution in $\mathbb{R}^d$, Assumption (3.2) is satisfied.

However, our proofs and experiments go through even if $\boldsymbol{c}$ is fixed, as follows. First, the theoretical results rest on $\boldsymbol{A}$'s distributional symmetry and are invariant to the linear transformation induced by $\boldsymbol{c}$. Second, observability is ensured since $\det(\boldsymbol{O})$ in expression (5) is not zero w.p. 1, as follows.

We use $\det(\boldsymbol{OP}) \neq 0$ w.p. 1 $\iff$ $\det(\boldsymbol{O}) \neq 0$ w.p. 1, since $\det(\boldsymbol{P}) \neq 0$ w.p. 1.

$$\det(\boldsymbol{OP}) \overset{\boldsymbol{z}:=\boldsymbol{c}^\top \boldsymbol{P}}{=} \det([\boldsymbol{z}; \mathrm{diag}(\boldsymbol{v})\boldsymbol{z}; \ldots \mathrm{diag}(\boldsymbol{v})^{d-1}\boldsymbol{z}]) \tag{56}$$

$$= \det(\mathrm{diag}(\boldsymbol{z}))\det\left(\begin{bmatrix} 1 & v_1 & \ldots & v_1^{d-1} \\ 1 & v_2 & \ldots & v_2^{d-1} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & v_d & \ldots & v_d^{d-1} \end{bmatrix}\right). \tag{57}$$

Since $\boldsymbol{P}$'s entries are drawn i.i.d. from an absolutely continuous distribution w.r.t. Lebesgue measure in $\mathbb{R}$, it holds that $z_i \neq 0$ w.p. 1. The remaining matrix is Vandermonde with $v_i \neq v_j$, $\forall i, j \in [d]$ w.p. 1. Hence, the determinant is nonzero w.p. 1 and observability holds almost surely.

E.2 RELATION OF TRANSFORMER FORWARD PASS WITH PCG

For convenience, we reproduce below the PCG iteration of Shewchuk et al. (1994) for minimizing an objective

$$f(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{A}\boldsymbol{w} + \boldsymbol{b}^\top \boldsymbol{w} + c$$

Algorithm 1 Preconditioned Conjugate Gradient

Input: preconditioner $\boldsymbol{H}$, $\boldsymbol{w}_0$, $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{w}_0$, $\boldsymbol{d}_0 = \boldsymbol{H}^{-1}\boldsymbol{r}_0$, $\delta_{\text{new}} = \boldsymbol{r}_0^\top \boldsymbol{d}_0$, $\delta_0 = \delta_{\text{new}}$
for $i = 0, 1, \ldots$ **do**
  $\boldsymbol{z}_i = \boldsymbol{A}\boldsymbol{d}_i$
  $\alpha_i = \frac{\delta_{\text{new}}}{\boldsymbol{d}_0^\top \boldsymbol{z}_0}$
  $\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \alpha_i \boldsymbol{d}_i$
  $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{z}_i$
  $\boldsymbol{v}_{i+1} = \boldsymbol{H}^{-1}\boldsymbol{r}_{i+1}$
  $\delta_{\text{old}} = \delta_{\text{new}}$, $\delta_{\text{new}} = \boldsymbol{r}_{i+1}^\top \boldsymbol{v}_{i+1}$,
  $\beta_{i+1} = \frac{\delta_{\text{new}}}{\delta_{\text{old}}}$
  $\boldsymbol{d}_{i+1} = \boldsymbol{v}_{i+1} + \beta_{i+1}\boldsymbol{d}_i$
end for
return θ_T

We compute the first two steps of the algorithm with respect to the loss (4), which can be rewritten as

$$\mathcal{L}_{AR(s)}(\boldsymbol{w}) \coloneqq \frac{1}{2(T-s-1)} \sum_{t=1}^{T-s-1} (y_{t+s} - \boldsymbol{w}^\top \bar{\boldsymbol{y}}_t)^2 \tag{58}$$

$$= \frac{1}{2(T-s-1)} \sum_{t=1}^{T-s-1} \boldsymbol{w}^\top \bar{\boldsymbol{y}}_t \bar{\boldsymbol{y}}_t^\top \boldsymbol{w} - 2y_{t+s}\boldsymbol{w}^\top \bar{\boldsymbol{y}}_t + y_{t+s}^2 \tag{59}$$

$$= \frac{1}{2}\boldsymbol{w}^\top \nabla^2 \mathcal{L}_{AR(s)} \boldsymbol{w} - \boldsymbol{w}^\top \nabla \mathcal{L}_{AR(s)}(0) + y_{t+s}^2 \tag{60}$$

Using the initializations proposed in the main text, $\boldsymbol{w}_0 = \boldsymbol{0}$ and $\boldsymbol{d}_0 = \boldsymbol{q}$, and $\boldsymbol{H} = \boldsymbol{P}^{-1}$ we get

$$\begin{aligned}
\boldsymbol{w}_1 &= \alpha_0 \boldsymbol{d}_0 = \alpha_0 \boldsymbol{q} \\
\boldsymbol{w}_2 &= \boldsymbol{w}_1 + \alpha_1 \boldsymbol{d}_1 \\
&= \alpha_0 \boldsymbol{q} + \alpha_1 \left[\boldsymbol{P}\boldsymbol{r}_1 + \beta_1 \boldsymbol{d}_0\right] \\
&= \alpha_0 \boldsymbol{q} + \alpha_1 \left[\boldsymbol{P}(\boldsymbol{r}_0 - \alpha_0 \boldsymbol{z}_0) + \beta_1 \boldsymbol{q}\right] \\
&= \alpha_0 \boldsymbol{q} + \alpha_1 \left[\boldsymbol{P}(\nabla \mathcal{L}_{AR(s)}(0) - \alpha_0 \nabla^2 \mathcal{L}_{AR(s)} \boldsymbol{q}) + \beta_1 \boldsymbol{q}\right] \\
&= \alpha_0 \boldsymbol{q} + \alpha_1 \boldsymbol{P} \nabla \mathcal{L}_{AR(s)}(0) - \alpha_0 \alpha_1 \boldsymbol{P} \nabla^2 \mathcal{L}_{AR(s)} \boldsymbol{q} + \alpha_1 \beta_1 \boldsymbol{q} \\
&= \left((\alpha_0 + \alpha_1 \beta_1)\boldsymbol{I} - \alpha_0 \alpha_1 \boldsymbol{P} \nabla^2 \mathcal{L}_{AR(s)}\right) \boldsymbol{q} + \alpha_1 \boldsymbol{P} \nabla \mathcal{L}_{AR(s)}(0) \\
&= \left[(\alpha_0 + \alpha_1 \beta_1) \nabla^2 \mathcal{L}_{AR(s)}^{-1} - \alpha_0 \alpha_1 \boldsymbol{P}\right] \nabla^2 \mathcal{L}_{AR(s)} \boldsymbol{q} + \alpha_1 \boldsymbol{P} \nabla \mathcal{L}_{AR(s)}(0)
\end{aligned}$$

E.3 Merging the $\hat{\gamma}(0)$ term into the Hessian preconditioner

We want to show that there exists a matrix $\mathbf{P}' \in \mathbb{R}^{s \times s}$ such that $c_N \mathbf{p} \hat{\gamma}_0 = \mathbf{P}' \nabla^2 \mathcal{L}_{AR(s)} \mathbf{q}$.

Let $\boldsymbol{v} = \nabla^2 \mathcal{L}_{AR(s)} \mathbf{q}$, then $\boldsymbol{P}' \coloneqq \frac{c_N \hat{\gamma}_0 \mathbf{p} \boldsymbol{v}^\top}{\boldsymbol{v}^\top \boldsymbol{v}}$ satisfies

$$\mathbf{P}' \nabla^2 \mathcal{L}_{AR(s)} \mathbf{q} = \frac{c_N \hat{\gamma}_0 \mathbf{p} \boldsymbol{v}^\top}{\boldsymbol{v}^\top \boldsymbol{v}} \nabla^2 \mathcal{L}_{AR(s)} \mathbf{q} = \frac{c_N \hat{\gamma}_0 \mathbf{p} \boldsymbol{v}^\top \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}} = c_N \hat{\gamma}_0 \mathbf{p}$$