Optimal In-Context Learning of Autoregressive Processes under Heterogeneous Second-Order Moments of the Prompts

Hanna Tseran*
RIKEN Center for Advanced Intelligence Project
Tokyo, Japan 103-0027
hanna.tseran@gmail.com

Masaaki Imaizumi
Graduate School of Arts and Sciences, The University of Tokyo
Tokyo, Japan 153-8902
RIKEN Center for Advanced Intelligence Project
Tokyo, Japan 103-0027
Kyoto University
Kyoto, Japan 606-8501
imaizumi@g.ecc.u-tokyo.ac.jp

Abstract

In-context learning enables large language models to adapt to tasks directly from input sequences, without parameter updates. We investigate the mechanism underlying in-context learning of autoregressive processes in transformers under heterogeneous second-order moments of prompt distributions. Our study considers a two-layer architecture consisting of a linear attention head and a nonlinear MLP, with the outer layer trained via gradient descent. We prove that there exists a fixed attention head parametrization under which the model converges to the unique function that approximates the optimal second-order moment-based AR(1) estimator. We analyze bounds on approximation and convergence rates and confirm our findings experimentally. These results extend the first-moment-centric understanding of estimators realized by in-context learning to second-order moment-based estimators, offering further insight into the robustness of ICL to new tasks.[2]

1 Introduction

In-context learning (ICL) in transformers (Vaswani et al., 2017) refers to the phenomenon in which a model learns a new task without changing its weights. It has become a topic of great interest since the rise of large language models (Brown et al., 2020). We focus on the autoregressive transformer training of the autoregressive process (AR) and consider the model that operates on sequences, called prompts, of inputs sampled from the AR(1) process $\mathcal{P}_\tau = (x_{\tau,1}, x_{\tau,2}, \ldots, x_{\tau,N})$, where $x_{\tau,i+1} = \alpha_\tau x_{\tau,i} + \xi_{\tau,i+1}$, $\xi_{\tau,i+1}$ is i.i.d. Gaussian noise and α_τ is different for each prompt $\mathcal{P}_\tau$, resulting in different second-order moments. The goal for an in-context learner is to use the prompt to form a prediction $\hat{x}_{\tau,N+1}$ such that $\hat{x}_{\tau,N+1} \approx x_{\tau,N+1}$. We take the perspective of a framework for in-context learning of function classes, as introduced in Garg et al. (2022), and characterize the function learned by a transformer.

*Work done while at RIKEN AIP. Now at Becoming.

[2]The code is available at https://github.com/hanna-tseran/icl_ar_heterogeneous

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Many recent theoretical works focus on analyzing in-context learning. There is particular emphasis on i.i.d. linear regression (Ahn et al., 2023; Mahankali et al., 2024; Zhang et al., 2024b; Huang et al., 2024; Zhang et al., 2025b). Autoregressive processes are a natural and theoretically tractable formalization of autoregressive next-token prediction, and ICL for AR processes has been studied by Sander et al. (2024); Zheng et al. (2024). Separately, prior work has examined the adaptation mechanisms of ICL under distribution shifts (Zhang et al., 2024a; Ahuja and Lopez-Paz, 2023). However, a systematic understanding of the ICL for noisy AR processes with general autoregressive coefficients, as well as convergence and approximation analysis, is still lacking. In particular, it remains unclear how existing results extend to heterogeneous prompt distributions in the AR setting, including heterogeneity in higher-order moments. Furthermore, recent results indicate that ICL in transformers can implement first-moment-based methods using count- or mean-based estimators, and can also behave as second-moment-based estimators (Akyürek et al., 2023; Edelman et al., 2024; Nichani et al., 2024). Nevertheless, it remains unclear whether ICL can explicitly realize estimators based on higher-order moments.

Contributions Our contributions are as follows.

- We specify the function map representing the optimal second-order moment-based estimator, in the sense of the estimator being consistent and asymptotically efficient, for ICL of an AR(1) process with heterogeneous distribution of the second-order moments of the prompts (Lemma 4.5).
- We prove that under the proposed fixed attention parametrization, a transformer realizes the unique function approximating the optimal AR(1) estimator, analyze convergence (Lemma 4.8) and approximation bounds (Lemma F.4 and Lemma 4.8), and summarize our main result in Theorem 4.1. With this, we extend the first-moment-centric understanding of estimators realized by in-context learning to second-order moment-based estimators, such as variances and covariances.
- We discuss an extension to the multidimensional VAR(1) process (Section 6.2) and confirm convergence and the function shape experimentally (Section 5).

2 Related work

Many recent theoretical works focus on in-context learning, particularly for i.i.d. linear regression. A prominent view is that during in-context learning, linear attention transformers implement a single step of gradient descent on the least-squares objective (Ahn et al., 2023; Mahankali et al., 2024; Zhang et al., 2024b). Other results establish near Bayes-optimal behavior (Zhang et al., 2024b; Wu et al., 2024), analyze training dynamics and convergence (Huang et al., 2024; Chen et al., 2024a; Zhang et al., 2025b), and study asymptotical behavior of in-context learning (Lu et al., 2025). Further, we discuss primary related works and review additional literature in Appendix A.

ICL of AR processes Extending these results further, we consider ICL of AR processes, which are a natural and theoretically tractable formalization of autoregressive next-token prediction. Moreover, the autoregressive model enables modeling a heterogeneous distribution of prompts via changes in the autoregressive coefficient, provides a closed-form optimal estimator, and, unlike i.i.d. linear regression, captures interdependencies between prompt tokens. Sander et al. (2024) analyze in-context learning of a noiseless AR(1) process with an orthogonal coefficient matrix, showing that a transformer trained on augmented tokens implements one step of gradient descent and characterizing the global minima for the non-augmented case. Gong et al. (2025) study how ICL can emerge from generalization bounds for autoregressive next-token prediction by employing PAC-Bayesian analysis. Zheng et al. (2024) consider a scenario that is most closely related to ours. The authors investigate the non-convex dynamics of a one-layer linear causal self-attention model autoregressively trained by gradient flow, where the sequences are generated by a noiseless AR(1) process. However, our work differs from existing studies on ICL of AR processes in several respects. We focus on the noisy AR process, a transformer consisting of a linear attention and a nonlinear MLP trained using gradient descent, discuss heterogeneity of the second-order moments of the prompts, show convergence to the optimal second-order moment-based estimator, and provide approximation bounds.

ICL adaptation to unseen tasks An additional fundamental question is the mechanism by which ICL adapts to unseen tasks. Ahuja and Lopez-Paz (2023) study covariate distribution shifts, and

Zhang et al. (2024a) consider task, query, and covariate distribution shifts with the train data distribution being the same for each task. Conversely, our approach allows for different covariance matrices between the prompts, a property we refer to as *heterogeneity of the second moments of the tasks*. In our case, a change in the autoregressive coefficient corresponding to a prompt leads to a change in the distribution of the datapoints sampled from the AR(1) process defined by it. Hence, our setup naturally allows us to consider distributions that vary across both training and test tasks. A similar setup to ours, but for the case of Markov chains, appears in Nichani et al. (2024), where each task is generated from a randomly chosen Markov chain.

Complexity of ICL estimators Another crucial question is the complexity of the methods that ICL implements. Published results demonstrate that transformers can implement first-moment-based methods using count- or mean-based estimators and can behave as second-moment-based estimators. Edelman et al. (2024) study transformers trained on Markov chain data and show that they learn in context by estimating n-gram statistics. Nichani et al. (2024) show that transformers can in-context estimate transition distributions, with the gradient of the first attention layer proportional to the χ^2-mutual information between pairs of tokens. For linear regression, Akyürek et al. (2023) constructively show that an autoregressive L-layer transformer can predict according to a single ShermanMorrison update, which is an iterative approximation of the OLS estimator. The other line of analysis establishes near-Bayes-optimal behavior (Zhang et al., 2024b; Wu et al., 2024) and is similar in that it shows the optimal estimator is learned for linear regression. However, it is not yet entirely clear whether ICL can explicitly realize higher-order moment-based estimators. Ahn et al. (2023) characterize the global minimizer of linear attention-only transformers for linear regression, showing that it approximates the inverse of the data covariance matrix, but does not consider convergence. In contrast, we show that a simple transformer can explicitly estimate the variance and covariance of the data and converge to a model that uses them to generate predictions with the optimal estimator. Another difficulty in methods using higher-moment estimators is demonstrating how a transformer can estimate the inverse of a scalar or matrix. We address this issue using a polynomial approximation of the inverse via a nonlinear MLP that uses variance and covariance aggregated in the attention layer.

3 Preliminaries

Notation Norms of vectors $\|\cdot\|$ denote ℓ_2 norms. Unless stated otherwise, we use lower case letters for scalars, bold lower case letters for vectors, and upper case letters for matrices. Subscript $[:, -1]$ denotes the rightmost column in the matrix, e.g., $A_{[:,-1]}$. We use $\xrightarrow{L^2}$ to denote L^2 convergence.

3.1 Autoregressive process

We focus on AR processes of dimension $d = 1$ and discuss extension to vector AR processes for $d \geq 2$ in Section 6.2.

Definition 3.1 (AR(1) process)**.** We consider a sequence of centered random variables $(x_1, \ldots, x_N, x_{N+1}), x_i \in \mathbb{R}, i = 1, \ldots, N+1$ that follows an autoregressive model of order 1, AR(1):

$$x_{i+1} = \alpha x_i + \xi_{i+1} \tag{1}$$

with a coefficient $\alpha \in (-1, 1)$ and a centered noise term $\xi_{i+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ independent of x_i.

We sample the first data point x_1 from the AR(1) stationary distribution, $x_1 \sim \mathcal{N}(0, 1/(1-\alpha^2))$. Hence, the considered AR(1) process is stationary. Further, we denote with $\text{AR}(1; \alpha)$ the AR(1) process with the coefficient α. The datapoints $x_i \in \mathbb{R} = \mathbb{R}^d$, where $d = 1$.

3.2 In-context learning of an autoregressive process

We consider an in-context learning problem of predicting the next data point in an AR process. Specifically, there are $\mathcal{T}$ prompts, to which we also refer as contexts, $\mathcal{P}_1, \ldots, \mathcal{P}_\mathcal{T}$, where each prompt consists of N data points generated from an autoregressive process, $\mathcal{P}_\tau = (x_{\tau,1}, \ldots, x_{\tau,N}), \tau = 1, \ldots, \mathcal{T}, x_{\tau,1:N} \sim \text{AR}(1, \alpha_\tau)$. Given access to these $\mathcal{T}$ contexts, we aim

to construct a learner that predicts x_{N+1} in a new context. Our setup is illustrated in Figure 4 in Appendix B.

Heterogeneous second-order moments Heterogeneous second-order moments naturally arise in this setup due to the assumption of a distribution over the autoregressive coefficients α_τ. This assumption implies that the covariance of the AR(1) process in Definition 3.1, which is equal to $\mathrm{cov}(x_{\tau,i}, x_{\tau,i+1}) = \alpha/(1-\alpha_\tau^2)$, changes between prompts. Note that while the term "heterogeneity" appears in slightly different contexts in the literature, in our work we use it to refer to the situation when the second-order moments of the distribution are different across all training and testing prompts. Since we further study the population loss, we omit the prompt index notation τ.

3.3 Model design

Embeddings Denote with $\mathcal{P}_N = (x_1, \ldots, x_N)$ a prompt of length N. The goal is for the model to predict x_{N+1} given P_N. For a given prompt $\mathcal{P}_N$ of length N, consider the input embedding $E_N = E(P_N)$ formed by stacking the input data consisting of the input-output pairs and adding two additional rows and one additional column, which will be useful in future computations:

$$E_N \coloneqq \begin{pmatrix} x_1 & x_2 & \cdots & x_{N-1} & 1 \\ x_2 & x_3 & \cdots & x_N & 0 \\ 0 & 0 & \cdots & 0 & x_N \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n \times N}. \tag{2}$$

We denote the number of rows in the embedding matrix (embedding dimension) with n. This embedding differs slightly from those commonly used in other works by adding two rows and modifying the last column (Zhang et al., 2024a; Chen et al., 2024a). Given the simplicity of this transformation, one might expect it to be learned by an embedding layer. This embedding simplifies the theoretical derivation by allowing the accumulation of variance, covariance, and the last observation in the attention output.

Transformer architecture We consider a two-layer transformer network in which the embedding is followed by a linear self-attention head and a nonlinear MLP layer. We denote all network parameters with $\theta \in \mathbb{R}^P$, specifically $\theta = \{\boldsymbol{\gamma}, \boldsymbol{\beta}, W_O, W_V, W_K, W_Q\}$, where $\boldsymbol{\gamma} \in \mathbb{R}^M$ and $\boldsymbol{\beta} \in \mathbb{R}^{M \times n}$ are weights of the MLP block, and $W_O, W_V, W_K, W_Q \in \mathbb{R}^{n \times n}$ are weights of the attention layer. For the activation function σ, we assume that $\sigma(x)$ is homogeneous of degree 1, i.e., $\sigma(\alpha x) = \alpha\sigma(x)$, L-Lipshitz, $\sigma'(x)$ is piecewise continuous on $\mathbb{R}$, $\int_{-\infty}^{\infty} \big(\sigma(x)\big)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x$ converges, $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^2] \le 1$ and $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^4] \le C_\sigma$, where $C_\sigma \in \mathbb{R}$ is a constant. These assumptions allow ReLU and leaky-ReLU type activation functions. Specifically, we consider

$$f(E_N; \theta) = f_{\text{MLP}}(f_{\text{att}}(E_N))) = \frac{1}{M}\sum_{i=1}^{M} \gamma_i \sigma(\boldsymbol{\beta}_i f_{\text{att}}(E_N)), \qquad f_{\text{MLP}} : \mathbb{R}^M \to \mathbb{R}, \tag{3}$$

where

$$f_{\text{att}}(E_N; \theta) = W_O^\top W_V E_N E_N^\top W_K^\top W_Q E_{N[:,-1]}, \qquad f_{\text{att}} : \mathbb{R}^{n\times N} \to \mathbb{R}^M, \tag{4}$$

is the attention layer with one linear attention head. Since we assume that the output appears in the last column of the output, we only use the rightmost column in the rightmost multiplication in (4).

3.4 Training setup

We initialize weights of the inner MLP layer as $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n), i = 1, \ldots, M$. We do not assume any specific initialization for the outer layer weights $\boldsymbol{\gamma} \in \mathbb{R}^M$. For instance, they can be set to zero. We assume that the autoregressive coefficient α is sampled from a uniform distribution on the open interval $(-1+\nu, 1-\nu)$, where $\nu \in \mathbb{R}, \nu > 0$ is a small constant, $\alpha \sim \mathcal{U}(-1+\nu, 1-\nu)$.

We consider the in-context learning problem and minimize the following population mean squared error (MSE) loss

$$\min_{\theta\in\mathbb{R}^P} \frac{1}{2} \mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu) \\ x_{1:N+1}\sim\text{AR}(1;\alpha)}} \left[|x_{N+1} - f(E_N;\theta)|^2\right] = \min_{\theta\in\mathbb{R}^P} \mathcal{L}(\theta), \tag{5}$$

where we take the expectation with respect to the data $x_{1:N+1}$ sampled from the AR(1) process, which is defined by the coefficient α, with the i.i.d. noise $\xi_{2:N+1}$. We omit noise in the expectation notation for brevity. Further, when clear from the context, we will be using either $\mathbb{E}_{\alpha,x_{1:N+1}}$ or simply $\mathbb{E}$. We say that a network is capable of in-context learning if the loss (5) is small on unseen prompts. A similar setup, for instance, appeared in Zhang et al. (2024a).

We train the network using gradient descent that optimizes only the outer layer weights $\boldsymbol{\gamma}$, while keeping all other weights fixed:

$$\gamma_i' = \gamma_i - \lambda \frac{\partial \mathcal{L}}{\partial \gamma_i}, \tag{6}$$

where $\lambda \in \mathbb{R}$ is a learning rate.

3.5 Challenges of the problem

The main challenge of this problem is analyzing how ICL handles heterogeneous second-order moments in prompts. This provides insight into ICL's ability to handle unseen tasks. Controlling for different covariances is challenging. To do it accurately, the learned estimator needs to use second-order moments estimated in context. Published results mainly show that transformers can implement first-moment-based methods using count- or mean-based estimators and can behave as second-moment-based estimators. However, these results do not explicitly demonstrate how a transformer can use higher-order moment-based estimators.

4 Convergence to the optimal estimator

We first present a simplified version of our result for readability.

Theorem 4.1 (Learning the optimal estimator with a transformer. Simplified)**.** *Assume the setup from Section 3. Fix a desired error ε and a parameter $p \in \mathbb{N}, p \geq 1$. Then, there exists such a fixed parametrization of the attention layer f^*_{att}, defined in Lemma 4.6, that for $M \geq \binom{n+p}{p}^{\frac{1}{2}} 3^{\frac{p}{2}} C_M/\varepsilon^2$, where $C_M \in \mathbb{R}$ is a constant, for a learning rate $\lambda < M/C_{\mathbf{u}}$, and sufficiently large N and p, parameters of the outer layer $\boldsymbol{\gamma}$ will converge to the unique solution $\boldsymbol{\gamma}^*$, such that the network will approximate the optimal estimator for the AR(1) process in* (7)*, and, with probability $\geq 1 - C_1 - C_2/(N-1)$,*

$$\mathbb{E}_{\alpha,x_{1:N+1},\boldsymbol{\beta}}\left[|x_{N+1} - f_{\text{MLP}}(f^*_{\text{att}}(E_N); \boldsymbol{\gamma}^*, \boldsymbol{\beta})|^2\right] \leq \left(\left(\frac{12}{N-6}+1\right)^{\frac{1}{2}} + C\left(\frac{1}{2}\right)^{(p+2)} + 2\varepsilon\right)^2,$$

where C, C_1, C_2 are constants.

Remark 4.2 (Interpretation of Theorem 4.1). Theorem 4.1 shows that there exists such a fixed parametrization of the attention layer that the network in (3), when its MLP layer is trained with gradient descent, converges to a consistent and asymptotically efficient OLS estimator, defined in Lemma 4.5, for AR(1)-based prompts with heterogeneous second-order moments. This finding enhances understanding of transformers' ability to generalize to unseen tasks during in-context learning. Additionally, the result indicates that the approximation error depends exponentially on the MLP width and linearly on the prompt length. Furthermore, the approximation error eventually approaches 1, that is, an irreducible error due to standard Gaussian noise. In the proof, we approximate the inverse function in the estimator using a polynomial, which is, in turn, approximated by a nonlinear MLP, suggesting a potential role of the MLP layer in architectures capable of realizing higher-order moment-based estimators.

Proof outline The proof has the following main steps:

1. We define the optimal estimator (Lemma 4.5), the learning goal, and the feature map needed to compute it. Then we show that attention can realize the prespecified feature map (Lemma 4.6).
2. We demonstrate equivalence of the optimization problems and reduce the problem to the inverse function approximation (Lemma 4.7).

3. We analyze approximation bounds for approximating the function with the polynomial and for the proposed estimator (Lemma F.4). We bound the approximation error and analyze the convergence of the MLP approximation of the polynomial (Lemma 4.8).
4. The results are combined in Theorem 4.1, which is a simplified version of the full result in Theorem C.1 proved in Appendix C.

Step 1. Attention realizes the feature map needed for the optimal estimator

Definition 4.3 (Consistent estimator)**.** When the probability limit of a sequence of estimators is equal to the true population parameter, the estimator is said to be consistent.

Definition 4.4 (Asymptotic efficiency, Definition 10.1.11 in Casella and Berger (2002))**.** A sequence of estimators $\hat{\alpha}_N$ is asymptotically efficient for a parameter α if $\sqrt{N}(\hat{\alpha}_N - \alpha) \to \mathcal{N}(0,\, v(\alpha))$ in distribution, and $v(\alpha) = 1/\mathcal{I}(\alpha)$, where $\mathcal{I}(\alpha)$ is Fisher information. That is, the asymptotic variance of $\hat{\alpha}_N$ attains the CramérRao lower bound.

Now we introduce a well-known result that defines the *optimal estimator* for the covariance-stationary AR(1) process in the sense of this estimator being *consistent and asymptotically efficient*.

Lemma 4.5 (Optimality of the OLS estimator)**.** *Consider a centered AR(1) model $x_{i+1} = \alpha x_i + \xi_{i+1},\ \ i = 1, \ldots, N-1$, where the autoregressive coefficient $\alpha \in (-1+\nu, 1-\nu)$, the noise term $\xi_{i+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, and $x_1 \sim \mathcal{N}(0, 1/(1-\alpha^2))$. Then, the ordinary least squares (OLS) estimator*

$$\hat{\alpha}_N = \frac{\sum_{i=1}^{N-1} x_i x_{i+1}}{\sum_{i=1}^{N-1} x_i^2}. \tag{7}$$

is a consistent and asymptotically efficient estimator of α.

Learning goal For a given prompt, the transformer should predict the next data point x_{N+1} given the embedding E_N of data $\{x_i\}_{i=1}^N$. We define $c_N := (1/(N-1)) \sum_{i=1}^{N-1} x_i x_{i+1}$ and $v_N := (1/(N-1)) \sum_{i=1}^{N-1} x_i^2$. Then, the model computes a prediction using the optimal estimator of α if it realizes the following expression

$$\hat{x}_{N+1} = \hat{\alpha}_N x_N = \frac{c_N}{v_N} x_N = \varphi(F(E_N)), \tag{8}$$

where we define $\varphi : \mathbb{R}^4 \to \mathbb{R},\ \ \varphi(\mathbf{u}) = (u_2/u_1) \cdot u_3,\ \ \mathbf{u} = (u_1, u_2, u_3, u_4)^\top$, and the feature map $F : \mathbb{R}^{n \times N} \to \mathbb{R}^4,\ F(E_N) = (v_N, c_N, x_N, 0)^\top$. Note that the function φ produces the desired output $(c_N/v_N) \cdot x_N$ for a vector $(v_N, c_N, x_N, 0)^\top$. Hence, further, we consider the situation where the role of f_{att} is to represent the feature map F, and f_{MLP} to approximate the function φ.

First, we identify an attention-head parametrization that leads to the accumulation of variances, covariances, and the last data point in the output for a given prompt. The proof is in Appendix D.

Lemma 4.6 (Attention can represent the prespecified feature map)**.** *There exists an attention layer $f^*_{\text{att}}(\cdot)$ defined in* (4) *which satisfies*

$$f^*_{\text{att}}(E_N) = (v_N, c_N, x_N, 0)^\top = F(E_N), \quad v_N = \frac{1}{N-1} \sum_{i=1}^{N-1} x_i^2,\ \ c_N = \frac{1}{N-1} \sum_{i=1}^{N-1} x_i x_{i+1} \tag{9}$$

for any E_N, where F is a feature map prespecified in (8)*.*

Specifically, we define attention weight matrices so that

$$W_O^\top W_V = \begin{pmatrix} \frac{1}{N-1} & 0 & 0 & -\frac{1}{N-1} \\ 0 & \frac{1}{N-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad W_K^\top W_Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Further we denote $\mathbf{u}_N = F(E_N) = f^*_{\text{att}}(E_N) \in \mathbb{R}^n$, where f^*_{att} is the attention head parametrized according to Lemma 4.6.

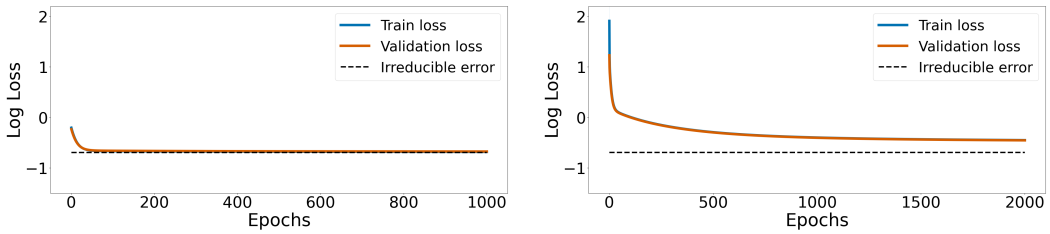


Figure 1: Convergence confirmation in the specified setup. Left: $d = 1$. Right: $d = 5$. For four training runs of the model, we plot the means and standard deviations, which are not well visible due to their small values. Irreducible error is due to the standard Gaussian error.

Step 2. Reduction to the polynomial approximation

In Lemma 4.7, we show that for sufficiently large N and degree p of a polynomial, the optimization problem with such attention definition is equivalent to the optimization problem of approximating the function φ using a polynomial. This problem is, in turn, equivalent to approximating a polynomial with a neural network. The proof of Lemma 4.7 is in Appendix E.

Lemma 4.7 (Equivalence of the optimization problems)**.** *Consider the network defined in Section 3.3 with the attention head parametrized according to Lemma 4.6. Consider the data generated using the AR(1) process specified in Definition 3.1. Re-index the sequence, so that* $x_{-(N+1)} \coloneqq x_1, \ldots, x_0 \coloneqq x_N, x_1 \coloneqq x_{N+1}$ *and denote* $\mathbf{u}_\infty = (v_\infty, c_\infty, x_0, 0)^\top$*, where* $v_\infty = \mathbb{E}[x_i^2]$ *and* $c_\infty = \mathbb{E}[x_i x_{i+1}]$*. Consider the polynomial* $h_p : \mathbb{R}^n \to \mathbb{R}$ *of degree* p *that approximates* φ *defined in* (8)*, so that* $h_p(\mathbf{u}) \xrightarrow[p\to\infty]{L^2} \varphi(\mathbf{u})$*. Then, almost surely,*

$$\operatorname*{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha,\\ x_{-(N+1):1}}} \left[|h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N)|^2\right] \xrightarrow[N,p\to\infty]{} \operatorname*{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha,\\ x_{-\infty:1}}} \left[|x_1 - f_{\text{MLP}}(\mathbf{u}_\infty)|^2\right]$$

with the convergence rate $O(\frac{1}{N} + |h_p(\mathbf{u}) - \varphi(\mathbf{u}))|L^2)$*. Furthermore, the solutions to both optimization problems are unique in the function space.*

Step 3. Approximation and convergence bounds

Now we obtain approximation and convergence results by combining results on the polynomial approximation of a function with those on the approximation of a polynomial by a neural network for our setup. The proof is in Appendix G.

Lemma 4.8 (Neural network approximates polynomials. Simplified)**.** *Fix a degree* p *polynomial* $h_p(\mathbf{u}) : \mathbb{R}^n \to \mathbb{R}$ *and a desired error* ε*. Assume that* $\mathbf{u} \in \mathbb{R}^n$ *is any vector such that* $\mathbf{u} \sim D_{\mathbf{u}}$*,* $\|\mathbf{u}\|^2 \leq C_{\mathbf{u}}$*. Consider a two-layer neural network:* $f_{\text{MLP}}(\mathbf{u}; \boldsymbol{\beta}, \boldsymbol{\gamma}) = (1/M)\sum_{i=1}^M \gamma_i \sigma(\boldsymbol{\beta}_i^\top \mathbf{u})$*. Assume that* $M \geq \binom{n+p}{p}^{\frac{1}{2}} 3^{\frac{p}{2}} C_M/\varepsilon^2$*, where* $C_M \in \mathbb{R}$ *is a constant. Consider the optimization problem* $\operatorname{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M}(1/2)\mathbb{E}_{\mathbf{u}\sim D_{\mathbf{u}}}\left[|h_p(\mathbf{u}) - f_{\text{MLP}}(\mathbf{u})|^2\right]$*, where the gradient descent is applied to the weights* $\boldsymbol{\gamma}$ *of the outer layer, while the other weights are fixed:* $\gamma_i' = \gamma_i - \lambda(\partial\mathcal{L})/(\partial\gamma_i)$*. Then, for a learning rate* $\lambda < M/C_{\mathbf{u}}$*, parameters of the outer layer* $\boldsymbol{\gamma}$ *will converge to* $\boldsymbol{\gamma}^*$ *such that*

$$\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M}\sim\mathcal{N}(0,I_n)\\ \mathbf{u}\sim D_{\mathbf{u}}}} \left[|f_{\text{MLP}}(\mathbf{u}; \boldsymbol{\beta}, \boldsymbol{\gamma}^*) - h_p(\mathbf{u})|\right] \leq 2\varepsilon.$$

5 Experiments

We perform a series of experiments to confirm our results. We verify convergence for 1- and multidimensional inputs in Figure 1. The MLP layer learns the theoretically predicted function from equation (8) under the given assumptions, as shown in Figure 3. We also demonstrate convergence of the attention layer in Figure 2. Additionally, we performed a trial run for an experiment on the synthetic Dyck-3 dataset with 1000 prompts. The experiment settings were similar to the convergence experiments with fixed attention in Figure 1, except for the hyperparameters choice. For a sequence length

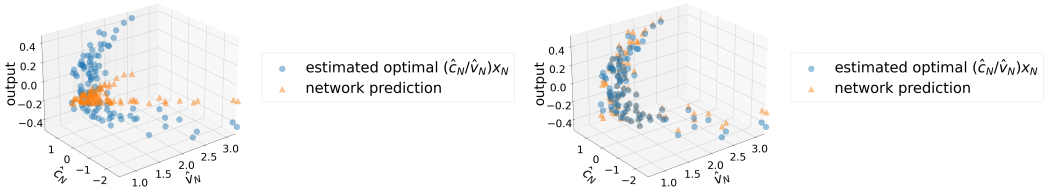


Figure 2: An example of the model converging to the optimal estimator when both non-zero weights of the attention head parametrization from Lemma 4.6 and outer weights of the MLP are not fixed during training. Left: At initialization. Right: After optimization. Note that convergence to the theoretical attention parametrization is not expected due to symmetries introduced by the combination of the attention head and MLP.

of $N = 21$, the model's training loss is about 0.005 for a nesting depth of 1, 0.026 for a depth of 3, and 0.049 for a depth of 10. Test loss is similar to train loss. Quality degrades as nesting depth increases, as expected, since the model is designed for AR(1) process estimation, which captures only dependency on the previous token. Experimental details and additional results are in Appendix I. The code is available at `https://github.com/hanna-tseran/icl_ar_heterogeneous`.

6 Discussion

We study the mechanism of in-context learning in transformers in the case of an AR(1) process. We prove that there exists a two-layer transformer with one fixed linear self-attention head and a nonlinear MLP trained by finite-step gradient descent that realizes the unique function approximating the optimal second-order moment-based AR predictor, which is reused during in-context learning. We explicitly characterize the learned function, show that the induced estimator is consistent and asymptotically efficient, and provide approximation and convergence rate bounds. Our results move beyond i.i.d. regression to autoregressive processes and extend previous work to accommodate heterogeneous second moments of the task distributions. With this, we extend the first-moment-centric understanding of estimators realized by in-context learning to second-order moment-based estimators, such as variances and covariances. Furthermore, the estimator implemented in the suggested setup generalizes to distributions different from those in the training set and thus can handle distribution shifts, providing an explanation for the robustness of the ICL to new tasks. We experimentally confirm convergence for both fixed and non-fixed attention heads, and the form of the learned function under the stated assumptions.

6.1 Limitations

This work focuses on the complexity of estimators learned by transformers under heterogeneous prompt distributions and leverages a combination of a hand-designed fixed attention parametrization to compute the required statistics and a trained MLP. While we confirm convergence of the full network experimentally, theoretical analysis would require consideration of the joint convergence behavior of the attention head and the MLP and is left for future work. Isolated convergence of the attention layer has been previously explored in other works, such as Zhang et al. (2024a), and we expect it to apply after adaptation to the setup differences, such as a noisy AR process and gradient descent. Additionally, since our results reveal a simple architecture that, despite its simplicity, suffices to represent the second-order estimator under heterogeneous prompt distributions, we do not explore multi-head or nonlinear attention. We also point out that the formal theoretical results are provided only for the scalar AR(1) process, while the VAR(1) results in Section 6.2 only discuss the possible approach to the extension.

6.2 Extension to multi-dimensional inputs

We focused on the $d = 1$ case in the formal proofs. Here, we highlight the main steps of the extension to multi-dimensional inputs, $d \geq 2$. We consider a sequence of centered random vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}_{N+1})$, $\mathbf{x}_j \in \mathbb{R}^d, j = 1, \ldots, N+1$ that follows a vector autoregressive model of

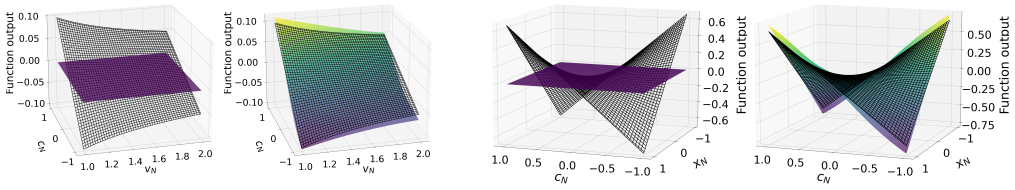


Figure 3: An example of the function represented by the fully-connected layer when $d = 1$. Left: c_N, v_N vary, $x_N = 0.1$. Right: c_N, x_N vary, $v_N = 1.5$. In each subfigure, the left panel shows the state before training, and the right panel shows the state after training. The colored surface shows the function estimated by the network, and the wireframe shows the theoretically predicted function $\varphi((v_N, c_N, x_N, 0)^\top) = (c_N/v_N)x_N$.

order 1, VAR(1):

$$\mathbf{x}_{j+1} = A\mathbf{x}_j + \boldsymbol{\xi}_{j+1},$$

with a coefficient matrix $A \in \mathbb{R}^{d\times d}$ satisfying the stability condition $\rho(A) < 1$, and a centered noise vector $\boldsymbol{\xi}_{j+1} \sim \mathcal{N}(0, I_d)$, independent of $\mathbf{x}_j$. We assume that entries of the matrix A are sampled from $\mathcal{N}(0, \omega^2/d)$, $\omega \in \mathbb{R}$. Thus, the covariance of the datapoints, which depends on the coefficient matrix A, is heterogeneous. We sample $\mathbf{x}_1$ from the VAR(1) stationary distribution $\mathbf{x}_1 \sim \mathcal{N}(0, \Sigma)$ where Σ satisfies the discrete-time Lyapunov equation: $\Sigma = A\Sigma A^\top + I_d$. Hence, the considered VAR(1) process is stationary (Hamilton, 1994). For a given prompt $\mathcal{P}_N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of length N, consider the following input embedding:

$$E_N \coloneqq \begin{pmatrix} X & I_d \\ X' & 0_d \\ 0_{d\times(N-1)} & \operatorname{diag}(\mathbf{x}_N) \\ 0_{d\times(N-1)} & I_d \end{pmatrix} \in \mathbb{R}^{4d\times(N-1+d)},$$

where $0_d \in \mathbb{R}^{d\times d}$ and $0_{d\times(N-1)} \in \mathbb{R}^{d\times(N-1)}$ denote matrices of zeros and $\operatorname{diag}(\mathbf{x})$ denotes a diagonal matrix with the vector $\mathbf{x}$ on its diagonal. Denote $n \coloneqq 4d$, $m \coloneqq N - 1 + d$ so that $E_N \in \mathbb{R}^{n\times m}$.

The network is defined as $f(E_N; \theta) = f_{\text{MLP}}(f_{\text{vec}}(f_{\text{att}}(E_N))) = \frac{1}{M}\sum_{i=1}^{M} \gamma_i \sigma(\boldsymbol{\beta}_i^\top (f_{\text{vec}}(f_{\text{att}}(E_N))))$, where $f_{\text{vec}} : \mathbb{R}^{n\times d} \to \mathbb{R}^{nd}$, $f_{\text{MLP}} : \mathbb{R}^{dn} \to \mathbb{R}^d$, $f_{\text{att}}(E_N) = W_O^\top W_V E_N E_N^\top W_K^\top W_Q E_{N[:,-d]}$, and $f_{\text{att}} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times d}$. Except for the map f_{vec} that vectorizes the attention output, the network has the same architecture as in the 1-dimensional case. Here, $\boldsymbol{\gamma} \in \mathbb{R}^{d\times M}$, $\boldsymbol{\beta} \in \mathbb{R}^{M\times nd}$, $W_O, W_V, W_K, W_Q \in \mathbb{R}^{n\times n}$. We assume the same training setup as for the 1-dimensional model.

The consistent estimator of the VAR(1) process coefficient matrix $A \in \mathbb{R}^{d\times d}$ is

$$\hat{A} = \left(\sum_{i=1}^{N-1} \mathbf{x}_{i+1}\mathbf{x}_i^\top\right)\left(\sum_{i=1}^{N-1} \mathbf{x}_i\mathbf{x}_i^\top\right)^{-1} = X'X^\top\left(XX^\top\right)^{-1},$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}) \in \mathbb{R}^{d\times(N-1)}$, $X' = (\mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathbb{R}^{d\times(N-1)}$. We define $C_N \coloneqq (1/(N-1))X'X^\top$ and $V_N \coloneqq (1/(N-1))XX^\top$. Then, the model computes a prediction using the optimal estimator of A if it realizes the following expression

$$\hat{\mathbf{x}}_{N+1} = \hat{A}\mathbf{x}_N = C_N(V_N)^{-1}\mathbf{x}_N = \varphi(F(E_N)),$$

where $\varphi\colon (U_1, U_2, U_3, U_4)^\top \mapsto U_2U_1^{-1}\operatorname{diag}(U_3)$, $\varphi\colon \mathbb{R}^{n\times d} \to \mathbb{R}^d$, $U_i \in \mathbb{R}^{d\times d}$, and the feature map $F : \mathbb{R}^{n\times m} \to \mathbb{R}^{4d\times d}$, $F(E_N) = (V_N, C_N, \operatorname{diag}(\mathbf{x}_N), 0_{d\times d})^\top$. Consider an approximation of φ via the network f, so that $U_2(U_1)^{-1}\operatorname{diag}(U_3) \approx U_2 h_p(U_1)\operatorname{diag}(U_3)$, where $h_p(U_1)$, $h_p : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ is a degree p polynomial approximating $(U_1)^{-1}$. It can be shown that there exists an attention layer f_{att}^* which satisfies $f_{\text{att}}^*(E_N) = F(E_N)$ for any E_N. We expect the equivalence of the optimization problems to follow the same reasoning as in the 1-dimensional case in Lemma 4.7.

Since $V_N = 1/(N-1)XX^\top$ is symmetric and almost surely positive definite for $N \geq d + 1$, it almost surely does not have zero eigenvalues, and we can use Embree et al. (2025, Proposition

3.2) based on the Crouzeix-Palencia theorem to bound the error of a matrix inverse approximation. Specifically, we conjecture that

$$\|V_N^{-1} - h_p(V_N)\| \leq (1+\sqrt{2}) \max_{\lambda \in W(V_N)} \left| \frac{1}{\lambda} - h_p(\lambda) \right|,$$

where $W(V_N) = \left\{ (v^\top V_N v)/(v^\top v) : 0 \neq v \in \mathbb{R}^d \right\}$ is the numerical range of V_N, which is the set of all Rayleigh quotients that is bounded by the largest and smallest eigenvalues of V_N. To bound $|1/\lambda - h_p(\lambda)|$, we can use the remainder of the Taylor polynomial approximation as in the 1-dimensional case. The result of the network approximation of the polynomial is currently obtained for the output in $\mathbb{R}$, but we expect it to be extendable to $\mathbb{R}^d$ based on Andoni et al. (2014). The rest of the proof is anticipated to apply with minimal changes. Additional notes and details on the extension are in Appendix H.

6.3 Future work

In future work, we would like to extend the analysis to long memory and nonlinear processes and to study how multiple layers and attention heads affect the complexity of the estimator a model can represent during in-context learning. If only the data-generating process changes, for instance, to an AR(p) process, while the model stays the same, the OLS estimator considered in this work might still approximate the optimal solution well, and one would then investigate convergence to this estimator for the new process and properly bound the approximation error. If both the process and the model change, an approach similar to ours would be to derive the optimal estimator for the new process and determine which transformer building blocks can represent its components, for example by using more heads, more layers, or nonlinear attention. We hypothesize that softmax attention may be particularly useful when input data exhibit nonlinear dependencies and that multi-head attention may help learn combinations of estimators or estimators for processes with long memory. Both could represent generally more complex estimators, such as those depending on statistics of order higher than two. We are also interested in connecting our analysis to mechanistic interpretability, particularly the induction head mechanism.

Acknowledgments and Disclosure of Funding

We thank Guillaume Braun and Shogo Nakakita for helpful discussions and for comments on an earlier draft of this manuscript. This study was supported by JSPS KAKENHI (24K02904), JST CREST (JPMJCR21D2), FOREST (JPMJFR216I), ASPIRE (JPMJAP2329), ERATO (JPMJER2601) and BOOST (JPMJBY24A9).

References

Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. (2023). Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650.

Ahuja, K. and Lopez-Paz, D. (2023). A closer look at in-context learning under distribution shifts. In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. (2023). What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

Andoni, A., Panigrahy, R., Valiant, G., and Zhang, L. (2014). Learning polynomials with neural networks. In *International Conference on Machine Learning*, pages 1908–1916. PMLR.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Buet-Golfouse, F. (2015). A multinomial theorem for hermite polynomials and financial applications. *Applied Mathematics*, 6(6):1017–1030.

Casella, G. and Berger, R. (2002). *Statistical Inference*. Duxbury Press.

Chen, S., Sheen, H., Wang, T., and Yang, Z. (2024a). Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality (extended abstract). In *Conference On Learning Theory*.

Chen, S., Sheen, H., Wang, T., and Yang, Z. (2024b). Unveiling induction heads: Provable training dynamics and feature learning in transformers. *Advances in Neural Information Processing Systems*, 37:66479–66567.

Cui, Y., Ren, J., He, P., Liu, H., Tang, J., and Xing, Y. (2025). Superiority of multi-head attention: A theoretical study in shallow transformers in in-context linear regression. In *The 28th International Conference on Artificial Intelligence and Statistics*.

Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., and Wei, F. (2023). Why can GPT learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Dange, A., Raj, V., Netrapalli, P., and Sarawagi, S. (2024). Transformer designs for in-context learning in foundation models for time series forecasting with covariates. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*.

Dunkl, C. F. and Xu, Y. (2014). *Orthogonal polynomials of several variables*. Cambridge University Press.

Edelman, E., Tsilivis, N., Edelman, B., Malach, E., and Goel, S. (2024). The evolution of statistical induction heads: In-context learning markov chains. *Advances in neural information processing systems*, 37:64273–64311.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Embree, M., Henningsen, J. A., Jackson, J., and Morgan, R. B. (2025). Polynomial approximation to the inverse of a large matrix. *SIAM Journal on Scientific Computing*, pages S258–S284.

Fu, D., Chen, T.-Q., Jia, R., and Sharan, V. (2024). Transformers learn to achieve second-order convergence rates for in-context linear regression. *Advances in Neural Information Processing Systems*, 37:98675–98716.

Fuller, W. A. (1995). *Introduction to statistical time series*. John Wiley & Sons.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. (2022). What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598.

Goddard, C., Smith, L. M., Ngampruetikorn, V., and Schwab, D. J. (2025). When can in-context learning generalize out of task distribution? In *Forty-second International Conference on Machine Learning*.

Gong, Z., Hu, X., Tang, H., and Liu, Y. (2025). Towards auto-regressive next-token prediction: In-context learning emerges from generalization. In *The Thirteenth International Conference on Learning Representations*.

González, R. A. and Rojas, C. R. (2020). Finite sample deviation and variance bounds for first order autoregressive processes. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5380–5384. IEEE.

Gradshteyn, I. S. and Ryzhik, I. M. (2007). Table of integrals, series, and products.

Greene, W. H. (2018). *Econometric analysis, Eighth edition*. Pearson.

Guo, T., Hu, W., Mei, S., Wang, H., Xiong, C., Savarese, S., and Bai, Y. (2024). How do transformers learn in-context beyond simple functions? a case study on learning with representations. In *The Twelfth International Conference on Learning Representations*.

Hahn, M. and Goyal, N. (2023). A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.

Huang, R., Liang, Y., and Yang, J. (2024). Non-asymptotic convergence of training transformers for next-token prediction. *Advances in Neural Information Processing Systems*, 37:80634–80673.

Hubinger, E., van Merwijk, C., Mikulik, V., Skalse, J., and Garrabrant, S. (2019). Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*.

Janson, S. (1983). On hypercontractivity for multipliers on orthogonal polynomials. *Arkiv för Matematik*, 21(1):97–110.

Lee, J. M. (2013). *Introduction to Smooth Manifolds*. Springer.

Li, H., Wang, M., Lu, S., Cui, X., and Chen, P.-Y. (2024). How do nonlinear transformers learn and generalize in in-context learning? In *Forty-first International Conference on Machine Learning*.

Liang, H., Balasubramanian, K., and Lai, L. (2025). Transformers handle endogeneity in in-context linear regression. In *The Thirteenth International Conference on Learning Representations*.

Lu, Y. M., Letey, M., Zavatone-Veth, J. A., Maiti, A., and Pehlevan, C. (2025). Asymptotic theory of in-context learning by linear attention. *Proceedings of the National Academy of Sciences*, 122(28):e2502599122.

Mahankali, A. V., Hashimoto, T., and Ma, T. (2024). One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*.

Nichani, E., Damian, A., and Lee, J. D. (2024). How transformers learn causal structure with gradient descent. In *Forty-first International Conference on Machine Learning*.

Oko, K., Song, Y., Suzuki, T., and Wu, D. (2024). Pretrained transformer efficiently learns low-dimensional target functions in-context. *Advances in Neural Information Processing Systems*, 37:77316–77365.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. (2022). In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Rahman, S. (2017). Wiener–hermite polynomial expansion for multivariate gaussian probability measures. *Journal of Mathematical Analysis and Applications*, 454(1):303–334.

Raventós, A., Paul, M., Chen, F., and Ganguli, S. (2023). Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in Neural Information Processing Systems*, 36:14228–14246.

Sander, M. E., Giryes, R., Suzuki, T., Blondel, M., and Peyré, G. (2024). How do transformers perform in-context autoregressive learning? In *Forty-first International Conference on Machine Learning*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. (2023a). Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Von Oswald, J., Schlegel, M., Meulemans, A., Kobayashi, S., Niklasson, E., Zucchet, N., Scherrer, N., Miller, N., Sandler, M., Vladymyrov, M., et al. (2023b). Uncovering mesa-optimization algorithms in transformers. *arXiv preprint arXiv:2309.05858*.

Wang, Q., Wang, Y., Ying, X., and Wang, Y. (2025). Can in-context learning really generalize to out-of-distribution tasks? In *The Thirteenth International Conference on Learning Representations*.

Wu, J., Zou, D., Chen, Z., Braverman, V., Gu, Q., and Bartlett, P. (2024). How many pretraining tasks are needed for in-context learning of linear regression? In *The Twelfth International Conference on Learning Representations*.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2022). An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Yang, T., Huang, Y., Liang, Y., and Chi, Y. (2024). In-context learning with representations: Contextual generalization of trained transformers. *Advances in Neural Information Processing Systems*, 37:85867–85898.

Yüksel, O. K., Even, M., and Flammarion, N. (2025). Long-context linear system identification. In *The Thirteenth International Conference on Learning Representations*.

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

Zhang, R., Wu, J., and Bartlett, P. (2024b). In-context learning of a linear transformer block: Benefits of the mlp component and one-step gd initialization. *Advances in Neural Information Processing Systems*, 37:18310–18361.

Zhang, X., Wang, H., Li, J., Xue, Y., Guan, S., Xu, R., Zou, H., Yu, H., and Cui, P. (2025a). Understanding the generalization of in-context learning in transformers: An empirical study. In *The Thirteenth International Conference on Learning Representations*.

Zhang, Y., Singh, A. K., Latham, P. E., and Saxe, A. M. (2025b). Training dynamics of in-context learning in linear attention. In *Forty-second International Conference on Machine Learning*.

Zheng, C., Huang, W., Wang, R., Wu, G., Zhu, J., and Li, C. (2024). On mesa-optimization in autoregressively trained transformers: Emergence and capability. *Advances in Neural Information Processing Systems*, 37:49081–49129.

Appendix

The appendix is organized as follows.

- Appendix A Additional related work.
- Appendix B Setup illustration.
- Appendix C Proof of the main statement on learning the representation of the optimal estimator with a transformer.
- Appendix D Proof of the existence of the attention representation.
- Appendix E Proof of the optimization problems convergence.
- Appendix F Proofs related to the bounds on the approximation of the optimal estimator with a polynomial.
- Appendix G Proofs related to the approximation of a polynomial with a neural network.
- Appendix H Notes on the extension to multi-dimensional inputs, $d \geq 2$.
- Appendix I Experimental details.

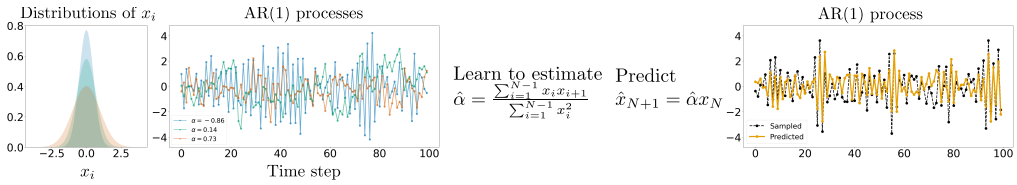
A Additional related work

Investigation of the mechanism of in-context learning has attracted significant attention. In this section, we outline additional related work.

Theoretical ICL studies Several works have considered ICL for problems beyond linear regression or AR process estimation, such as single-index nonlinear target functions (Oko et al., 2024), binary classification (Li et al., 2024), time-series forecasting (Dange et al., 2024), and complex functions with representations Guo et al. (2024). Many works focus on gradient flow for the ICL convergence analysis (Chen et al., 2024a; Zhang et al., 2024a). While most works focus on the analysis of a single linear attention head, Cui et al. (2025) compare single- and multi-head attention for linear regression. System identification for long-context linear systems based on autoregressive processes of rank $p \geq 1$ is theoretically analyzed in (Yüksel et al., 2025). Second-order learning methods are considered in Fu et al. (2024). Several publications investigate whether in-context learning implements standard algorithms, such as gradient descent and ridge regression (Akyürek et al., 2023), as well as implicit Bayesian inference (Xie et al., 2022), and provide information-theoretic bounds Hahn and Goyal (2023). Training single-layer attention models by gradient descent on linear regression tasks has been shown to lead them to perform in-context as a Bayes-optimal ridge regression estimator, both in finite-sample settings (Yang et al., 2024; Wu et al., 2024) and in the asymptotic regime Lu et al. (2025), while other analyses characterized their behavior under gradient flow in terms of kernel regression (Chen et al., 2024a).

Empirical ICL studies Empirical studies (Raventós et al., 2023; Zhang et al., 2025a) explored in-context learning behavior under the distribution shifts. Goddard et al. (2025) empirically investigate the conditions on the pretraining distribution under which ICL emerges and generalizes out-of-distribution, finding that as task diversity increases, transformers undergo a transition from a specialized solution to one that generalizes to the entire task space.Raventós et al. (2023) examine in-context learning on linear regression while varying pretraining task diversity and empirically demonstrate a task diversity threshold for the emergence of ICL. Garg et al. (2022) empirically considered ICL extrapolation beyond the training distribution focusing on the cases when distribution of the prompt inputs or functions are different, $\mathcal{D}^{\text{train}}_{x/\mathcal{F}} \neq \mathcal{D}^{\text{test}}_{x/\mathcal{F}}$, and when distributions of in-context examples and query inputs are different, $\mathcal{D}^{\text{test}}_{\text{query}} \neq \mathcal{D}^{\text{test}}_{x}$. The authors observed that the model error degrades gracefully and remains close to that of the least squares estimator.

ICL generalization Wang et al. (2025) study ICL on out-of-distribution (OOD) tasks not encountered during training and show that the model implements algorithm selection in the case of multitask pre-training. Liang et al. (2025) study a related aspect of endogeneity in which the additive noise is correlated with the explanatory variables. They consider instrumental-variable regression and show that a looped transformer can efficiently implement two-stage least squares regression via a bi-level



(a) During training. (b) During in-context learning.

Figure 4: Illustration of the setup for the 1-dimensional input and 3 different prompts.

gradient descent procedure with an exponential convergence rate. Zhang et al. (2025a) show that transformers can generalize to tasks similar to those seen during training and to familiar tasks after receiving minimal training, but fail on entirely new problems not encountered during training.

Related notions Elhage et al. (2021) introduce the concept of induction heads to mechanistically interpret transformers, attention heads implementing a simple algorithm. Olsson et al. (2022) further investigate this idea, presenting preliminary evidence that induction heads may be the source of in-context learning in transformers. They also observe that first-layer heads are often responsible for copying tokens and supplying information to induction heads. Chen et al. (2024b) prove that gradient flow on a cross-entropy ICL loss with n-gram Markov chain data in a two-attention-layer transformer converges to a model implementing a generalized induction head with contribution of all building blocks, where the first attention layer acts as a copier and the second as a classifier. Our work can be interpreted as introducing a first-layer attention head that goes beyond simple copying and aggregates statistics from an autoregressive process. Von Oswald et al. (2023a) show how trained transformers learn models by gradient descent in their forward pass with linear self-attention layers learning linear models and deeper layers realizing curvature correction. Von Oswald et al. (2023b) provide empirical evidence that training autoregressive transformers induces mesa-optimization, a gradient-based optimization algorithm that runs inside the forward pass and enables in-context learning (Hubinger et al., 2019). Both works connect their findings to the induction heads, noting that in their construction, the first attention layer implements a method similar to copying. Experiments in Von Oswald et al. (2023b) also include autoregressive processes with orthogonal coefficient matrices. In our work, we formalize in-context learning of autoregressive processes in transformers and identify a network parametrization under which we prove convergence to an explicitly defined function that might arise from the mesa-optimization dynamics. Dai et al. (2023) propose explaining language models as meta-optimizers and view in-context learning as implicit fine-tuning, identifying a dual form between transformer attention and gradient descent.

Additional connections of this work to other literature Additionally, results in this paper connect to the concepts of induction heads (Elhage et al., 2021; Olsson et al., 2022; Chen et al., 2024b) and mesa-optimization (Hubinger et al., 2019; Von Oswald et al., 2023b; Dai et al., 2023). Our findings suggest that mesa-optimization might yield an optimal AR process estimator and uncover a form of an induction head that surpasses simple copying and aggregates second moments of an AR process.

B Setup illustration

For additional clarity, we provide an illustration for our setup that is discussed in Section 1 and Section 3 in Figure 4.

C Proof of the main statement on learning the representation of the optimal estimator with a transformer

In this section, we provide and prove the full version of Theorem 4.1.

Theorem C.1 (Learning the representation of the optimal estimator with a transformer)**.** *Consider prompts of length N constructed using the data sampled from the AR(1) model* (1)*: $x_{i+1} = \alpha_\tau x_i + \xi_{i+1}, i = 1, \ldots, N$, where N is sufficiently large with $N \geq 7$. An autoregressive coefficient α is*

sampled from a uniform distribution on the open interval $(-1+\nu, 1+\nu)$, $\alpha \sim \mathcal{U}(-1+\nu, 1-\nu)$, *and a centered noise term* $\xi_{i+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ *independent of* x_i. *We also sample* $x_1 \sim \mathcal{N}(0, 1/(1-\alpha^2))$. *The prompts* $E_N \in \mathbb{R}^{n\times n}$ *are constructed according to* (2).

Consider a two-layer transformer model defined in (3)*:*

$$f(E_N;\theta) = f_{\text{MLP}}(f_{\text{att}}(E_N))) = \sum_{i=1}^{M} \gamma_i \sigma(\boldsymbol{\beta}_i f_{\text{att}}(E_N)),$$

where

$$f_{\text{att}}(E_N;\theta) = W_O^\top W_V E_N E_N^\top W_K^\top W_Q E_{N[:,-1]},$$

is one linear attention head.

Here θ *denotes all network parameters, specifically* $\theta = \{\boldsymbol{\gamma}, \boldsymbol{\beta}, W_O, W_V, W_K, W_Q\}$, $W_O, W_V, W_K, W_Q \in \mathbb{R}^{n\times n}$. *Assume that the inner layer MLP weights* $\boldsymbol{\beta} \in \mathbb{R}^{M\times n}$ *are initialized with* $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n)$. *Assume that for any input embedding* E_N, $\|\mathbf{u}_N\|^2 = \|F(E_N)\|^2 \le C_{\mathbf{u}}$, *and* $\|\mathbf{u}_N\| \neq 0$. *For the activation function* σ, *we assume that* $\sigma(x)$ *is homogeneous of degree* 1, *i.e.,* $\sigma(\alpha x) = \alpha\sigma(x)$, *L-Lipshitz,* $\sigma'(x)$ *is piecewise continuous on* $\mathbb{R}$, $\int_{-\infty}^{\infty} \big(\sigma(x)\big)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x$ *converges,* $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^2] \le 1$ *and* $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^4] \le C_\sigma$, *where* $C_\sigma \in \mathbb{R}$ *is a constant.*

Consider the following minimization problem from (5)

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\text{AR}(1;\alpha)}} \left[|x_{N+1} - f(E_N;\theta)|^2\right] = \min_{\boldsymbol{\gamma}\in\mathbb{R}^M} \mathcal{L}(\theta). \tag{10}$$

We run the gradient descent on the outer layer weights $\boldsymbol{\gamma}$, *while keeping other weights fixed, according to* (6)*:*

$$\gamma_i' = \gamma_i - \lambda \frac{\partial \mathcal{L}}{\partial \gamma_i}.$$

Fix a desired error ε *and assume that* $M \ge \binom{n+p}{p}^{\frac{1}{2}} 3^{\frac{p}{2}} C_M/\varepsilon^2$, *where* $C_M \in \mathbb{R}$ *is a constant. Then, there exist such parametrization of the attention layer* f_{att}^* *and a degree* p *polynomial* $h_p(\mathbf{u}) : \mathbb{R}^n \to \mathbb{R}$ *approximating* φ, *that for a learning rate* $\lambda < M/C_{\mathbf{u}}$, *and sufficiently large* N *and* p, *parameters of the outer layer* $\boldsymbol{\gamma}$ *will converge to the unique solution* $\boldsymbol{\gamma}^*$, *such that the network will approximate the optimal estimator for the AR process in* (7), *and, with probability* $\ge 1 - C_1 - C_2/(N-1)$,

$$\mathbb{E}_{\alpha, x_{1:N+1}, \boldsymbol{\beta}} \left[|x_{N+1} - f_{\text{MLP}}(f_{\text{att}}^*(E_N); \boldsymbol{\gamma}^*, \boldsymbol{\beta})|^2\right] \le \left(\left(\frac{12}{N-6}+1\right)^{\frac{1}{2}} + C\left(\frac{1}{2}\right)^{(p+2)} + 2\varepsilon\right)^2.$$

where C, C_1, C_2 *are constants.*

Proof. Based on Lemma 4.5, the consistent estimator of x_{N+1} is given by

$$\hat{x}_{N+1} = \frac{\sum_{i=1}^{N-1} x_i x_{i+1}}{\sum_{i=1}^{N-1} x_i^2} x_N = \frac{c_N}{v_N} x_N = \varphi(F(E_N)),$$

where $c_N = (1/(N-1))\sum_{i=1}^{N-1} x_i x_{i+1}$, $v_N = (1/(N-1))\sum_{i=1}^{N-1} x_i^2$, $\varphi : \mathbb{R}^4 \to \mathbb{R}$, $\varphi : (u_1, u_2, u_3, u_4)^\top \mapsto \frac{u_1}{u_2} u_3$, and a feature map $F : \mathbb{R}^{(2d+2)\times(N+1)} \to \mathbb{R}^4$, $F(E_N) := (v_N, c_N, x_N, 0)^\top$.

According to Lemma 4.6, there exists a parametrization of the attention layer f_{att}^* such that

$$f_{\text{att}}^*(E_N) = (v_N, c_N, x_N, 0)^\top = F(E_N)$$

for any E_N.

Define $\mathbf{u}_N = (v_N, c_N, x_N, 0)^\top = (\frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2, \frac{1}{N-1}\sum_{i=1}^{N-1} x_i x_{i+1}, x_N, 0)^\top = F(E_N) = f_{\text{att}}^*(E_N)$, where x_i are samples from the stationary AR(1) process as specified in Section 3.1. We

will denote with $D_{\mathbf{u}_N}$ the distribution of $\mathbf{u}_N$ induced by this AR(1) process. Recall that $f = f_{\text{MLP}} \circ f_{\text{att}}$. Hence, we consider the following optimization problem

$$\min_{\boldsymbol{\gamma} \in \mathbb{R}^M} \frac{1}{2} \mathbb{E}_{\mathbf{u}_N \sim D_{\mathbf{u}_N}} \left[|x_{N+1} - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right]. \tag{11}$$

By Lemma 4.7, if a polynomial $h_p : \mathbb{R}^n \to \mathbb{R}$ of degree p approximates φ so that $h_p(\mathbf{u}_N) \xrightarrow[p \to \infty]{L^2} \varphi(\mathbf{u}_N)$, then, almost surely,

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^M} \frac{1}{2} \mathbb{E}_{\mathbf{u}_N \sim D_{\mathbf{u}_N}} \left[|h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N; \boldsymbol{\gamma})|^2 \right] \\ &\qquad \xrightarrow[N,p \to \infty]{} \operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^M} \frac{1}{2} \mathbb{E}_{\mathbf{u}_\infty \sim D_{\mathbf{u}_\infty}} \left[|x_1 - f_{\text{MLP}}(\mathbf{u}_\infty; \boldsymbol{\gamma})|^2 \right] \end{aligned}$$

with the convergence rate $O(\frac{1}{N} + \|h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N))\|_{L^2})$, where we re-index the sequence, so that $x_{-(N+1)} \coloneqq x_1, \ldots, x_0 \coloneqq x_N, x_1 \coloneqq x_{N+1}$ and denote $\mathbf{u}_\infty = (v_\infty, c_\infty, x_0, 0)^\top$, where $v_\infty = \mathbb{E}[x_i^2]$ and $c_\infty = \mathbb{E}[x_i x_{i+1}]$.

Then, for sufficiently large N, p,

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^M} \frac{1}{2} \mathbb{E}_{\mathbf{u}_N \sim D_{\mathbf{u}_N}} \left[|h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right] \\ &\qquad \approx \operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^M} \frac{1}{2} \mathbb{E}_{\mathbf{u}_N \sim D_{\mathbf{u}_N}} \left[|x_{N+1} - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right] \end{aligned} \tag{12}$$

Hence, now we consider the following optimization problem:

$$\operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^M} \frac{1}{2} \mathbb{E}_{\mathbf{u}_N \sim D_{\mathbf{u}_N}} \left[|h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right]. \tag{13}$$

Fix a degree p polynomial $h_p(\mathbf{u}_N) : \mathbb{R}^n \to \mathbb{R}$ with bounded ℓ^2 norm, i.e., $\|h_p\|_{\ell^2} = \sqrt{\sum_{\mathbf{j}} b_{\mathbf{j}}^2} \leq C_h$, where $b_{\mathbf{j}}$ are the polynomial coefficients. By Lemma 4.8, for any desired small error ε, the outer layer parameters $\boldsymbol{\gamma}$ will converge to $\boldsymbol{\gamma}^*$ such that

$$\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M} \sim \mathcal{N}(0, I_n) \\ \mathbf{u}_N \sim D_{\mathbf{u}_N}}} \left[|f_{\text{MLP}}(\mathbf{u}_N; \boldsymbol{\beta}, \boldsymbol{\gamma}^*) - h_p(\mathbf{u}_N)| \right] \leq 2\varepsilon. \tag{14}$$

Furthermore, by Lemma F.4, with probability $\geq 1 - C_1 - C_2/(N-1)$,

$$\mathbb{E}_{\alpha, x_{1:N+1}, \mathbf{u}_N} \left[|x_{N+1} - h_p(\mathbf{u}_N)|^2 \right] \leq 2 \left(\frac{12}{N-6} + C \left(\frac{1}{2} \right)^{2(p+2)} + 1 \right), \tag{15}$$

where C, C_1, C_2 are constants.

Combining (14) with (15), the final approximation bound is

$$\begin{aligned} &\mathbb{E}_{\alpha, x_{1:N+1}, \mathbf{u}_N, \boldsymbol{\beta}_{1:M}} \left[|x_{N+1} - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right] \\ &= \mathbb{E}_{\alpha, x_{1:N+1}, \mathbf{u}_N, \boldsymbol{\beta}_{1:M}} \left[|x_{N+1} - h_p(\mathbf{u}_N) + h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right] \\ &\leq \left(\left(\mathbb{E}_{\alpha, x_{1:N+1}, \mathbf{u}_N} \left[|x_{N+1} - h_p(\mathbf{u}_N)|^2 \right] \right)^{\frac{1}{2}} + \left(\mathbb{E}_{\mathbf{u}_N, \boldsymbol{\beta}_{1:M}} \left[|h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N)|^2 \right] \right)^{\frac{1}{2}} \right)^2 \\ &\leq \left(\left(\frac{12}{N-6} + 1 \right)^{\frac{1}{2}} + C \left(\frac{1}{2} \right)^{(p+2)} + 2\varepsilon \right)^2. \end{aligned}$$

where we used Minkowski inequality.

Furthermore, note that by Lemma 4.7, the solution of both optimization problems in (12) is unique in function space. Hence, the gradient descent will recover the approximation of the optimal estimator from Lemma 4.5. □

D Proof of the existence of the attention representation

Lemma 4.6 (Attention can represent the prespecified feature map)**.** *There exists an attention layer* $f^*_{\text{att}}(\cdot)$ *defined in* (4) *which satisfies*

$$f^*_{\text{att}}(E_N) = (v_N, c_N, x_N, 0)^\top = F(E_N), \quad v_N = \frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2, \;\; c_N = \frac{1}{N-1}\sum_{i=1}^{N-1} x_i x_{i+1} \tag{9}$$

for any E_N*, where* F *is a feature map prespecified in* (8)*.*

Proof. The proof is by construction. We define the parameter matrices in the following way:

$$W_O \coloneqq \begin{pmatrix} \frac{1}{N-1} & 0 & 0 & 0 \\ 0 & \frac{1}{N-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad W_V \coloneqq \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$W_K \coloneqq I_4, \qquad W_Q \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Recall the definition (2) of embedding E_N. Then

$$E_N E_N^\top = \begin{pmatrix} \sum_{i=1}^{N-1} x_i^2 + 1 & \sum_{i=1}^{N-1} x_i x_{i+1} & x_N & 1 \\ \sum_{i=1}^{N-1} x_i x_{i+1} & \sum_{i=2}^{N} x_i^2 & 0 & 0 \\ x_N & 0 & x_N^2 & x_N \\ 1 & 0 & x_N & 1 \end{pmatrix},$$

$$E_N E_N^\top W_K^\top W_Q = \begin{pmatrix} \sum_{i=1}^{N-1} x_i^2 + 1 & 0 & 0 & 0 \\ \sum_{i=1}^{N-1} x_i x_{i+1} & 0 & 0 & 0 \\ x_N & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$W_O^\top W_V E_N E_N^\top W_K^\top W_Q = \begin{pmatrix} \frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2 & 0 & 0 & 0 \\ \frac{1}{N-1}\sum_{i=1}^{N-1} x_i x_{i+1} & 0 & 0 & 0 \\ x_N & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$W_O^\top W_V E_N E_N^\top W_K^\top W_Q E_{N[:,-1]} = \begin{pmatrix} \frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2 & 0 & 0 & 0 \\ \frac{1}{N-1}\sum_{i=1}^{N-1} x_i x_{i+1} & 0 & 0 & 0 \\ x_N & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ x_N \\ 1 \end{pmatrix} = \begin{pmatrix} v_N \\ c_N \\ x_N \\ 0 \end{pmatrix},$$

and we obtain the desired result. □

E Proof of the optimization problems convergence

We use the notion of equivalence of optimization problems informally. We call two problems *equivalent* if from a solution of one, a solution of the other is readily found, and vice versa (Boyd and Vandenberghe, 2004).

We will use $\xrightarrow{L^2}$ to denote L^2 convergence.

Lemma 4.7 (Equivalence of the optimization problems)**.** *Consider the network defined in Section 3.3 with the attention head parametrized according to Lemma 4.6. Consider the data generated using the AR(1) process specified in Definition 3.1. Re-index the sequence, so that* $x_{-(N+1)} \coloneqq x_1, \ldots, x_0 \coloneqq x_N, x_1 \coloneqq x_{N+1}$ *and denote* $\mathbf{u}_\infty = (v_\infty, c_\infty, x_0, 0)^\top$*, where* $v_\infty = \mathbb{E}[x_i^2]$ *and* $c_\infty = \mathbb{E}[x_i x_{i+1}]$*. Consider the polynomial* $h_p : \mathbb{R}^n \to \mathbb{R}$ *of degree* p *that approximates* φ *defined*

in (8)*, so that* $h_p(\mathbf{u}) \xrightarrow[p\to\infty]{L^2} \varphi(\mathbf{u})$*. Then, almost surely,*

$$\operatorname*{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha,\\ x_{-(N+1):1}}} \left[|h_p(\mathbf{u}_N) - f_{\text{MLP}}(\mathbf{u}_N)|^2\right] \xrightarrow[N,p\to\infty]{} \operatorname*{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha,\\ x_{-\infty:1}}} \left[|x_1 - f_{\text{MLP}}(\mathbf{u}_\infty)|^2\right]$$

with the convergence rate $O(\frac{1}{N} + |h_p(\mathbf{u}) - \varphi(\mathbf{u}))|L^2)$*. Furthermore, the solutions to both optimization problems are unique in the function space.*

Proof. **Part I. The optimization problem with the estimator converges to the original problem.**

Consider the following minimization problem

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{-(N+1):1}\sim\text{AR}(1;\alpha)}} \left[|x_1 - f_{\text{MLP}}\left(\mathbf{u}_N;\boldsymbol{\gamma}\right))|^2\right]. \tag{16}$$

Note that since $x_{N+1} = \alpha x_N + \xi_{N+1}$ and $\xi_{i+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), i = 1,\ldots,N$, the optimization problem (16) is equivalent to

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{-N+1:1}\sim\text{AR}(1;\alpha)}} \left[|\alpha x_0 - f_{\text{MLP}}\left(\mathbf{u}_N;\boldsymbol{\gamma}\right))|^2\right]. \tag{17}$$

Hence, further we consider the optimization problem (17).

For sufficiently large N, $\hat{\alpha}_N$ is approximately normally distributed with mean α and variance $(1-\alpha^2)/N$ (Box et al., 2015, Appendix A.6). Then, for a fixed α, $\mathbb{E}[(\hat{\alpha}_N - \alpha)^2] \approx (1-\alpha^2)/N \xrightarrow[N\to\infty]{} 0$. Hence, $\hat{\alpha}_N \xrightarrow[N\to\infty]{L^2} \alpha$.

Note that,

$$\mathbf{u}_N \xrightarrow[N\to\infty]{L^2} \mathbf{u}_\infty = (v_\infty, c_\infty, x_0, 0)^\top.$$

Denote $\boldsymbol{\zeta}_N \coloneqq (\sigma(\boldsymbol{\beta}_1^\top \mathbf{u}_N), \ldots, \sigma(\boldsymbol{\beta}_M^\top \mathbf{u}_N))^\top \in \mathbb{R}^M$. We rewrite the network as

$$f_{\text{MLP}}(\mathbf{u}_N;\boldsymbol{\gamma}) = \sum_{i=1}^M \gamma_i \sigma(\boldsymbol{\beta}_i^\top \mathbf{u}_N) = \boldsymbol{\gamma}^\top \zeta.$$

Similarly, denote $\boldsymbol{\zeta}_\infty \coloneqq (\sigma(\boldsymbol{\beta}_1^\top \mathbf{u}_\infty), \ldots, \sigma(\boldsymbol{\beta}_M^\top \mathbf{u}_\infty))^\top \in \mathbb{R}^M$, and denote vectors $\boldsymbol{\gamma}$ corresponding to $\mathbf{u}_N$ and $\mathbf{u}_\infty$ with $\boldsymbol{\gamma}_N$ and $\boldsymbol{\gamma}_\infty$ respectively.

Hence, we are considering the minimizers of the following two problems:

$$\mathbb{M}_N(\boldsymbol{\gamma}_N) = \frac{1}{2}\mathbb{E}_{\alpha, x_{-(N-1):0}}\left[|\hat{\alpha}_N x_0 - \boldsymbol{\gamma}_N^\top \boldsymbol{\zeta}_N|^2\right],$$
$$\mathbb{M}_\infty(\boldsymbol{\gamma}_\infty) = \frac{1}{2}\mathbb{E}_{\alpha, x_{-\infty:0}}\left[|\alpha x_0 - \boldsymbol{\gamma}_\infty^\top \boldsymbol{\zeta}_\infty|^2\right].$$

Since the matrices $\mathbb{E}[\boldsymbol{\zeta}_N \boldsymbol{\zeta}_N^T]$ and $\mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^T]$ are positive definite, they are invertible. Hence, the solutions are unique and are explicitly given as (Greene, 2018, Chapter 3):

$$\boldsymbol{\gamma}_N^* = \mathbb{E}[\boldsymbol{\zeta}_N \boldsymbol{\zeta}_N^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_N \hat{\alpha}_N x_0]$$

and

$$\boldsymbol{\gamma}_\infty^* = \mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_\infty \alpha x_0].$$

Now we will show that solutions of these two problems converge in the ℓ_2 norm, $\boldsymbol{\gamma}_N^* \xrightarrow[N\to\infty]{} \boldsymbol{\gamma}_\infty^*$. Consider the ℓ_2 norm:

$$\begin{aligned}
\|\boldsymbol{\gamma}_N^* - \boldsymbol{\gamma}_\infty^*\| &= \|\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_\infty\alpha x_0]\| \\
&= \|\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] \\
&\quad + \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\mathbb{E}[\boldsymbol{\zeta}_\infty\alpha x_0]\| \\
&= \| \left(\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1} - \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\right)\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] \\
&\quad + \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\left(\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty\alpha x_0]\right)\| \\
&\le \| \left(\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1} - \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\right)\| \|\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0]\| \\
&\quad + \|\mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\| \| \left(\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty\alpha x_0]\right)\|,
\end{aligned}$$

where we used Cauchy-Schwarz inequality.

Now consider each term separately:

$$\|\mathbb{E}[\boldsymbol{\zeta}_N\hat{\alpha}_N x_0]\| \le \sqrt{\|\mathbb{E}[\boldsymbol{\zeta}_N^2]\|}\sqrt{\|\mathbb{E}[\hat{\alpha}_N^2 x_0^2]\|},$$

where we used Cauchy-Schwarz inequality. Additionally,

$$\begin{aligned}
&\| \left(\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1} - \mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\right)\| \\
&= \|\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1}\| \| \left(\mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top] - \mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]\right)\| \|\mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\|.
\end{aligned}$$

We can bound $\|\mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]^{-1}\| \le G_1$ and $\|\mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top]^{-1}\| \le G_2$.

Consider

$$\begin{aligned}
&\| \left(\mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top] - \mathbb{E}[\boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]\right)\| \\
&= \| \left(\mathbb{E}[\boldsymbol{\zeta}_\infty\boldsymbol{\zeta}_\infty^\top - \boldsymbol{\zeta}_N\boldsymbol{\zeta}_\infty^\top + \boldsymbol{\zeta}_N\boldsymbol{\zeta}_\infty^\top - \boldsymbol{\zeta}_N\boldsymbol{\zeta}_N^\top]\right)\| \\
&= \|\mathbb{E}[\left(\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N\right)\boldsymbol{\zeta}_\infty^\top + \boldsymbol{\zeta}_N\left(\boldsymbol{\zeta}_\infty^\top - \boldsymbol{\zeta}_N^\top\right)]\| \\
&\le \|\mathbb{E}[(\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N)]\|\|\mathbb{E}[\boldsymbol{\zeta}_\infty^\top]\| + \|\mathbb{E}[\boldsymbol{\zeta}_N]\|\|\mathbb{E}[\left(\boldsymbol{\zeta}_\infty^\top - \boldsymbol{\zeta}_N^\top\right)]\| \\
&= \|\mathbb{E}[(\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N)]\| \left(\|\mathbb{E}[\boldsymbol{\zeta}_\infty]\| + \|\mathbb{E}[\boldsymbol{\zeta}_N]\|\right) \\
&\le (g_1 + g_2)\|\mathbb{E}[(\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N)]\| = O(\|E_N - E_\infty\|),
\end{aligned}$$

where we used Cauchy-Schwarz inequality.

Consider

$$\begin{aligned}
&\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N \\
&= \sigma(\boldsymbol{\beta} f_{\text{att}}(E_\infty)) - \sigma(\boldsymbol{\beta} f_{\text{att}}(E_N)) \\
&= \sigma(\boldsymbol{\beta} f_{\text{att}}(E_\infty) - \boldsymbol{\beta} f_{\text{att}}(E_N)) \\
&= \sigma\left(\boldsymbol{\beta}\left(f_{\text{att}}(E_\infty) - f_{\text{att}}(E_N)\right)\right) \\
&= \sigma\left(\boldsymbol{\beta} f_{\text{att}}(E_\infty - E_N)\right).
\end{aligned}$$

Then

$$\mathbb{E}[\sigma\left(\boldsymbol{\beta} f_{\text{att}}(E_\infty - E_N)\right)^2] = \mathbb{E}[\sigma\left(\boldsymbol{\beta}(\mathbf{u}_\infty - \mathbf{u}_N)\right)^2] = \mathbb{E}[\sigma\left(\boldsymbol{\beta}_1(v_\infty - v_N) + \boldsymbol{\beta}_2(c_\infty - c_N)\right)^2].$$

Assuming σ is L-Lipshitz,

$$\begin{aligned}
|\sigma(\boldsymbol{\beta} f_{\text{att}}(E_\infty)) - \sigma(\boldsymbol{\beta} f_{\text{att}}(E_N))| &\le L|\boldsymbol{\beta} f_{\text{att}}(E_\infty) - \boldsymbol{\beta} f_{\text{att}}(E_N)| \\
&= L|\boldsymbol{\beta}\mathbf{u}_\infty - \boldsymbol{\beta}\mathbf{u}_N| = L|\boldsymbol{\beta}||\mathbf{u}_\infty - \mathbf{u}_N|.
\end{aligned}$$

Bound $\|\mathbb{E}[(\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N)]\|$:

$$\begin{aligned}
\mathbb{E}\left[\|\sigma(\boldsymbol{\beta} f_{\text{att}}(E_\infty)) - \sigma(\boldsymbol{\beta} f_{\text{att}}(E_N))\|^2\right] &\le \mathbb{E}\left[L^2\|\boldsymbol{\beta}\|^2\|\mathbf{u}_\infty - \mathbf{u}_N\|^2\right] \le L^2\|\boldsymbol{\beta}\|^2\mathbb{E}\left[\|\mathbf{u}_\infty - \mathbf{u}_N\|^2\right] \\
&= L^2\|\boldsymbol{\beta}\|^2\mathbb{E}\left[\|\mathbf{u}_\infty - \mathbf{u}_N\|^2\right] = O\left(\frac{1}{N}\right).
\end{aligned}$$

Consider the second summand. First, similar to above denote

$$\|\mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1}\| \leq G_2.$$

Finally,

$$\begin{aligned}
&\| \left(\mathbb{E}[\boldsymbol{\zeta}_N \hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty \alpha x_0]\right) \| \\
&= \| \left(\mathbb{E}[\boldsymbol{\zeta}_N \hat{\alpha}_N x_0 - \boldsymbol{\zeta}_N \alpha x_0 + \boldsymbol{\zeta}_N \alpha x_0 - \boldsymbol{\zeta}_\infty \alpha x_0]\right) \| \\
&= \| \left(\mathbb{E}[\boldsymbol{\zeta}_N \left(\hat{\alpha}_N x_0 - \alpha x_0\right) + \left(\boldsymbol{\zeta}_N - \boldsymbol{\zeta}_\infty\right) \alpha x_0]\right) \| \\
&\leq \|\mathbb{E}[\boldsymbol{\zeta}_N]\| \|\mathbb{E}[\left(\hat{\alpha}_N x_0 - \alpha x_0\right]) \| + \|\mathbb{E}[\boldsymbol{\zeta}_N - \boldsymbol{\zeta}_\infty]\| \|\mathbb{E}[\alpha x_0]\| \\
&= \|\mathbb{E}[\boldsymbol{\zeta}_N]\| \|\mathbb{E}[\hat{\alpha}_N x_0 - \alpha x_0]\| \\
&\leq g_1 \|\mathbb{E}[\hat{\alpha}_N x_0 - \alpha x_0]\| \\
&= g_1 \|\mathbb{E}[\left(\hat{\alpha}_N - \alpha\right) x_0]\| \\
&\leq g_1 \sqrt{\mathbb{E}[\left(\hat{\alpha}_N - \alpha\right)^2]} \sqrt{\mathbb{E}[x_0^2]} \\
&= g_1 C \sqrt{\mathbb{E}[\left(\hat{\alpha}_N - \alpha\right)^2]} \\
&= O(\sqrt{\mathbb{E}[\left(\hat{\alpha}_N - \alpha\right)^2]}) \xrightarrow[N \to \infty]{} 0
\end{aligned}$$

by L_2 convergence.

Combining,

$$\begin{aligned}
\|\boldsymbol{\gamma}_N^* - \boldsymbol{\gamma}_\infty^*\| &\leq KV(g_1 + g_2) \|\mathbb{E}[(\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N)]\| + G_2 g_1 C \sqrt{\mathbb{E}[\left(\hat{\alpha}_N - \alpha\right)^2]} \\
&= O(\|\boldsymbol{\zeta}_\infty - \boldsymbol{\zeta}_N\|_{L_2}) + O(\|\hat{\alpha}_N - \alpha\|_{L_2}) = O\left(\frac{1}{N}\right) \xrightarrow[N \to \infty]{} 0
\end{aligned}$$

Part II. The optimization problem with the polynomial converges to the problem with the estimator.

We will use the same argument as in Part I. Now the processes are given by

$$\mathbb{L}_p = \frac{1}{2} \mathbb{E}_{\substack{\alpha \sim \mathcal{U}(-1+\nu, 1-\nu) \\ x_{1:N+1} \sim \text{AR}(1;\alpha)}} \left[|h_p(F(E_N)) - f_{\text{MLP}}(F(E_N))|^2 \right]$$

and

$$\mathbb{L} = \frac{1}{2} \mathbb{E}_{\substack{\alpha \sim \mathcal{U}(-1+\nu, 1-\nu) \\ x_{1:N+1} \sim \text{AR}(1;\alpha)}} \left[|\hat{\alpha}_N x_0 - f_{\text{MLP}}(F(E_N))|^2 \right]$$

defined on the space $\Theta_{\boldsymbol{\gamma}}$.

We assume that $h_p \xrightarrow[p \to \infty]{L_2} \varphi$. Note that the part with f_{MLP} is not affected by p. Note that $\hat{\alpha}_N x_0 = \varphi(\mathbf{u})$.

Assuming that the matrices are invertible, the solutions are explicitly given by (Greene, 2018):

$$\boldsymbol{\gamma}_\alpha^* = \mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1} \mathbb{E}[\boldsymbol{\zeta}_\infty \hat{\alpha}_N x_0]$$

and

$$\boldsymbol{\gamma}_p^* = \mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1} \mathbb{E}[\boldsymbol{\zeta}_\infty h_p(F(E_N))] = \mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1} \mathbb{E}[\boldsymbol{\zeta}_\infty h_p(\mathbf{u})].$$

Then, consider the ℓ_2 norm:

$$\begin{aligned}
\|\boldsymbol{\gamma}_\alpha^* - \boldsymbol{\gamma}_p^*\| &= \|\mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1} \mathbb{E}[\boldsymbol{\zeta}_\infty \hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1} \mathbb{E}[\boldsymbol{\zeta}_\infty h_p(\mathbf{u})]\| \\
&= \|\mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1}\| \|\mathbb{E}[\boldsymbol{\zeta}_\infty \hat{\alpha}_N x_0] - \mathbb{E}[\boldsymbol{\zeta}_\infty h_p(\mathbf{u})]\| \\
&= \|\mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1}\| \|\mathbb{E}[\boldsymbol{\zeta}_\infty \varphi(\mathbf{u})] - \mathbb{E}[\boldsymbol{\zeta}_\infty h_p(\mathbf{u})]\|.
\end{aligned}$$

As in Step 1,

$$\|\mathbb{E}[\boldsymbol{\zeta}_\infty \boldsymbol{\zeta}_\infty^\top]^{-1}\| \leq G_2.$$

For the second argument, we get

$$\begin{aligned}
&\| \left(\mathbb{E}[\boldsymbol{\zeta}_\infty \varphi(\mathbf{u})] - \mathbb{E}[\boldsymbol{\zeta}_\infty h_p(\mathbf{u})]\right) \| \\
&= \| \left(\mathbb{E}[\boldsymbol{\zeta}_\infty (\varphi(\mathbf{u}) - h_p(\mathbf{u}))]\right) \| \\
&\leq \|\mathbb{E}[\boldsymbol{\zeta}_\infty]\| \|\mathbb{E}[(\varphi(\mathbf{u}) - h_p(\mathbf{u}))]\| \\
&\leq g_2 \|\mathbb{E}[(\varphi(\mathbf{u}) - h_p(\mathbf{u}))]\| \\
&\leq g_2 \sqrt{\mathbb{E}[(\varphi(\mathbf{u}) - h_p(\mathbf{u}))^2]} \\
&= O(\sqrt{\mathbb{E}[(\varphi(\mathbf{u}) - h_p(\mathbf{u}))^2]}) \xrightarrow[p\to\infty]{} 0.
\end{aligned}$$

Combining, we get the final result. Furthermore, note that solutions of all considered argmin problems are unique in the function space as discussed above. □

F Proofs related to the bounds on the approximation of the optimal estimator with a polynomial

For reference, we provide the following the result on the Taylor approximation.

Lemma F.1 (Upper bound on the remainder of the Taylor series approximation. Corollary C.16 in Lee (2013))**.** *Suppose $U \subseteq \mathbb{R}^n$ is an open subset, $a \in U$, and $f \in C^{k+1}(U)$ for some $k \geq 0$. If W is a convex subset of U containing a, on which all of the $(k+1)$st partial derivatives of f are bounded in absolute value by a constant M, then for all $x \in W$,*

$$|f(x) - P_k(x)| \leq \frac{n^{k+1} M}{(k+1)!} \|x - a\|^{k+1},$$

where P_k is the kth Taylor polynomial of f at a.

Further we denote L^2 convergence with $\xrightarrow{L^2}$.

Lemma F.2 (Upper bound on the approximation of the function φ by a polynomial h_p)**.** *Consider the function of the form*

$$\varphi : (u_1, u_2, u_3, u_4)^\top \mapsto \frac{u_2}{u_1} u_3, \qquad \varphi : \mathbb{R}^4 \to \mathbb{R}.$$

Then φ can be approximated by a degree p polynomial h_p, $p \geq 1$ so that with probability $\geq 1 - C_1 - C_2/(N-1)$,

$$\mathbb{E}\left[\left(h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\right)^2\right] \leq C \left(\frac{1}{2}\right)^{2(p+2)} \xrightarrow[p\to\infty]{} 0,$$

where C, C_1, C_2 are constants.

Proof. Recall that $\mathbf{u}_N = (v_N, c_N, x_N, 0)^\top = (\frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2, \frac{1}{N-1}\sum_{i=1}^{N-1} x_i x_{i+1}, x_N, 0)^\top$, where x_i are samples from the stationary AR(1) process. We will construct the polynomial in the following way. Consider L^2 norm:

$$\begin{aligned}
\|h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\|_{L^2}^2 &= \mathbb{E}\left[\left(h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\right)^2\right] = \mathbb{E}\left[\left(c_N \cdot x_N \cdot g_{p-2}(v_N) - \frac{c_N x_N}{v_N}\right)^2\right] \\
&= \mathbb{E}\left[c_N^2 \cdot x_N^2 \left(g_{p-2}(v_N) - \frac{1}{v_N}\right)^2\right],
\end{aligned}$$

where $g_q = g_{p-2}$ is the Taylor polynomial of degree $q = p-2$ we will use to approximate $1/v_N$ so that the total degree of the polynomial h_p is p. We will expand around the mean $\mathbb{E}_{\alpha,x_i}[v_N] = v_m$.

Note that

$$v_m = \mathbb{E}[v_N] = \mathbb{E}\left[\frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2\right] = \mathbb{E}\left[x_i^2\right] = \mathbb{E}\left[\frac{1}{1-\alpha^2}\right]$$
$$= \frac{1}{2-2\nu}\int_{-1+\nu}^{1-\nu}\frac{1}{1-\alpha^2}d\alpha = \frac{\ln\left(\frac{2}{\nu}-1\right)}{2-2\nu}. \tag{18}$$

For now, assume that $v_N \in [v_m - \delta, v_m + \delta] = [2v_m/3, 4v_m/3]$. According to Lemma F.1,

$$\left|g_q(v_N) - \frac{1}{v_N}\right| \leq \frac{K\delta^{q+1}}{(q+1)!},$$

where K is an upper bound on the absolute values of all derivatives of $1/v_N$ on the interval $[v_m - \delta, v_m + \delta]$ of order $q+1$. Assume that $\delta < v_m$, so that $v_m - \delta > 0$. Since

$$\left|\frac{d^{q+1}(\frac{1}{v_N})}{v_N^{q+1}}\right| = (q+1)!v_N^{-(q+2)} \leq \frac{(q+1)!}{(v_m-\delta)^{q+2}},$$

we get

$$\left|g_q(v_N) - \frac{1}{v_N}\right| \leq \frac{(q+1)!\delta^{q+1}}{(q+1)!(v_m-\delta)^{q+2}} = \frac{1}{v_m-\delta}\left(\frac{\delta}{v_m-\delta}\right)^{q+1}.$$

Taking $\delta = v_m/3$, we get

$$\left|g_q(v_N) - \frac{1}{v_N}\right| \leq \frac{3}{2v_m}\left(\frac{1}{2}\right)^{q+1}.$$

Then for the full L^2 norm, given that the part linear in the inputs can be approximated perfectly, we get

$$\|h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\|_{L^2}^2 = \mathbb{E}\left[\left(h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\right)^2\right] = \mathbb{E}\left[c_N^2 \cdot x_N^2\left(g_{p-2}(v_N) - \frac{1}{v_N}\right)^2\right]$$
$$\leq \frac{9}{4v_m^2}\left(\frac{1}{2}\right)^{2p+2}\mathbb{E}\left[c_N^2 \cdot x_N^2\right] \leq \frac{9}{4v_m^2}\left(\frac{1}{2}\right)^{2p+2}\sqrt{\mathbb{E}\left[c_N^4\right]\mathbb{E}\left[x_N^4\right]},$$

where we used Cauchy-Schwarz inequality.

Recall that $\alpha \sim U(-1+\nu, 1-\nu)$. Then

$$\mathbb{E}[x_N^4] = 3\mathbb{E}_\alpha\left[\frac{1}{(1-\alpha^2)^2}\right] = 3\int_{-1+\nu}^{1-\nu}\frac{1}{(1-\alpha^2)^2}\frac{d\alpha}{2-2\nu} = \frac{3}{2-2\nu}\int_0^{(1-\nu)^2}\frac{x^{-\frac{1}{2}}}{(1-x)^2}dx$$
$$= 3\,{}_2F_1\left(2, \frac{1}{2}, \frac{3}{2}, (1-\nu)^2\right) < \infty, \tag{19}$$

where ${}_2F_1$ is a Gauss hypergeometric function, see Gradshteyn and Ryzhik (2007) for the integral computation. Similarly,

$$\mathbb{E}[c_N^4] = \mathbb{E}\left[\left(\frac{1}{N-1}\sum_{i=1}^{N-1} x_i x_{i+1}\right)^4\right] = \frac{1}{(N-1)^4}\mathbb{E}\left[\left(\sum_{i=1}^{N-1} x_i x_{i+1}\right)^4\right]$$
$$\leq \frac{1}{N-1}\mathbb{E}\left[\sum_{i=1}^{N-1} x_i^4 x_{i+1}^4\right] \leq \sqrt{\mathbb{E}\left[x_i^8\right]\mathbb{E}\left[x_{i+1}^8\right]}$$
$$= \mathbb{E}\left[x_i^8\right] = 105\,{}_2F_1\left(4, \frac{1}{2}, \frac{3}{2}, (1-\nu)^2\right) < \infty$$

where we repeatedly used Cauchy-Schwarz inequality. Combining,

$$\begin{aligned}
&\|h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\|_{L^2}^2 \\
&\leq \left(\frac{1}{2}\right)^{2p+2} \frac{9\,(1-\nu)}{\left(\ln\left(\frac{2}{\nu}-1\right)\right)^2} \left(3{}_2F_1\left(2, \frac{1}{2}, \frac{3}{2}, (1-\nu)^2\right)\right)^{\frac{1}{2}} \left(105{}_2F_1\left(4, \frac{1}{2}, \frac{3}{2}, (1-\nu)^2\right)\right)^{\frac{1}{2}} \\
&= C\left(\frac{1}{2}\right)^{2p+2} \xrightarrow[p\to\infty]{} 0,
\end{aligned}$$

where C is a constant.

Now we estimate the probability that $v_N \in [v_m - \delta, v_m + \delta]$, where $\delta = v_m/3$. Based on (18),

$$v_m = \mathbb{E}[v_N] = \frac{\ln\left(\frac{2}{\nu}-1\right)}{2-2\nu} = C_1.$$

For the variance of v_N we get

$$\mathrm{Var}[v_N] = \mathrm{Var}\left[\frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2\right] = \mathbb{E}\left[\left(\frac{1}{N-1}\sum_{i=1}^{N-1} x_i^2\right)^2\right] - v_m^2 \leq \frac{1}{(N-1)^2}\mathbb{E}\left[\left(\sum_{i=1}^{N-1} x_i^2\right)^2\right].$$

Consider

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_{i=1}^{N-1} x_i^2\right)^2\right] &= \mathbb{E}\left[\sum_{i=1}^{N-1} x_i^4 + 2\sum_{1\leq i<j\leq N-1} x_i^2 x_j^2\right] \\
&= \sum_{i=1}^{N-1} \mathbb{E}\left[x_i^4\right] + 2\sum_{1\leq i<j\leq N-1} \mathbb{E}\left[x_i^2 x_j^2\right] = [\text{using (19)}] \\
&= 3(N-1){}_2F_1\left(2, \frac{1}{2}, \frac{3}{2}, (1-\nu)^2\right) + 2\sum_{1\leq i<j\leq N-1} \mathbb{E}\left[x_i^2 x_j^2\right].
\end{aligned}$$

Consider

$$2\sum_{1\leq i<j\leq N-1} \mathbb{E}\left[x_i^2 x_j^2\right] = 2\sum_{k=1}^{N-2}\sum_{i=1}^{N-1-k} \mathbb{E}\left[x_i^2 x_{i+k}^2\right] \leq 2\sum_{k=1}^{N-2}(N-1-k)\mathbb{E}\left[x_1^2 x_{1+k}^2\right].$$

Recall that for a stationary AR(1) process observations have Gaussian distribution. Using Isserils's theorem, we get

$$\begin{aligned}
&\mathbb{E}\left[x_1^2 x_{1+k}^2\right] = \mathbb{E}_{\alpha,x_i}\left[x_1 x_1 x_{1+k} x_{1+k}\right] = \mathbb{E}_\alpha\left[\mathbb{E}_{x_i}\left[x_1 x_1 x_{1+k} x_{1+k}\right]\right] \\
&= \mathbb{E}_\alpha\left[\left(\mathbb{E}_{x_i}\left[x_1^2\right]\right)^2 + 2\left(\mathbb{E}_{x_i}\left[x_1 x_{1+k}\right]\right)^2\right] = \mathbb{E}_\alpha\left[\left(\mathbb{E}_{x_i}\left[x_1^2\right]\right)^2\right] + 2\mathbb{E}_\alpha\left[\left(\mathbb{E}_{x_i}\left[x_1 x_{1+k}\right]\right)^2\right] \\
&= \mathbb{E}_\alpha\left[\left(\frac{1}{1-\alpha^2}\right)^2\right] + 2\mathbb{E}_\alpha\left[\left(\frac{\alpha^k}{1-\alpha^2}\right)^2\right].
\end{aligned}$$

It is known that kth autocovariance for the stationary AR(1) process with the standard Gaussian noise is $(\alpha^k/1-\alpha^2)$ (Hamilton, 1994, Chapter 3).

Then,

$$
\begin{aligned}
&2\sum_{k=1}^{N-2}(N-1-k)\mathbb{E}\left[x_1^2x_{1+k}^2\right] = 2\mathbb{E}_\alpha\left[\sum_{k=1}^{N-2}(N-1-k)\frac{1+2\alpha^{2k}}{(1-\alpha^2)^2}\right] \\
&= 2\mathbb{E}_\alpha\left[\frac{1}{(1-\alpha^2)^2}\left(\sum_{k=1}^{N-2}(N-1-k)+2\sum_{k=1}^{N-2}(N-1-k)\alpha^{2k}\right)\right] \\
&= 2\mathbb{E}_\alpha\left[\frac{1}{(1-\alpha^2)^2}\left(\frac{(N-2)(N-1)}{2}+2\sum_{k=1}^{N-2}(N-1-k)\alpha^{2k}\right)\right] \\
&\leq 2\mathbb{E}_\alpha\left[\frac{1}{(1-\alpha^2)^2}\left(\frac{(N-2)(N-1)}{2}+2(N-2)\sum_{k=0}^{\infty}\alpha^{2k}\right)\right] \\
&= 2(N-2)\mathbb{E}_\alpha\left[\frac{1}{(1-\alpha^2)^2}\left(\frac{(N-1)}{2}+\frac{2}{1-\alpha^2}\right)\right] = (N-2)\left((N-1)C_2+C_3\right).
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\mathbb{E}\left[\left(\sum_{i=1}^{N-1}x_i^2\right)^2\right] &\leq 3(N-1){}_2F_1\left(2,\frac{1}{2},\frac{3}{2},(1-\nu)^2\right)+(N-2)\left((N-1)C_2+C_3\right) \\
&\leq (N-1)^2C_2+(N-1)C_3.
\end{aligned}
$$

Then for the variance we get

$$
\mathrm{Var}[v_N] \leq C_1+\frac{1}{N-1}C_2.
$$

According to the Chebyshev inequality

$$
\mathbb{P}\left(|v_N-\mathbb{E}[v_N]| \leq \delta = \frac{\mathbb{E}[v_N]}{3} = k\sqrt{\mathrm{Var}[v_N]}\right) \geq 1-\frac{1}{k^2},
$$

where $k = \mathbb{E}[v_N]/(3\sqrt{\mathrm{Var}[v_N]})$. Then

$$
\frac{1}{k^2} = \frac{9\,\mathrm{Var}[v_N]}{\left(\mathbb{E}[v_N]\right)^2} \leq C_1+\frac{1}{N-1}C_2.
$$

Hence,

$$
\mathbb{P}\left(|v_N-\mathbb{E}[v_N]| \leq \frac{\mathbb{E}[v_N]}{3} = k\sqrt{\mathrm{Var}[v_N]}\right) \geq 1-\frac{1}{k^2} \geq 1-C_1-\frac{1}{N-1}C_2.
$$

□

For reference we cite the following result as a lemma.

Lemma F.3 (Bounds for stable AR(1) processes, Theorem 1 in González and Rojas (2020))**.** *Consider the AR(1) process and the least-squares estimate of α. Assume that $\{x_i\}$ is stationary and $|\alpha|<1$. Then, for all $\varepsilon>0$ and sample size $N \geq 2$,*

$$
\mathbb{P}(\hat{\alpha}_N-\alpha>\varepsilon) \leq \left(\frac{1-\alpha^2}{1-\alpha^2+\varepsilon^2}\right)^{1/4} \times \left(1+\frac{\alpha^2+\varepsilon^2}{2}+\sqrt{\left(1+\frac{\alpha^2+\varepsilon^2}{2}\right)^2-\alpha^2}\right)^{-(N-2)/4}.
$$

Furthermore, for all $N \geq 7$,

$$
\mathbb{E}[(\hat{\alpha}_N-\alpha)^2] \leq \frac{8}{N-6}-\frac{8\alpha^2}{N+2}.
$$

We bound the approximation error of the target function φ approximation with a polynomial.

Lemma F.4 (Approximation bounds)**.** *Assume that N is sufficiently large with $N \geq 7$. Then, for $\mathbf{u}_N = (v_N, c_N, x_N, 0)^\top = (1/(N-1)\sum_{i=1}^{N-1} x_i^2, 1/(N-1)\sum_{i=1}^{N-1} x_i x_{i+1}, x_N, 0)^\top$, where x_i are samples from the stationary AR(1) process as specified in Section 3.1. Then, there exists a degree $p \geq 1$, polynomial h_p, such that with probability $\geq 1 - C_1 - C_2/(N-1)$,*

$$\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[|x_{N+1} - h_p(\mathbf{u}_N)|^2\right] \leq \left(\left(\frac{12}{N-6}+1\right)^{\frac{1}{2}} + C\left(\frac{1}{2}\right)^{(p+2)}\right)^2,$$

where C, C_1, C_2 are constants.

Proof. **Part I. Estimation error.**

We defined $\varphi(\mathbf{u}_N) = \hat{x}_{N+1} = \hat{\alpha}_N x_N$. Consider

$$\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[|x_{N+1} - \hat{\alpha}_N x_N|^2\right] = \mathbb{E}\left[|x_{N+1} - \varphi(\mathbf{u}_N)|^2\right] = \mathbb{E}\left[|\alpha x_N + \xi_{N+1} - \hat{\alpha}_N x_N|^2\right]$$
$$= \mathbb{E}\left[(\alpha - \hat{\alpha}_N)^2 x_N^2\right] + 2\mathbb{E}\left[(\alpha - \hat{\alpha}_N) x_N \xi_{N+1}\right] + \mathbb{E}\left[\xi_{N+1}^2\right] = \mathbb{E}\left[(\alpha - \hat{\alpha}_N)^2\right]\mathbb{E}\left[x_N^2\right] + 1,$$

where we used that for sufficiently large N, $\hat{\alpha}_N$ and x_N are essentially independent for a stationary AR(1) process; see Lemma F.5 for the proof and (Box et al., 2015, Appendix A7.6) for a usage example.

Since we are considering the zero mean AR process, for sufficiently large N, using the formula for the variance of the stationary AR(1) process, we get $\mathbb{E}\left[|x_N|^2\right] = 1/(1-\alpha^2)$ (Hamilton, 1994, Section 3.4). Then, using the law of total expectation and combining it with the bound on the estimation error from Lemma F.3, we obtain

$$\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[|x_{N+1} - \hat{\alpha}_N x_N|^2\right] \leq \mathbb{E}_{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)}\left[\left(\frac{8}{N-6} - \frac{8\alpha^2}{N+2}\right)\cdot\frac{1}{1-\alpha^2}\right] + 1$$
$$\leq \mathbb{E}_{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)}\left[\frac{8}{N-6}\cdot\frac{1}{1-\alpha^2}\right] + 1 = \frac{12}{N-6} + 1,$$

where we also used that $\mathbb{E}_{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)}[\alpha^2] = \frac{1}{3}$. Note that $\mathbb{E}[\frac{1}{1-\alpha^2}]$ is finite since we assumed a uniform distribution on the open interval $\alpha \in (-1+\nu, 1-\nu)$.

Part II. Taylor polynomial approximation error.

By Lemma F.2, φ can be approximated by a degree p polynomial h_p, $p \geq 1$ so that with probability $\geq 1 - C_1 - C_2/(N-1)$,

$$\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[\left(h_p(\mathbf{u}_N) - \varphi(\mathbf{u}_N)\right)^2\right] \leq C\left(\frac{1}{2}\right)^{2(p+2)} \xrightarrow[p\to\infty]{} 0,$$

where C, C_1, C_2 are constants.

Part III. Final result.

Combining, we get that with probability $\geq 1 - C_1 - C_2/(N-1)$,

$$\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[|x_{N+1} - h_p(\mathbf{u}_N)|^2\right]$$
$$\leq \left(\left(\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[|x_{N+1} - \varphi(\mathbf{u}_N)|^2\right]\right)^{\frac{1}{2}} + \left(\mathbb{E}_{\substack{\alpha\sim\mathcal{U}(-1+\nu,1-\nu)\\ x_{1:N+1}\sim\mathrm{AR}(1;\alpha)}}\left[|\varphi(\mathbf{u}_N) - h_p(\mathbf{u}_N)|^2\right]\right)^{\frac{1}{2}}\right)^2$$
$$\leq \left(\left(\frac{12}{N-6}+1\right)^{\frac{1}{2}} + \sqrt{C}\left(\frac{1}{2}\right)^{(p+2)}\right)^2,$$

where we used Minkowski inequality. In the final bound, we absorb the square root into the constant and write C in place of $\sqrt{C}$. □

Below we present the proof of the technical Lemma used in Lemma F.4.

Lemma F.5. *For one prompt we consider a sequence of centered random variables $(x_1, \dots, x_N, x_{N+1}), x_i \in \mathbb{R}, i = 1, \dots, N+1$ that follows an autoregressive model of order 1, AR(1)*

$$x_{i+1} = \alpha x_i + \xi_{i+1}$$

with a coefficient $\alpha \in (-1, 1)$ and a centered noise term $\xi_{i+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ independent of x_i.

The ordinary least squares (OLS) estimator is given by

$$\hat{\alpha}_N = \frac{\sum_{i=1}^{N-1} x_i x_{i+1}}{\sum_{i=1}^{N-1} x_i^2}.$$

Then, $\hat{\alpha}_N$ and x_N are asymptotically independent.

Proof. To show that $\hat{\alpha}_N$ and x_N are asymptotically independent., we show that for a fixed α, $\sqrt{N}(\hat{\alpha}_N - \alpha)$ and x_N are asymptotically independent.

The estimation error is

$$\hat{\alpha}_N - \alpha = \frac{\sum_{i=1}^{N-1} x_i \xi_{i+1}}{\sum_{i=1}^{N-1} x_i^2} = \frac{S_N}{V_N}.$$

Now consider $\sqrt{N}(\hat{\alpha}_N - \alpha) = \frac{S_N/\sqrt{N}}{V_N/N}$.

$V_N/N \xrightarrow[N\to\infty]{} 1/(1-\alpha^2)$ in probability (Fuller, 1995, Theorem 6.3.5). $\sqrt{N}(\hat{\alpha}_N - \alpha) \xrightarrow[N\to\infty]{} \mathcal{N}(0, 1-\alpha^2)$ in distribution (Fuller, 1995, Theorem 8.2.1). Hence,

$$S_N/\sqrt{N} = \sqrt{N}(\hat{\alpha}_N - \alpha) \cdot (V_N/N) \xrightarrow[N\to\infty]{} \mathcal{N}(0, 1/(1-\alpha^2)) \text{ in distribution.} \tag{20}$$

Fix $k \in \mathbb{N}, k > 0$. For S_N we get

$$S_N = \sum_{i=1}^{N-1} x_i \xi_{i+1} = \sum_{i=1}^{N-k-1} x_i \xi_{i+1} + \sum_{i=N-k}^{N-1} x_i \xi_{i+1} = s_1 + s_2,$$

and for x_N,

$$x_N = \alpha^k x_{N-k} + \sum_{i=0}^{k-1} \alpha^i \xi_{N-i} = x_N^{(1)} + x_N^{(2)}.$$

Notice that s_1 and $x_N^{(2)}$ are statistically independent, since s_1 depends on $x_i, \xi_{i+1}, i = 1, \dots, N-k-1$, and $x_N^{(2)}$ depends on $\xi_i, i = N-k+1, \dots N$. Using the Chebyshev inequality, for a fixed k, $s_2/\sqrt{N} = \sum_{i=N-k}^{N-1} x_i \xi_{i+1}/\sqrt{N} \xrightarrow[N\to\infty]{} 0$ in probability, since s_2 is a sum of k terms with zero expectation and bounded variance. Therefore, combining with (20), $s_1/\sqrt{N} \xrightarrow[N\to\infty]{} \mathcal{N}(0, 1/(1-\alpha^2))$ in distribution and is independent from $x_N^{(2)}$.

By stationarity, x_{N-k} follows stationary distribution $\mathcal{N}(0, 1/(1-\alpha^2))$. Then, using $|\alpha| < 1$ and the Chebyshev inequality once again, we get $x_N^{(1)} = \alpha^k x_{N-k} \xrightarrow[k\to\infty]{} 0$ in probability.

Combining, we get that $\sqrt{N}(\hat{\alpha}_N - \alpha)$ is asymptotically independent from x_N. □

G Proofs related to the approximation of a polynomial with a neural network

Further I_n denotes an identity matrix in $\mathbb{R}^{n\times n}$. $\|\|$ stands for the ℓ_2 vector norm unless said otherwise. Denote with $n \coloneqq d+3 = 4$ the input dimension of the MLP layer, which is also the embedding

dimension. We use the standard multi-index notation, specifically $|\mathbf{j}| = \sum_{k=1}^{n} j_k$, $\mathbf{u}^{\mathbf{j}} = \prod_{k=1}^{n} u_k^{j_k}$, $\mathbf{j}! = \prod_{k=1}^{n} j_k!$.

Further discussion uses Hermite polynomials. For the discussion of the multivariate Hermite polynomials see, for instance Dunkl and Xu (2014, Section 5.1.3) or Buet-Golfouse (2015); Rahman (2017). Univariate probabilist's Hermite polynomials $He_j(x), x \in \mathbb{R}, j \in \mathbb{N}$ are defined as

$$He_j(x) = (-1)^j e^{\frac{x^2}{2}} \frac{d^j}{dx^j} e^{-\frac{x^2}{2}}.$$

Multivariate Hermite polynomials are defined as

$$\mathrm{He}_{\mathbf{j}}(\mathbf{x}) = (-1)^{|\mathbf{j}|} e^{\frac{\mathbf{x}^T\mathbf{x}}{2}} \left(\frac{\partial}{\partial \mathbf{x}}\right)^{\mathbf{j}} e^{-\frac{\mathbf{x}^T\mathbf{x}}{2}} = \prod_{k=1}^{n} \mathrm{He}_{j_k}(x_k),$$

where $\mathbf{x} \in \mathbb{R}^n, \mathbf{j} \in \mathbb{N}^n$.

The proof of the following result builds on Andoni et al. (2014, Theorem 3.1).

Lemma G.1 (Representation lemma)**.** *Fix a degree p polynomial $h_p(\mathbf{u}) : \mathbb{R}^n \to \mathbb{R}$ and a desired error ε. Assume that ℓ^2 norm of h_p is bounded, i.e., $\|h_p\|_{\ell^2} = \sqrt{\sum_{\mathbf{j}} b_{\mathbf{j}}^2} \leq C_h$, where $b_{\mathbf{j}}$ are the polynomial coefficients. Assume that $\mathbf{u} \in \mathbb{R}^n$ is any vector such that $\mathbf{u} \sim D_{\mathbf{u}}$, $\|\mathbf{u}\|^2 \leq C_{\mathbf{u}}$, and $\|\mathbf{u}\| \neq 0$. Consider a two-layer neural network:*

$$f_{\text{MLP}}(\mathbf{u}; \boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{1}{M} \sum_{i=1}^{M} \gamma_i \sigma(\boldsymbol{\beta}_i^\top \mathbf{u}),$$

where $\boldsymbol{\beta} \in \mathbb{R}^{M \times n}$, $\boldsymbol{\gamma} \in \mathbb{R}^M$ and σ is the activation function such that $\sigma(x)$ is continuous, homogeneous of degree 1, $\sigma(\alpha x) = \alpha\sigma(x)$, $\sigma'(x)$ is piecewise continuous on $\mathbb{R}$, $\int_{-\infty}^{\infty} \big(\sigma(x)\big)^2 e^{-\frac{x^2}{2}} \,\mathrm{d}x$ converges and $\sqrt{\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^4]} \leq C_\sigma$, where $C_\sigma \in \mathbb{R}$ is a constant. Assume that $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n)$, and $M \geq \binom{n+p}{p}^2 3^p C/\varepsilon^2$, where $C \in \mathbb{R}$ is a constant. Then, there exists $\boldsymbol{\gamma}^*$ such that

$$\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M} \sim \mathcal{N}(0, I_n) \\ \mathbf{u} \sim D_{\mathbf{u}}}} \left[|f_{\text{MLP}}(\mathbf{u}; \boldsymbol{\beta}, \boldsymbol{\gamma}^*) - h_p(\mathbf{u})| \right] \leq \varepsilon.$$

Proof. Define $\hat{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|$. For the i-th row $\boldsymbol{\beta}_i \in \mathbb{R}^n$ of $\boldsymbol{\beta}$, using the assumption that σ is homogeneous of degree 1, we have

$$\sigma(\mathbf{u}; \boldsymbol{\beta}_i) = \sigma\left(\sum_{k=1}^{n} \beta_{i,k} u_k\right) = \|\mathbf{u}\| \sigma\left(\sum_{k=1}^{n} \beta_{i,k} \frac{u_k}{\|\mathbf{u}\|}\right) = \|\mathbf{u}\| \sigma(\hat{\mathbf{u}}; \boldsymbol{\beta}_i).$$

We will use probabilist's Hermite polynomials to show that there is a choice of the outer layer weights such that the neural network approximates the target polynomial.

Since we assumed that $\sigma(x)$ is continuous, $\sigma'(x)$ is piecewise continuous on $\mathbb{R}$ and $\int_{-\infty}^{\infty} \big(\sigma(x)\big)^2 e^{-\frac{x^2}{2}} \,\mathrm{d}x$ converges, we can represent the activation function as

$$\sigma(\hat{\mathbf{u}}; \boldsymbol{\beta}_i) = \sigma\left(\sum_{k=1}^{n} \beta_{i,k} \hat{u}_k\right) = \sum_{j=0}^{\infty} a_j He_j\left(\sum_{k=1}^{n} \beta_{i,k} \hat{u}_k\right).$$

Given that $\|\hat{\mathbf{u}}\| = 1$, and using the multinomial theorem for Hermite polynomials (Buet-Golfouse, 2015, Theorem 4), the activation function can be represented using Hermite polynomials:

$$\begin{aligned} \sigma(\hat{\mathbf{u}}; \boldsymbol{\beta}_i) &= \sum_{j=0}^{\infty} a_j He_j\left(\sum_{k=1}^{n} \beta_{i,k} \hat{u}_k\right) = \sum_{j=0}^{\infty} a_j \sum_{\mathbf{j}, |\mathbf{j}|=j} \frac{j!}{j_1! \cdots j_n!} \prod_{k=1}^{n} (\hat{u}_k)^{j_k} \mathrm{He}_{j_k}(\beta_{i,k}) \\ &= \sum_{\mathbf{j} \in J} a_{\mathbf{j}} \mathrm{He}_{\mathbf{j}}(\boldsymbol{\beta}_i) (\hat{\mathbf{u}})^{\mathbf{j}}, \end{aligned} \tag{21}$$

where all coefficients a_j are incorporated into $a_{\mathbf{j}}$ and J is a collection of all possible $\mathbf{j} \in \mathbb{N}^n$.

Since $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n)$, we can use the orthogonality property of Hermite polynomials, which states

$$\mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[\mathrm{He}_{\mathbf{r}}(\boldsymbol{\beta}_i) \, \mathrm{He}_{\mathbf{j}}(\boldsymbol{\beta}_i) \right] = \delta_{\mathbf{r},\mathbf{j}} \mathbf{r}! . \tag{22}$$

From (21) and (22) follows that

$$\mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[\mathrm{He}_{\mathbf{r}}(\boldsymbol{\beta}_i) \sigma(\hat{\mathbf{u}}; \boldsymbol{\beta}_i) \right] = \mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[\mathrm{He}_{\mathbf{r}}(\boldsymbol{\beta}_i) \sum_{\mathbf{j}} a_{\mathbf{j}} \, \mathrm{He}_{\mathbf{j}}(\boldsymbol{\beta}_i) \hat{\mathbf{u}}^{\mathbf{j}} \right] = a_{\mathbf{r}} \mathbf{r}! \hat{\mathbf{u}}^{\mathbf{r}}. \tag{23}$$

Now we will constructively show how to choose the outer layer network weights to approximate any given polynomial $h_p(\mathbf{u})$ of degree p.

Note that for any $\mathbf{u}$, such that $\|\mathbf{u}\| \neq \mathbf{0}$, we have $\varphi(\mathbf{u}) = \|\mathbf{u}\|(u_1 u_3)/(u_2 \|\mathbf{u}\|) = \|\mathbf{u}\| \varphi(\hat{\mathbf{u}})$. We will define the polynomial as $h_p(\mathbf{u}) = \|\mathbf{u}\| h_p(\hat{\mathbf{u}})$, where h_p approximates φ.

Any multivariate polynomial can be written as a finite sum $h_p(\hat{\mathbf{u}}) = \sum_{|\mathbf{j}| \leq p} b_{\mathbf{j}} \hat{\mathbf{u}}^{\mathbf{j}}$. Define $T_h(\boldsymbol{\beta}_i) \coloneqq \sum_{\mathbf{j}} c_{\mathbf{j}} b_{\mathbf{j}} \, \mathrm{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)$, where $c_{\mathbf{j}} \coloneqq 1/(a_{\mathbf{j}} \mathbf{j}!)$. Using (23), we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right] &= \mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[T_h(\boldsymbol{\beta}_i) \left(\|\mathbf{u}\| \sigma(\hat{\mathbf{u}}; \boldsymbol{\beta}_i) \right) \right] \\
&= \mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[\left(\sum_{\mathbf{j}} c_{\mathbf{j}} b_{\mathbf{j}} \, \mathrm{He}_{\mathbf{j}}(\boldsymbol{\beta}_i) \right) \cdot \left(\|\mathbf{u}\| \sum_{\mathbf{j}} a_{\mathbf{j}} \, \mathrm{He}_{\mathbf{j}}(\boldsymbol{\beta}_i) \hat{\mathbf{u}}^{\mathbf{j}} \right) \right] \\
&= \|\mathbf{u}\| \sum_{\mathbf{j}} b_{\mathbf{j}} \hat{\mathbf{u}}^{\mathbf{j}} = h_p(\mathbf{u}).
\end{aligned} \tag{24}$$

Hence, $T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i)$ is centered around $h_p(\mathbf{u})$ in the function space. Now we will bound the approximation error for the case when $\gamma_i = T_h(\boldsymbol{\beta}_i)$. Denote the error by $\Delta \coloneqq (1/M) \sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) - h_p(\mathbf{u})$. The change in the order of expectations further is justified by Fubini's theorem. We obtain

$$\begin{aligned}
\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M} \sim \mathcal{N}(0,I_n) \\ \mathbf{u} \sim D_{\mathbf{u}}}} \left[|\Delta|^2 \right] &= \mathbb{E}_{\boldsymbol{\beta}_{1:M}, \mathbf{u}} \left[\left| \frac{1}{M} \sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) - h_p(\mathbf{u}) \right|^2 \right] \\
&= \mathbb{E}_{\boldsymbol{\beta}_{1:M}, \mathbf{u}} \left[\left(\frac{1}{M} \sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right)^2 \right] - \mathbb{E}_{\boldsymbol{\beta}_{1:M}, \mathbf{u}} \left[\frac{2}{M} \left(\sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right) h_p(\mathbf{u}) \right] \\
&\qquad + \mathbb{E}_{\mathbf{u}} \left[h_p(\mathbf{u})^2 \right].
\end{aligned}$$

Note that by linearity of expectation, $\mathbb{E}_{\boldsymbol{\beta}_{1:M} \sim \mathcal{N}(0,I_n)} \left[\sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right] = M \mathbb{E}_{\boldsymbol{\beta}_i \sim \mathcal{N}(0,I_n)} \left[T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right] = M h_p(\mathbf{u})$, where we used that all $\boldsymbol{\beta}_i$ are i.i.d. and (24). Hence,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}_{1:M}, \mathbf{u}} \left[|\Delta|^2 \right] &= \mathbb{E}_{\boldsymbol{\beta}_{1:M}, \mathbf{u}} \left[\left(\frac{1}{M} \sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right)^2 \right] - 2 \mathbb{E}_{\mathbf{u}} \left[h_p(\mathbf{u})^2 \right] + \mathbb{E}_{\mathbf{u}} \left[h_p(\mathbf{u})^2 \right] \\
&= \mathbb{E}_{\boldsymbol{\beta}_{1:M}, \mathbf{u}} \left[\left(\frac{1}{M} \sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right)^2 \right] - \mathbb{E}_{\mathbf{u}} \left[h_p(\mathbf{u})^2 \right].
\end{aligned}$$

Note that when $i \neq j$,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}_i, \boldsymbol{\beta}_j, \mathbf{u}} \left[T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) T_h(\boldsymbol{\beta}_j) \sigma(\mathbf{u}; \boldsymbol{\beta}_j) \right] &= \mathbb{E}_{\mathbf{u}} \left[\mathbb{E}_{\boldsymbol{\beta}_i} \left[T_h(\boldsymbol{\beta}_i) \sigma(\mathbf{u}; \boldsymbol{\beta}_i) \right] \mathbb{E}_{\boldsymbol{\beta}_j} \left[T_h(\boldsymbol{\beta}_j) \sigma(\mathbf{u}; \boldsymbol{\beta}_j) \right] \right] \\
&= [\text{using (24)}] = \mathbb{E}_{\mathbf{u}} \left[h_p(\mathbf{u})^2 \right].
\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\beta}_{1:M},\mathbf{u}}\left[|\Delta|^2\right] &= \mathbb{E}_{\boldsymbol{\beta}_{1:M},\mathbf{u}}\left[\left(\frac{1}{M}\sum_{i=1}^{M} T_h(\boldsymbol{\beta}_i)\sigma(\mathbf{u};\boldsymbol{\beta}_i)\right)^2\right] - \mathbb{E}_{\mathbf{u}}\left[\left(h_p(\mathbf{u})\right)^2\right] \\ &= \frac{1}{M^2}\left(M\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[T_h(\boldsymbol{\beta}_i)^2\sigma(\mathbf{u};\boldsymbol{\beta}_i)^2\right] + M(M-1)\mathbb{E}_{\mathbf{u}}\left[h_p(\mathbf{u})^2\right]\right) - \mathbb{E}_{\mathbf{u}}\left[\left(h_p(\mathbf{u})\right)^2\right] \\ &\leq \frac{1}{M}\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[T_h(\boldsymbol{\beta}_i)^2\sigma(\mathbf{u};\boldsymbol{\beta}_i)^2\right].\end{aligned}$$

Using CauchySchwarz inequality,

$$\mathbb{E}_{\boldsymbol{\beta}_i}\left[T_h(\boldsymbol{\beta}_i)^2\mathbb{E}_{\mathbf{u}}\left[\sigma(\mathbf{u};\boldsymbol{\beta}_i)^2\right]\right] = \mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[T_h(\boldsymbol{\beta}_i)^2\sigma(\mathbf{u};\boldsymbol{\beta}_i)^2\right] \leq \sqrt{\mathbb{E}_{\boldsymbol{\beta}_i}\left[T_h(\boldsymbol{\beta}_i)^4\right]}\cdot\sqrt{\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[\sigma(\mathbf{u};\boldsymbol{\beta}_i)^4\right]}.$$

Now we bound $\mathbb{E}_{\boldsymbol{\beta}_i}\left[T_h(\boldsymbol{\beta}_i)^4\right]$. Using Jensen's inequality,

$$\mathbb{E}_{\boldsymbol{\beta}_i}\left[\left(\sum_{\mathbf{j}} c_{\mathbf{j}} b_{\mathbf{j}} \operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)\right)^4\right] \leq D^3\sum_{\mathbf{j}}\mathbb{E}_{\boldsymbol{\beta}_i}\left[c_{\mathbf{j}}^4 b_{\mathbf{j}}^4 \operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^4\right],$$

where $D = \binom{n+p}{p}$ is the total number of elements in the sum.

Defining $a(p) \coloneqq \min_{|\mathbf{j}|=p}|a_{\mathbf{j}}|$, we get

$$\mathbb{E}_{\boldsymbol{\beta}_i}\left[c_{\mathbf{j}}^4 b_{\mathbf{j}}^4 \operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^4\right] = \mathbb{E}_{\boldsymbol{\beta}_i}\left[\frac{b_{\mathbf{j}}^4}{a_{\mathbf{j}}^4(\mathbf{j}!)^4}\operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^4\right] \leq \frac{b_{\mathbf{j}}^4}{a_*^4(\mathbf{j}!)^4}\mathbb{E}_{\boldsymbol{\beta}_i}\left[\operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^4\right],$$

where $a_* \coloneqq \min_{\mathbf{j},\mathbf{j}\in J} a_{\mathbf{j}} \leq \min_{\mathbf{j},|\mathbf{j}|\leq p} a_{\mathbf{j}}$.

Using hypercontractivity of the Hermite polynomials, see e.g., Janson (1983, Theorem 4), we get

$$\mathbb{E}_{\boldsymbol{\beta}_i}\left[\operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^4\right] \leq 3^{|\mathbf{j}|}\left(\mathbb{E}_{\boldsymbol{\beta}_i}\left[\operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^2\right]\right)^2 = 3^{|\mathbf{j}|} \leq 3^p,$$

where we used that by orthogonality of Hermite polynomials $\mathbb{E}_{\boldsymbol{\beta}_i}\left[\operatorname{He}_{\mathbf{j}}(\boldsymbol{\beta}_i)^2\right] = 1$.

Then,

$$\mathbb{E}_{\boldsymbol{\beta}_i}\left[T_h(\boldsymbol{\beta}_i)^4\right] \leq D^3 3^p \frac{1}{a_*^4}\sum_{|\mathbf{j}|\leq p}\frac{b_{\mathbf{j}}^4}{(\mathbf{j}!)^4} \leq D^3 3^p \frac{1}{a_*^4}\sum_{|\mathbf{j}|\leq p} b_{\mathbf{j}}^4 \leq \frac{D^3 3^p C_h^4}{a_*^4},$$

where we used the assumption that $\|h_p\|_{\ell^2} \leq C_h$ and monotonicity of ℓ_p norms, according to which $\|\mathbf{x}\|_4^4 \leq \|\mathbf{x}\|_2^4 \leq 1$. Hence,

$$\begin{aligned}\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M}\sim\mathcal{N}(0,I_n)\\ \mathbf{u}\sim D_{\mathbf{u}}}}\left[\|\Delta\|^2\right] &\leq \frac{1}{M}\sqrt{\mathbb{E}_{\boldsymbol{\beta}_i}\left[T_h(\boldsymbol{\beta}_i)^4\right]}\cdot\sqrt{\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[\sigma(\mathbf{u};\boldsymbol{\beta}_i)^4\right]} \\ &\leq \frac{1}{M}\frac{D^{\frac{3}{2}}3^{\frac{p}{2}}C_h^2}{a_*^2}\cdot\sqrt{\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[\sigma(\mathbf{u};\boldsymbol{\beta}_i)^4\right]}.\end{aligned}$$

Since $\|\mathbf{u}\| = 1$ and $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n)$, we get $\boldsymbol{\beta}_i^\top\mathbf{u}/\|\mathbf{u}\| \sim \mathcal{N}(0,1)$. Using the assumptions $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^2] \leq 1$, $\|\mathbf{u}\|^2 \leq C_{\mathbf{u}}C_\sigma$ and homogenuity of σ, we obtain $\sqrt{\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[\sigma(\mathbf{u};\boldsymbol{\beta}_i)^4\right]} \leq C_{\mathbf{u}}\sqrt{3}$. Thus,

$$\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M}\sim\mathcal{N}(0,I_n)\\ \mathbf{u}\sim D_{\mathbf{u}}}}\left[\|\Delta\|^2\right] \leq \frac{1}{M}\frac{D^{\frac{3}{2}}3^{\frac{p}{2}}C_h^2 C_{\mathbf{u}}C_\sigma}{a_*^2}.$$

Therefore, if the width of the network $M \geq D^{\frac{3}{2}}3^{\frac{p}{2}}C/\varepsilon^2$, where $C \in \mathbb{R}$ is a constant, and the outer layer weights are chosen as $\gamma_i = T_h(\boldsymbol{\beta}_i)$, then $\mathbb{E}_{\boldsymbol{\beta}_i,\mathbf{u}}\left[\|\Delta\|\right] \leq \varepsilon$. □

Now we provide and prove the full version of Lemma 4.8.

Lemma G.2 (Neural network approximates polynomials)**.** *Fix a degree p polynomial $h_p(\mathbf{u}) : \mathbb{R}^n \to \mathbb{R}$. Assume that ℓ^2 norm of h_p is bounded, i.e., $\|h_p\|_{\ell^2} = \sqrt{\sum_{\mathbf{j}} b_{\mathbf{j}}^2} \leq C_h$, where $b_{\mathbf{j}}$ are the polynomial coefficients. Fix a desired error ε. Assume that $\mathbf{u} \in \mathbb{R}^n$ is any vector such that $\mathbf{u} \sim D_{\mathbf{u}}$, $\|\mathbf{u}\|^2 \leq C_{\mathbf{u}}$, and $\|\mathbf{u}\| \neq 0$. Consider a two-layer neural network: $f_{\text{MLP}}(\mathbf{u}; \boldsymbol{\beta}, \boldsymbol{\gamma}) = (1/M)\sum_{i=1}^M \gamma_i \sigma(\boldsymbol{\beta}_i^\top \mathbf{u})$, where $\boldsymbol{\beta} \in \mathbb{R}^{M\times n}$, $\boldsymbol{\gamma} \in \mathbb{R}^M$ and the activation function $\sigma(x)$ is homogeneous of degree 1, i.e., $\sigma(\alpha x) = \alpha\sigma(x)$, continuous, $\sigma'(x)$ is piecewise continuous on $\mathbb{R}$, $\int_{-\infty}^{\infty} \big(\sigma(x)\big)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x$ converges, $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^2] \leq 1$ and $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^4] \leq C_\sigma$, where $C_\sigma \in \mathbb{R}$ is a constant. Assume that $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n)$ and $M \geq \binom{n+p}{p}^{\frac{1}{2}} 3^{\frac{p}{2}} C_M/\varepsilon^2$, where $C_M \in \mathbb{R}$ is a constant.*

Consider the following optimization problem: $\operatorname{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M} (1/2)\mathbb{E}_{\mathbf{u}\sim D_{\mathbf{u}}}\left[|h_p(\mathbf{u}) - f_{\text{MLP}}(\mathbf{u})|^2\right] = \operatorname{argmin}_{\boldsymbol{\gamma}\in\mathbb{R}^M} \mathcal{L}(\boldsymbol{\gamma})$*, where the gradient descent is applied to the weights $\boldsymbol{\gamma}$ of the outer layer, while the other weights are fixed: $\gamma_i' = \gamma_i - \lambda(\partial\mathcal{L})/(\partial\gamma_i)$.*

Then, for a learning rate $\lambda < M/C_{\mathbf{u}}$, parameters of the outer layer $\boldsymbol{\gamma}$ will converge to $\boldsymbol{\gamma}^*$ such that

$$\mathbb{E}_{\substack{\boldsymbol{\beta}_{1:M}\sim\mathcal{N}(0,I_n)\\ \mathbf{u}\sim D_{\mathbf{u}}}} \left[|f_{\text{MLP}}(\mathbf{u}; \boldsymbol{\beta}, \boldsymbol{\gamma}^*) - h_p(\mathbf{u})|\right] \leq 2\varepsilon.$$

Proof. The proof idea stems from Andoni et al. (2014, Theorem 4.1.). In this proof we work in the Hilbert space $L^2(\mathbb{R}^n)$ and think of all functions in the base of monomials $\mathbf{u}^{\mathbf{j}}$. We define the corresponding inner product for $f, g \in L^2(\mathbb{R}^n)$ as $\langle f, g\rangle_{D_{\mathbf{u}}} = \mathbb{E}_{\mathbf{u}\sim D_{\mathbf{u}}}[f(\mathbf{u})g(\mathbf{u})]$ and the norm is $\|f\| = \sqrt{f^* f} = \sqrt{\mathbb{E}_{\mathbf{u}\sim D_{\mathbf{u}}}[f^2(\mathbf{u})]}$. We use superscript star to denote the linear functional associated with the function, for instance $f^* g = \langle f, g\rangle_{D_{\mathbf{u}}}$.

We denote with $p_i \coloneqq \sigma(\boldsymbol{\beta}_i^\top \mathbf{u})$, a vector in the Hilbert space $L^2(\mathbb{R}^n)$. Then, $f_{\text{MLP}}(\mathbf{u}) = (1/M)\sum_{i=1}^M \gamma_i p_i$. Denote the error function with $e = h_p - f_{\text{MLP}}$. The gradient descent minimizes $\mathcal{L} = (1/2)\mathbb{E}_{\mathbf{u}}[e^2] = (1/2)\langle e, e\rangle_{D_{\mathbf{u}}} = (1/2)e^* e$. Taking derivative of the loss with respect to the outer layer weights γ_i, we get

$$\frac{\partial\mathcal{L}}{\partial\gamma_i} = -\frac{1}{M}\langle e, p_i\rangle_{D_{\mathbf{u}}} = -\frac{1}{M} p_i^* e.$$

The gradient descent update is $\hat{\gamma}_i = \gamma_i - \lambda\frac{\partial\mathcal{L}}{\partial\gamma_i}$, and the updated function after one step of the gradient descent is

$$\hat{f}_{\text{MLP}} = \frac{1}{M}\sum_{i=1}^M p_i\left(\gamma_i - \lambda\frac{\partial\mathcal{L}}{\partial\gamma_i}\right) = \frac{1}{M}\sum_{i=1}^M p_i\left(\gamma_i + \frac{\lambda}{M} p_i^* e\right) = f_{\text{MLP}} + \frac{\lambda}{M^2}\sum_{i=1}^M p_i p_i^* e.$$

The new error function is

$$\hat{e} = h_p - \hat{f}_{\text{MLP}} = h_p - f_{\text{MLP}} - \frac{\lambda}{M^2}\sum_{i=1}^M p_i p_i^* e = \left(I - \frac{\lambda}{M^2}\sum_{i=1}^M p_i p_i^*\right) e.$$

Let P be the matrix $P \coloneqq (p_1, \ldots, p_M)$ which columns are p_i-s. Then, $\sum_{i=1}^M p_i p_i^* = PP^*$. After t iterations of the gradient descent the error is

$$e^{(t)} = \left(I - \frac{\lambda}{M^2} PP^*\right)^t e^{(0)},$$

where I is an identity operator, $e^{(0)} = h_p(\mathbf{u}) - f_{\text{MLP}}^{(0)}(\mathbf{u}) = h_p(\mathbf{u}) - \frac{1}{M} P\boldsymbol{\gamma}^{(0)}$ is the starting error function, and $\boldsymbol{\gamma}^{(0)}$ are the inner layer weights at initialization.

According to Lemma G.1, $e^{(0)}$ can be represented as $e^{(0)} = x^{(0)} + \Delta^{(0)}$, where $x^{(0)} = \frac{1}{M}\sum_{i=1}^M \eta_i^{(0)} p_i = \frac{1}{M} P\boldsymbol{\eta}^{(0)}$ and $\|\Delta^{(0)}\|_{\beta_{1:M}\sim\mathcal{N}(0,I_n),\mathbf{u}\sim D_{\mathbf{u}}} \leq \varepsilon$. Then the error at the step t can be rewritten as

$$e^{(t)} = \left(I - \frac{\lambda}{M^2} PP^*\right)^t (x^{(0)} + \Delta^{(0)}) = \left(I - \frac{\lambda}{M^2} PP^*\right)^t \frac{1}{M} P\boldsymbol{\eta}^{(0)} + \left(I - \frac{\lambda}{M^2} PP^*\right)^t \Delta^{(0)}.$$

Note that $\left(I - \frac{\lambda}{M^2} PP^*\right) \frac{1}{M} P\boldsymbol{\eta}^{(0)} = \frac{1}{M} P\boldsymbol{\eta}^{(0)} - \frac{\lambda}{M^3} PP^* P\boldsymbol{\eta}^{(0)} = \frac{1}{M} P\left(I - \frac{\lambda}{M^2} P^*P\right)\boldsymbol{\eta}^{(0)}$. Hence, $\left(I - \frac{\lambda}{M^2} PP^*\right)^t \frac{1}{M} P\boldsymbol{\eta}^{(0)} = \frac{1}{M} P\left(I - \frac{\lambda}{M^2} P^*P\right)^t \boldsymbol{\eta}^{(0)}$.

Define $\boldsymbol{\eta}^{(t)} \coloneqq \left(I - \frac{\lambda}{M^2} P^*P\right)^t \boldsymbol{\eta}^{(0)}$. Then,

$$e^{(t)} = \frac{1}{M} P\boldsymbol{\eta}^{(t)} + \Delta^{(t)}, \qquad \text{where } \Delta^{(t)} = \left(I - \frac{\lambda}{M^2} PP^*\right)^t \Delta^{(0)}. \tag{25}$$

Then for the ℓ_2 norm of $\boldsymbol{\eta}^{(t+1)} = \left(I - \frac{\lambda}{M^2} P^*P\right)\boldsymbol{\eta}^{(t)}$ we get

$$\begin{aligned}
\|\boldsymbol{\eta}^{(t+1)}\|^2 = \boldsymbol{\eta}^{(t+1)*}\boldsymbol{\eta}^{(t+1)} &= \left(\left(I - \frac{\lambda}{M^2} PP^*\right)\boldsymbol{\eta}^{(t)}\right)^* \left(I - \frac{\lambda}{M^2} P^*P\right)\boldsymbol{\eta}^{(t)} \\
&= \boldsymbol{\eta}^{(t)*}\left(I - \frac{\lambda}{M^2} P^*P\right)^* \left(I - \frac{\lambda}{M^2} P^*P\right)\boldsymbol{\eta}^{(t)} \\
&= \|\boldsymbol{\eta}^{(t)}\|^2 - \frac{2\lambda}{M^2}\|P\boldsymbol{\eta}^{(t)}\|^2 + \frac{\lambda^2}{M^4}\|P^*P\boldsymbol{\eta}^{(t)}\|^2,
\end{aligned}$$

where we used that P^*P is self-adjoint.

Note that $\|P\| \le \|P\|_F = \sqrt{\sum_{i=1}^M \|p_i\|^2}$, where $\|\cdot\|_F$ denotes Frobenius norm. Since $\boldsymbol{\beta}_i \sim \mathcal{N}(0, I_n)$, we get $\boldsymbol{\beta}_i^\top \mathbf{u}/\|\mathbf{u}\| \sim \mathcal{N}(0,1)$. Using the assumptions $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\sigma(x)^2] \le 1$, $\|\mathbf{u}\|^2 \le C_\mathbf{u}$ and homogeneity of σ, we obtain $\|p_i\|^2 = \mathbb{E}_{\boldsymbol{\beta}_i^\top \mathbf{u}\sim\mathcal{N}(0,1)}[\sigma(\boldsymbol{\beta}_i^\top \mathbf{u})^2] \le C_\mathbf{u}$. Hence, $\|P\| = \|P^*\| \le C_\mathbf{u}\sqrt{M}$. Then, given the assumption on the learning rate $\lambda \le M/C_\mathbf{u}$, we get $(\lambda/M^2)\|P\|^2 \le 1$. For now suppose that $\|P\boldsymbol{\eta}^{(t)}\| \ge M\varepsilon$. Therefore,

$$\begin{aligned}
\|\boldsymbol{\eta}^{(t+1)}\|^2 &\le \|\boldsymbol{\eta}^{(t)}\|^2 - \frac{2\lambda}{M^2}\|P\boldsymbol{\eta}^{(t)}\|^2 + \frac{\lambda^2}{M^4}\|P^*\|^2\|P\boldsymbol{\eta}^{(t)}\|^2 \\
&\le \|\boldsymbol{\eta}^{(t)}\|^2 - \frac{\lambda}{M^2}\|P\boldsymbol{\eta}^{(t)}\|^2 \le \|\boldsymbol{\eta}^{(t)}\|^2 - \lambda\varepsilon^2.
\end{aligned}$$

Now we will show that $\|P\boldsymbol{\eta}^{(t)}\| \le M\varepsilon$. Suppose $\|P\boldsymbol{\eta}^{(t)}\| \ge M\varepsilon$. Then, $\|\boldsymbol{\eta}^{(t+1)}\|^2 \le \|\boldsymbol{\eta}^{(t)}\|^2 - \lambda\varepsilon^2 \le \|\boldsymbol{\eta}^{(0)}\|^2 - \lambda(t+1)\varepsilon^2$. Hence $\|\boldsymbol{\eta}^{(0)}\|^2 - \lambda(t+1)\varepsilon^2 \ge \|\boldsymbol{\eta}^{(t)}\|^2 - \lambda\varepsilon^2 \ge -\lambda\varepsilon^2$ and we get that we can have at most $t \le \|\boldsymbol{\eta}^{(0)}\|^2/(\lambda\varepsilon^2)$ steps. Then after $O(\|\boldsymbol{\eta}^{(0)}\|^2/(\lambda\varepsilon^2))$ steps $\|P\boldsymbol{\eta}^{(t)}\| \le M\varepsilon$ and using (25),

$$\|e^{(t)}\| \le \frac{1}{M}\|P\boldsymbol{\eta}^{(t)}\| + \|\Delta^{(t)}\| \le 2\varepsilon,$$

where we used that $\|\left(I - \frac{\lambda}{M^2} PP^*\right)\| \le 1$ is a contraction operator. To see this, observe that $\|\left(I - \frac{\lambda}{M^2} PP^*\right) f\|^2 = f^*\left(I - \frac{\lambda}{M^2} PP^*\right)^*\left(I - \frac{\lambda}{M^2} PP^*\right) f = \|f\|^2 - \frac{2\lambda}{M^2}\|P^*f\|^2 + \frac{\lambda^2}{M^4}\|PP^*f\|^2 \le \|f\|^2 - \frac{2\lambda}{M^2}\|P^*f\|^2 + \frac{\lambda^2}{M^4}\|P\|^2\|P^*f\|^2 \le \|f\|^2$, where we used $\frac{\lambda}{M^2}\|P\|^2 \le 1$. □

H Notes on the extension to multi-dimensional inputs, $d \ge 2$.

In this section, we discuss the approach to extending our analysis to multi-dimensional inputs, $d \ge 2$. We focus on the $d = 1$ case in the formal proofs and highlight the main steps of the extension here. Further, we denote by $\mathrm{diag}(\mathbf{x})$ the diagonal matrix with the vector $\mathbf{x}$ on its diagonal. We use $\otimes$ to denote the Kronecker product.

H.1 Preliminaries

Definition H.1 (VAR(1) process)**.** We consider a sequence of centered random vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}_{N+1})$, $\mathbf{x}_j \in \mathbb{R}^d, j = 1, \ldots, N+1$ that follows a vector autoregressive model of order 1, VAR(1): $\mathbf{x}_{j+1} = A\mathbf{x}_j + \boldsymbol{\xi}_{j+1}$, with a coefficient matrix $A \in \mathbb{R}^{d\times d}$ satisfying the stability condition $\rho(A) < 1$, and a centered noise vector $\boldsymbol{\xi}_{j+1} \sim \mathcal{N}(0, I_d)$, independent of $\mathbf{x}_j$.

We assume that entries of the matrix A are sampled from $\mathcal{N}(0, \omega^2/d),\ \omega \in \mathbb{R}$ and that the stability condition $\rho(A) < 1$ is satisfied. Thus, the covariance of the datapoints, which depends on the coefficient matrix A, is heterogeneous. We sample the first data point $\mathbf{x}_1$ from the VAR(1) stationary distribution $\mathbf{x}_1 \sim \mathcal{N}(0, \Sigma)$ where Σ satisfies the discrete-time Lyapunov equation: $\Sigma = A\Sigma A^\top + I_d$. Hence, the considered VAR(1) process is stationary, see Hamilton (1994).

For a given prompt $\mathcal{P}_N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of length N, consider the input embedding formed by stacking the input data consisting of the input-output pairs and adding $2d$ additional rows and d additional columns, which will be useful in the future computations. We get

$$E_N \coloneqq \begin{pmatrix} X & I_d \\ X' & 0_d \\ 0_{d\times(N-1)} & \operatorname{diag}(\mathbf{x}_N) \\ 0_{d\times(N-1)} & I_d \end{pmatrix} \in \mathbb{R}^{4d\times(N-1+d)}, \tag{26}$$

where $0_d \in \mathbb{R}^{d\times d}$ and $0_{d\times(N-1)} \in \mathbb{R}^{d\times(N-1)}$ denote matrices of zeros. Denote $n \coloneqq 4d$, $m \coloneqq N-1+d$ so that $E_N \in \mathbb{R}^{n\times m}$.

The network is defined as $f(E_N;\theta) = f_{\text{MLP}}(f_{\text{vec}}(f_{\text{att}}(E_N))) = \frac{1}{M}\sum_{i=1}^{M}\gamma_i\sigma(\boldsymbol{\beta}_i^\top(f_{\text{vec}}(f_{\text{att}}(E_N))))$, where $f_{\text{vec}} : \mathbb{R}^{n\times d} \to \mathbb{R}^{nd}$, $f_{\text{MLP}} : \mathbb{R}^{dn} \to \mathbb{R}^d$, $f_{\text{att}}(E_N) = W_O^\top W_V E_N E_N^\top W_K^\top W_Q E_{N[:,-d]}$, and $f_{\text{att}} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times d}$. Except for the map f_{vec} that vectorizes the attention output, the network has the same architecture as in the 1-dimensional case. Here, $\boldsymbol{\gamma} \in \mathbb{R}^{d\times M}$, $\boldsymbol{\beta} \in \mathbb{R}^{M\times nd}$, $W_O, W_V, W_K, W_Q \in \mathbb{R}^{n\times n}$. We assume the same training setup as for the 1-dimensional model.

H.2 Convergence to the optimal estimator

Lemma H.2 (Specification of the optimal estimator)**.** *Consider a centered VAR(1) model $\mathbf{x}_{j+1} = A\mathbf{x}_j + \boldsymbol{\xi}_{j+1}$, $\mathbf{x}_j \in \mathbb{R}^d, j = 1, \ldots, N+1$ where for the autoregressive coefficient matrix A, $\rho(A) < 1$. Then, the consistent estimator of the VAR(1) process coefficient matrix $A \in \mathbb{R}^{d\times d}$ is*

$$\hat{A} = \left(\sum_{i=1}^{N-1}\mathbf{x}_{i+1}\mathbf{x}_i^\top\right)\left(\sum_{i=1}^{N-1}\mathbf{x}_i\mathbf{x}_i^\top\right)^{-1} = X'X^\top\left(XX^\top\right)^{-1},$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}) \in \mathbb{R}^{d\times(N-1)}$, $X' = (\mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathbb{R}^{d\times(N-1)}$.

For the given prompt, the transformer should predict data point $\mathbf{x}_{N+1}$ given the embedding E_N of data $\{\mathbf{x}_j\}_{j=1}^N$. We define $C_N \coloneqq (1/(N-1))X'X^\top$ and $V_N \coloneqq (1/(N-1))XX^\top$. Then, the model computes a prediction using the optimal estimator of A if it realizes the following expression

$$\hat{\mathbf{x}}_{N+1} = \hat{A}\mathbf{x}_N = C_N(V_N)^{-1}\mathbf{x}_N = \varphi(F(E_N)),$$

where we define $\varphi\colon (U_1, U_2, U_3, U_4)^\top \mapsto U_2U_1^{-1}\operatorname{diag}(U_3)$, $\varphi\colon \mathbb{R}^{n\times d} \to \mathbb{R}^d$, $U_i \in \mathbb{R}^{d\times d}$, and the feature map $F : \mathbb{R}^{n\times m} \to \mathbb{R}^{4d\times d}$, $F(E_N) = (V_N, C_N, \operatorname{diag}(\mathbf{x}_N), 0_{d\times d})^\top$.

Consider the network decomposition $f(E_N) = f_{\text{MLP}}(f_{\text{att}}(E_N))$, where f_{att} corresponds to the attention head, the role of which is to represent the feature map F, and f_{MLP} is the MLP, which approximates the function φ. To simplify the discussion, consider an approximation of φ via the network f, so that $U_2(U_1)^{-1}\operatorname{diag}(U_3) \approx U_2h_p(U_1)\operatorname{diag}(U_3)$, where $h_p(U_1)$, $h_p : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ is a degree p polynomial approximating $(U_1)^{-1}$.

Lemma H.3 (Attention can represent the prespecified feature map)**.** *There exists an attention layer f_{att}^* which satisfies $f_{\text{att}}^*(E_N) = (V_N, C_N, \operatorname{diag}(\mathbf{x}_N), 0_d)^\top = F(E_N)$, where $V_N = 1/(N-1)XX^\top$, $C_N = 1/(N-1)X'X^\top$, for any E_N, where F is a predefined feature map.*

Proof. The proof is by construction. We define the parameter matrices in the following way:

$$W_K \coloneqq I_n, \qquad W_Q \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \otimes I_d,$$

$$W_O \coloneqq \begin{pmatrix} \frac{1}{N-1} & 0 & 0 & 0 \\ 0 & \frac{1}{N-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \otimes I_d, \qquad W_V \coloneqq \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \otimes I_d.$$

Recall the definition (26) of embedding E_N. Then

$$E_N E_N^\top = \begin{pmatrix} XX^\top + I_d & XX'^\top & \operatorname{diag}(\mathbf{x}_N) & I_d \\ X'X^\top & X'X'^\top & 0_d & 0_d \\ \operatorname{diag}(\mathbf{x}_N) & 0_d & \operatorname{diag}(\mathbf{x}_N) \cdot \operatorname{diag}(\mathbf{x}_N) & \operatorname{diag}(\mathbf{x}_N) \\ I_d & 0_d & \operatorname{diag}(\mathbf{x}_N) & I_d \end{pmatrix},$$

$$E_N E_N^\top W_K^\top W_Q = \begin{pmatrix} XX^\top + I_d & 0_d & 0_d & 0_d \\ X'X^\top & 0_d & 0_d & 0_d \\ \operatorname{diag}(\mathbf{x}_N) & 0_d & 0_d & 0_d \\ I_d & 0_d & 0_d & 0_d \end{pmatrix},$$

$$W_O^\top W_V E_N E_N^\top W_K^\top W_Q = \begin{pmatrix} \frac{1}{N-1}XX^\top & 0_d & 0_d & 0_d \\ \frac{1}{N-1}X'X^\top & 0_d & 0_d & 0_d \\ \operatorname{diag}(\mathbf{x}_N) & 0_d & 0_d & 0_d \\ 0_d & 0_d & 0_d & 0_d \end{pmatrix},$$

$$W_O^\top W_V E_N E_N^\top W_K^\top W_Q E_{N[:,-d]} = \begin{pmatrix} \frac{1}{N-1}XX^\top & 0_d & 0_d & 0_d \\ \frac{1}{N-1}X'X^\top & 0_d & 0_d & 0_d \\ \operatorname{diag}(\mathbf{x}_N) & 0_d & 0_d & 0_d \\ 0_d & 0_d & 0_d & 0_d \end{pmatrix} \begin{pmatrix} I_d \\ 0_d \\ \operatorname{diag}(\mathbf{x}_N) \\ I_d \end{pmatrix} = \begin{pmatrix} V_N \\ C_N \\ \operatorname{diag}(\mathbf{x}_N) \\ 0_d \end{pmatrix},$$

and we obtain the desired result. □

We expect the equivalence of the optimization problems to follow the same reasoning as in the 1-dimensional case in Lemma 4.7.

H.3 Approximation and convergence bounds

Since $V_N = 1/(N-1)XX^\top$ is symmetric and almost surely positive definite for $N \geq d+1$, it almost surely does not have zero eigenvalues, and we can use Embree et al. (2025, Proposition 3.2) based on the CrouzeixPalencia theorem to bound the error of a matrix inverse approximation.

Conjecture H.4 (Upper bound on the matrix inverse approximation with a polynomial)**.**

$$\|V_N^{-1} - h_p(V_N)\| \leq (1+\sqrt{2}) \max_{\lambda \in W(V_N)} \left| \frac{1}{\lambda} - h_p(\lambda) \right|,$$

where $W(V_N) = \left\{ \frac{v^\top V_N v}{v^\top v} : 0 \neq v \in \mathbb{R}^d \right\}$ *is the numerical range of* V_N*, which is the set of all Rayleigh quotients that is bounded by the largest and smallest eigenvalues of* V_N*.*

Proof. Here we outline the sketch of a proof.

We will use a known bound on the remainder of the approximation of a multivariable function with a Taylor polynomial provided in Lemma F.1, but for a one-variable function. Since $\lambda \in \mathbb{R}$, an upper bound on the remainder term of the pth-order Taylor polynomial of φ at λ_0 is given by

$$|\varphi(\lambda) - h_p(\lambda)| \leq \frac{K}{(p+1)!} |\lambda - \lambda_0|^{p+1}, \tag{27}$$

where K is an upper bound on the absolute values of all partial derivatives of φ on $\mathbb{R}$ of order $p+1$.

Take $\delta = |\lambda - \lambda_0| > 0$. The p-th derivative of $1/\lambda$ with respect to λ is given by $\frac{(-1)^p p!}{\lambda^{p+1}}$. Then, we can bound the absolute value of the $(p+1)$-th order derivative with

$$|D^{(p+1)}\varphi(\lambda)| \leq \frac{(p+1)!}{C_2^{p+2}}$$

where $0 < C_2 \leq |\lambda|$.

Taking $K = \frac{(p+1)!}{C_2^{p+2}}$ and combining with (27), we get

$$|\varphi(\lambda) - h_p(\lambda)| \leq \frac{1}{(p+1)!}\frac{(p+1)!}{C_2^{p+2}}\delta^{p+1} = \frac{1}{C_2}\left(\frac{\delta}{C_2}\right)^{p+1}.$$

What is left is to combine it with the matrix inverse approximation bound. The final bound is

$$\|V_N^{-1} - h_p(V_N)\| \leq (1+\sqrt{2}) \max_{\lambda \in W(V_N)} \left(\frac{1}{C_2}\left(\frac{\delta}{C_2}\right)^{p+1}\right).$$

The expression above will converge to 0 as $p \to \infty$ if $\delta/C_2 < 1$. □

The result of the network approximation of the polynomial is currently obtained for the output in $\mathbb{R}$, but we expect it to be extendable to $\mathbb{R}^d$ based on Andoni et al. (2014). The rest of the proof is anticipated to apply with minimal changes.

I Experimental details

The experiments were implemented in Python using PyTorch (Paszke et al., 2019). Experiments in Figure 1 and Figure 5 were run on an NVIDIA DGX-1 with 8 NVIDIA Tesla P100 GPUs, each with 16 GB of memory. Other experiments were performed on an Apple MacBook Pro equipped with an Apple M3 Max chip with a 14-core CPU, 30-core GPU, 36 GB unified memory, and a 1 TB SSD, running macOS Sequoia 15.

Details for the convergence experiment in Figure 1 The network was defined as in (3), with one attention head, one fully connected layer, $M = 10,000$, and the activation function was GeLU activation function. The attention layer was defined according to Lemma 4.6. AR(1) of dimension 1, as defined in Definition 3.1, was used to generate $10,000$ tasks of length $N = 100$, and the training-validation-test split was $8/1/1$. The weights $\boldsymbol{\beta}$ were sampled from the standard Gaussian distribution, and $\boldsymbol{\gamma}$ was initialized to zero. We trained only on the prediction of the last token, and thus did not use an autoregressive masking needed to predict each sequence element. We observed no notable difference in performance across activation functions after running experiments with ReLU, GeLU, and tanh. Additional results are in Figure 5. Test loss was $0.52 \pm .02$ in Figure 1 for $d = 1$, $0.51 \pm .02$ in Figure 5, and 0.51 ± 0.02 in Figure 1 for $d = 5$. Note that the lower bound of 0.5 is expected in the loss and is due to $1/2$ of the standard Gaussian error in the loss. This result also supports the approximation bound in Theorem 4.1 (Theorem C.1).

Details for the experiment in Figure 2 In this experiment, we optimize both the outer weight of the MLP, and weights of the attention matrices that are not zero based on Lemma 4.6. The experiment settings were similar to the convergence experiments with fixed attention in Figure 1, except for the hyperparameters choice and the switch from full-batch gradient descent to Adam. Specifically, we used $d = 1$, $M = 500$, $N = 50$, and 2000 generated tasks. The model was trained for 50 epochs. In Figure 2, we plot an example of the predicted values for the $\hat{x}_{N+1}$ in the slice of the input space where $x_N \in [0.4, 0.6]$ as the value depending on $\hat{v}_N$ and $\hat{c}_N$.

Experiment with a synthetic language dataset As a sanity check, we performed a trial run for an experiment on the synthetic Dyck-3 dataset with 1000 prompts. The experiment settings were similar to the convergence experiments with fixed attention in Figure 1, except for the hyperparameters choice. Specifically, we used $d = 6$ and $M = 100$. For a sequence length of $N = 21$, the model's training loss is about 0.005 for a nesting depth of 1, 0.026 for a depth of 3, and 0.049 for a depth of 10. Test loss is similar to train loss. Quality degrades as nesting depth increases, as expected, since the model is designed for AR(1) process estimation, which captures only dependency on the previous token.

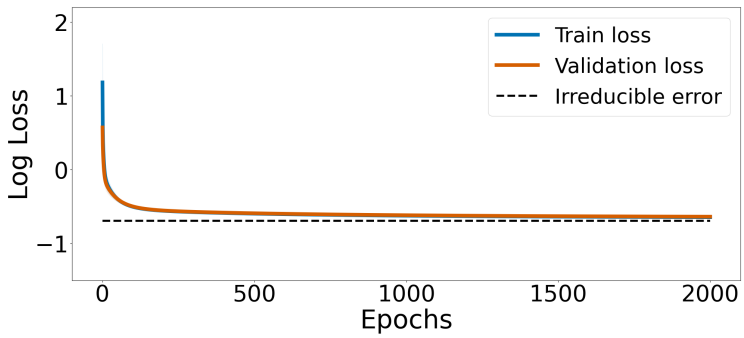


Figure 5: Confirming convergence for the specified setup. Additional results for 2-dimensional inputs. We plot the means and standard deviations, which are not well visible due to their small values, for 4 training runs of the model. Irreducible error is due to the standard Gaussian error.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We ensure that abstract claims match the contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a separate Limitations paragraph in the Discussion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide assumptions, statements and proof intuition in the main part with full proofs for lemmas and theorems in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully describe experimental settings in the Appendix "Experimental details" and share the code at `https://anonymous.4open.science/r/aricl_submission-E0E3`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is available at `https://anonymous.4open.science/r/aricl_submission-E0E3`. We either generate the data using the code we share or rely on the publicly available datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We fully describe experimental settings in the Appendix "Experimental details".

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   We plot means and standard deviations in the experiments reporting the loss, while other plots show output examples. These settings are clearly stated and are consistent with the discussion in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information on the computer resources in the Appendix "Experimental details".

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We have confirmed that the research in the paper conforms to the NeurIPS Code of Ethics. The research is of a theoretical nature, uses either synthetic or publicly available datasets, and does not directly raise the described concerns.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper focuses on the foundational research in deep learning theory and does not have a direct path to negative applications or societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release data or models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide information on the used assets in the Appendix "Experimental details".

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.