Why Do Transformers Fail to Forecast Time Series In-Context?

Yufa Zhou*, Yixiao Wang*, Surbhi Goel, Anru Zhang

Duke University

Penn University of Pennsylvania

Motivation: Time Series Forecasting

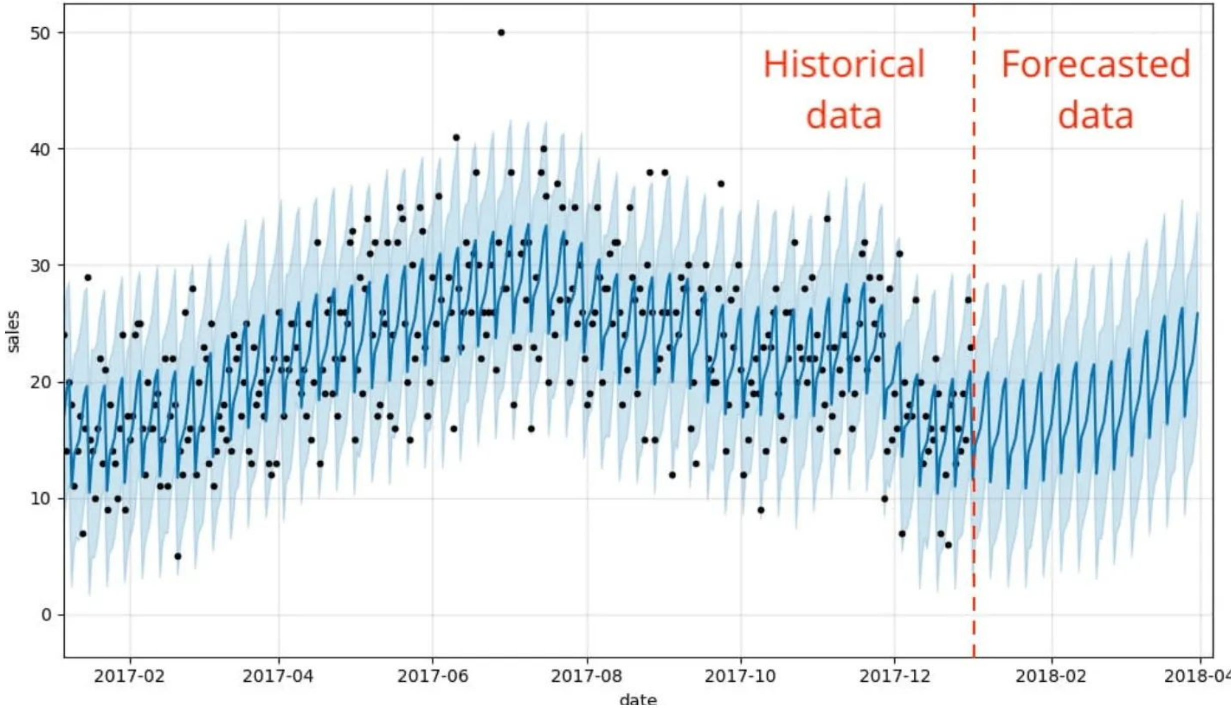
What is Time Series Forecasting (TSF)?

Predicts future values using *historical observations*.

Analyzes temporal patterns:

- trends
- seasonality
- periodicity
- short-term dynamics

Uses these patterns to generate accurate forecasts for upcoming time steps.



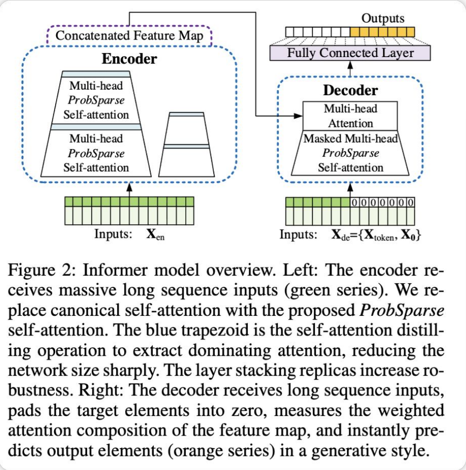
Motivation: Transformers for Time Series

Phase 1: Transformers (2021-2023)

Informer (ProbSparse Attention)
Pyraformer (Pyramidal Attention)
Autoformer (Auto-Correlation)
FEDformer (Frequency Enhanced)

Phase 2: LLMs & Foundation Models (2023-Present)

Zero-Shot (Pretrained LLMs)
Reprogramming (Time-LLM)
Foundation Models (Chronos)



Concatenated Feature Map
Encoder
Multi-head *ProbSparse* Self-attention
Multi-head *ProbSparse* Self-attention
Inputs: $\mathbf{X}_{en}$
Outputs
Fully Connected Layer
Decoder
Multi-head Attention
Masked Multi-head *ProbSparse* Self-attention
0 0 0 0 0 0 0
Inputs: $\mathbf{X}_{de}=\{\mathbf{X}_{token}, \mathbf{X}_{\mathbf{0}}\}$

Figure 2: Informer model overview. Left: The encoder receives massive long sequence inputs (green series). We replace canonical self-attention with the proposed *ProbSparse* self-attention. The blue trapezoid is the self-attention distilling operation to extract dominating attention, reducing the network size sharply. The layer stacking replicas increase robustness. Right: The decoder receives long sequence inputs, pads the target elements into zero, measures the weighted attention composition of the feature map, and instantly predicts output elements (orange series) in a generative style.

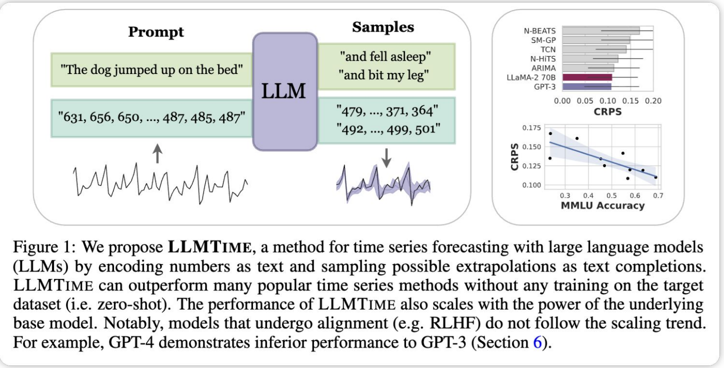


Figure 1: We propose **LLMTime**, a method for time series forecasting with large language models (LLMs) by encoding numbers as text and sampling possible extrapolations as text completions. LLMTime can outperform many popular time series methods without any training on the target dataset (i.e. zero-shot). The performance of LLMTime also scales with the power of the underlying base model. Notably, models that undergo alignment (e.g. RLHF) do not follow the scaling trend. For example, GPT-4 demonstrates inferior performance to GPT-3 (Section 6).

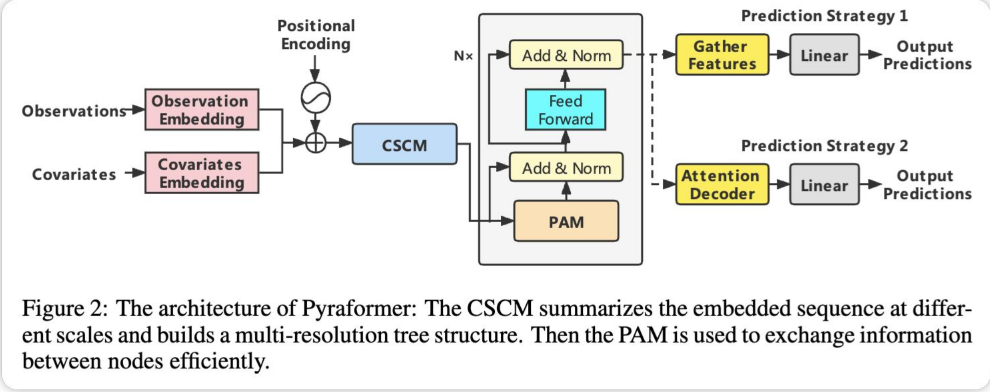


Figure 2: The architecture of Pyraformer: The CSCM summarizes the embedded sequence at different scales and builds a multi-resolution tree structure. Then the PAM is used to exchange information between nodes efficiently.

Motivation: Counterintuitive Underperformance

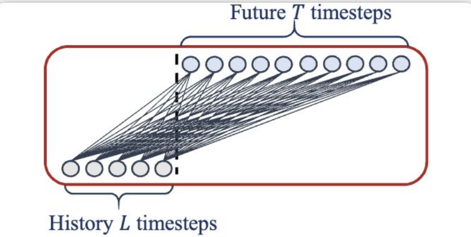
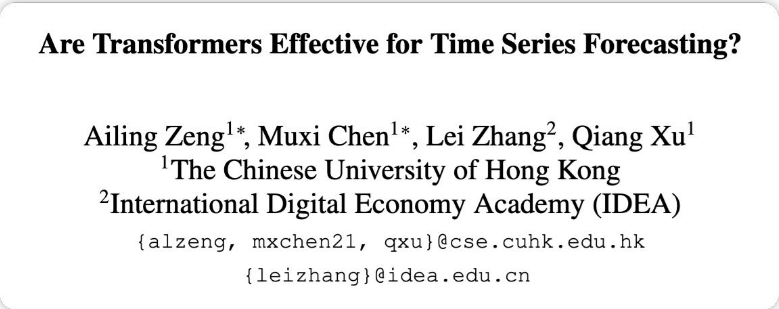
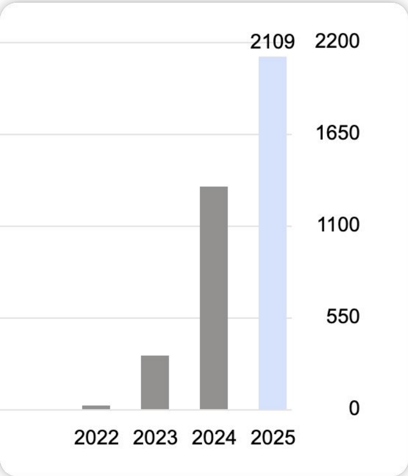


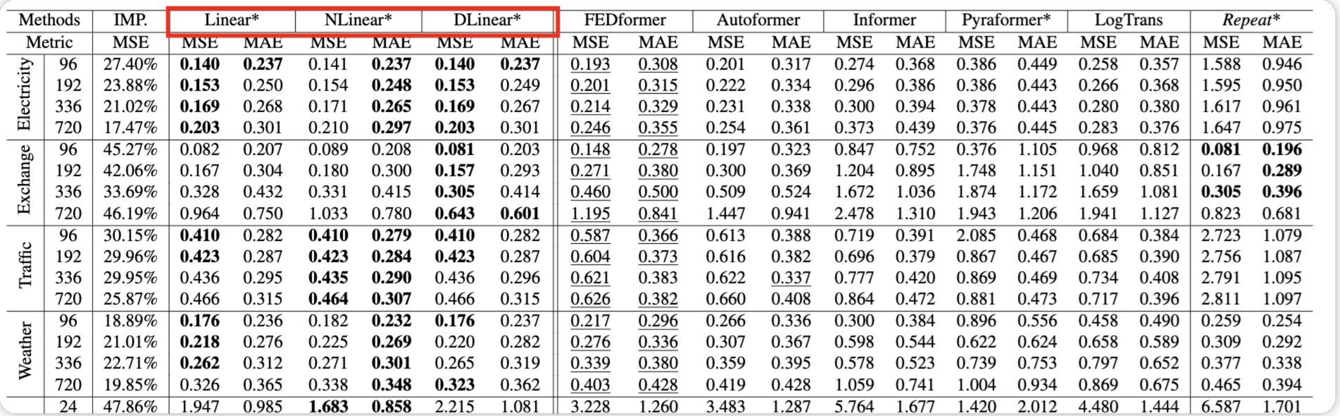
Figure 2. Illustration of the basic linear model.



Are Transformers Effective for Time Series Forecasting?

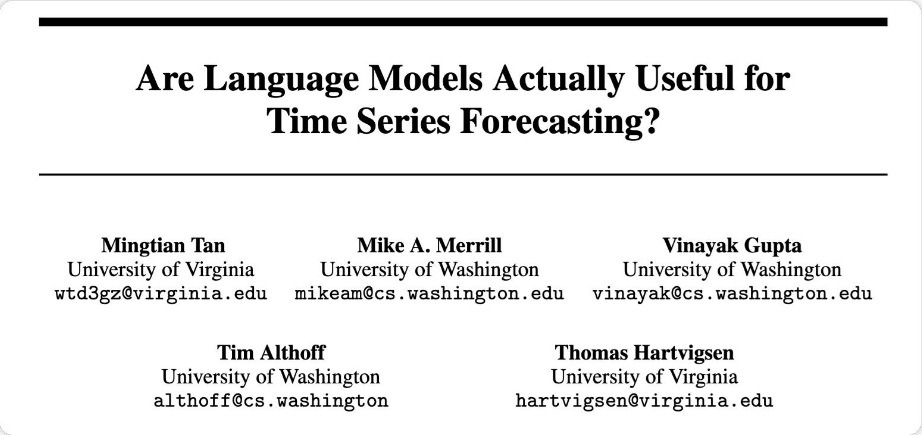
Ailing Zeng[1]*, Muxi Chen[1]*, Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy (IDEA)
{alzeng, mxchen21, qxu}@cse.cuhk.edu.hk
{leizhang}@idea.edu.cn





Methods		IMP.	Linear*		NLinear*		DLinear*		FEDformer		Autoformer		Informer		Pyraformer*		LogTrans		*Repeat**	
Metric		MSE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	27.40%	**0.140**	**0.237**	0.141	**0.237**	**0.140**	**0.237**	0.193	0.308	0.201	0.317	0.274	0.368	0.386	0.449	0.258	0.357	1.588	0.946
	192	23.88%	**0.153**	0.250	0.154	**0.248**	**0.153**	0.249	0.201	0.315	0.222	0.334	0.296	0.386	0.386	0.443	0.266	0.368	1.595	0.950
	336	21.02%	**0.169**	0.268	0.171	**0.265**	**0.169**	0.267	0.214	0.329	0.231	0.338	0.300	0.394	0.378	0.443	0.280	0.380	1.617	0.961
	720	17.47%	**0.203**	0.301	0.210	**0.297**	**0.203**	0.301	0.246	0.355	0.254	0.361	0.373	0.439	0.376	0.445	0.283	0.376	1.647	0.975
Exchange	96	45.27%	0.082	0.207	0.089	0.208	**0.081**	0.203	0.148	0.278	0.197	0.323	0.847	0.752	0.376	1.105	0.968	0.812	**0.081**	**0.196**
	192	42.06%	0.167	0.304	0.180	0.300	**0.157**	0.293	0.271	0.380	0.300	0.369	1.204	0.895	1.748	1.151	1.040	0.851	0.167	**0.289**
	336	33.69%	0.328	0.432	0.331	0.415	**0.305**	0.414	0.460	0.500	0.509	0.524	1.672	1.036	1.874	1.172	1.659	1.081	**0.305**	**0.396**
	720	46.19%	0.964	0.750	1.033	0.780	**0.643**	**0.601**	1.195	0.841	1.447	0.941	2.478	1.310	1.943	1.206	1.941	1.127	0.823	0.681
Traffic	96	30.15%	**0.410**	0.282	**0.410**	**0.279**	**0.410**	0.282	0.587	0.366	0.613	0.388	0.719	0.391	2.085	0.468	0.684	0.384	2.723	1.079
	192	29.96%	**0.423**	0.287	**0.423**	**0.284**	**0.423**	0.287	0.604	0.373	0.616	0.382	0.696	0.379	0.867	0.467	0.685	0.390	2.756	1.087
	336	29.95%	0.436	0.295	**0.435**	**0.290**	0.436	0.296	0.621	0.383	0.622	0.337	0.777	0.420	0.869	0.469	0.734	0.408	2.791	1.095
	720	25.87%	0.466	0.315	**0.464**	**0.307**	0.466	0.315	0.626	0.382	0.660	0.408	0.864	0.472	0.881	0.473	0.717	0.396	2.811	1.097
Weather	96	18.89%	**0.176**	0.236	0.182	**0.232**	**0.176**	0.237	0.217	0.296	0.266	0.336	0.300	0.384	0.896	0.556	0.458	0.490	0.259	0.254
	192	21.01%	**0.218**	0.276	0.225	**0.269**	0.220	0.282	0.276	0.336	0.307	0.367	0.598	0.544	0.622	0.624	0.658	0.589	0.309	0.292
	336	22.71%	**0.262**	0.312	0.271	**0.301**	0.265	0.319	0.339	0.380	0.359	0.395	0.578	0.523	0.739	0.753	0.797	0.652	0.377	0.338
	720	19.85%	0.326	0.365	0.338	**0.348**	**0.323**	0.362	0.403	0.428	0.419	0.428	1.059	0.741	1.004	0.934	0.869	0.675	0.465	0.394
	24	47.86%	1.947	0.985	**1.683**	**0.858**	2.215	1.081	3.228	1.260	3.483	1.287	5.764	1.677	1.420	2.012	4.480	1.444	6.587	1.701

Motivation: Counterintuitive Underperformance



Are Language Models Actually Useful for Time Series Forecasting?

Mingtian Tan
University of Virginia
wtd3gz@virginia.edu

Mike A. Merrill
University of Washington
mikeam@cs.washington.edu

Vinayak Gupta
University of Washington
vinayak@cs.washington.edu

Tim Althoff
University of Washington
althoff@cs.washington

Thomas Hartvigsen
University of Virginia
hartvigsen@virginia.edu

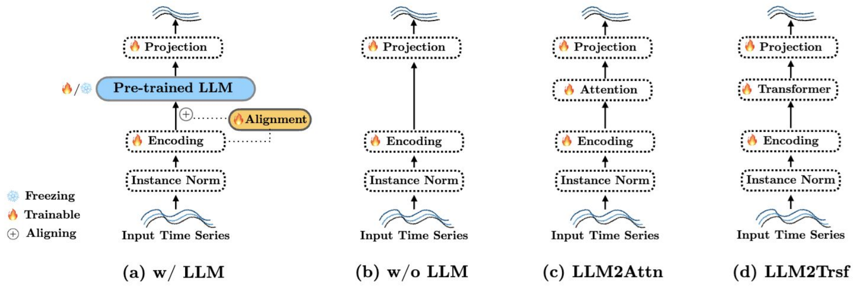
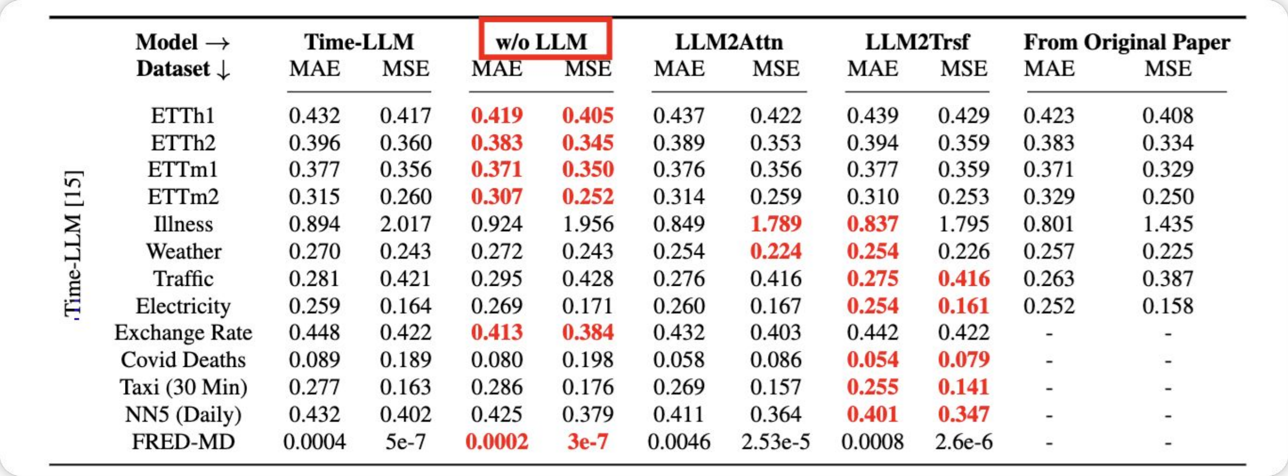


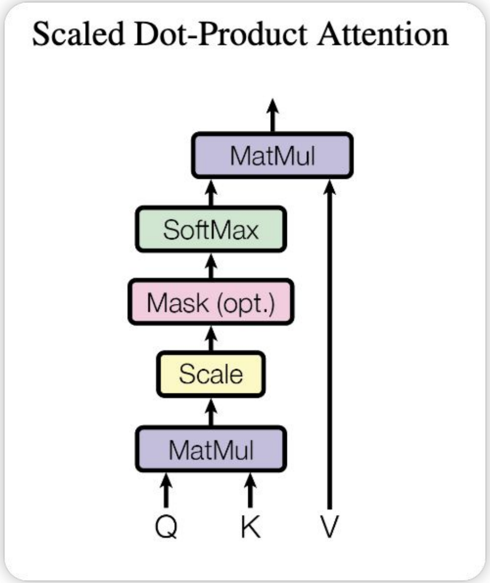
Figure 1: Overview of all LLM ablation methods. Figure (a) represents time series forecasting using an LLM as the base model. In some works, the LLM components are frozen [15, 14], while in others, they undergo fine-tuning [50, 22, 4]. Figure (b) shows the model with the LLM components removed, retaining only the remaining structure. Figure (c) replaces the LLM components with a single-layer self-attention mechanism. Figure (d) replaces the LLM components with a simple Transformer.

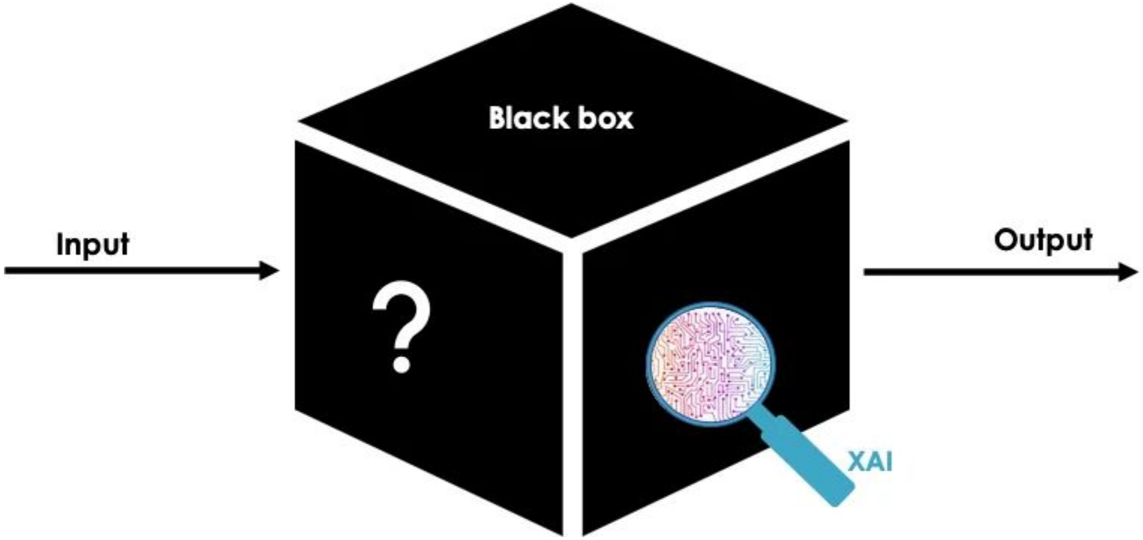


	Model →	Time-LLM		w/o LLM		LLM2Attn		LLM2Trsf		From Original Paper	
	Dataset ↓	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Time-LLM [15]	ETTh1	0.432	0.417	**0.419**	**0.405**	0.437	0.422	0.439	0.429	0.423	0.408
	ETTh2	0.396	0.360	**0.383**	**0.345**	0.389	0.353	0.394	0.359	0.383	0.334
	ETTm1	0.377	0.356	**0.371**	**0.350**	0.376	0.356	0.377	0.359	0.371	0.329
	ETTm2	0.315	0.260	**0.307**	**0.252**	0.314	0.259	0.310	0.253	0.329	0.250
	Illness	0.894	2.017	0.924	1.956	0.849	**1.789**	**0.837**	1.795	0.801	1.435
	Weather	0.270	0.243	0.272	0.243	0.254	**0.224**	**0.254**	0.226	0.257	0.225
	Traffic	0.281	0.421	0.295	0.428	0.276	0.416	**0.275**	**0.416**	0.263	0.387
	Electricity	0.259	0.164	0.269	0.171	0.260	0.167	**0.254**	**0.161**	0.252	0.158
	Exchange Rate	0.448	0.422	**0.413**	**0.384**	0.432	0.403	0.442	0.422	-	-
	Covid Deaths	0.089	0.189	0.080	0.198	0.058	0.086	**0.054**	**0.079**	-	-
	Taxi (30 Min)	0.277	0.163	0.286	0.176	0.269	0.157	**0.255**	**0.141**	-	-
	NN5 (Daily)	0.432	0.402	0.425	0.379	0.411	0.364	**0.401**	**0.347**	-	-
	FRED-MD	0.0004	5e-7	**0.0002**	**3e-7**	0.0046	2.53e-5	0.0008	2.6e-6	-	-

Why This Happen?

Attention Mechanism might be the Key!





Defining the AR(p) Process

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \varepsilon_t$$

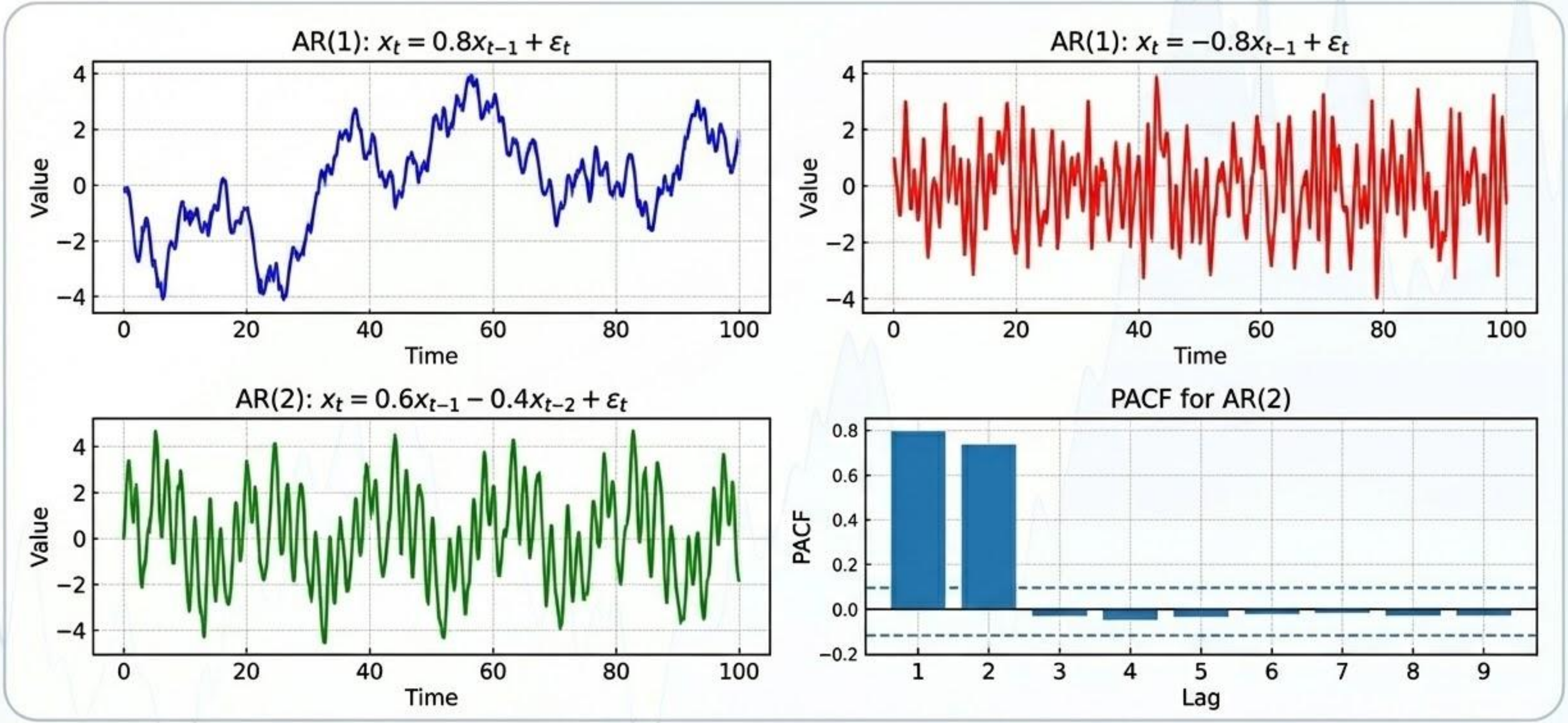
- x_t: The value of the time series at time t.
- p: The order of the autoregressive process.
- $\phi_1, \ldots, \phi_p$: The autoregressive coefficients.
- ε_t: White noise term (random error) at time t, $\varepsilon_t \sim N(0, \sigma^2)$.

Intuitive Definition: An AR(p) process models the current value as a linear combination of its p past values plus a random error term.

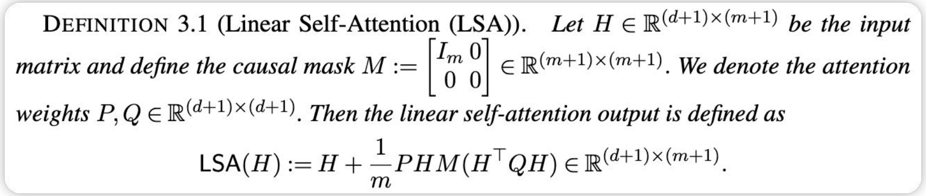
Stationarity Condition:

For the process to be weakly stationary, the roots of the characteristic polynomial $1 - \phi_1 z - \phi_2 z^2 - \ldots - \phi_p z^p = 0$ must lie outside the unit circle.

Visualizing AR(p) Process Dynamics

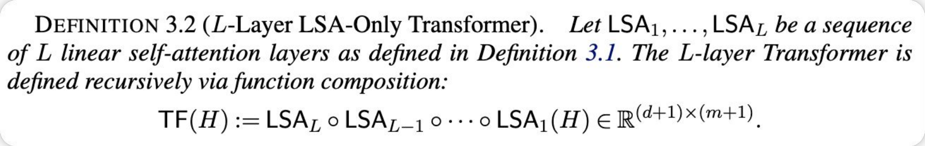


Our Theoretical Setting: In-Context Forecasting with LSA



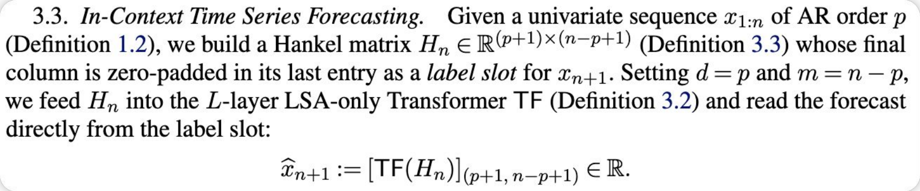
DEFINITION 3.1 (Linear Self-Attention (LSA)). *Let $H \in \mathbb{R}^{(d+1)\times(m+1)}$ be the input matrix and define the causal mask $M := \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}$. We denote the attention weights $P, Q \in \mathbb{R}^{(d+1)\times(d+1)}$. Then the linear self-attention output is defined as*

$$\mathsf{LSA}(H) := H + \frac{1}{m} PHM(H^\top QH) \in \mathbb{R}^{(d+1)\times(m+1)}.$$



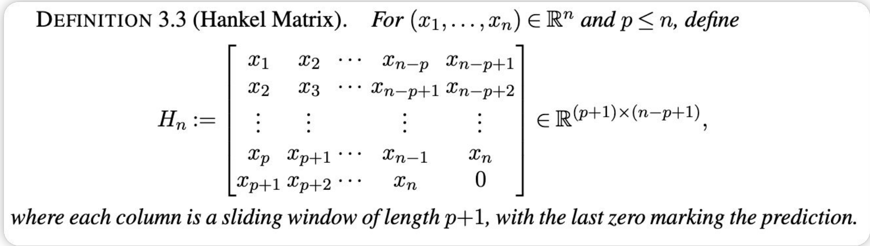
DEFINITION 3.2 (L-Layer LSA-Only Transformer). *Let $\mathsf{LSA}_1, \ldots, \mathsf{LSA}_L$ be a sequence of L linear self-attention layers as defined in Definition 3.1. The L-layer Transformer is defined recursively via function composition:*

$$\mathsf{TF}(H) := \mathsf{LSA}_L \circ \mathsf{LSA}_{L-1} \circ \cdots \circ \mathsf{LSA}_1(H) \in \mathbb{R}^{(d+1)\times(m+1)}.$$



3.3. *In-Context Time Series Forecasting.* Given a univariate sequence $x_{1:n}$ of AR order p (Definition 1.2), we build a Hankel matrix $H_n \in \mathbb{R}^{(p+1)\times(n-p+1)}$ (Definition 3.3) whose final column is zero-padded in its last entry as a *label slot* for x_{n+1}. Setting $d = p$ and $m = n - p$, we feed H_n into the L-layer LSA-only Transformer TF (Definition 3.2) and read the forecast directly from the label slot:

$$\widehat{x}_{n+1} := [\mathsf{TF}(H_n)]_{(p+1,\, n-p+1)} \in \mathbb{R}.$$



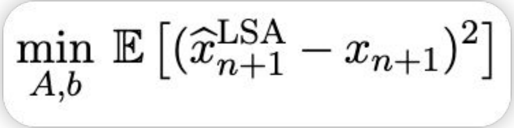
DEFINITION 3.3 (Hankel Matrix). *For $(x_1, \ldots, x_n) \in \mathbb{R}^n$ and $p \leq n$, define*

$$H_n := \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-p} & x_{n-p+1} \\ x_2 & x_3 & \cdots & x_{n-p+1} & x_{n-p+2} \\ \vdots & \vdots & & \vdots & \vdots \\ x_p & x_{p+1} & \cdots & x_{n-1} & x_n \\ x_{p+1} & x_{p+2} & \cdots & x_n & 0 \end{bmatrix} \in \mathbb{R}^{(p+1)\times(n-p+1)},$$

where each column is a sliding window of length $p+1$, with the last zero marking the prediction.

The Hankel matrix **implicitly encodes position** by fixing the relative order of observations, capturing autoregressive structure **without the need for external positional embeddings**.

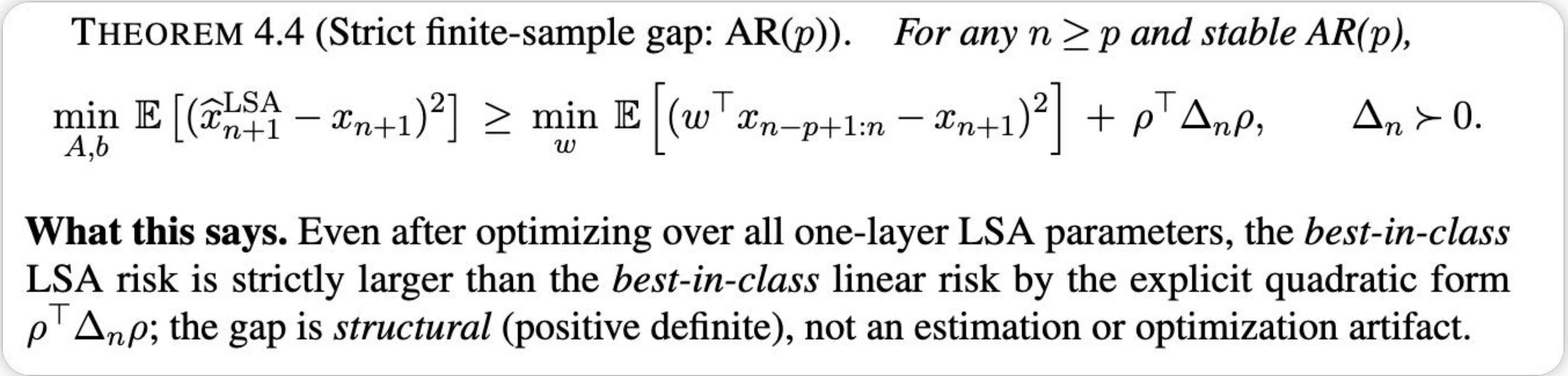
MSE Loss:



$$\min_{A,b} \mathbb{E}\left[(\widehat{x}^{\mathrm{LSA}}_{n+1} - x_{n+1})^2\right]$$

Overview of our theoretical formulation. See more details in our paper preliminaries.

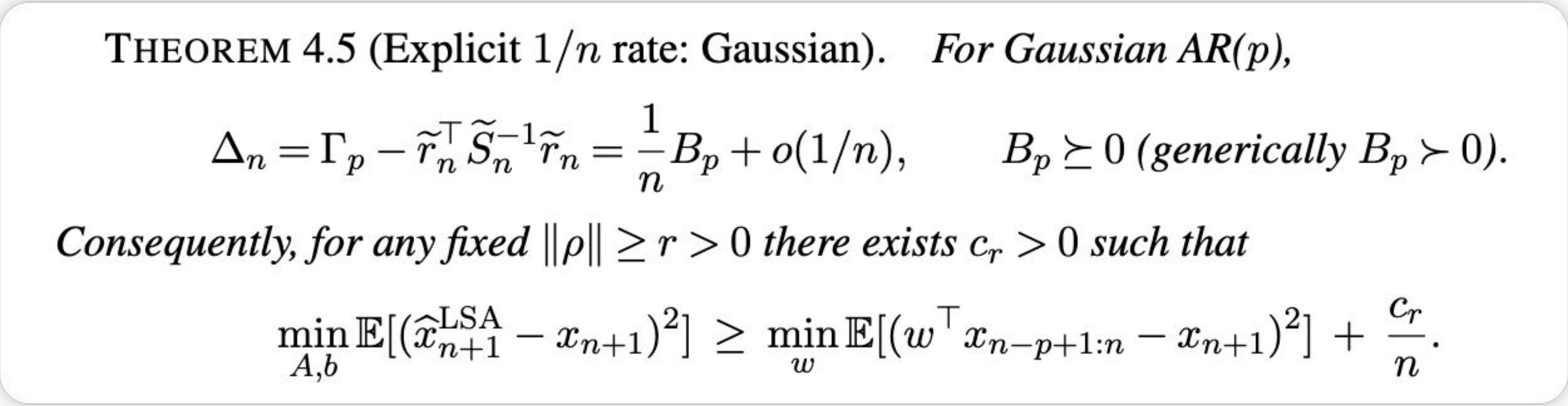
Our Theoretical Results: Strict Finite-Sample Gap



THEOREM 4.4 (Strict finite-sample gap: AR(p)). *For any $n \ge p$ and stable AR(p),*

$$\min_{A,b} \mathbb{E}\left[(\widehat{x}^{\mathrm{LSA}}_{n+1} - x_{n+1})^2\right] \ge \min_{w} \mathbb{E}\left[\left(w^\top x_{n-p+1:n} - x_{n+1}\right)^2\right] + \rho^\top \Delta_n \rho, \qquad \Delta_n \succ 0.$$

What this says. Even after optimizing over all one-layer LSA parameters, the *best-in-class* LSA risk is strictly larger than the *best-in-class* linear risk by the explicit quadratic form $\rho^\top \Delta_n \rho$; the gap is *structural* (positive definite), not an estimation or optimization artifact.



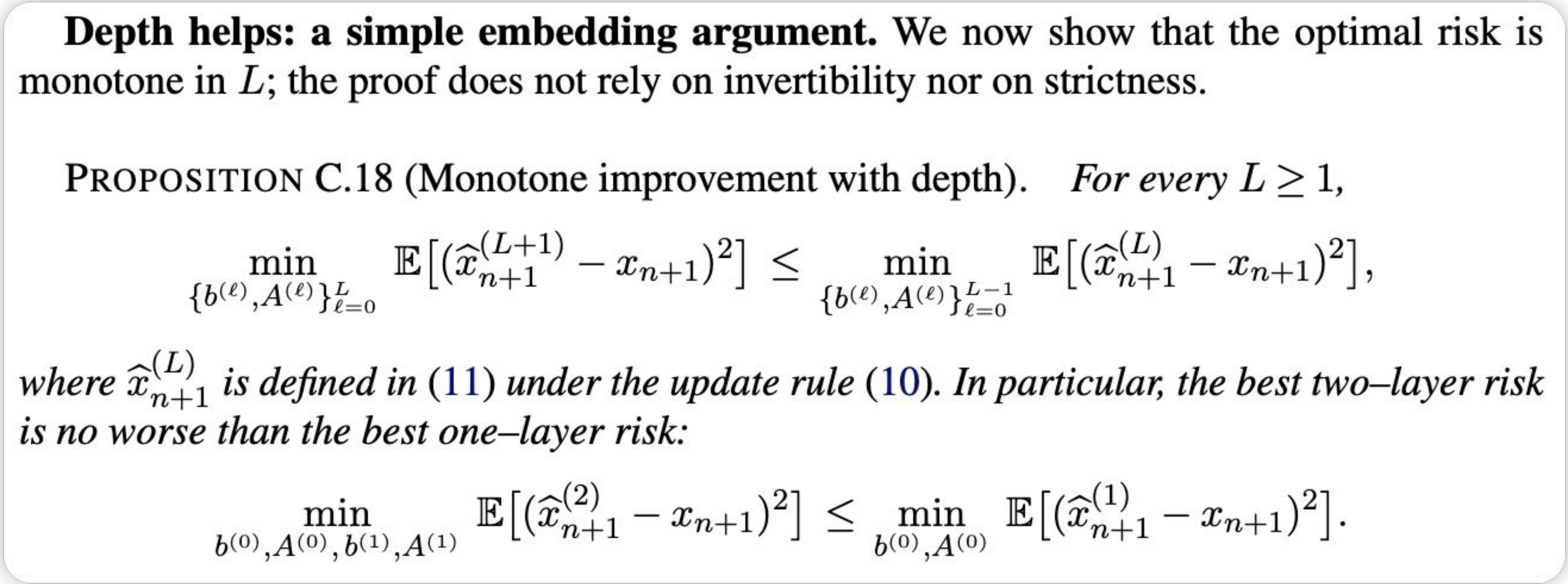
THEOREM 4.5 (Explicit $1/n$ rate: Gaussian). *For Gaussian AR(p),*

$$\Delta_n = \Gamma_p - \widetilde{r}_n^\top \widetilde{S}_n^{-1} \widetilde{r}_n = \frac{1}{n} B_p + o(1/n), \qquad B_p \succeq 0 \; \textit{(generically } B_p \succ 0).$$

Consequently, for any fixed $\|\rho\| \ge r > 0$ there exists $c_r > 0$ such that

$$\min_{A,b} \mathbb{E}[(\widehat{x}^{\mathrm{LSA}}_{n+1} - x_{n+1})^2] \ge \min_{w} \mathbb{E}[(w^\top x_{n-p+1:n} - x_{n+1})^2] + \frac{c_r}{n}.$$

Our Theoretical Results: Multi-Layer Improvment



Depth helps: a simple embedding argument. We now show that the optimal risk is monotone in L; the proof does not rely on invertibility nor on strictness.

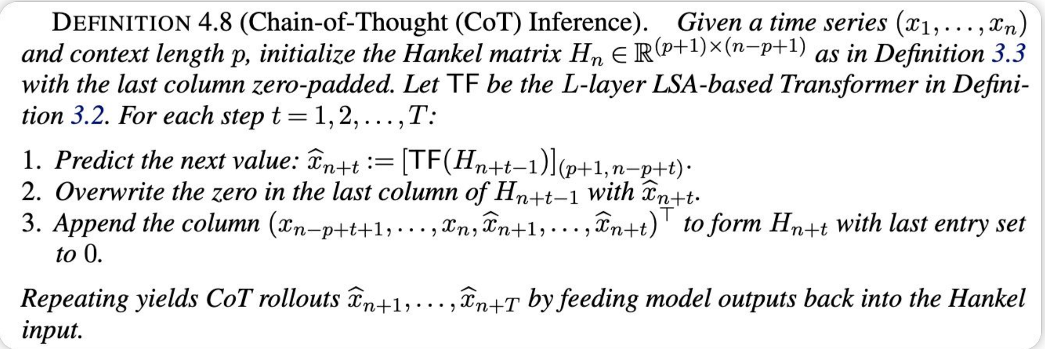
PROPOSITION C.18 (Monotone improvement with depth). *For every $L \geq 1$,*

$$\min_{\{b^{(\ell)},A^{(\ell)}\}_{\ell=0}^{L}} \mathbb{E}\big[(\widehat{x}_{n+1}^{(L+1)} - x_{n+1})^2\big] \leq \min_{\{b^{(\ell)},A^{(\ell)}\}_{\ell=0}^{L-1}} \mathbb{E}\big[(\widehat{x}_{n+1}^{(L)} - x_{n+1})^2\big],$$

where $\widehat{x}_{n+1}^{(L)}$ is defined in (11) *under the update rule* (10)*. In particular, the best two–layer risk is no worse than the best one–layer risk:*

$$\min_{b^{(0)},A^{(0)},b^{(1)},A^{(1)}} \mathbb{E}\big[(\widehat{x}_{n+1}^{(2)} - x_{n+1})^2\big] \leq \min_{b^{(0)},A^{(0)}} \mathbb{E}\big[(\widehat{x}_{n+1}^{(1)} - x_{n+1})^2\big].$$

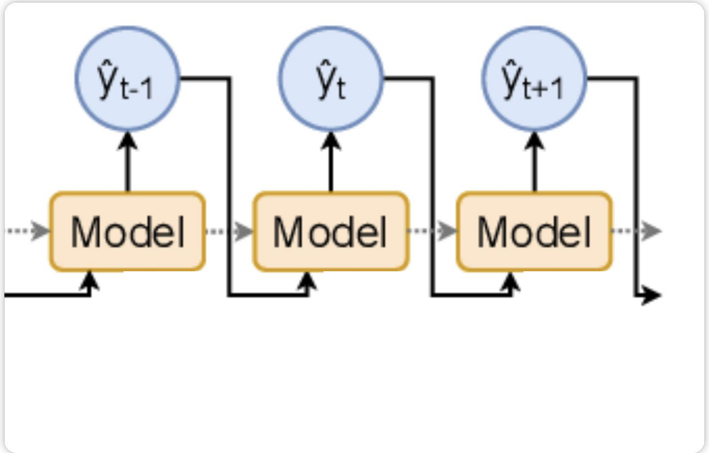
Our Theoretical Results: Chain-of-Thought Collapse

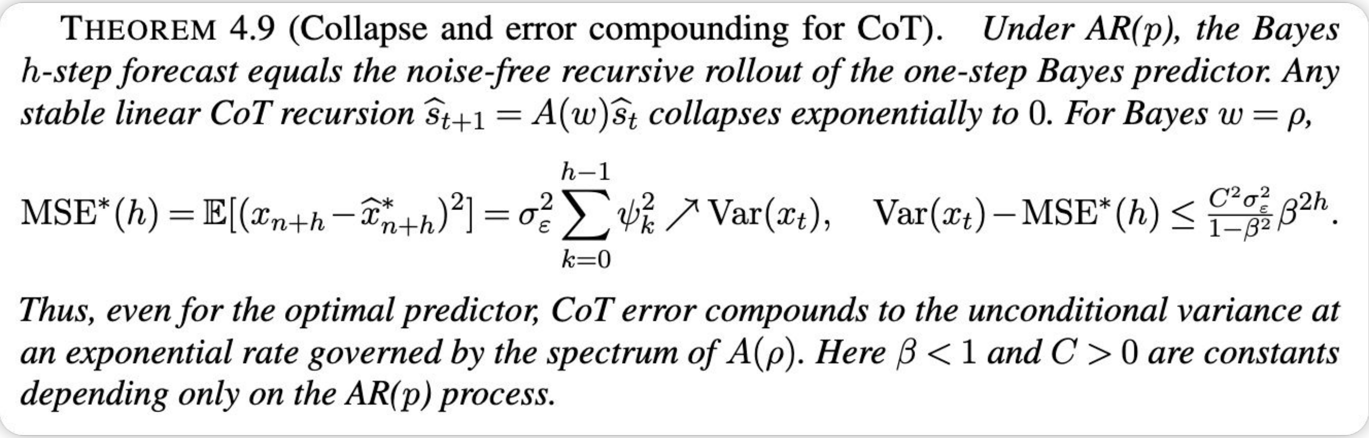


DEFINITION 4.8 (Chain-of-Thought (CoT) Inference). *Given a time series $(x_1, \dots, x_n)$ and context length p, initialize the Hankel matrix $H_n \in \mathbb{R}^{(p+1)\times(n-p+1)}$ as in Definition 3.3 with the last column zero-padded. Let TF be the L-layer LSA-based Transformer in Definition 3.2. For each step $t = 1, 2, \dots, T$:*

1. *Predict the next value:* $\widehat{x}_{n+t} := [\mathsf{TF}(H_{n+t-1})]_{(p+1,n-p+t)}$.
2. *Overwrite the zero in the last column of* H_{n+t-1} *with* $\widehat{x}_{n+t}$.
3. *Append the column* $(x_{n-p+t+1}, \dots, x_n, \widehat{x}_{n+1}, \dots, \widehat{x}_{n+t})^\top$ *to form* H_{n+t} *with last entry set to* 0.

Repeating yields CoT rollouts $\widehat{x}_{n+1}, \dots, \widehat{x}_{n+T}$ *by feeding model outputs back into the Hankel input.*



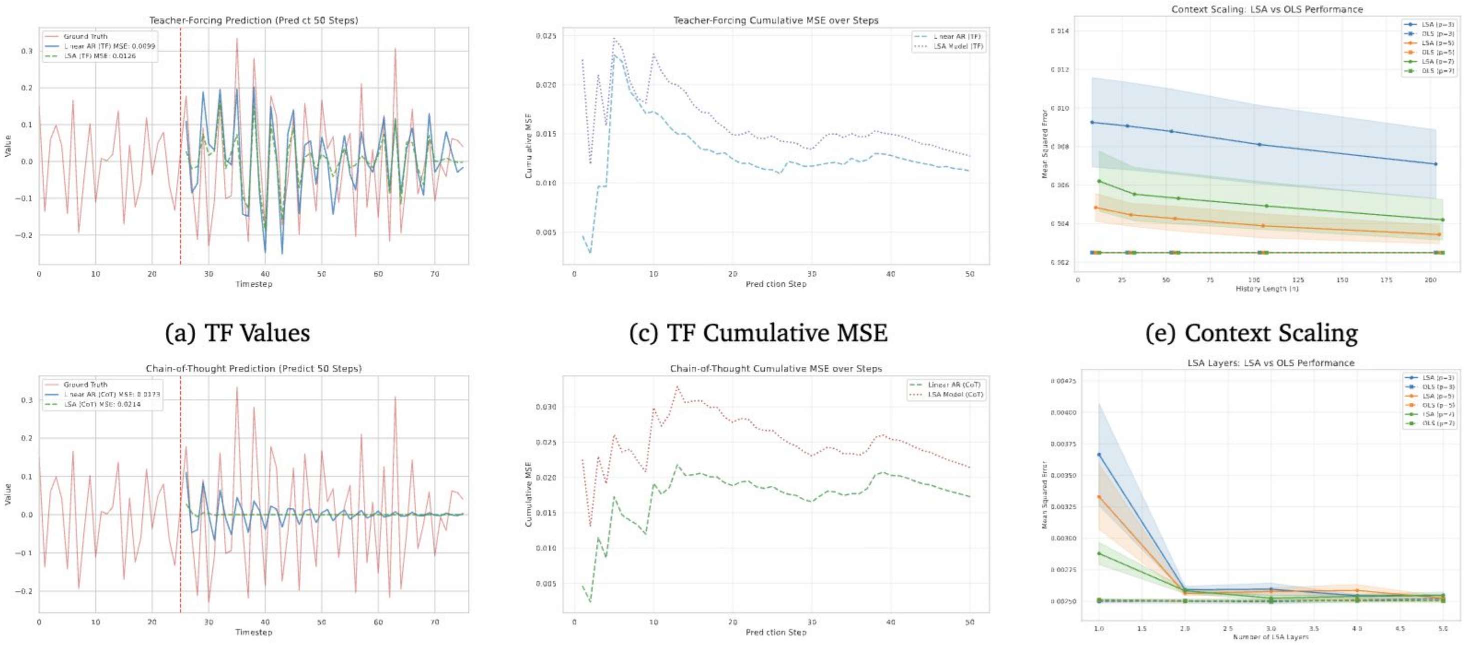


THEOREM 4.9 (Collapse and error compounding for CoT). *Under AR(p), the Bayes h-step forecast equals the noise-free recursive rollout of the one-step Bayes predictor. Any stable linear CoT recursion $\widehat{s}_{t+1} = A(w)\widehat{s}_t$ collapses exponentially to 0. For Bayes $w = \rho$,*

$$\mathrm{MSE}^*(h) = \mathbb{E}[(x_{n+h} - \widehat{x}^*_{n+h})^2] = \sigma_\varepsilon^2 \sum_{k=0}^{h-1} \psi_k^2 \nearrow \mathrm{Var}(x_t), \quad \mathrm{Var}(x_t) - \mathrm{MSE}^*(h) \le \tfrac{C^2\sigma_\varepsilon^2}{1-\beta^2}\beta^{2h}.$$

Thus, even for the optimal predictor, CoT error compounds to the unconditional variance at an exponential rate governed by the spectrum of $A(\rho)$. Here $\beta < 1$ and $C > 0$ are constants depending only on the AR(p) process.

Our Empirical Validation: Predictions and Scaling Effect



(a) TF Values

(c) TF Cumulative MSE

(e) Context Scaling

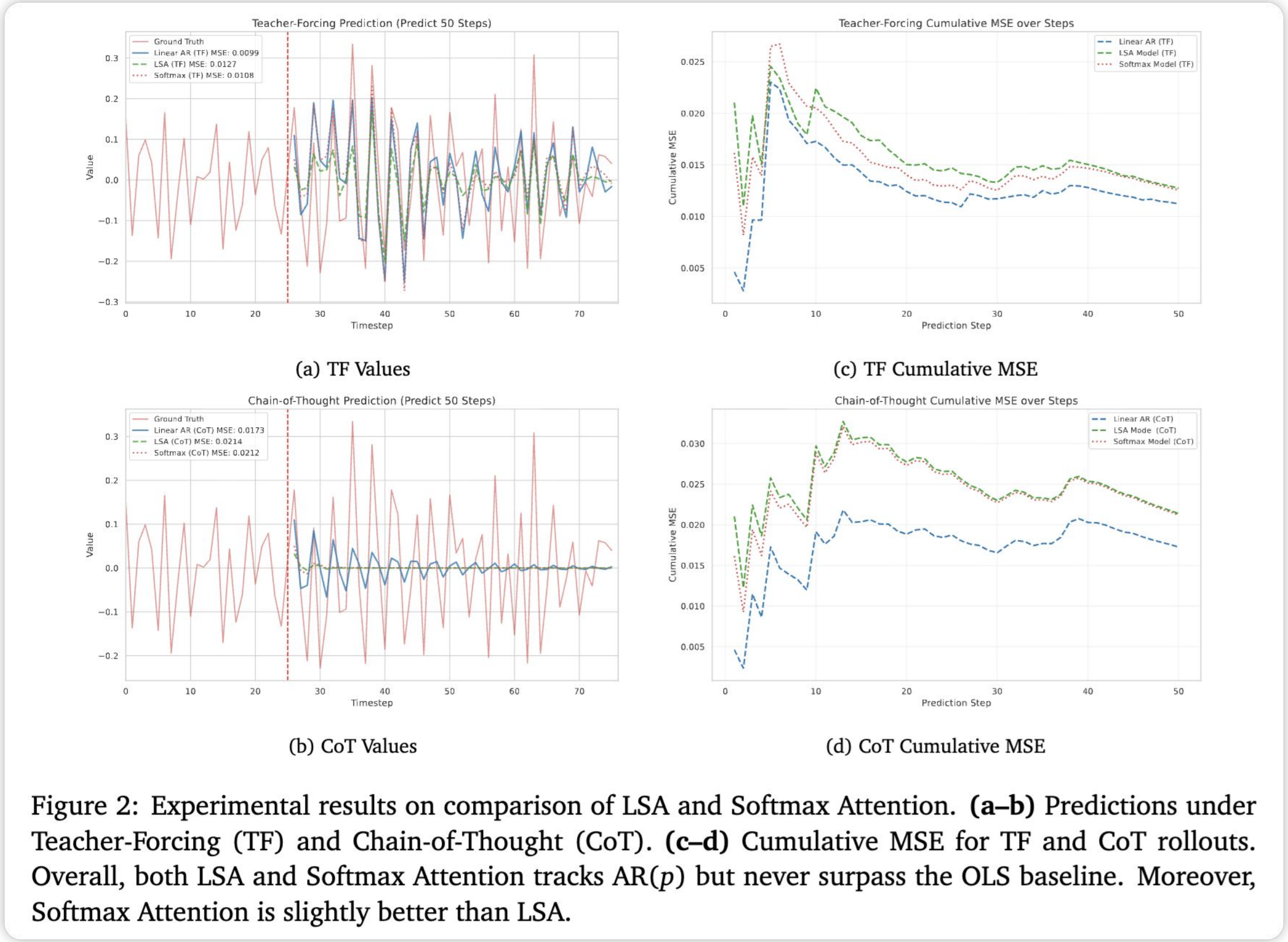
(b) CoT Values

(d) CoT Cumulative MSE

(f) LSA Layer Scaling

Figure 1: Experimental results. **(a–b)** Predictions under Teacher-Forcing (TF) and Chain-of-Thought (CoT). **(c–d)** Cumulative MSE for TF and CoT rollouts. **(e–f)** Scaling experiments varying the history length and the number of LSA layers. Overall, LSA tracks AR(p) but never surpasses the OLS baseline, confirming its representational limits.

Our Empirical Validation: Softmax vs LSA



(a) TF Values

(c) TF Cumulative MSE

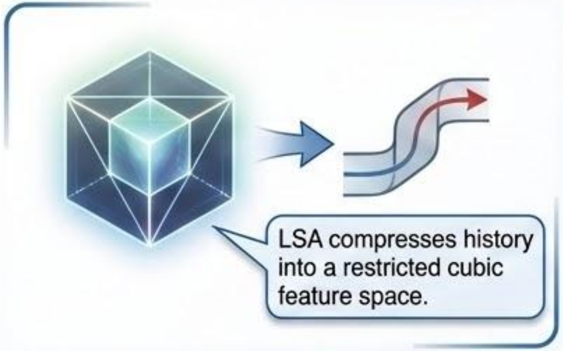
(b) CoT Values

(d) CoT Cumulative MSE

Figure 2: Experimental results on comparison of LSA and Softmax Attention. **(a–b)** Predictions under Teacher-Forcing (TF) and Chain-of-Thought (CoT). **(c–d)** Cumulative MSE for TF and CoT rollouts. Overall, both LSA and Softmax Attention tracks AR(p) but never surpass the OLS baseline. Moreover, Softmax Attention is slightly better than LSA.

Significance

1. First Rigorous Theoretical Justification



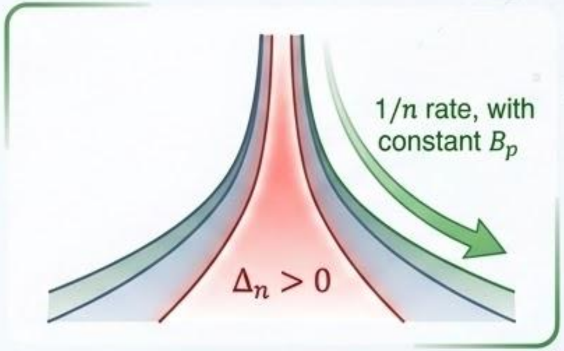
LSA compresses history into a restricted cubic feature space.

Structural impossibility: We give the first rigorous proof that LSA cannot outperform linear models on AR(p).

Beyond empirical observation: Prior work only showed empirical underperformance or relied on loose approximations.

Key insight: LSA compresses history into a restricted cubic feature space that contains no more information than the last p lags.

2. Precise Quantification of the Gap

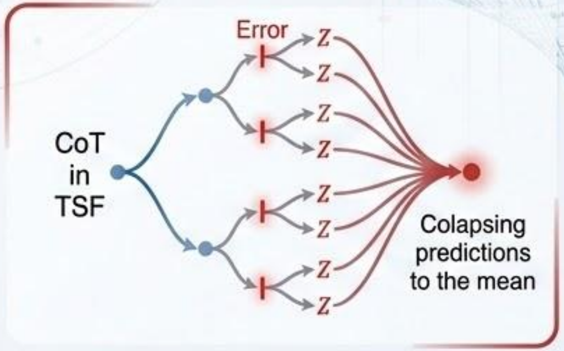


Strict finite-sample gap: For any finite n, the LSA predictor has a strictly positive excess risk (Δ_n>0).

Explicit convergence rate: The gap shrinks only at a 1/n rate, with constant B_p determined by process moments.

Implication: This gap is structural, not due to optimization or estimation error.

3. Demystifying Inference Dynamics



CoT collapse: Unlike NLP, CoT in TSF causes errors to compound exponentially, collapsing predictions to the mean.

Implications for practice: Highlights the limits of blindly applying LLM-style architectures to low-order dynamical systems and questions the "bigger is better" assumption in Deep Learning.