Transformer Designs for In-Context Learning in Foundation Models for Time Series Forecasting with Covariates

Anonymous Authors[1]

Abstract

Recent foundation models (FMs) for time series forecasting (TSF) have shown promising results in zero-shot generalization to new series. However, when time series are associated with input covariates, these models are incapable of modeling series-specific dependence of the forecasted values on the covariates. We identify that historical values in TSF implicitly provide labeled data, which can be leveraged for in-context learning (ICL). While transformers have demonstrated ICL capabilities for regression tasks, when harnessing them as FMs we need to analyze the impact of what constitutes a token in the transformer, the type of attention, and the placement of loss functions during pre-training. We study three existing tokenization schemes for regression tasks in terms of their training convergence and ICL capacity. We propose a modified shifted causal attention designed for faster convergence during pre-training since it allows imposition of next-token loss at multiple positions. Further, it combines the covariates and target such that ICL is achievable for linear regression in just one layer. For time-series data, a popular tokenization method in existing FMs is patching the input series. Our theoretical analysis shows that such tokenization is suboptimal for ICL on time series with covariates.

1. Introduction

In time series forecasting (TSF) with covariates, each series i at time t is characterized by a real-valued output y_t^τ and a vector of input features $\mathbf{x}_t^\tau$. Our goal is to train a model to predict a set of future values $y_{H+1}, \ldots, y_{H+F}$ given the history $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_H, y_H)\} \stackrel{\text{def}}{=} (X_{1:H}, Y_{1:H})$ and corresponding future covariates $\mathbf{x}_{H+1}, \ldots, \mathbf{x}_{H+F}$.

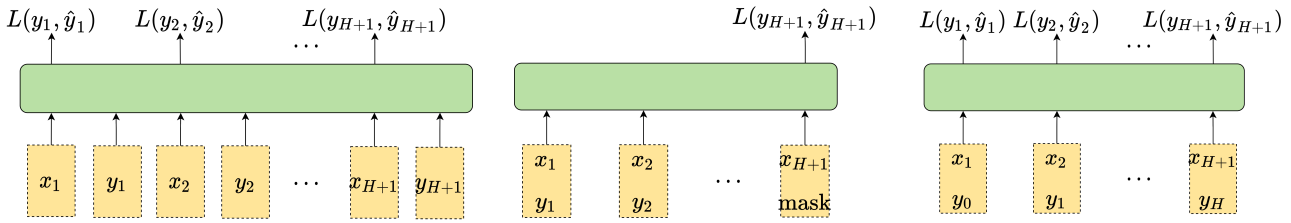
Following the success of foundation models in NLP, several foundation models have been proposed for TSF (Das et al., 2024; Jin et al., 2024; Ansari et al., 2024; Rasul et al., 2023; Ekambaram et al., 2024). These transformer-based models are pre-trained on diverse time series datasets and can generalize to new series without fine-tuning, typically by minimizing loss on future values given the preceding sequence of y values.

A limitation is that these models primarily focus on forecasts based on previous y values and do not adequately handle series-specific covariates $\mathbf{x}$ without fine-tuning[1]. In many applications, such as retail, covariates like holidays, discounts, promotions, and weather conditions are crucial. A major challenge with building foundation models with covariates is handling arbitrary series-specific dependence of y values on the covariates. Without covariates, it is unclear if test series differ enough from the large collection of series used for pre-training to require series-specific learning. However, with covariates where each series is associated with its own distribution $P_\tau(y_t|X_{1:t}, Y_{1:t-1})$, some form of series-specific adaptation is essential.

Recently, transformers have demonstrated the capability of in-context learning (ICL) for the regression task, a special case of the TSF task where y_t only depends on $\mathbf{x}_t$. Given few-shot examples in-context $\mathbf{x}_1, y_1 \ldots, \mathbf{x}_H, y_H$ where each y_t is a series-specific function $f_\tau(\mathbf{x}_t)$ of the corresponding input, prior work (Garg et al., 2022; Ahn et al., 2023; Akyürek et al., 2023; Panwar et al., 2024; Mahankali et al., 2024; Zhang et al., 2023; Von Oswald et al., 2023) has shown that causal transformers can estimate f via gradient-like updates across the transformer layers. However, the layout of the covariates and targets and placements of loss functions significantly impact ICL capacity and pre-training convergence. We analyze these options and propose an alternative Shifted Causal Attention model that allows pre-training with next-token loss while supporting ICL with just one layer.

For time-series data, although most foundation models are also transformer-based, these are modified to better represent the local context of time series and handle long series efficiently. A common modification is patching (Das et al., 2024; Nie et al., 2023; Jin et al., 2024) where instead of treating each time step as a separate transformer token, the series

[1]The model in (Garza & Mergenthaler-Canseco, 2023) may be an exception, but it lacks detailed information on covariate handling.



(a) Interleaved (b) Aligned (c) Lagged

Figure 1. Different choices of tokenization of in-context examples along with positions of loss terms during pre-training a foundation model for regression. Our proposed SCA model's loss and input tokenization is identical to Lagged, the only difference is internally the attention is Shifted Causal Attention instead of standard causal attention in the above three variants.

is disjointly partitioned into equal-sized contiguous patches, and each patch is treated as a token. These models do not handle covariates, and we show that patched transformers reduce the capacity of in-context learning with covariates in Section 3.2. We empirically study the impact of these design with two classes of regression functions.

2. Related Work

Recently, there has been a surge in building foundation models (FM) for zero-shot forecasting on new time series. These models, often variations of the causal transformer trained with next-token prediction loss, include **TimeGPT-1** (Garza & Mergenthaler-Canseco, 2023), **Lag-Llama** (Rasul et al., 2023), and **TimesFM** (Das et al., 2024). While details of TimeGPT-1 are unavailable, TimesFM relies heavily on patching. Other models, like Chronos (Ansari et al., 2024), LLMTime (Gruver et al., 2023), and TimeLLM (Jin et al., 2024), treat y values as discrete tokens. The first non-transformer-based FM, TTM (Ekambaram et al., 2024), uses feed-forward layers and shows superior performance, indicating generalization without explicit ICL for time-series data without covariates.

The ICL abilities of transformers have been extensively studied for regression tasks (Garg et al., 2022; Ahn et al., 2023; Akyürek et al., 2023; Panwar et al., 2024; Mahankali et al., 2024; Zhang et al., 2023; Von Oswald et al., 2023). Transformers can learn various function classes in-context, such as linear functions, 2-layer neural networks, and decision trees, with performance comparable to standard algorithms like XGBoost and SGD (Garg et al., 2022). Subsequent work shows transformers can simulate steps of gradient descent and Sherman-Morrison updates (Akyürek et al., 2023). ICL matches ordinary least-squares predictions on noiseless datasets and the minimum Bayes risk predictor on noisy datasets. Zhang et al. (Zhang et al., 2023) demonstrate that linear attention converges to meta-learning linear regression functions.

3. ICL for Time-Series with Covarates

For training the foundation models we assume that we are given a large diverse collection of series $\mathcal{S} = \{(\mathbf{x}_t^\tau, y_t^\tau) : t = 1 \dots T_\tau, i = 1 \dots S\}$ where each $\mathbf{x}_t^\tau \in \Re^{d_\tau}$. Each series i follows its own distribution $y_t^\tau = f_\tau(X_{1:t}^\tau, Y_{1:t-1}^\tau) + \eta$, where $\eta \sim N(0, \sigma^2)$, $X_{1:t}^\tau$ and $Y_{1:t}^\tau$ denote the sequence of covariates and output y values between 1 and t, and $f_\tau \sim \mathcal{F}$ a broad class of functions.

When designing a transformer as a foundation model, there are several design choices to be made in terms of what constitutes a token in the transformer, the type of attention, and the placement of loss functions during pre-training. These decisions arise even for the regression task, a special case of TSF where y_t only depends on $\mathbf{x}_t$ and we discuss these in Section 3.1. We then analyze patched transformers since they are commonplace for time-series in Section 3.2.

3.1. ICL for Regression with Transformers

For regression, each value y_t is a function of only $\mathbf{x}_t$ (i.e. $y_t = f(\mathbf{x}_t)$), and given $\{(\mathbf{x}_s, y_s) : s = 1 \dots H\}$ we need to in-context learn to output $f(\mathbf{x}_{H+1})$. The training data $\mathcal{S}$ covers examples with diverse f_τ, and we need to train parameters θ of M_θ using these. In prior work, many different methods have been proposed for how the in-context examples can be tokenized for input to a transformer and where loss terms are introduced. These are categorized into three types, as shown in Figure 1.

1. **Interleaved** (Garg et al., 2022) This option interleaves the position of covariates and labels after padding to equal-sized vectors (Figure 1a). The attention is causal, and this allows imposing a loss for each in-context example during training as $\sum_{i \in \mathcal{S}} \sum_{t \in [H]} (y_t^\tau - o_{2t-1})^2$ where we use o_t to denote the output from position t of the transformer. Since for each sequence i, the loss is introduced for each y_t, the convergence of θ potentially accelerates.
2. **Aligned** Zhang et al. (2023) analyzes an Aligned setting (Figure 1b), where the inputs and labels are concate-

nated, while the label of the last query token x_{H+1} is masked to 0. The loss is only on this last token since other positions already know the labels. The training loss is $\sum_{i\in\mathcal{S}}(y_{H+1}^{\tau} - o_{H+1})^2$. An advantage of this setting is that since the inputs and labels are part of the same token, ICL is possible only with a single linear transformer layer, as shown in (Zhang et al., 2023).

3. **Lagged** To get the best of both worlds, one might be motivated to employ a lagged setting (variant 1c), where each x_t is concatenated with y_{t-1} thus making the sequence compact while allowing the imposition of loss for every example as $\sum_{i\in\mathcal{S}}\sum_{t\in[H]}(y_t^{\tau} - o_t)^2$. However, in the lagged variant, a single transformer layer does not suffice for ICL since this ability crucially depends on the self-attention matrix being able to compute $\sum_{t=1}^{H} x_t^T y_t$ within the single layer.

An ideal variant would aim to include the inductive bias of the Aligned variant for ICL while imposing next-token loss at multiple positions for faster convergence. To do so, we propose a tweak to causal attention, which we refer to as *shifted causal attention* (SCA). Given the query, key and value vectors $\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t \in \mathbb{R}^D$ for $t \in [H]$, we define the output of SCA as:

$$[SCA] \qquad \frac{\sum_{j=1}^{t-1}\exp(\langle \mathbf{q}_t, \mathbf{k}_j\rangle)\mathbf{v}_{j+1} + \exp(\langle \mathbf{q}_t, \mathbf{k}_t\rangle)\mathbf{v}_t}{\sum_{j=1}^{t}\exp(\langle \mathbf{q}_t, \mathbf{k}_j\rangle)} \tag{1}$$

Compared to standard causal attention, we see two differences. The value vector is obtained not from the same position j as key but from a position shifted by 1 position ahead, and there is no attention over t itself. Since the indices in the summation never exceed t, shifted causal attention is strictly causal in nature, i.e., there is no leakage of information regarding y_t at the t^{th} position, thus allowing us to impose a loss for each in-context example during training as $\sum_{i\in\mathcal{S}}\sum_{t\in[H]}(y_t^{\tau} - o_t)^2$.

3.2. Patched Transformers

For time-series data, a more popular transformer variant is to tokenize the input series into disjoint partitions, called patches of a fixed length (Nie et al., 2023; Das et al., 2024; Jin et al., 2024). Here, we analyze why patching could hinder in-context learning with series-specific covariates. Following prior work (Zhang et al., 2023; Ahn et al., 2023; Von Oswald et al., 2023) consider a single-headed, one-layer transformer with linear self-attention in the Aligned setting (Figure 1b).

If the input sequence length is $(\mathbf{x}_1, y_1)\ldots(\mathbf{x}_H, y_H)$, p is the patch length, and covariate $\mathbf{x} \in \Re^d$, number of patches $N = \frac{H}{p}$, the input to the transformer can be written as

$$G = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_{p+1} & \ldots & \mathbf{x}_{Np-p+1} & \mathbf{x}_{H+1} \\ y_1 & y_{p+1} & \ldots & y_{Np-p+1} & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ \mathbf{x}_p & \mathbf{x}_{2p} & \ldots & \mathbf{x}_{Np} & 0 \\ y_p & y_{2p} & \ldots & y_{Np} & 0 \end{bmatrix} \in \Re^{(d+1)p\times N+1}$$

Let Z be a submatrix of G comprising only its first N columns. As shown in (Zhang et al., 2023) a one-layer linear transformer $M_{\text{lsa}}(G)$ has parameters $U, v \in \Re^{r\times r}$ where $r = (d+1)p$ the size of an input token, and for the query point $\mathbf{x}_{H+1}$, the output of the transformer is:

$$\hat{y}_{H+1} = \frac{1}{N} U Z Z^T v \mathbf{x}_{H+1} \stackrel{\text{def}}{=} \hat{\mathbf{w}}\,\mathbf{x}_{H+1}$$

Theorem 3.1. *Consider a data distribution where $P(\mathbf{x}) \sim \mathcal{N}(0, \Sigma)$ where $\Sigma \in \Re^{d\times d}$ is a positive definite co-variance matrix, and for each series i, let $P_\tau(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{w}_\tau \mathbf{x}_t, \sigma^2)$ where $\mathbf{w}_\tau \sim \mathcal{N}(0, I_{d\times d})$. Consider a new input sequence τ, $\mathbf{x}_1, y_1, \ldots \mathbf{x}_H, y_H, \mathbf{x}_{H+1}$ where $y = \mathbf{w}^*\mathbf{x} + \eta$ sampled from this distribution. We show that if the U, v parameters lead to an unbiased estimate for $\mathbf{w}^*$, then the error of the estimate is lower bounded by $\frac{\sigma^2 dp}{2H}$. This is suboptimal by a factor of $\frac{p}{2}$, compared to an unpatched transformer where $p = 1$. (Proof in Appendix)*

4. Experiments

We first present experiments on the regression task to answer the following research question: 1. Does imposing loss at each in-context position accelerate pre-training compared to imposing loss on a single masked position? 2. How does shifted causal attention compare with standard causal attention? 3. How does patching impact ICL?

To answer these questions, we train the four transformer variants on two tasks following the setup of (Garg et al., 2022). We train the transformer variants using squared error as mentioned in Section 3.1. We optimize the parameters using gradient descent by training for 200K steps with a batch size of 64. We compare performance across one and four layers with dimensions of $\mathbf{x}$ d set to 10 and the number of in-context examples H set to 40. For each sequence i, first, we sample $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_H, \mathbf{x}_{H+1} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, I)$, and then sample $f_\tau \in \mathcal{F}$ and assign $y_t = f_\tau(\mathbf{x}_t)$ for $t = 1 \ldots H+1$. We considered two different classes $\mathcal{F}$: 1. **Linear Regression** where $f_\tau(\mathbf{x}_t) = \mathbf{w}_\tau \mathbf{x}_t$ where weight $\mathbf{w}_\tau$ is sampled independently from the isotropic Gaussian distribution $\mathcal{N}(0, I_{d\times d})$. 2. **2NN** Two-layer neural network where $f_\tau(\mathbf{x}_t) = (\mathbf{w}_2^\tau)^T \text{ReLU}((\mathbf{w}_1^\tau)^T \mathbf{x}_t)$ where each parameter is sampled independently from the Gaussian distribution. The hidden state size is set to 100.

In our experiments, we utilize two loss functions: last index loss and validation loss, and we plot by averaging loss

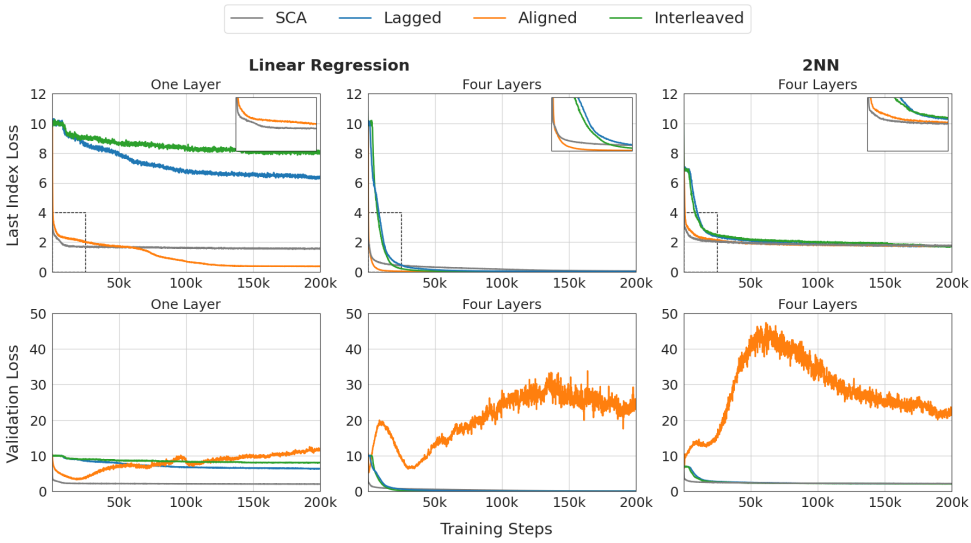


Figure 2. Comparison of Regression Networks. This figure compares four different regression networks: Interleaved, Aligned, Lagged, and SCA, across increasing training epochs for (a) one layer and (b) four layers. The first two plots from the left represent the Linear Regression Task, while the rightmost plot represents the Two-Layer Neural Network Task.

over five independent model runs. The last index loss focuses solely on the prediction at the final position $(H+1)$, corresponding to a single example following H in-context examples. In contrast, the validation loss calculates the mean squared error across a wider range, encompassing positions $2d$ to $H+1$. The validation loss reflects on the model's ability to predict multiple y's given $2d$ in-context examples. In the depicted figures, the last index loss is computed at the 41st position, and the validation loss spans positions 21 to 41.

In Figure 2, we compare four regression networks: Interleaved, Aligned , Lagged, and SCA as shown in Figure 1, across increasing training epochs for linear regression task with one and four layers and 2NN task with four layers. We can make the following observations: (1) SCA converges slightly faster than the Aligned setting, and both achieve similar final loss values during training, except in the one-layer transformer case for the linear regression task. (2) Although both SCA and Aligned models achieve similar final loss values, the Aligned model performs significantly worse in the case of predictions at multiple positions. This is due to the aligned model applying loss only at the last position during training, whereas the other variants apply loss at multiple positions. (3) The shifting in SCA allows for a direct interaction between the y's and $\mathbf{x}$'s in the Lagged setting, leading to better convergence compared to the standard causal attention.

Figure 3 investigates the impact of patch size on the convergence of the SCA variant with one and four-layer transformers. We employed linear attention for this set of experiments.

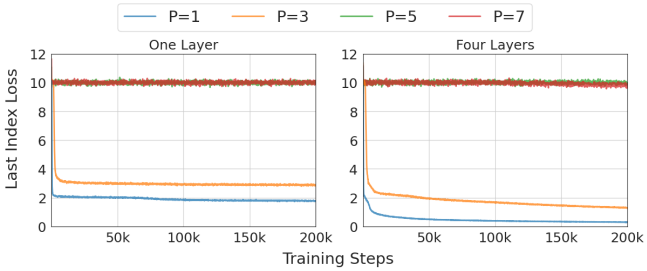


Figure 3. Impact of Patching on Convergence of Regression Networks. Compares SCA with increasing patch size using (a) one layer and (b) four layers.

We observe that increasing patch size hinders in-context learning for a one-layer transformer. This is evident from the slower convergence (less improvement in loss with more training steps) with a larger patch size.

5. Conclusion

In this work, we investigate the in-context learning capabilities of transformer variants based on different training losses and tokenization strategies for in-context examples. We introduce the shifted causal attention (SCA) mechanism, which enables faster convergence and an improved in-context learning capability with just one transformer layer. We empirically and theoretically show that patching, a widely used technique in time series foundation models, hinders in-context learning. As ongoing work, we are experimenting with an FM for TSF that incorporates SCA and a hybrid set of linear and patched layers to provide efficient ICL for time-series data.

References

Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=LziniAXEI9.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., and Wang, Y. Chronos: Learning the language of time series, 2024.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *ICML*, 2024.

Ekambaram, V., Jati, A., Nguyen, N. H., Dayama, P., Reddy, C., Gifford, W. M., and Kalagnanam, J. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series, 2024.

Garg, S., Tsipras, D., Liang, P., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=flNZJ2eOet.

Garza, A. and Mergenthaler-Canseco, M. Timegpt-1, 2023.

Gruver, N., Finzi, M. A., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=md68e8iZK1.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., and Wen, Q. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Lehmann, E. L. and Casella, G. *Theory of point estimation*. Springer Science & Business Media, 2006.

Mahankali, A. V., Hashimoto, T., and Ma, T. 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*, 2024. URL https://openreview.net/forum?id=8p3fu56lKc.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Panwar, M., Ahuja, K., and Goyal, N. In-context learning through the bayesian prism. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HX5ujdsSon.

Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Biloš, M., Ghonia, H., Hassen, N. V., Schneider, A., Garg, S., Drouin, A., Chapados, N., Nevmyvaka, Y., and Rish, I. Lag-llama: Towards foundation models for time series forecasting, 2023.

Van der Vaart, A. W. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *ArXiv*, abs/2306.09927, 2023. URL https://api.semanticscholar.org/CorpusID:259187776.

A. Theoretical results

Let us first set some notation. Let p be the patch size, and n be the total number of in-context examples. So, the in-context examples are given by $z_i := (x_{i1}, y_{i1}, \cdots, x_{ip}, y_{ip}) \in \mathbb{R}^{p(d+1)}$ for $i = 1, \cdots, \frac{n}{p}$. Given a test example x, its prediction by a single layer, single linear attention head is given by:

$$\hat{y} = P \sum_{i=1}^{n/p} (Vz_i) \left((Qx)^\top Kz_i \right),$$

where $Q \in \mathbb{R}^{h \times d}, K \in \mathbb{R}^{h \times p(d+1)}, V \in \mathbb{R}^{h \times p(d+1)}, P \in \mathbb{R}^{1 \times h}$ are the query, key, value and projection matrices respectively, and h is the head dimension. Denoting $U := Q^\top K \in \mathbb{R}^{d \times p(d+1)}$ and $v := V^\top P^\top \in \mathbb{R}^{p(d+1)}$, the above equation can equivalently be written as:

$$\hat{y} = \langle x, \sum_{i=1}^{n/p} U z_i z_i \top v \rangle. \tag{2}$$

We will also denote $\hat{w} := \sum_{i=1}^{n/p} U z_i z_i \top v \in \mathbb{R}^d$. Our main result is the following.

Theorem A.1. *Let x_i for $i \in [n/p]$ be sampled independently from $\mathcal{N}(0, I)$. Let $y_i = \langle w^*, x_i \rangle + \eta_i$ where the noise $\eta_i \sim \sigma \cdot \mathcal{N}(0, 1)$ for some arbitrary w^*. Suppose further that the predictor of a single linear self attention head (2) is unbiased i.e.,*

$$\mathbb{E}_{x_{ij}, \eta_{ij}} \left[\hat{w} | w^* \right] = w^*.$$

Then, we have that $\mathbb{E}_{x_{ij}, \eta_{ij}} \left[\|\hat{w} - w^*\|^2 \right] \geq \frac{p}{2} \cdot \frac{\sigma^2 d}{n}$.

Remark: Note that this rate is suboptimal by a factor of $\frac{p}{2}$ compared to the optimal rate of $\frac{\sigma^2 d}{n}$ for linear least squares regression (Van der Vaart, 2000; Lehmann & Casella, 2006).

Proof. Let us first denote $U = \left[U_{x_1}, u_{y_1}, \cdots, U_{x_p}, u_{y_p} \right]$, where each $U_{x_j} \in \mathbb{R}^{d \times d}$ and $u_{y_j} \in \mathbb{R}^d$ and $v = \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ \vdots \\ v_{x_p} \\ v_{y_p} \end{bmatrix}$, where each $v_{x_j} \in \mathbb{R}^d$ and $v_{y_j} \in \mathbb{R}$ for each $j = 1, \cdots, p$. We now compute $\mathbb{E}[\hat{w}]$ and see what the unbiasedness property tells us. We have that (after dropping the subscript i in x_{ij} and η_{ij}due to linearity of expectation):

$$\begin{aligned}
\mathbb{E}[\hat{w}] &= \frac{n}{p} \cdot \mathbb{E} \left[\left(\sum_{j=1}^{p} v_{y_j} \left(\langle w^*, x_j \rangle + \eta_j \right) + \langle v_{x_j}, x_j \rangle \right) \left(\sum_{j=1}^{p} U_{x_j} x_j + \left(\langle w^*, x_j \rangle + \eta_j \right) u_{y_j} \right) \right] \\
&= \frac{n}{p} \cdot \sum_{j,k=1}^{p} \left(v_{y_k} U_{x_j} + u_{y_k} v_{x_j}^\top \right) \mathbb{E} \left[x_j x_k^\top \right] w^* + \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right) \mathbb{E} \left[\eta_j x_k \right] \\
&\qquad + U_{x_j} \mathbb{E} \left[x_j x_k^\top \right] v_{x_k} + v_{y_j} u_{y_k} \mathbb{E} \left[\left(\langle w^*, x_j \rangle + \eta_j \right) \left(\langle w^*, x_k \rangle + \eta_k \right) \right] \\
&= \frac{n}{p} \cdot \left(\sum_{j=1}^{p} \left(v_{y_j} U_{x_j} + u_{y_j} v_{x_j}^\top \right) w^* + U_{x_j} v_{x_j} + v_{y_j} u_{y_j} \left(1 + \|w^*\|^2 \right) \right).
\end{aligned}$$

The unbiasedness condition that $\mathbb{E}\left[\hat{w}|w^*\right] = w^*$ for any w^* implies that:

$$\sum_{j=1}^{p} \left(v_{y_j} U_{x_j} + u_{y_j} v_{x_j}^\top \right) = \frac{p}{n} \cdot Id \tag{3}$$

$$\sum_{j=1}^{p} U_{x_j} v_{x_j} = 0, \text{ and} \tag{4}$$

$$\sum_{j=1}^{p} u_{y_j} v_{y_j} = 0, \tag{5}$$

where Id is the $d \times d$ identity matrix. We now wish to compute $\|\hat{w} - w^*\|^2$. To do so, let us first decompose $\hat{w} - w^*$ as:

$$\hat{w} - w^* = \alpha + \beta,$$

where:

$$\begin{aligned}
\alpha :=& \left(\left(\sum_{i=1}^{n/p} \sum_{j,k=1}^{p} \left(v_{y_k} U_{x_j} + u_{y_k} v_{x_j}^\top \right) x_{ij} x_{ik}^\top \right) - Id \right) w^* + \left(\sum_{i=1}^{n/p} \sum_{j,k=1}^{p} U_{x_j} x_{ij} x_{ik}^\top v_{x_k} \right) \\
&+ \left(\sum_{i=1}^{n/p} \sum_{j,k=1}^{p} v_{y_j} u_{y_k} \left(\langle w^*, x_{ij} \rangle + \eta_{ij} \right) \left(\langle w^*, x_{ik} \rangle + \eta_{ik} \right) \right), \text{ and} \\
\beta :=& \sum_{i=1}^{n/p} \sum_{j,k=1}^{p} \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right) \eta_{ij} x_{ik}.
\end{aligned}$$

Note that $\mathbb{E}_{x_{ij},\eta_{ij}}\left[\alpha\right] = \mathbb{E}_{x_{ij},\eta_{ij}}\left[\beta\right] = 0$ by the unbiasedness assumption, while $\mathbb{E}_{x_{ij},\eta_{ij}}\left[\langle \alpha, \beta \rangle\right] = 0$ since x_{ij}, η_{ij} are all independent for all $i \in [n/p]$ and $j \in [p]$ with $x_{ij} \sim \mathcal{N}(0, Id)$ and $\eta_{ij} \sim \mathcal{N}(0, \sigma^2)$. So, we have that:

$$\mathbb{E}\left[\|\hat{w} - w^*\|^2 \right] = \mathbb{E}\left[(\alpha + \beta)^2 \right] = \mathbb{E}\left[\alpha^2\right] + \mathbb{E}\left[\beta^2\right] \geq \mathbb{E}\left[\beta^2\right].$$

So, it suffices to lower bound $\mathbb{E}\left[\beta^2\right]$. We have that:

$$\begin{aligned}
\mathbb{E}\left[\beta^2\right] &= \sum_{i=1}^{n/p} \sum_{j,k=1}^{p} \mathbb{E}\left[\eta_{ij}^2 x_{ik}^\top \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right)^\top \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right) x_{ik} \right] \\
&= \frac{\sigma^2 n}{p} \sum_{j,k=1}^{p} \langle \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right) \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right)^\top, \mathbb{E}\left[x_{ik} x_{ik}^\top \right] \rangle \\
&= \frac{\sigma^2 n}{p} \sum_{j,k=1}^{p} \mathrm{Tr}\left(\left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right) \left(v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right)^\top \right) \\
&= \frac{\sigma^2 n}{p} \sum_{j,k=1}^{p} \left\| v_{y_j} U_{x_k} + u_{y_j} v_{x_k}^\top \right\|_F^2 \\
&= \frac{\sigma^2 n}{p} \left(\left(\sum_{j=1}^{p} v_{y_j}^2 \right) \left(\sum_{k=1}^{p} \|U_{x_k}\|_F^2 \right) + \left(\sum_{j=1}^{p} \left\| u_{y_j} \right\|^2 \right) \left(\sum_{k=1}^{p} \|v_{x_k}\|^2 \right) + \langle \sum_{j=1}^{p} u_{y_j} v_{y_j}, \sum_{k=1}^{p} U_{x_k} v_{x_k} \rangle \right) \\
&= \frac{\sigma^2 n}{p} \left(\left(\sum_{j=1}^{p} v_{y_j}^2 \right) \left(\sum_{k=1}^{p} \|U_{x_k}\|_F^2 \right) + \left(\sum_{j=1}^{p} \left\| u_{y_j} \right\|^2 \right) \left(\sum_{k=1}^{p} \|v_{x_k}\|^2 \right) \right),
\end{aligned}$$

where we used the fact that $\sum_{j=1}^{p} u_{y_j} v_{y_j} = 0$ from (5) in the last step. Denoting by $v_y := \begin{bmatrix} v_{y_1} \\ \vdots \\ v_{y_p} \end{bmatrix} \in \mathbb{R}^p$, $v_x(\ell) := \begin{bmatrix} v_{x_1}(\ell) \\ \vdots \\ v_{x_p}(\ell) \end{bmatrix} \in \mathbb{R}^p$, $u_y(\ell) := \begin{bmatrix} u_{y_1}(\ell) \\ \vdots \\ u_{y_p}(\ell) \end{bmatrix} \in \mathbb{R}^p$ and $U_x(\ell) := \begin{bmatrix} U_{x_1}(\ell,\ell) \\ \vdots \\ U_{x_p}(\ell,\ell) \end{bmatrix} \in \mathbb{R}^p$ for every $\ell \in [d]$, where $A(\ell_1, \ell_2)$ the $(\ell_1, \ell_2)^{\text{th}}$ element of the matrix A and by $b(\ell)$, the ℓ^{th} element of the vector b, we recase Equation (3) as:

$$\langle v_y, U_x(\ell) \rangle + \langle u_y(\ell), v_x(\ell) \rangle = \frac{p}{n}, \tag{6}$$

for every $\ell \in [d]$.

We now have that:

$$\begin{aligned}
\mathbb{E}\left[\beta^2\right] &\geq \frac{\sigma^2 n}{p} \left(\left(\sum_{j=1}^{p} v_{y_j}^2 \right) \left(\sum_{k=1}^{p} \|U_{x_k}\|_F^2 \right) + \left(\sum_{j=1}^{p} \left\|u_{y_j}\right\|^2 \right) \left(\sum_{k=1}^{p} \|v_{x_k}\|^2 \right) \right) \\
&\geq \frac{\sigma^2 n}{p} \left(\|v_y\|^2 \left(\sum_{\ell=1}^{d} \|U_x(\ell)\|_F^2 \right) + \left(\sum_{\ell=1}^{d} \|u_y(\ell)\|^2 \right) \left(\sum_{\ell=1}^{d} \|v_x(\ell)\|^2 \right) \right) \\
&\geq \frac{\sigma^2 n}{p} \sum_{\ell=1}^{d} \|v_y\|^2 \|U_x(\ell)\|_F^2 + \|u_y(\ell)\|^2 \|v_x(\ell)\|^2 \\
&\geq \frac{\sigma^2 n}{2p} \sum_{\ell=1}^{d} \langle v_y, U_x(\ell) \rangle + \langle u_y(\ell), v_x(\ell) \rangle \geq \frac{\sigma^2 d n}{2p} \cdot \left(\frac{p}{n} \right)^2 \\
&= \frac{p}{2} \cdot \frac{\sigma^2 d}{n},
\end{aligned}$$

where the last step follows from Equation (6). This proves the theorem. □