TEAFormers: TEnsor-Augmented Transformers for Multi-Dimensional Time Series Forecasting

Linghang Kong[1] , **Elynn Chen**[2*] , **Yuzhou Chen**[3] and **Yuefeng Han**[4]
[1]New York University, New York, NY 10012, USA
[2]New York University, New York, NY 10012, USA
[3]University of California, Riverside, Riverside, CA 92521, USA
[4]Notre Dame University, Notre Dame, IN 46556, USA

Abstract

Multi-dimensional time series data, common in fields like economics, finance, and climate science, pose challenges for traditional Transformers, which flatten multi-dimensional structures and lose critical relationships. To address this, we propose TEAFormer, a novel Transformer framework that preserves and leverages multi-dimensional structures through a Tensor-Augmentation (TEA) module. The TEA module employs tensor expansion for enhanced multi-view feature learning and tensor compression for efficient information aggregation, reducing computational costs while improving prediction accuracy. As a versatile component, the TEA module integrates seamlessly with existing Transformer architectures. Experiments incorporating TEA into three popular Transformer models across real-world benchmarks demonstrate significant performance gains, showcasing TEAFormer's potential for advanced time series forecasting.

1 Introduction

In the era of big data, multi-dimensional time series data, such as matrix and tensor-valued time series, are increasingly common in applications like economics, finance, and climate science. For instance, policymakers track quarterly economic indicators like GDP growth and inflation across multiple countries [Chen *et al.*, 2020a]; investors monitor financial metrics such as asset/equity ratios and revenue from various companies [Chen and Fan, 2023]; and scientists observe hourly environmental variables like PM2.5 and ozone levels at multiple stations [Chen *et al.*, 2020b; Chen *et al.*, 2024f]. All these datasets naturally present themselves as time series of matrices (order-2 tensors). Transformers [Vaswani *et al.*, 2017] have demonstrated substantial promise in modeling sequential data such as language[Zhang *et al.*, 2018], text[Guo *et al.*, 2022], and time series [Wen *et al.*, 2023]. However, existing Transformer models fail to preserve the tensor structure of data. The calculations within these architectures effectively flatten multi-dimensional observations into vectors, losing the inherent multi-dimensional relationships and patterns. This paper fills the gap by introducing TEAFormer, a novel approach that preserves and leverages multi-dimensional tensor structures through a Tensor-Augmented (TEA) module, offering multiple benefits: **1)** Tensor expansion aggregates multi-view features from multiple channels, including those from multi-head attention, enhancing information capture. **2)** Tensor compression reduces computational costs via automatic tensor decomposition, compressing expanded features into a core tensor for efficient self-attention without sacrificing performance. **3)** The TEA module is a versatile component compatible with attention mechanisms and encoder-decoder structures, making it adaptable to any Transformer architecture. TEAFormer achieves significant computational savings while effectively modeling intricate tensor dependencies. We incorporate the Tensor-Augmentation module into three widely used time series Transformer models: Transformer [Vaswani *et al.*, 2017], Informer [Zhou *et al.*, 2021], and Autoformer [Wu *et al.*, 2021]. We conduct extensive experiments across a diverse range of datasets, demonstrating that our approach not only reduces computational costs but also enhances prediction accuracy. Our results show substantial improvements in evaluation metrics, such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) scores, compared to baseline models. This underscores the potential of tensor decomposition as a powerful tool for advancing matrix and tensor time series forecasting. Recent statistical studies have explored the benefits of maintaining the multi-dimensional structure in tensor time series using linear factor models [Wang *et al.*, 2019; Chen *et al.*, 2020a; Chen and Fan, 2023; Chen *et al.*, 2024f; Chen *et al.*, 2022b; Han *et al.*, 2020; Han *et al.*, 2022; Han and Zhang, 2023; Han *et al.*, 2024b] and linear vector auto-regressive (VAR) models [Chen *et al.*, 2021; Xiao *et al.*, 2023; Li and Xiao, 2021]. They have shown great advantages of preserving the multi-dimensional structure in tensor time series. However, all of them adopt linear approaches and have limited capacity to capture potential nonlinear relationships, especially in applications involving language and text. This work is the first to formally introduce a TEnsor-Augmented Transformer (TEAFormer), distinct from statistical VAR models. Maintaining tensor structure in Transformers is much more challenging due to the architecture's complexity, including encoding, decoding, and multi-head atten-

*Corresponding author

tion mechanisms, as detailed in Section 2. Our contributions are summarized in three key points:

- *Introduction of TEAFormer.* We introduce TEAFormer, a novel multi-dimensional (tensor) time series Transformer architecture that integrates a Tensor-Augmentation module with any Transformer-based models. This is the first work to utilize tensor learning in Transformer.
- *Development of Tensor-Augmentation Module.* We develop a Tensor-Augmentation module that involves tensor expansion and compression, effectively aggregating information through multi-view feature learning while reducing the computational burden associated with self-attention and information aggregation.
- *Evaluation of Appliction to Existing Tranformers.* Our extensive experimental results show that all implemented TEAFormers outperform their baseline counterparts in 61 out of 78 trials across three real-world benchmarks.

The rest of this paper is organized as follows: Section 1.1 shows related work on Neural Networks and Transformer models for time series forecasting. Section 2 provides a detailed description of the TEAFormer architecture and its integration of tensor decomposition. In Section 3, we present our experimental setup, including datasets, evaluation metrics, and implementation details, followed by a discussion of the results. Finally, Section 4 concludes the paper and outlines potential directions for future research.

1.1 Related Work

Multi-dimensional Time Series Analysis. The key challenge in multi-dimensional time series analysis is capturing complex inter-variable relationships. Traditional methods address high dimensionality by vectorizing tensor time series and applying factor models, which effectively capture common dependencies among variables [Bai, 2013; Bai, 2003; Bai and Ng, 2002; Chen *et al.*, 2024a]. Recent research has extended factor model methodologies to matrix- and tensor-variate time series analysis. These studies predominantly employ Tucker low-rank structures [Chen and Fan, 2023; Chen *et al.*, 2023; Chen *et al.*, 2020a; Chen *et al.*, 2024f; Chen *et al.*, 2022a; Chen *et al.*, 2022b; Han *et al.*, 2020; Han *et al.*, 2022; Yu *et al.*, 2024; Zhou *et al.*, 2024] and Canonical Polyadic (CP) low-rank structures [Chang *et al.*, 2023; Chen *et al.*, 2024b; Han *et al.*, 2024b; Han and Zhang, 2023]. Additionally, researchers have developed matrix- and tensor-variate autoregressive models with bilinear or multilinear forms, supported by theoretical guarantees demonstrating that preserving multi-dimensional structure enhances performance [Chen *et al.*, 2021; Hoff, 2015; Li and Xiao, 2021; Li and Xiao, 2024; Xiao *et al.*, 2023; Chen *et al.*, 2020b; Liu and Chen, 2022; Chen and Chen, 2022; Chen *et al.*, 2024c; Han *et al.*, 2024a]. The computer science literature focuses primarily on empirical approaches, leveraging neural network architectures for forecasting. Variables exhibit distinct patterns and cycles, interacting through both lagged relationships and simultaneous influences. While CNNs and RNNs have been traditional tools, recent innovations include models like DeepGLO [Sen *et al.*, 2019], which employs multiple Temporal Convolutional Networks (TCNs) to capture global and local dependencies. Graph Neural Networks (GNNs) represent another promising direction: STGCL [Liu *et al.*, 2022] utilizes contrastive learning at both graph and node levels for spatiotemporal GNNs, while GNN-RNN [Fan *et al.*, 2022] processes diverse data types through CNNs and RNNs before using GraphSage GNN to extract geospatial representations. CausalGNN [Wang *et al.*, 2022] combines embeddings into latent matrix representations and integrates them with causal embeddings using GNN-based non-linear transformations for disease forecasting. However, incorporating tensor structures, such as Tucker or CP low-rank constraints, into transformer architectures remains an unexplored area, highlighting a key gap in the literature. **Transformer for Multivariate Time Series Forecasting.** Transformers [Vaswani *et al.*, 2017] offer significant advantages for handling sequence data, particularly due to their attention mechanisms, which allow for greater parallelization and faster training times on modern hardware. Informer [Zhou *et al.*, 2021] modifies the traditional transformer architecture for time series forecasting by using a *probsparse* attention mechanism for improved computational efficiency and a generative decoder to reduce complexity and prevent cumulative errors. Building on Informer, Spacetimeformer [Grigsby *et al.*, 2021] incorporates spatial correlations by adding time and space information during embedding, calculating both global and local attention. Autoformer [Wu *et al.*, 2021] enhances efficiency and decomposition ability through an autocorrelation mechanism and time series decomposition methods. Fedformer [Zhou *et al.*, 2022] leverages signal processing techniques, using Fourier transformation for global information decomposition. ETSformer [Woo *et al.*, 2022] combines traditional time series decomposition with exponential smoothing, creating interpretable and decomposable forecasting outputs. Crossformer [Zhang and Yan, 2022] introduces a two-stage attention mechanism, computing attention along the time axis first and then along the dimension axis, with an innovative router mechanism to manage attention distribution and reduce time complexity. DSformer [Yu *et al.*, 2023] collects global and local information through sampling methods, rearranging data along the time and dimension axes. **Tensor-Augmented Neural Networks.** While neural networks can effectively process high-dimensional tensor inputs, most existing approaches utilize tensors primarily for computational efficiency rather than leveraging their statistical properties. [Cohen *et al.*, 2016] established a foundational connection between deep networks and hierarchical tensor factorizations. Building on this, tensor contraction layers [Kossaifi *et al.*, 2017] and regression layers [Kossaifi *et al.*, 2020] were developed to regularize models, successfully reducing parameter counts while preserving model accuracy. Recent innovations, including the Graph Tensor Network [Xu *et al.*, 2023] and Tensor-view Topological Graph Neural Network [Wen *et al.*, 2024], have introduced novel frameworks for processing large-scale, multi-dimensional data. Additionally, significant progress has been made in uncertainty quantification and representation learning for tensor-augmented neural networks

[Chen *et al.*, 2024d; Chen *et al.*, 2024e; Wu *et al.*, 2024a; Wu *et al.*, 2024b; Wu *et al.*, 2024c]. Despite these advances, the development of tensor-augmented transformers specifically designed for multi-dimensional time series remains an open challenge.

2 Tensor-Augmented Transformer

For a clear presentation, we consider the most common case of a two-dimensional matrix-variate time series $\boldsymbol{X}_t \in \mathbb{R}^{D_1 \times D_2}$ for $t \in [L]$. In sequential forecasting, one aims to predict the future values $\boldsymbol{X}_{t+1}$ based on historical observations $\boldsymbol{X}_1, ..., \boldsymbol{X}_t$.

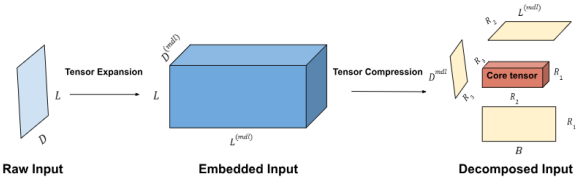

Figure 1: The structure of tensor-augmentation module when $E = 1$ and $M = 1$.

We preserve the inherent multi-dimensional structure in the TEAFormer. For matrix-variate time series, we let $\mathcal{X}^{(raw)} \in \mathbb{R}^{L \times D_1 \times D_2}$ be one sample of raw input data, where L denotes the sequence length, D_1 and D_2 denote the sizes of two-dimensional matrix observation.

Embedding with Tensor Mode and Size Expansion. During embedding, we transform the raw input $\mathcal{X}^{(raw)}$ to feature $\mathcal{X} \in \mathbb{R}^{L \times L^{(mdl)} \times D^{(mdl)}}$. Here, the temporal embedded dimension $L^{(mdl)}$ is not restricted to a scalar value of one-dimension. Instead, it can represent multiple dimensions $L^{(mdl)} = L_1^{(mdl)} \times \cdots \times L_E^{(mdl)}$, where $E \geq 1$ is the total number of modes extended from the temporal mode of $\mathcal{X}^{(raw)}$. For example, CrossFormer [Zhang and Yan, 2022] expands the raw temporal mode of length L to $L^{(mdl)} = L_1^{(mdl)} \times L_2^{(mdl)}$ where $L_1^{(mdl)}$ represents the number of sub-sequences and $L_2^{(mdl)}$ represents the embedding of each sub-sequence. This corresponding to *Tensor mode expansion*.

Analogously, the feature embedded dimensions $D^{(mdl)} := D_1^{(mdl)} \times \cdots \times D_M^{(mdl)}$ can be multi-dimensional with $M \geq 2$. If $M > 2$ where 2 is the number of dimensions of the raw time series observation $\boldsymbol{X}_t$, the embedding results in a *mode-expanded feature space*, or *Tensor mode expansion*, that is the number of modes increases. If $\Pi_{m=1}^{M} D_m^{(mdl)} > \Pi_{m=1}^{2} D_m$, this embedding results in a *size-expanded feature space*, or *Tensor size expansion*, that is the size of the feature space increases. For example, CrossFormer [Zhang and Yan, 2022] expands the raw temporal mode of length L to $L^{(mdl)} = L_1^{(mdl)} \times L_2^{(mdl)}$ by segmenting the L-length sequence to S-number of sub-sequences of size L/S. The embedding also results in a *size-expanded feature space* since $L_1^{(mdl)} = S$ and $L_2^{(mdl)} > L/S$. Finally, we note that this setting also include the classic transformer as sub-cases. Specifically, for vector (order-1 tensor) data, $D^{(mdl)} := D_1^{(mdl)}$ corresponds to the common transformer; for matrix (order-2 tensor) data $D^{(mdl)} := D_1^{(mdl)} \times D_2^{(mdl)}$ corresponding to matrix time series without *Tensor mode expansion*.

Tensor-Augmented Sequence-to-Sequence Multi-Head Attention (MHA). We first characterize the formulation of a Single-Head Attention (SHA). The embedded feature is $\mathcal{X} \in \mathbb{R}^{L \times L^{(mdl)} \times D^{(mdl)}}$. One attention head is structured as a tensor with dimension expressed as $D^{(attn)} := D_1^{(attn)} \times \cdots \times D_M^{(attn)}$. The value weight tensor is $\mathcal{W}_V \in \mathbb{R}^{L^{(mdl)} \times D^{(mdl)} \times D^{(attn)}}$, the query and key weight tensors are $\mathcal{W}_Q, \mathcal{W}_K \in \mathbb{R}^{L^{(mdl)} \times D^{(mdl)} \times D^{(attn)}}$, and the output weight tensor is $\mathcal{W}_O \in \mathbb{R}^{D^{(attn)} \times L^{(mdl)} \times D^{(mdl)}}$, all of which are trainable. The output of one layer of the Transformer is a $(L \times L^{(mdl)} \times D^{(mdl)})$-dimensional tensor, and can be described as follows:

$$\mathcal{T}^{\mathrm{SHA}}(\mathcal{X}; \mathcal{W}_Q, \mathcal{W}_K, \mathcal{W}_V, \mathcal{W}_O) := \sigma\left(\mathrm{RowSoftmax}\left(\left\langle \langle \mathcal{X}, \mathcal{W}_Q \rangle, \langle \mathcal{X}, \mathcal{W}_K \rangle^{\top} \right\rangle\right) \langle \mathcal{X}, \mathcal{W}_V \rangle\right) \mathcal{W}_O, \tag{1}$$

where $\langle \mathcal{X}, \mathcal{W}_Q \rangle$, $\langle \mathcal{X}, \mathcal{W}_K \rangle$, $\langle \mathcal{X}, \mathcal{W}_V \rangle$ denote tensor inner products that result in a $(L \times D^{(attn)})$-dimensional matrix, $[\mathrm{RowSoftmax}(\boldsymbol{M})]_{\ell,:} := \mathrm{softmax}(\boldsymbol{M}_{\ell,:})$ that runs softmax on row of its input, and σ is a L_σ Lipshitz activation function that is applied element-wise and has the property $\sigma(0) = 0$.

For MHA, let H denote the number of head, we collect all weight tensors in $\mathcal{W} := \{\mathcal{W}_{Q,h}, \mathcal{W}_{K,h}, \mathcal{W}_{V,,h}, \mathcal{W}_{O,h}\}_{h=1}^{H}$ for all heads $h \in [H]$ and define an extra head weight vector $\boldsymbol{w}_H$ of dimension H. One layer of the Transformer with MHA can be described as $\mathcal{T}$

$$\mathcal{T}(\mathcal{X}; \mathcal{W}, \boldsymbol{w}_H) = [\mathcal{T}^{\mathrm{SHA}}(\mathcal{X}; \mathcal{W}_{Q,h}, \mathcal{W}_{K,h}, \mathcal{W}_{V,,h}, \mathcal{W}_{O,h})]_{:::h} \times_4 \boldsymbol{w}_H. \tag{2}$$

The dimension of $\mathcal{T}(\mathcal{X})$ is $L \times L^{(mdl)} \times D^{(mdl)}$. Thus, a S-multi-layer Transformer can be constructed by iteratively compose $\mathcal{T}(\mathcal{X})$ for S times, denoted as

$$\mathcal{T}_S(\mathcal{X}; \{\mathcal{W}^{(s)}, \boldsymbol{w}_H^{(s)}\}_{s=0}^{S}) := \mathcal{T}^{(S)} \circ \cdots \circ \mathcal{T}^{(0)}(\mathcal{X}), \tag{3}$$

where $\mathcal{T}^{(s)} = (\cdot; \mathcal{W}^{(s)}, \boldsymbol{w}_H^{(s)})$.

In our study, multi-head self-attention (MSA) is used. MSA is a specific type of MHA, in which queries, keys, and values all come from the same sequence, i.e. $\mathcal{W}_Q = \mathcal{W}_K = \mathcal{W}_V$, thus allowing the model to capture dependencies within that sequence.

Automated Information Aggregation and Compression. The initial feature input and the hidden throughputs are all of order-$(1 + E + M)$ tensor structure of dimension $L \times L^{(mdl)} \times D^{(mdl)}$ where E is the number of modes of the hidden temporal embedding and M is the number of modes of the feature model embedding. Current transformers treat all dimensions equally and carry out calculation by flattening all the tensors. The idea of this paper is to preserve the tensor (multi-dimensional) structure. As such, we are able to auto-

mate information aggregation and information compression through tensor expansion and compression.

Tensor compression refers to the procedure that incorporates low-rank tensor decomposition in the procedure, which can only be achieved when we keep the tensor structure in (2). We incorporate tensor low-rank structures such as *CP low-rank*, *Tucker low-rank*, and *Tensor Train low-rank*.

Tucker low-rank structure is defined by

$$\mathcal{X} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \cdots \times_M \boldsymbol{U}_M + \mathcal{E}, \tag{4}$$

where $\mathcal{E} \in \mathbb{R}^{D_1^{(mdl)} \times \cdots \times D_M^{(mdl)}}$ is the tensor of the idiosyncratic component (or noise) and $\mathcal{C}$ is the latent core tensor representing the true low-rank feature tensors and $\boldsymbol{U}_m$, $m \in [M]$, are the loading matrices.

CP low-rank is a special case where the core tensor $\mathcal{C}$ has the same dimensions over all modes, that is $R_m = R$ for all $m \in [M]$, and is super-diagonal. TT low-rank is a different kind of low-rank structure, which inherits advantages from both CP and Tucker decomposition. Specifically, TT decomposition can compress tensors as significantly as CP decomposition, while its calculation is as stable as Tucker decomposition.

Practical Experiment Implementation. To our knowledge, no Transformer-based time series forecasting model incorporates more than one hidden dimension in the $D^{(mdl)}$ set. Despite variations in embedding methods, models consistently produce three-dimensional embeddings based on sequence length or features. While tensor-augmented multi-head attention could theoretically handle higher dimensions, our experiments are limited to scenarios with a single hidden dimension and three-dimensional embedded data.

Before passing into the attention block, the embedded input as a three-dimensional tensor will be passed into a Tensor-Augmentation module to be decomposed using Tucker decomposition method and be transformed into a group of factorized tensors.

$$\begin{aligned} \mathcal{X} &= Embed(\mathcal{X}^{(raw)}), \\ \mathcal{X} &= \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3, \end{aligned} \tag{5}$$

where $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times R_3}, R_1 \ll L, R_2 \ll L^{(mdl)}, R_3 \ll D^{(mdl)}$ is a core tensor that acts as a framework to capture the complex interactions between cross-dimensional and temporal features across the dataset, thus could be leveraged as a latent representation for global information in MTS. $\boldsymbol{U}_1 \in \mathbb{R}^{L \times R_1}, \boldsymbol{U}_2 \in \mathbb{R}^{L^{(mdl)} \times R_2}, \boldsymbol{U}_3 \in \mathbb{R}^{D^{(mdl)} \times R_3}$ are factor matrices. They can extract the most predominant features in each dimension while maintaining the original dimension's structure.

To reduce the computational cost of self-attention while capturing global interactions within the data, we pass the smaller core tensor, which preserves essential information, into the attention block for computation. To be more specific, in each encoder layer, we have,

$$\begin{aligned} \widehat{\mathcal{C}} &= LayerNorm(\mathcal{C} + MSA(\mathcal{C}, \mathcal{C}, \mathcal{C})), \\ \widehat{\mathcal{X}}^{enc} &= \widehat{\mathcal{C}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3, \end{aligned} \tag{6}$$

where $\widehat{\mathcal{X}}^{enc}$ is the output of the current encoder layer. And the rest of the model structure will be the widely adopted Transformer encoder-decoder architecture.

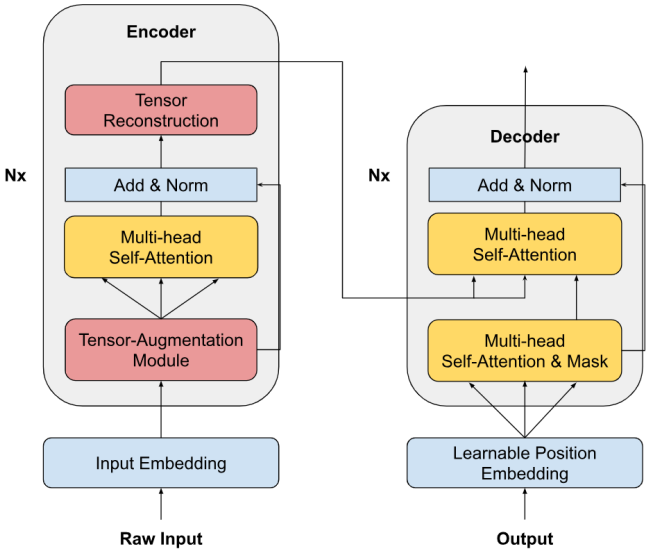

Figure 2: The architecture of TEAFormer. The input embedding is a general reference to any positional, segment, or temporal embedding that transforms raw input into 3D shape.

3 Experiments

Datasets We conduct experiments on three real world datasets that are very popular in recent multi-dimensional time series forecasting studies: (1) ETTh1: The temperature of electricity transformers (ETT)[1] is a vital metric in the long-term deployment of electric power. This dataset contains two years of data from two different counties in China. The ETTh1 subset is the data on 1-hour-level granularity; (2) ETTm1: The subset of ETT dataset on 15-minutes-level granularity; (3) WTH[2]: This dataset includes local climatological data for almost 1,600 locations across the United States, spanning four years from 2010 to 2013. (4) Exchange[Lai *et al.*, 2018]: This dataset records the daily exchange rates of 8 different countries from 1990 to 2016; (5) ILI[3]: National illness dataset that records influenza-likeillness(ILI) patients data between 2002 and 2021 in the United States; (6) Electricity [4]: Dataset that contains the hourly electricity consumption of 321 customers from 2012 to 2014.

Baselines In this paper, we use three popular attention-based encoder-decoder structure multi-dimensional time series forecasting models as our baseline: (1) Transformer [Vaswani *et al.*, 2017], (2) Informer [Zhou *et al.*, 2021], and (3) Autoformer [Wu *et al.*, 2021].

[1]https://github.com/zhouhaoyi/ETDataset
[2]https://www.ncei.noaa.gov/data/local-climatological-data/
[3]https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
[4]https://pems.dot.ca.gov/

Table 1: Performance comparison between baselines and TEA-based models.

Dataset	SeqLen	Informer		TEA-Informer		Autoformer		TEA-Autoformer		Transformer		TEA-Transformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	0.577	0.549	**0.500**	**0.517**	0.384	0.425	**0.372**	**0.412**	0.592	0.572	**0.587**	**0.545**
	48	0.685	0.625	**0.589**	**0.578**	0.392	**0.419**	**0.388**	0.431	0.854	0.721	**0.784**	**0.660**
	168	**0.931**	**0.752**	0.968	0.775	0.490	0.481	**0.480**	**0.479**	1.047	0.833	**0.944**	**0.789**
	336	**1.128**	0.873	1.134	**0.857**	0.505	0.487	**0.495**	**0.484**	1.168	**0.880**	**1.147**	0.884
	720	1.215	0.896	**1.200**	**0.883**	**0.498**	**0.500**	0.527	0.526	**1.073**	**0.822**	1.107	0.869
ETTm1	24	0.323	0.369	**0.291**	**0.367**	**0.383**	**0.403**	0.404	0.429	0.316	0.371	**0.312**	**0.371**
	48	0.494	0.503	**0.400**	**0.434**	**0.454**	**0.459**	0.462	0.461	0.492	0.477	**0.479**	**0.475**
	96	0.678	0.614	**0.485**	**0.470**	0.481	0.463	**0.470**	**0.459**	0.619	0.581	**0.566**	**0.528**
	288	1.056	0.786	**0.954**	**0.740**	0.634	0.528	**0.560**	**0.510**	0.962	0.767	**0.884**	**0.731**
	672	1.192	0.926	**1.016**	**0.792**	0.606	0.542	**0.546**	**0.519**	1.168	0.837	**0.897**	**0.720**
WTH	96	0.552	0.535	**0.548**	**0.526**	0.266	0.366	**0.263**	**0.336**	0.450	0.466	**0.350**	**0.407**
	192	0.616	0.584	**0.601**	**0.551**	0.307	0.367	**0.306**	**0.366**	0.587	0.543	**0.583**	**0.543**
	336	0.702	0.620	**0.599**	**0.562**	0.359	0.395	**0.358**	**0.395**	0.648	0.576	**0.635**	**0.549**
	720	0.831	0.731	**0.610**	**0.581**	**0.419**	**0.428**	0.467	0.461	0.932	0.708	**0.711**	**0.586**
Exchange	96	**0.914**	**0.768**	1.113	0.829	0.143	0.274	**0.142**	**0.272**	0.658	0.618	**0.636**	**0.611**
	192	1.288	0.830	**1.210**	**0.830**	0.272	0.380	**0.261**	**0.375**	**1.162**	**0.832**	1.251	0.856
	336	1.794	1.071	**1.483**	**0.998**	0.455	0.505	**0.452**	**0.502**	1.795	1.083	**1.683**	**1.060**
	720	2.940	1.415	**1.326**	**0.910**	1.094	0.813	**1.088**	**0.811**	2.217	1.166	**1.735**	**1.075**
Illness	24	**5.086**	**1.540**	5.964	1.704	3.491	1.307	**3.226**	**1.264**	4.670	1.414	**4.306**	**1.343**
	36	**5.038**	**1.551**	5.328	1.583	3.573	1.297	**3.282**	**1.269**	4.560	1.396	**4.551**	**1.396**
	48	**5.012**	**1.551**	5.297	1.580	3.406	1.291	**3.243**	**1.236**	4.860	1.469	**4.704**	**1.425**
	60	5.388	1.604	**5.248**	**1.604**	2.884	1.184	**2.867**	**1.160**	5.018	1.500	**4.789**	**1.453**
Electricity	96	0.289	0.386	**0.283**	**0.386**	**0.201**	**0.316**	0.213	0.328	0.306	0.399	**0.261**	**0.371**
	192	0.335	0.422	**0.334**	**0.421**	0.232	0.344	**0.231**	**0.342**	0.326	0.416	**0.310**	**0.400**
	336	**0.340**	**0.455**	0.520	0.599	0.241	0.350	**0.240**	**0.349**	0.459	0.498	**0.347**	**0.427**
	720	**0.574**	**0.580**	0.619	0.657	**0.259**	**0.364**	0.308	0.397	0.508	0.536	**0.439**	**0.493**

Experiment Setup To better demonstrate our tensor-augmented attention module's performance, we directly utilized the hyperparameter settings in the baseline papers so that we are able to make comparisons on the performance of the baseline model's finest state. In Autoformer [Wu *et al.*, 2021], Informer was used as a baseline to be compared with, but the hyperparameters were tuned significantly different from the original Informer [Zhou *et al.*, 2021]. In this case, we use the hyperparameter settings from the Informer paper. The prediction windows size L_y is set as follow: 1d, 2d, 7d, 14d, 30d for ETTh1 and ETTm1, and 4d, 8d, 14d, 30d for Weather.

Evaluation Metrics We use two evaluation matrics: Mean Square Error and Mean Absolute Error,

$$\text{MSE} = \frac{1}{LL^{(mdl)}D^{(mdl)}} \sum_{t=1}^{L} \sum_{i=1}^{L^{(mdl)}} \tag{7}$$

$$\sum_{j=1}^{D^{(mdl)}} \|\boldsymbol{X}_{t,ij} - \widehat{\boldsymbol{X}}_{t,ij}\|_2^2, \tag{8}$$

$$\text{MAE} = \frac{1}{LL^{(mdl)}D^{(mdl)}} \sum_{t=1}^{L} \sum_{i=1}^{L^{(mdl)}} \sum_{j=1}^{D^{(mdl)}} \tag{9}$$

$$|\boldsymbol{X}_{t,ij} - \widehat{\boldsymbol{X}}_{t,ij}| \tag{10}$$

3.1 Experiment Results and Analysis

As shown in Table 1, 61 out of 78 trials with our tensor-augmented models outperform the baselines, demonstrating significant performance improvements. TEAFormer consistently achieves better results across most window sizes, with its best performance on the WTH dataset. This is likely due to WTH's higher feature count and the regularity of weather patterns. In contrast, datasets like ETT, electricity, and illness pose challenges due to severe distribution shifts, which may limit TEAFormer's forecasting accuracy. Additionally, TEAFormer performs best with simpler model structures; for instance, its application to the Transformer, which has the simplest architecture among the models studied, consistently yields the most significant improvements over baseline counterparts.

4 Conclusions and Future Work

We propose TEAFormer, a novel approach to multi-dimensional time series forecasting that integrates tensor decomposition with transformers via a Tensor-Augmented Attention module. Embedded data is expanded and Tucker-decomposed into a core tensor and factor matrices, with the core tensor representing latent cross-dimensional and temporal interactions. By applying multi-head self-attention to

the core tensor, TEAFormer significantly reduces computational cost while enhancing the performance of Transformer, Informer, and Autoformer.

While core tensor-based attention improves efficiency, decomposition and reconstruction introduce additional computational costs. Solely relying on the core tensor poses challenges, including dimension mismatches, limited information retention, and incompatibility with attention masking. Future work should explore alternative tensor augmentation methods to optimize performance further.

TEAFormer's adaptability to diverse data and its ability to balance performance and efficiency make it valuable for real-world applications, particularly in resource-constrained settings. This work highlights the potential of integrating tensor decomposition with neural networks, paving the way for more efficient and powerful models.

References

[Bai and Ng, 2002] Jushan Bai and Serena Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

[Bai, 2003] Jushan Bai. Inferential theory for factor models of large dimensions. *Econometrica*, 71(1):135–171, 2003.

[Bai, 2013] J Bai. Panel data model: Factor analysis. *Advances in Economics and Econometrics*, 3:437–484, 2013.

[Chang *et al.*, 2023] Jinyuan Chang, Jing He, Lin Yang, and Qiwei Yao. Modelling matrix time series via a tensor CP-decomposition. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(1):127–148, 2023.

[Chen and Chen, 2022] Elynn Y Chen and Rong Chen. Modeling dynamic transport network with matrix factor models: an application to international trade flow. *Journal of Data Science*, 21(3):490–507, 2022.

[Chen and Fan, 2023] Elynn Y Chen and Jianqing Fan. Statistical inference for high-dimensional matrix-variate factor models. *Journal of the American Statistical Association*, 118(542):1038–1055, 2023.

[Chen *et al.*, 2020a] Elynn Y Chen, Ruey S Tsay, and Rong Chen. Constrained factor models for high-dimensional matrix-variate time series. *Journal of the American Statistical Association*, 115(530):775–793, 2020.

[Chen *et al.*, 2020b] Elynn Y Chen, Xin Yun, Rong Chen, and Qiwei Yao. Modeling multivariate spatial-temporal data with latent low-dimensional dynamics. *arXiv preprint arXiv:2002.01305*, 2020.

[Chen *et al.*, 2021] Rong Chen, Han Xiao, and Dan Yang. Autoregressive models for matrix-valued time series. *Journal of Econometrics*, 222(1):539–560, 2021.

[Chen *et al.*, 2022a] Rong Chen, Yuefeng Han, Zebang Li, Han Xiao, Dan Yang, and Ruofan Yu. Analysis of tensor time series: tensorts. *Journal of Statistical Software*, 2022.

[Chen *et al.*, 2022b] Rong Chen, Dan Yang, and Cun-Hui Zhang. Factor models for high-dimensional tensor time series. *Journal of the American Statistical Association*, 117(537):94–116, 2022.

[Chen *et al.*, 2023] Elynn Y Chen, Jianqing Fan, and Xuening Zhu. Community network auto-regression for high-dimensional time series. *Journal of Econometrics*, 235(2):1239–1256, 2023.

[Chen *et al.*, 2024a] Bin Chen, Elynn Y Chen, Stevenson Bolivar, and Rong Chen. Time-varying matrix factor models. *arXiv preprint arXiv:2404.01546*, 2024.

[Chen *et al.*, 2024b] Bin Chen, Yuefeng Han, and Qiyang Yu. Estimation and inference for cp tensor factor models. *arXiv preprint arXiv:2406.17278*, 2024.

[Chen *et al.*, 2024c] Elynn Chen, Jianqing Fan, and Xiaonan Zhu. Factor augmented matrix regression. *arXiv preprint arXiv:2405.17744*, 2024.

[Chen *et al.*, 2024d] Elynn Chen, Yuefeng Han, and Jiayu Li. High-dimensional tensor classification with cp low-rank

discriminant structure. *arXiv preprint arXiv:2409.14397*, 2024.

[Chen *et al.*, 2024e] Elynn Chen, Yuefeng Han, and Jiayu Li. High-dimensional tensor discriminant analysis with incomplete tensors. *arXiv preprint arXiv:2410.14783*, 2024.

[Chen *et al.*, 2024f] Elynn Y Chen, Dong Xia, Chencheng Cai, and Jianqing Fan. Semi-parametric tensor factor analysis by iteratively projected singular value decomposition. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, page qkae001, 2024.

[Cohen *et al.*, 2016] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[Fan *et al.*, 2022] Joshua Fan, Junwen Bai, Zhiyun Li, Ariel Ortiz-Bobea, and Carla P Gomes. A gnn-rnn approach for harnessing geospatial and temporal information: Application to crop yield prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11873–11881, 2022.

[Grigsby *et al.*, 2021] Jake Grigsby, Zhe Wang, Nam Nguyen, and Yanjun Qi. Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*, 2021.

[Guo *et al.*, 2022] Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. Longt5: Efficient text-to-text transformer for long sequences, 2022.

[Han and Zhang, 2023] Yuefeng Han and Cun-Hui Zhang. Tensor principal component analysis in high dimensional CP models. *IEEE Transactions on Information Theory*, 69(2):1147–1167, 2023.

[Han *et al.*, 2020] Yuefeng Han, Rong Chen, Dan Yang, and Cun-Hui Zhang. Tensor factor model estimation by iterative projection. *arXiv preprint arXiv:2006.02611*, 2020.

[Han *et al.*, 2022] Yuefeng Han, Rong Chen, and Cun-Hui Zhang. Rank determination in tensor factor model. *Electronic Journal of Statistics*, 16(1):1726–1803, 2022.

[Han *et al.*, 2024a] Yuefeng Han, Rong Chen, Cun-Hui Zhang, and Qiwei Yao. Simultaneous decorrelation of matrix time series. *Journal of the American Statistical Association*, 119(546):957–969, 2024.

[Han *et al.*, 2024b] Yuefeng Han, Dan Yang, Cun-Hui Zhang, and Rong Chen. CP factor model for dynamic tensors. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, page qkae036, 2024.

[Hoff, 2015] Peter D. Hoff. Multilinear tensor regression for longitudinal relational data. *Annals of Applied Statistics*, 9:1169–1193, 2015.

[Kossaifi *et al.*, 2017] Jean Kossaifi, Aran Khanna, Zachary C. Lipton, Tommaso Furlanello, and Anima Anandkumar. Tensor contraction layers for parsimonious deep nets. *CVPR*, pages 1940–1946, 2017.

[Kossaifi *et al.*, 2020] Jean Kossaifi, Zachary C. Lipton, Arinbjorn Kolbeinsson, Aran Khanna, Tommaso Furlanello, and Anima Anandkumar. Tensor regression networks. *JMLR*, 21:1–21, 2020.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks, 2018.

[Li and Xiao, 2021] Zebang Li and Han Xiao. Multilinear tensor autoregressive models. *arXiv preprint arXiv:2110.00928*, 2021.

[Li and Xiao, 2024] Zebang Li and Han Xiao. Cointegrated matrix autoregression models. *arXiv preprint arXiv:2409.10860*, 2024.

[Liu and Chen, 2022] Xialu Liu and Elynn Y Chen. Identification and estimation of threshold matrix-variate factor models. *Scandinavian Journal of Statistics*, 49(3):1383–1417, 2022.

[Liu *et al.*, 2022] Xu Liu, Yuxuan Liang, Chao Huang, Yu Zheng, Bryan Hooi, and Roger Zimmermann. When do contrastive learning signals help spatio-temporal graph forecasting? In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, New York, NY, USA, 2022. Association for Computing Machinery.

[Sen *et al.*, 2019] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in Neural Information Processing Systems*, 32, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[Wang *et al.*, 2019] Dong Wang, Xialu Liu, and Rong Chen. Factor models for matrix-valued high-dimensional time series. *Journal of Econometrics*, 208(1):231–248, 2019.

[Wang *et al.*, 2022] Lijing Wang, Aniruddha Adiga, Jiangzhuo Chen, Adam Sadilek, Srinivasan Venkatramanan, and Madhav Marathe. Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12191–12199, Jun. 2022.

[Wen *et al.*, 2023] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. In *International Joint Conference on Artificial Intelligence(IJCAI)*, 2023.

[Wen *et al.*, 2024] Tao Wen, Elynn Chen, and Yuzhou Chen. Tensor-view topological graph neural network. In *International Conference on Artificial Intelligence and Statistics, 2024, València SPAIN*, 2024.

[Woo *et al.*, 2022] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[Wu *et al.*, 2024a] Yujia Wu, Junyi Mo, Elynn Chen, and Yuzhou Chen. Tensor-fused multi-view graph contrastive learning. *arXiv preprint arXiv:2410.15247*, 2024.

[Wu *et al.*, 2024b] Yujia Wu, Bo Yang, Elynn Chen, Yuzhou Chen, and Zheshi Zheng. Conditional prediction roc bands for graph classification. *arXiv preprint arXiv:2410.15239*, 2024.

[Wu *et al.*, 2024c] Yujia Wu, Bo Yang, Yang Zhao, Elynn Chen, Yuzhou Chen, and Zheshi Zheng. Conditional uncertainty quantification for tensorized topological neural networks. *arXiv preprint arXiv:2410.15241*, 2024.

[Xiao *et al.*, 2023] Han Xiao, Yuefeng Han, Rong Chen, and Chengcheng Liu. Reduced rank autoregressive models for matrix time series. *working paper*, 2023.

[Xu *et al.*, 2023] Yao Lei Xu, Kriton Konstantinidis, and Danilo P Mandic. Graph tensor networks: An intuitive framework for designing large-scale neural learning systems on multiple domains. *arXiv preprint arXiv:2303.13565*, 2023.

[Yu *et al.*, 2023] Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3062–3072, 2023.

[Yu *et al.*, 2024] Ruofan Yu, Rong Chen, Han Xiao, and Yuefeng Han. Dynamic matrix factor models for high dimensional time series. *arXiv preprint arXiv:2407.05624*, 2024.

[Zhang and Yan, 2022] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

[Zhang *et al.*, 2018] Jiacheng Zhang, Huanbo Luan, Maosong Sun, FeiFei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. Improving the transformer translation model with document-level context, 2018.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pages 11106–11115. AAAI Press, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

[Zhou *et al.*, 2024] Guanhao Zhou, Yuefeng Han, and Xiufan Yu. Factor augmented tensor-on-tensor neural networks. *arXiv preprint arXiv:2405.19610*, 2024.