LETO: Modeling Multivariate Time Series with Memorizing at Test Time

Ali Behrouz [*1] **Daniel Yiming Cao** [*1] **Ali Parviz** [*2 3] **Michele Santacatterina** [4] **Ramin Zabih** [1]

Abstract

Modeling multivariate time series remains a core challenge due to complex temporal and cross-variate dependencies. While sequence models like Transformers, CNNs, and RNNs have been adapted from NLP and vision tasks, they often struggle with multivariate structure, long-range dependencies, or error propagation. We introduce Leto, a 2D memory module that leverages temporal inductive bias while preserving variate permutation equivariance. By combining in-context memory with cross-variate attention, Leto effectively captures temporal patterns and inter-variate signals. Experiments across diverse benchmarks—forecasting, classification, and anomaly detection—demonstrate its strong performance.

1. Introduction

Modeling multivariate time series data is a well-established problem in the literature with a diverse set of applications ranging from healthcare (Ivanov et al., 1999; Tang et al., 2023) and neuroscience (Behrouz & Hashemi, 2024a) to finance (Gajamannage et al., 2023; Pincus & Kalman, 2004), energy (Zhou et al., 2021), transportation management (Durango-Cohen, 2007), and weather forecasting (Allen et al., 2025; Price et al., 2025). Classical shallow models—such as State Space Models (Harvey, 1990; Aoki, 2013), ARIMA (Bartholomew, 1971), SARIMA (Bender & Simonovic, 1994), Exponential Smoothing (ETS) (Winters, 1960)—have long been the de-facto mathematical models for time series prediction, modeling diverse complex patterns (such as seasonal and trend patterns). Deploying these models at scale in real-world settings remains challenging due to their reliance on manual data preprocessing, sensitive model selection, and inherently sequential, non-parallelizable computations. Additionally, these models often fail to capture (1) the inter-dependencies of different variates, and (2) the complex *non-linear* dynamics inherent to multivariate time series data.

The emergence of deep learning has shifted the focus of recent time series research away from traditional statistical methods toward deep neural network architectures such as Transformer-based (Zhou et al., 2021; Wu et al., 2021), recurrence-based (Behrouz et al., 2024d;e; Patro & Agneeswaran, 2024; Jia et al., 2023), and temporal convolutional-based (Bai et al., 2018; Sen et al., 2019; Luo & Wang, 2024) models. Despite the outstanding performance of Transformers (Vaswani et al., 2017) across various diverse domains (Du et al., 2023; Nguyen et al., 2024; Wu et al., 2021), recent studies have highlighted their frequent suboptimal performance compared to even linear methods, mainly due to their inherent permutation equivariance that contradicts the causal nature of time series (Zeng et al., 2023c). Additionally, their quadratic time and memory complexity is a notable bottleneck for their use in large-scale long real-world settings with long-range prediction horizon.

While modern linear RNNs offer efficient alternatives to Transformers (Peng et al., 2023a; Katharopoulos et al., 2020; Kacham et al., 2023; Smith et al., 2023), their use in multivariate time series poses key challenges. First, the non-stationary and noisy nature of time series data can lead to error accumulation in additive recurrent models, requiring careful design (Jia et al., 2023; Behrouz et al., 2024d). Second, these models are inherently single-sequence and often neglect cross-variate dependencies, which are crucial but not always beneficial (Zeng et al., 2023a; Zhang et al., 2023; Nie et al., 2023; Chen et al., 2023). Finally, recent 2D recurrent approaches (Jia et al., 2023; Behrouz et al., 2024d) are sensitive to the order of variates, lacking permutation equivariance.

Contributions. In this paper, to mitigate the above-mentioned limitations in existing time series models, we present LETO, a novel 2-dimensional architecture based on two meta in-context memory modules—called time and variate memory modules—that learns how to memorize cross-time and cross-variate patterns at test time, respectively. While LETO updates the time memory module using a re-

[*]Equal contribution [1]Cornell University [2]Mila - Quebec AI Institute [3]New Jersey Institute of Technology [4]New York University. Correspondence to: Daniel Cao <dyc33@cornell.edu>, Ali Behrouz <ab2947@cornell.edu>, Ali Parviz <ali.parviz@mila.quebec>, Michele Santacatterina <santam13@nyu.edu>, Ramin Zabih <rdz@cs.cornell.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

current rule to take advantage of its temporal inductive bias, it uses an attention-like (with `Softmax`) non-parametric memory module across variates to accurately consider their permutation equivariance property. To capture the dynamics of dependencies across variates, LETO needs to mix the states of both time and variate memories at each time stamps. However, the non-parametric nature of variate memory module makes it state-less, empowering the memory to learn the dynamics of variate dependencies across time. To overcome this challenge, LETO uses a parametric approximation of the non-parametric memory and expresses the `Softmax` attention using its Taylor series. To the best of our knowledge, LETO is the first native 2-dimensional hybrid model. In our experiments, we perform various evaluations and compare LETO with state-of-the-art time series models on diverse downstream tasks, including: (1) short-, long-, and ultra-long-term forecasting, (2) classification, and (3) anomaly detection tasks. We further demonstrate the effectiveness of LETO for longer horizons and support the significance of LETO's design by performing ablation studies.

A more detailed discussion of background concepts and related work is provided in Appendix B.

Notation. We let matrix $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_V\} \in \mathbb{R}^{V \times T \times d_{\text{in}}}$ denote a multivariate time series, where T and V are the number of time stamps and variates, respectively, and d_{in} is the feature dimension of the input (often $d_{\text{in}} = 1$). We use $x_{v,t} \in \mathbb{R}^{d_{\text{in}}}$ to refer to the value of the time series in v-th variate at time t.

2. LETO: Learning to Memorize at Test Time with 2-Dimensional Memory

We present our model: LETO, a native 2-dimensional architecture that takes advantage of two separate memory modules, each of which learns how to memorize patterns across either time or variate dimensions. Figure 2 illustrates the architectural design of LETO.

2.1. How to Memorize 2-Dimensional Data?

While sequence modeling with test-time memorization is effective for univariate time series, multivariate data requires two memory modules—one for each dimension (time and variate). Naively memorizing training data risks overfitting and fails under distribution shifts. To address this, we propose a *meta in-context memory* that learns *how* to memorize at test time. Instead of storing training samples, it captures generalizable patterns, selectively retaining or discarding information based on training-time dynamics.

Cross Time Dynamic. To illustrate the modeling of cross-time patterns, we fix the variate v and omit it from subscripts when clear. This setup defines a meta-learning problem over the memory parameters, where the goal is to reconstruct projected inputs $\mathbf{v}_i = W_v \mathbf{x}_i$ from corrupted versions $\mathbf{k}_i = W_k \mathbf{x}_i$. Given a reconstruction loss $\ell(\cdot)$, training involves two nested loops. In the *inner loop*, only the memory is updated to minimize reconstruction error via gradient descent:

$$\mathcal{M}_t = \alpha_t \mathcal{M}_{t-1} - \eta_t \nabla \ell(\mathcal{M}_{t-1}; \mathbf{x}_{v,t}). \tag{1}$$

All other parameters remain fixed. The *outer loop* then updates the full model (excluding memory) for the downstream task—e.g., forecasting, classification, or anomaly detection. Using a reconstruction loss, i.e., $\ell(\mathcal{M}; \mathbf{x}_t) = \|\mathcal{M}\mathbf{k}_t - \mathbf{v}_t\|_2^2$, where $\mathbf{k}_t$ and $\mathbf{v}_t$ are defined as previously, gives us a memory module with delta update rule (recurrence) (Schlag et al., 2021) as:

$$\begin{aligned}\mathcal{M}_t &= \mathcal{M}_{t-1} - \eta_t \nabla \ell(\mathcal{M}_{t-1}; \mathbf{x}_t) \\ &= (\mathbf{I} - \eta_t \mathbf{k}_t \mathbf{k}_t^\top)\mathcal{M}_{t-1} + \eta_t \mathbf{v}_t \mathbf{k}_t^\top \end{aligned} \tag{2}$$

where $(\mathbf{I} - \mathbf{k}_t \mathbf{k}_t^\top)$ is the transition matrix from state $\mathcal{M}_{t-1}$ to $\mathcal{M}_t$ and $\mathbf{v}_t \mathbf{k}_t^\top$ is the transformation of the input data. This linear recurrent process is equivalent to a linear dynamical system with non-diagonal transition matrix, which is more expressive than its counterpart dynamical systems with diagonal transition (Behrouz et al., 2024d; Patro & Agneeswaran, 2024; Li et al., 2024). In our later design of LETO in Equation Variant 2, we further enhance the above formulation by incorporating a gating mechanism from the Titans architecture (Behrouz et al., 2024e) as:

$$\mathcal{M}_t = (\alpha_t \mathbf{I} - \eta_t \mathbf{k}_t \mathbf{k}_t^\top)\mathcal{M}_{t-1} + \eta_t \mathbf{v}_t {\mathbf{k}_t}^\top, \tag{3}$$

where α controls the retention from the previous state of the memory. When $\alpha \to 1$, it fully retains the past state and when $\alpha \to 0$ it erases the past state of the memory.

Cross Variate Dynamic. In the previous section, we discuss a neural memory module that learns how to memorize cross-time patterns. While our memory module captures cross-time patterns, multivariate time series often contain richer cross-variate dependencies (Tang et al., 2023; Behrouz et al., 2024a; Liu et al., 2024a). To model these, one might transpose the input and apply the same memory mechanism (Equation 3) across variates. However, this approach is sensitive to variate order. Unlike time, variate dimensions are typically unordered, so models must be *permutation equivariant*—producing outputs that permute consistently with input permutations.

Transformers are one of the most powerful architectures with the permutation equivariance property (Yun et al., 2020; Xu et al., 2024). Although this property makes their direct applicability to time series data limited, it makes them a great choice of architectural backbone for use in learning the cross-variate information (Liu et al., 2024a). To this end, given the input data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_V\} \in \mathbb{R}^{V \times T \times d_{\text{in}}}$, one can define $\tilde{\mathbf{X}} = \mathbf{X}^\top = \{\tilde{\mathbf{x}}_1, \dots \tilde{\mathbf{x}}_T\} \in \mathbb{R}^{T \times V \times d_{\text{in}}}$ and then pass it to a Transformer block to capture the cross-variate dependencies: $\mathbf{Y} = \texttt{Transformer}\left(\tilde{\mathbf{X}}\right)$. While the above method can satisfy both (1) fusing information across variates, and (2) preserving the robustness to the permutation of variates, it only models cross-variate patterns and misses the dynamics of variates dependencies (Behrouz et al., 2024d; Jia et al., 2023).

2.2. LETO: A Native 2-Dimensional Memory System

Previously we discussed how one can design an effective memory module that learns how to map underlying patterns across time *or* variate dimensions in the data. A simple and commonly used method in the literature is to use two different modules, each for one of the dimensions, and then mix their outputs for the final prediction (Ahamed & Cheng, 2024b; Christou et al., 2024). That is, given input $\mathbf{X} \in \mathbb{R}^{V \times T \times d_{\text{in}}}$, one can use $\texttt{Module}_1(\cdot)$ and $\texttt{Module}_2(\cdot)$ to fuse information across time and variates, respectively, and then combine them for the final output:

$$Y_{\text{time}} = \texttt{Module}_1(\mathbf{X}), \qquad Y_{\text{variate}} = \texttt{Module}_2(\tilde{\mathbf{X}}),$$
$$Y_{\text{output}} = \texttt{Combine}\left(Y_{\text{time}}, Y_{\text{variate}}\right). \qquad \text{(Variant 1)}$$

Another commonly used method is to employ $\texttt{Module}_1(\cdot)$ and $\texttt{Module}_2(\cdot)$ in a sequential manner (instead of the above parallel manner). However, all these models treat each dimension separately and thus miss the inter-dependencies of time and variate dimensions at each state of the system, resulting in less expressive power in modeling time series data. We present a native 2-D memory system that not only has the temporal inductive bias across time, but also has the permutation equivariance property across variates.

We use two memory modules $\mathcal{M}^{(1)}(\cdot)$ and $\mathcal{M}^{(2)}(\cdot)$ to learn the underlying mappings/patterns across time and variate dimensions, respectively. To design such memory modules it is appropriate to use a reconstruction objective $\ell(\cdot)$ for the memory and then optimize this objective with an optimization algorithm (such as gradient descent). However, to capture the inter-dependencies of dimensions at each step of optimization, it is necessary to fuse the information between the memory modules as well. Therefore, the state of each memory module not only depends on its time stamp, but it also depends on its variate. Given $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_V\}$ as the input, and arbitrary $v \in \{1, \dots, V\}$ we define the cross-time memory update as:

$$\mathcal{M}^{(1)}_{t,v} = \alpha_{t,v}\mathcal{M}^{(1)}_{t-1,v} - \eta_{t,v}\nabla\ell(\mathcal{M}^{(1)}_{t-1,v}, \mathbf{x}_{t,v}) + \beta_{t,v}\mathcal{M}^{(2)}_{t-1,v} - \gamma_{t,v}\nabla\ell(\mathcal{M}^{(2)}_{t-1,v}, \mathbf{x}_{t,v}), \quad (4)$$

where $\ell(\mathcal{M}^{(j)}_{t-1,v}, \mathbf{x}_{t,v}) = \|\mathcal{M}^{(j)}_{t-1,v}\mathbf{k}_{t.v} - \mathbf{v}_{t,v}\|_2^2$ for $j \in \{1, 2\}$ and $v \in \{1, \dots, V\}$ and $\mathbf{k}_{t,v} = W_k\mathbf{x}_{t,v}$ and $\mathbf{v}_{t,v} = W_v\mathbf{x}_{t,v}$. Expanding the gradient for the above formulation results in the recurrent update rule for the cross-time memory module as follows:

$$\mathcal{M}^{(1)}_{t,v} = (\alpha_{t,v}\mathbf{I} - \eta_{t,v}\mathbf{k}_{t,v}\mathbf{k}_{t,v}^\top)\mathcal{M}_{t-1,v} + \eta_{t,v}\mathbf{v}_{t,v}\mathbf{k}_{t,v}^\top + (\beta_{t,v}\mathbf{I} - \gamma_{t,v}\mathbf{k}_{t,v}\mathbf{k}_{t,v}^\top)\mathcal{M}_{t-1,v} + \gamma_{t,v}\mathbf{v}_{t,v}\mathbf{k}_{t,v}^\top. \quad (5)$$

The above formulation demonstrates how to update the cross-time memory. To get the final output from this memory, we need to multiply it by the input data $\mathbf{x}_{t,v}$ to achieve the $\mathbf{x}_{t,v}$'s corresponding information in the memory: i.e., $\mathbf{Y}^{(1)}_{t,v} = \mathcal{M}^{(1)}_{t,v}\mathbf{x}_{t,v}$. One can similarly define the recurrence for the cross-variate memory module $\mathcal{M}^{(2)}_{t,v}$ as:

$$\mathcal{M}^{(2)}_{t,v} = \theta_{t,v}\mathcal{M}^{(1)}_{t,v-1} - \lambda_{t,v}\nabla\ell(\mathcal{M}^{(1)}_{t,v-1}, \mathbf{x}_{t,v}) + \mu_{t,v}\mathcal{M}^{(2)}_{t,v-1} - \omega_{t,v}\nabla\ell(\mathcal{M}^{(2)}_{t,v-1}, \mathbf{x}_{t,v}). \quad (6)$$

However, it is still sensitive to the order of variates. This sensitivity to variate ordering comes from the parametric nature of gradient descent algorithm as its iterations requires a series of ordered steps. Therefore, the use of any other parametric optimizer can cause such sensitivity to the order. To overcome this issue, we use the non-parametric estimate of our objective. Interestingly, with a small modification and using Nadaraya-Watson estimators (Fan, 2018; Zhang et al., 2022b), the non-parametric estimate of the objective is equivalent to softmax attention mechanism in Transformers (Vaswani et al., 2017), as also discussed in previous studies (Sun et al., 2024; Behrouz et al., 2025). Therefore, due to this theoretical connection, we use an attention module for the cross-variate information mixing. The final output of this block can simply be defined as:

$$\mathbf{Y}^{(2)}_{t,v} = \theta_{t,v}\,\texttt{Attention}\left(\{\mathcal{M}^{(1)}_{t,i}\,\mathbf{x}_{t,i}\}_{i=1}^{V}\right) + \mu_{t,v}\,\texttt{Attention}\left(\{\mathbf{x}_{t,i}\}_{i=1}^{V}\right). \quad (7)$$

Note that $\mathcal{M}^{(1)}_{t,i}\,\mathbf{x}_{t,i}$ provides the $\mathbf{x}_{t,i}$'s corresponding information in cross-time memory module and so the first term combines the cross-time dynamic of all variates at the same time. While computation of the final output for the cross-variate memory is clear, we need to access its memory (i.e., $\mathcal{M}^{(2)}_{t,v}$) to use in the update of cross-time memory (i.e., Equation 4). The memory of Transformers are known to be the pair of key and value matrices $(\mathbf{K}, \mathbf{V})$ in the attention mechanism (Zhang & Cai, 2022; Wu et al., 2022b; Behrouz et al., 2024e; Bietti et al., 2023). However, incorporating a pair of matrices into the recurrence update rule of Equation 4 is unclear and challenging. Therefore, we utilize a kernelized variant of attention, in which we replace $\texttt{Softmax}$ with a separable kernel $\phi(\cdot)$ (Katharopoulos et al., 2020; Kacham et al., 2023; Arora et al., 2024) (see Appendix A for the corresponding background and detailed formulation). This allows us to concretely define the memory of

Table 1: Average performance on long term forecasting tasks over four prediction lengths: {96, 192, 336, 720}. A lower MAE and MSE indicates a better prediction. The best performance is highlighted in **red**, and the second-best is underlined.

Models	LETO (Ours)		TimeMixer		Simba		ModernTCN		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.347**	**0.375**	0.381	0.385	0.383	0.396	0.351	0.381	0.407	0.410	0.414	0.407	0.387	0.400	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407
ETTm2	**0.249**	**0.302**	0.275	0.323	0.271	0.327	0.253	0.314	0.288	0.332	0.286	0.327	0.281	0.326	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401
ETTh1	**0.393**	**0.401**	0.447	0.440	0.441	0.432	0.404	0.420	0.454	0.447	0.446	0.434	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452
ETTh2	**0.318**	0.381	0.364	0.395	0.361	0.391	0.322	0.379	0.383	0.407	0.374	0.398	0.387	0.407	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515
Exchange	**0.297**	0.364	0.391	0.453	0.298	**0.363**	0.302	0.366	0.360	0.403	0.378	0.417	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414
Traffic	0.408	**0.267**	0.484	0.297	0.493	0.291	**0.398**	0.270	0.428	0.282	0.626	0.378	0.481	0.304	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383

Table 2: Average performance on Ultra long-term forecasting tasks (MSE / MAE)

Dataset	Metric	LETO		MICN		TimesNet		PatchTST		DLinear		FiLM		FEDformer		Autoformer		Informer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	720–1440	0.4782	0.5614	1.0460	0.7765	0.6119	0.5962	0.8243	0.6704	0.4923	0.5473	**0.4730**	0.5336	0.4833	0.5393	1.4957	0.9533	0.5064	**0.5317**
	1440–1440	**0.4639**	**0.5387**	0.8262	1.2207	0.5720	0.5712	0.9053	0.7328	0.5146	0.5615	0.4849	0.5429	0.5142	0.5571	1.7873	1.0283	0.7247	0.6920
	1440–2880	**0.6047**	**0.5868**	2.8936	1.3717	0.7683	0.6846	1.1282	0.8087	0.8355	0.7193	0.6847	0.6493	3.9018	1.5276	1.2867	0.8878	0.6152	0.5953
Traffic	720–1440	0.1672	0.2431	0.2876	0.3916	0.1882	0.2656	0.1904	0.2685	0.1639	**0.2412**	**0.1638**	0.2448	0.2753	0.3650	0.3104	0.4095	0.7614	0.6496
	1440–1440	**0.1521**	0.2497	0.2905	0.3923	0.2081	0.2712	0.1917	0.2764	0.1590	**0.2411**	0.1602	0.2437	0.2848	0.3681	0.2970	0.3999	0.7375	0.6414
	1440–2880	**0.1425**	0.2433	0.2823	0.3874	0.1560	**0.2409**	0.1819	0.2761	0.1550	0.2421	0.1744	0.2693	0.2952	0.3844	0.3035	0.3982	0.9408	0.7618
ETTh1	720–1440	**0.1331**	**0.2943**	0.4640	0.5836	0.1391	0.3049	0.3708	0.4906	0.2952	0.4370	0.2949	0.4388	0.1768	0.3409	0.3298	0.4741	0.1378	0.3051
	1440–1440	**0.1359**	0.3120	0.5188	0.6075	0.1404	**0.3093**	0.4475	0.5392	0.2200	0.3714	0.3226	0.4678	0.1928	0.3576	0.3618	0.5507	0.1402	0.3192
	1440–2880	**0.2591**	0.3949	0.7591	0.7215	0.2732	0.4094	0.9617	0.8072	0.3773	0.4794	0.3624	0.4705	0.2627	**0.3754**	0.3177	0.4733	0.3495	0.4111

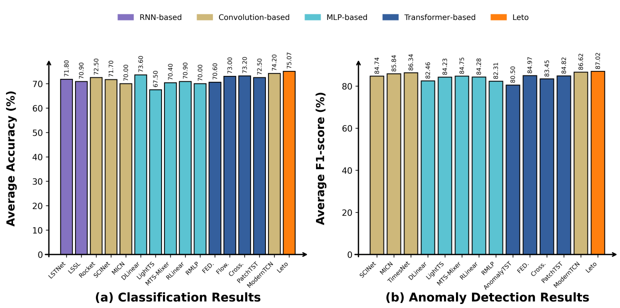


Figure 1: Anomaly detection and classification results of LETO and baselines. Higher accuracy/F1-score indicate better performance.

the Transformer with keys and values of $\{\hat{\mathbf{k}}_i\}$ and $\{\hat{\mathbf{v}}_i\}$ as (Katharopoulos et al., 2020) $\mathcal{M}^{(2)}_{t,v} = \sum_{i=1}^{V} \hat{\mathbf{v}}_{t,i}\phi(\hat{\mathbf{k}}^\top_{t,i})$.

The question about what would be the optimal kernel $\phi(\cdot)$ to use in the above formulation remains. To answer this, we recall the formulation of `Softmax` attention that is proportional to $\texttt{softmax}(\mathbf{q}_t^\top \mathbf{k}_t)\mathbf{v}_t$. To replace softmax $\texttt{softmax}(\cdot)$ with a separable kernel $\phi(\cdot)$, we can choose the kernel to approximate the exponential term in softmax with its Taylor series. Accordingly, we use the first four terms of the Taylor series of $\exp(\cdot)$ defined as: $\exp(x) \approx \phi(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!}$. Combining the prior expressions, we can define our native 2-dimensional update rule as:

$$\begin{aligned} \mathcal{M}^{(1)}_{t,v} &= \alpha_{t,v}\mathcal{M}^{(1)}_{t-1,v} - \eta_{t,v}\nabla\ell(\mathcal{M}^{(1)}_{t-1,v}, \mathbf{x}_{t,v}) \\ &\quad + \beta_{t,v}\mathcal{M}^{(2)}_{t-1,v} - \gamma_{t,v}\nabla\ell(\mathcal{M}^{(2)}_{t-1,v}, \mathbf{x}_{t,v}), \quad \text{(Variant 2)} \end{aligned}$$

where $\mathcal{M}^{(2)}_{t,v} = \sum_{i=1}^{V} \hat{\mathbf{v}}_{t,i}\phi(\hat{\mathbf{k}}^\top_{t,i})$ and $\phi(x) = x + \frac{x^2}{2} + \frac{x^3}{3!}$. In the above formulation $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{k}}_i$ are keys and values of the Transformer block, coming from the keys and values of the cross-variate dynamic attention mentioned in Equation 7.

3. Experiments

Goals and Baselines. In this section, we evaluate LETO on a wide range of time series tasks, comparing with the state-of-the-art multivariate time series models (Wu et al., 2023; Luo & Wang, 2024; Lim & Zohren, 2021; Woo et al., 2022; Wu et al., 2021; Zhou et al., 2022b; Zhang & Yan, 2023; Liu et al., 2024a; Dehghani et al., 2023; Das et al., 2023; Liu et al., 2022a; Patro & Agneeswaran, 2024; Zeng et al., 2023b; Xu et al., 2021; Wang et al., 2024) on forecasting: long, ultra-long, and short term, classification, and anomaly detection tasks. Detailed dataset descriptions and complete experimental results are provided in Appendix E.

3.1. Main Results: Classification and Forecasting

Long-Term Forecasting. We conduct experiments on the long-term forecasting tasks using commonly used benchmark datasets used by Zhou et al. (2021). The average performance across different horizons is summarized in Table 1. LETO consistently delivers strong results across different datasets, highlighting its robustness compared to recurrent, convolutional, SSM, and Transformer-based models.

Ultra Long-term Forecasting. We further extend the evaluation to ultra-long-range forecasting on the same benchmark datasets (Zhou et al., 2021) to observe the effectiveness of LETO in longer horizons. The tasks on the left side of the Table 2 retain the same interpretation as in the standard long-term forecasting setting. The results in Table 2 demonstrate LETO's ability to capture long-term dependencies from extremely long historical inputs, maintaining its strong performance across various extended prediction horizons.

Classification and Anomaly Detection. We evaluate the performance of LETO on 10 multivariate datasets from the UEA Time Series Classification (Bagnall et al., 2018) (see Figure 1 and Table 10). For anomaly detection, we conduct experiments on five widely-used benchmarks: SMD (Su et al., 2019), SWaT (Mathur & Tippenhauer, 2016), PSM (Abdulaal et al., 2021), and SMAP (Hundman et al., 2018) and observe the effectiveness of our approach.

Impact Statement

LETO delivers strong, general-purpose performance across forecasting, classification, and anomaly detection tasks. Its adaptability makes it suitable for real-world applications like energy forecasting, weather prediction, financial modeling, and supply chain demand estimation. Notably, it performs well in industrial anomaly detection, where robustness to noise and structural shifts is critical—underscoring its potential as a foundational model for time series analysis.

References

Abdulaal, A., Liu, Z., and Lancewicki, T. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2485–2494, 2021.

Ahamed, M. A. and Cheng, Q. Mambatab: A simple yet effective approach for handling tabular data. *arXiv preprint arXiv:2401.08867*, 2024a.

Ahamed, M. A. and Cheng, Q. Timemachine: A time series is worth 4 mambas for long-term forecasting. In *ECAI 2024: 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain-Including 13th Conference on Prestigious Applications of Intelligent Systems. European Conference on Artificial Intelli*, volume 392, pp. 1688, 2024b.

Allen, A., Markou, S., Tebbutt, W., Requeima, J., Bruinsma, W. P., Andersson, T. R., Herzog, M., Lane, N. D., Chantry, M., Hosking, J. S., et al. End-to-end data-driven weather prediction. *Nature*, pp. 1–3, 2025.

Aoki, M. *State space modeling of time series*. Springer Science & Business Media, 2013.

Arora, S., Eyuboglu, S., Zhang, M., Timalsina, A., Alberti, S., Zou, J., Rudra, A., and Re, C. Simple linear attention language models balance the recall-throughput tradeoff. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1763–1840. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/arora24a.html.

Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Baron, E., Zimerman, I., and Wolf, L. A 2-dimensional state space layer for spatial inductive bias. In *The Twelfth International Conference on Learning Representations*, 2024.

Bartholomew, D. J. Time series analysis forecasting and control., 1971.

Behrouz, A. and Hashemi, F. Brain-mamba: Encoding brain activity via selective state space models. In Pollard, T., Choi, E., Singhal, P., Hughes, M., Sizikova, E., Mortazavi, B., Chen, I., Wang, F., Sarker, T., McDermott, M., and Ghassemi, M. (eds.), *Proceedings of the fifth Conference on Health, Inference, and Learning*, volume 248 of *Proceedings of Machine Learning Research*, pp. 233–250. PMLR, 27–28 Jun 2024a.

Behrouz, A. and Hashemi, F. Graph Mamba: Towards learning on graphs with state space models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 119–130, 2024b.

Behrouz, A., Delavari, P., and Hashemi, F. Unsupervised representation learning of brain activity via bridging voxel activity and functional connectivity. In *International conference on machine learning (ICML)*, 2024a.

Behrouz, A., Parviz, A., Karami, M., Sanford, C., Perozzi, B., and Mirrokni, V. S. Best of both worlds: Advantages of hybrid graph sequence models. *arXiv preprint arXiv:2411.15671*, 2024b.

Behrouz, A., Santacatterina, M., and Zabih, R. MambaMixer: Efficient selective state space models with dual token and channel selection. *arXiv preprint arXiv:2403.19888*, 2024c.

Behrouz, A., Santacatterina, M., and Zabih, R. Chimera: Effectively modeling multivariate time series with 2-dimensional state space models. In *Thirty-eighth Conference on Advances in Neural Information Processing Systems*, 2024d.

Behrouz, A., Zhong, P., and Mirrokni, V. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024e.

Behrouz, A., Razaviyayn, M., Zhong, P., and Mirrokni, V. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. *arXiv preprint arXiv:2504.13173*, 2025.

Bender, M. and Simonovic, S. Time-series modeling for long-range stream-flow forecasting. *Journal of Water Resources Planning and Management*, 120(6):857–870, 1994.

Bietti, A., Cabannes, V., Bouchacourt, D., Jegou, H., and Bottou, L. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36: 1560–1588, 2023.

Box, G. E. and Jenkins, G. M. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

Cao, D. Y., Behrouz, A., Parviz, A., Karami, M., Santacatterina, M., and Zabih, R. Effectively designing 2-dimensional sequence models for multivariate time series. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=xkFRduCUNz.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler, M., and Dubrawski, A. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

Christou, P., Chen, S., Chen, X., and Dube, P. Test time learning for time series forecasting. *arXiv preprint arXiv:2409.14012*, 2024.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Das, A., Kong, W., Leach, A., Mathur, S. K., Sen, R., and Yu, R. Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A. P., Puigcerver, J., Geirhos, R., Alabdulmohsin, I., Oliver, A., Padlewski, P., Gritsenko, A. A., Lucic, M., and Houlsby, N. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Du, D., Su, B., and Wei, Z. Preformer: Predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096881.

Durango-Cohen, P. L. A time series analysis framework for transportation infrastructure management. *Transportation Research Part B: Methodological*, 41(5):493–505, 2007.

Dwivedi, V. P., Rampaek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. *ArXiv*, abs/2206.08164, 2022. URL https://api.semanticscholar.org/CorpusID:249712241.

Fan, J. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Routledge, 2018.

Franceschi, J.-Y., Dieuleveut, A., and Jaggi, M. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS*, 2019.

Gajamannage, K., Park, Y., and Jayathilake, D. I. Real-time forecasting of time series in financial markets using sequentially trained dual-lstms. *Expert Systems with Applications*, 223:119879, 2023.

Gastinger, J., Huang, S., Galkin, M., Loghmani, E., Parviz, A., Poursafaei, F., Danovitch, J., Rossi, E., Koutis, I., Stuckenschmidt, H., et al. Tgb 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. *Advances in neural information processing systems*, 37:140199–140229, 2024.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., and Ré, C. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585, 2021.

Gu, A., Goel, K., Gupta, A., and Ré, C. On the parameterization and initialization of diagonal state space models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=yJE7iQSAep.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Harvey, A. C. Forecasting, structural time series models and the kalman filter. *Cambridge university press*, 1990.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 1997.

Huang, Y., Miao, S., and Li, P. What can we learn from state space models for machine learning on graphs? *ArXiv*, abs/2406.05815, 2024.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Ilbert, R., Odonnat, A., Feofanov, V., Virmaux, A., Paolo, G., Palpanas, T., and Redko, I. Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. *arXiv preprint arXiv:2402.10198*, 2024.

Ivanov, P. C., Amaral, L. A. N., Goldberger, A. L., Havlin, S., Rosenblum, M. G., Struzik, Z. R., and Stanley, H. E. Multifractality in human heartbeat dynamics. *Nature*, 399 (6735):461–465, 1999.

Jia, Y., Lin, Y., Hao, X., Lin, Y., Guo, S., and Wan, H. WITRAN: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kacham, P., Mirrokni, V., and Zhong, P. Polysketchformer: Fast transformers via sketches for polynomial kernels. *arXiv preprint arXiv:2310.01655*, 2023.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Kitaev, N., Kaiser, L., and Levskaya, A. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*, 2018.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *NeurIPS*, 2019.

Li, S., Singh, H., and Grover, A. Mamba-nd: Selective state space modeling for multi-dimensional data. *arXiv preprint arXiv:2402.05892*, 2024.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv: Learning*, 2017.

Li, Z., Qi, S., Li, Y., and Xu, Z. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Liang, D., Zhou, X., Wang, X., Zhu, X., Xu, W., Zou, Z., Ye, X., and Bai, X. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

Lim, B. and Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., and Xu, Q. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022a.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., and Dustdar, S. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations (ICLR)*, 2021.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Rethinking the stationarity in time series forecasting. In *NeurIPS*, 2022b.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022c.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., and Liu, Y. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024b.

Luo, D. and Wang, X. ModernTCN: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

Ma, J., Li, F., and Wang, B. U-Mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Mathur, A. P. and Tippenhauer, N. O. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pp. 31–36. IEEE, 2016.

Nguyen, E., Poli, M., Faizi, M., Thomas, A., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C., Bengio, Y., et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ICLR*, 2019.

Patro, B. N. and Agneeswaran, V. S. SiMBA: Simplified Mamba-based architecture for vision and multivariate time series, 2024.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023a.

Peng, B., Alcaide, E., Anthony, Q. G., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Grella, M., Kranthikiran, G., Du, X., He, X., Hou, H., Kazienko, P., Kocoń, J., Kong, J., Koptyra, B., Lau, H., Mantri, K. S. I., Mom, F., Saito, A., Tang, X., Wang, B., Wind, J. S., Wozniak, S., Zhang, R., Zhang, Z., Zhao, Q., Zhou, P., Zhu, J., and Zhu, R. Rwkv: Reinventing rnns for the transformer era. In *Conference on Empirical Methods in Natural Language Processing*, 2023b.

Pincus, S. and Kalman, R. E. Irregularity, volatility, risk, and financial market time series. *Proceedings of the National Academy of Sciences*, 101(38):13709–13714, 2004.

Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T. R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P., et al. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Schlag, I., Irie, K., and Schmidhuber, J. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pp. 9355–9366. PMLR, 2021.

Sen, R., Yu, H.-F., and Dhillon, I. S. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Smith, J. T., Warrington, A., and Linderman, S. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to transformer for large language models. *ArXiv*, abs/2307.08621, 2023.

Sun, Y., Li, X., Dalal, K., Xu, J., Vikram, A., Zhang, G., Dubois, Y., Chen, X., Wang, X., Koyejo, S., et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Tang, S., Dunnmon, J. A., Liangqiong, Q., Saab, K. K., Baykaner, T., Lee-Messer, C., and Rubin, D. L. Modeling multivariate biosignals with graph neural networks and structured state space models. In *Conference on health, inference, and learning*, pp. 50–71. PMLR, 2023.

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

Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J. Y., and ZHOU, J. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=7oLshfEIC2`.

Winters, P. R. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. C. H. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Wu, H., Wu, J., Xu, J., Wang, J., and Long, M. Flowformer: Linearizing transformers with conservation flows. In *ICML*, 2022a.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Wu, Y., Rabe, M. N., Hutchins, D., and Szegedy, C. Memorizing transformers. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=TrjbxzRcnf-.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence*, 2019.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Xu, H., Xiang, L., Ye, H., Yao, D., Chu, P., and Li, B. Permutation equivariance of transformers and its applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5987–5996, 2024.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In *ICLR*, 2021.

Yi, K., Zhang, Q., Fan, W., He, H., Hu, L., Wang, P., An, N., Cao, L., and Niu, Z. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

Yu, T., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting. *ArXiv*, abs/1709.04875, 2017.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S., and Kumar, S. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *AAAI*, 2023a.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023b.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023c.

Zhang, M., Saab, K. K., Poli, M., Dao, T., Goel, K., and Re, C. Effectively modeling time series with simple discrete state spaces. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., and Li, J. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022a.

Zhang, Y. and Cai, D. Linearizing transformer with key-value memory. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 346–359, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.24. URL https://aclanthology.org/2022.emnlp-main.24/.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Zhang, Y., Liu, B., Cai, Q., Wang, L., and Wang, Z. An analysis of attention via the lens of exchangeability and latent variable models. *arXiv preprint arXiv:2212.14852*, 2022b.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Zhou, T., Ma, Z., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R., et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems*, 35:12677–12690, 2022a.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning (ICML)*, 2022b.

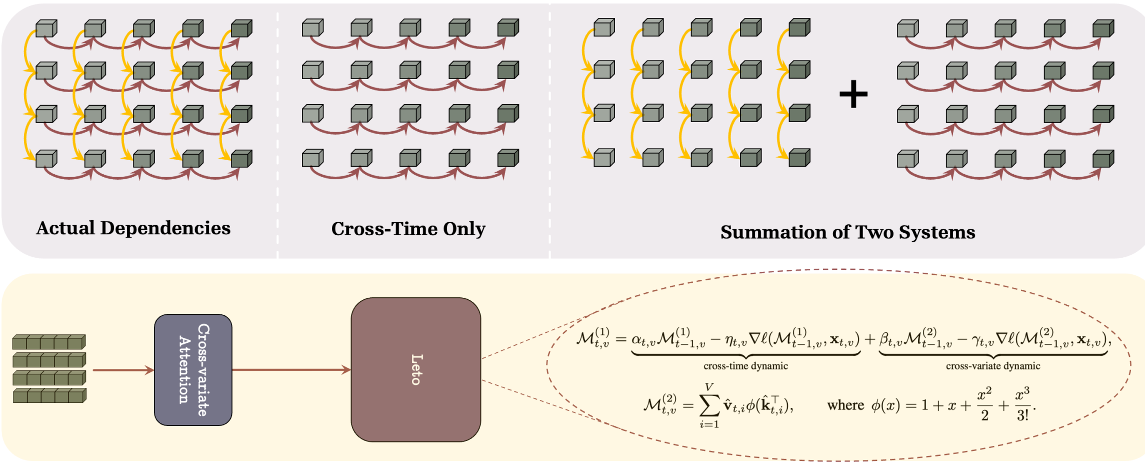


Figure 2: **An Overview of LETO's Architecture**: We define two inter-connected memory blocks M^1, M^2 corresponding to time and variate axes, where the recurrence is updated by fusing together both cross-time and cross-variate information, using an approximation of softmax attention for M^2.

A. Preliminaries and Background

Transformers and their Permutation Equivariance Property. Transformers (Vaswani et al., 2017) have been the de facto backbone for many deep learning models and are based on attention module. Let $x \in \mathbb{R}^{N \times d_{\text{in}}}$ be the input, attention computes output $\mathbf{y} \in \mathbb{R}^{N \times d_{\text{in}}}$ based on softmax over input dependent key, value, and query matrices:

$$\mathbf{Q} = x\mathbf{W_Q}, \qquad \mathbf{K} = x\mathbf{W_K}, \qquad \mathbf{V} = x\mathbf{W_V}, \tag{8}$$

$$\mathbf{y}_i = \sum_{j=1}^{N} \frac{\exp\left(\mathbf{Q}_i^{\top}\mathbf{K}_j/\sqrt{d_{\text{in}}}\right)\mathbf{V}_j}{\sum_{\ell=1}^{N}\exp\left(\mathbf{Q}_i^{\top}\mathbf{K}_\ell/\sqrt{d_{\text{in}}}\right)}, \tag{9}$$

where $\mathbf{W_Q}, \mathbf{W_K}$, and $\mathbf{W_V} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ are learnable parameters. This formulation of attention makes it permutation equivariant, meaning that the permutation of the input cannot change the output but permute it. That is, let $\pi(.)$ be a permutation, and $\mathcal{A}(\cdot)$ be the above attention module, we have:

$$\mathcal{A}(\pi(x)) = \pi(\mathcal{A}(x)). \tag{10}$$

The property, which is called permutation equivariance, is a desirable property for the data that is permutation equivariant, such as variates in the multivariate time series. When encoding the multivariate time series, we do not want the output of the model to be sensitive to the order of the input (variates) and so transformers are great architectures as any change to the order, does not change the output, but just permute it.

Learning to Memorize at Test Time. The concept of learning to memorize at test time is derived from the learning at test time or learning to learn, which backs to very early studies on local learning (**?**): i.e., training each test sample on its neighbors before making a prediction (**??**). Later, test time training shows promising results in vision tasks (**??**), mainly because of the ability to properly address out-of-distribution cases. Using this perspective, recently this idea has been applied on sequence modeling (Sun et al., 2024; Behrouz et al., 2024e; 2025). These methods that aim to train a memory module that learns how to memorize the context at test time, have shown promising results in language and sequence modeling tasks. In this work, we also take this perspective and design a 2-dimensional test time memorizer that generalizes all these methods to 2-dimensional data modality.

B. Additional Related Work

Classical Approach. Time series modeling has been a fundamental research topic, Classical approaches include a range of statistical models such as exponential smoothing (Winters, 1960), ARIMA (Bartholomew, 1971), SARIMA (Bender

& Simonovic, 1994), and the Box-Jenkins methodology (Box & Jenkins, 1968), with later advancements introducing state-space models (Harvey, 1990; Aoki, 2013). While these models offer interpretability, they often fall short in capturing complex non-linear dynamics and typically rely on manual inspection of time series characteristics—such as trend and seasonality—limiting their adaptability across diverse datasets.

Transformer-based models. Transformer-based architectures have become increasingly prominent in multivariate time series forecasting, particularly when modeling complex inter-variable and temporal dependencies (Zhou et al., 2022b; Kitaev et al., 2020; Zhang & Yan, 2023; Zeng et al., 2023a; Zhou et al., 2021; Liu et al., 2021; Wu et al., 2021; Ilbert et al., 2024; Nie et al., 2023). A line of research has focused on designing specialized attention mechanisms that leverage the unique structure of time series data (Woo et al., 2022), while others have explored strategies for capturing long-term temporal patterns to improve forecasting accuracy (Nie et al., 2023; Zhou et al., 2022a).

In parallel, recent works have revisited linear recurrent neural networks (Linear RNNs) as efficient alternatives to Transformers, aiming to reduce the quadratic complexity while maintaining competitive performance on long-range dependency modeling (Sun et al., 2023; Peng et al., 2023b; Wu et al., 2023). For instance, Chen et al. (2023) introduce TSMixer, a purely MLP-based model that demonstrates strong performance on time series forecasting tasks. Notably, the expressive capacity of certain linear models aligns with 2D state space models (SSMs), suggesting that these architectures can be interpreted as specific instances within the broader 2D SSM framework. Additionally, convolution-based models have shown renewed promise (Luo & Wang, 2024), where the use of global convolutional kernels facilitates an expanded receptive field for capturing long-range dynamics.

Recurrent-based models. Another line of research closely related to our work involves deep sequence modeling. Recurrent neural networks (RNNs), including variants such as GRUs (Chung et al., 2014), LSTMs (Hochreiter & Schmidhuber, 1997), and DeepAR (Salinas et al., 2020), have been widely used for sequential data. However, these models suffer from well-known limitations such as vanishing and exploding gradients, along with inherently sequential computation that slows down training and inference. To address these inefficiencies, recent efforts have explored linear attention mechanisms as faster alternatives (Katharopoulos et al., 2020; Schlag et al., 2021; Kacham et al., 2023). For instance, Katharopoulos et al. (2020) propose a linear attention model with a recurrent formulation, enabling efficient inference and reduced computational complexity.

In parallel, deep state space models (SSMs) have gained momentum as a compelling alternative to Transformer-based architectures (Vaswani et al., 2017), offering improved scalability and training efficiency (Gu et al., 2020). These models blend classical state space formulations with deep learning by parameterizing neural network layers using multiple linear SSMs. This hybrid formulation leverages the convolutional interpretation of SSMs to mitigate the optimization challenges typically associated with RNNs (Gu et al., 2020; 2021; 2022a;b; Smith et al., 2023). Recently, Gu & Dao (2023) introduced Mamba, a novel deep SSM architecture where parameters dynamically depend on input features. This approach has been successfully extended to various modalities—including images (Ma et al., 2024; Liu et al., 2024b; Behrouz et al., 2024c), point clouds (Liang et al., 2024), tabular data (Ahamed & Cheng, 2024a), graphs (Behrouz & Hashemi, 2024b; Behrouz et al., 2024b; Huang et al., 2024), and time series (Behrouz et al., 2024d; Cao et al., 2025)—demonstrating strong capabilities in modeling long-range dependencies across domains.

Other Methods. Graph-based models have emerged as powerful tools for time series forecasting(Wu et al., 2020; Yi et al., 2024), especially when the data exhibits spatial or relational structure across variables or entities. Approaches such as graph neural networks (GNNs) model dependencies through learned graph representations, enabling effective spatiotemporal forecasting in domains like traffic (Yu et al., 2017; Li et al., 2017) and sensor networks (Wu et al., 2019). Recent work has extended these ideas by incorporating dynamic graphs (Wu et al., 2023; Dwivedi et al., 2022; Gastinger et al., 2024), learning graph structures jointly with temporal dynamics to better capture evolving relationships over time. These methods offer strong performance in settings where explicit or latent graph structure underpins multivariate time series behavior.

C. Parallelizable Training of LETO

While the recurrence-based formulation of LETO enables it to better capture joint temporal and variate dependencies, as well as their independent dynamics, it introduces sequential dependencies that can hinder training efficiency. To address this, we develop a parallelizable training strategy inspired by recent advances in test-time memorization frameworks (Sun et al., 2024; Behrouz et al., 2024e).

Specifically, for a given variate v, we divide its time series $\{x_{1,v}, \ldots, x_{T,v}\}$ into C disjoint chunks of length $b = T/C$. Each chunk $S_i = \{x_{(i-1)b+1,v}, \ldots, x_{ib,v}\}$ can be treated as an independent subsequence for computing the inner-loop updates of the memory module. This chunking allows us to approximate the gradient $\nabla\ell(M^{(1)}_{t-1,v}, x_{t,v})$ with $\nabla\ell(M^{(1)}_{t',v}, x_{t,v})$, where $t' = \lfloor t/b \rfloor \cdot b$ is the last time step of the previous chunk. Since t' is fixed for each chunk, this gradient can be computed in parallel for all time steps within a chunk.

Moreover, the cross-variate dynamic component—modeled via the attention mechanism—is independent of time and can be computed in advance. We precompute the attention-based memory $M^{(2)}_{t,v}$ for all variates using equation above with a Taylor-approximated softmax kernel. This enables us to also precompute $\nabla\ell(M^{(2)}_{t,v}, x_{t,v})$, further decoupling the cross-variate dynamics from the sequential recurrence.

With the cross-variate memory and its corresponding gradient terms available, the remaining computation in each chunk reduces to a linear update over the cross-time memory using the precomputed components. As a result, we obtain a recurrence that is linear within chunks and can be parallelized across both time and variates.

D. Dataset and Experimental Details

The experimental details are reported in Table 3.

E. Additional Experimental Results

E.1. Metrics

We utilize the mean square error (MSE) and mean absolute error (MAE) for long-term forecasting. For short-term forecasting on the M4 datasets, we follow the methodology of N-BEATS (Oreshkin et al., 2019) and utilize the symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA) as metrics. It is worth noting that OWA is a specific metric utilized in the M4 competition. The calculations of these metrics are:

$$\mathrm{RMSE} = (\sum_{i=1}^{F} (\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2)^{\frac{1}{2}}, \qquad \mathrm{MAE} = \sum_{i=1}^{F} |\mathbf{X}_i - \widehat{\mathbf{X}}_i|,$$

$$\mathrm{SMAPE} = \frac{200}{F} \sum_{i=1}^{F} \frac{|\mathbf{X}_i - \widehat{\mathbf{X}}_i|}{|\mathbf{X}_i| + |\widehat{\mathbf{X}}_i|}, \qquad \mathrm{MAPE} = \frac{100}{F} \sum_{i=1}^{F} \frac{|\mathbf{X}_i - \widehat{\mathbf{X}}_i|}{|\mathbf{X}_i|},$$

$$\mathrm{MASE} = \frac{1}{F} \sum_{i=1}^{F} \frac{|\mathbf{X}_i - \widehat{\mathbf{X}}_i|}{\frac{1}{F-s} \sum_{j=s+1}^{F} |\mathbf{X}_j - \mathbf{X}_{j-s}|}, \qquad \mathrm{OWA} = \frac{1}{2} \left[\frac{\mathrm{SMAPE}}{\mathrm{SMAPE}_{\text{Naïve2}}} + \frac{\mathrm{MASE}}{\mathrm{MASE}_{\text{Naïve2}}} \right],$$

where s is the periodicity of the data. $\mathbf{X}, \widehat{\mathbf{X}} \in \mathbb{R}^{F \times C}$ are the ground truth and prediction results of the future with F time pints and C dimensions. $\mathbf{X}_i$ means the i-th future time point. For classification, we use accuracy as the metric. Lastly for anomaly detection, we use F1-Score as the metric.

E.2. Short Term Forecasting

The complete results of short term forecasting are reported in Table 6.

E.3. Long Term Forecasting

The complete results of long term forecasting are reported in 7.

E.4. Anomaly Detection

The complete results of Anomaly Detection are reported in Table 9.

E.5. Classification

The complete results of Classification are reported in 10.

Table 3: Dataset descriptions. The dataset size is organized in (Train, Validation, Test).

Tasks	Dataset	Dim	Series Length	Dataset Size	Information (Frequency)
Forecasting (Long-term)	ETTm1, ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Electricity (15 mins)
	Electricity	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Electricity (Hourly)
	Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Transportation (Hourly)
	Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	Weather (10 mins)
	Exchange	8	{96, 192, 336, 720}	(5120, 665, 1422)	Exchange rate (Daily)
Forecasting (short-term)	M4-Yearly	1	6	(23000, 0, 23000)	Demographic
	M4-Quarterly	1	8	(24000, 0, 24000)	Finance
	M4-Monthly	1	18	(48000, 0, 48000)	Industry
	M4-Weakly	1	13	(359, 0, 359)	Macro
	M4-Daily	1	14	(4227, 0, 4227)	Micro
	M4-Hourly	1	48	(414, 0, 414)	Other
Imputation	ETTm1, ETTm2	7	96	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	96	(8545, 2881, 2881)	Electricity (15 mins)
	Weather	21	96	(36792, 5271, 10540)	Weather (10 mins)
Classification (UEA)	EthanolConcentration	3	1751	(261, 0, 263)	Alcohol Industry
	FaceDetection	144	62	(5890, 0, 3524)	Face (250Hz)
	Handwriting	3	152	(150, 0, 850)	Handwriting
	Heartbeat	61	405	(204, 0, 205)	Heart Beat
	JapaneseVowels	12	29	(270, 0, 370)	Voice
	PEMS-SF	963	144	(267, 0, 173)	Transportation (Daily)
	SelfRegulationSCP1	6	896	(268, 0, 293)	Health (256Hz)
	SelfRegulationSCP2	7	1152	(200, 0, 180)	Health (256Hz)
	SpokenArabicDigits	13	93	(6599, 0, 2199)	Voice (11025Hz)
	UWaveGestureLibrary	3	315	(120, 0, 320)	Gesture
Anomaly Detection	SMD	38	100	(566724, 141681, 708420)	Server Machine
	MSL	55	100	(44653, 11664, 73729)	Spacecraft
	SMAP	25	100	(108146, 27037, 427617)	Spacecraft
	SWaT	51	100	(396000, 99000, 449919)	Infrastructure
	PSM	25	100	(105984, 26497, 87841)	Server Machine

Table 5: Standard deviation and statistical tests for our **LETO** method and the strongest baseline **ModernTCN** on the M4 dataset (short-term forecasting). Lower is better. Confidence is derived from a paired two-tailed t-test over five runs.

Frequency	LETO (Ours)			ModernTCN (2024)			Confidence
	SMAPE	MASE	OWA	SMAPE	MASE	OWA	
Yearly	13.183 ± 0.115	2.941 ± 0.028	0.754 ± 0.022	13.226 ± 0.118	2.957 ± 0.031	0.777 ± 0.025	99%
Quarterly	9.953 ± 0.101	1.150 ± 0.015	0.851 ± 0.015	9.971 ± 0.105	1.167 ± 0.017	0.878 ± 0.018	95%
Monthly	12.517 ± 0.115	0.935 ± 0.014	0.853 ± 0.014	12.556 ± 0.120	0.917 ± 0.015	0.866 ± 0.016	95%
Others	4.583 ± 0.084	2.797 ± 0.027	0.900 ± 0.021	4.715 ± 0.090	3.107 ± 0.028	0.986 ± 0.024	99%
Averaged	11.658 ± 0.112	1.541 ± 0.022	0.832 ± 0.018	11.698 ± 0.120	1.556 ± 0.024	0.838 ± 0.020	95%

Table 6: Full results for the short-term forecasting task in the M4 dataset. $*$. in the Transformers indicates the name of $*$former. *Stationary* means the Non-stationary Transformer. A lower SMAPE, MASE, and OWA indicate a better prediction. As a convention for all experimental results, best performance is highlighted in **red**, and the second-best is underlined. We take the average of 5 separate runs for each prediction frequency.

Models		LETO (Ours)	ModernTCN (2024)	PatchTST (2023)	TimesNet (2023)	N-HiTS (2023)	N-BEATS* (2022)	ETS* (2019)	LightTS (2022)	DLinear (2022a)	FED* (2023a)	Stationary (2022b)	Auto* (2022b)	Pyra* (2021)	In* (2021)	Re* (2021)
Yearly	SMAPE	**13.183**	13.226	13.258	13.387	13.418	13.436	18.009	14.247	16.965	13.728	13.717	13.974	15.530	14.727	16.169
	MASE	**2.941**	2.957	2.985	2.996	3.045	3.043	4.487	3.109	4.283	3.048	3.078	3.134	3.711	3.418	3.800
	OWA	**0.754**	0.777	0.781	0.786	0.793	0.794	1.115	0.827	1.058	0.803	0.807	0.822	0.942	0.881	0.973
Quarterly	SMAPE	**9.953**	9.971	10.179	10.100	10.202	10.124	13.376	11.364	12.145	10.792	10.958	11.338	15.449	11.360	13.313
	MASE	**1.150**	1.167	0.803	1.182	1.194	1.169	1.906	1.328	1.520	1.283	1.325	1.365	2.350	1.401	1.775
	OWA	0.851	0.878	**0.803**	0.890	0.899	0.886	1.302	1.000	1.106	0.958	0.981	1.012	1.558	1.027	1.252
Monthly	SMAPE	**12.517**	12.556	12.641	12.670	12.791	12.677	14.588	14.014	13.514	14.260	13.917	13.958	17.642	14.062	20.128
	MASE	0.935	**0.917**	0.930	0.933	0.969	0.937	1.368	1.053	1.037	1.102	1.097	1.103	1.913	1.141	2.614
	OWA	**0.853**	0.866	0.876	0.878	0.899	0.880	1.149	0.981	0.956	1.012	0.998	1.002	1.511	1.024	1.927
Others	SMAPE	**4.583**	4.715	4.946	4.891	5.061	4.925	7.267	15.880	6.709	4.954	6.302	5.485	24.786	24.460	32.491
	MASE	**2.797**	3.107	2.985	3.302	3.216	3.391	5.240	11.434	4.953	3.264	4.064	3.865	18.581	20.960	33.355
	OWA	**0.9001**	0.986	1.044	1.035	1.040	1.053	1.591	3.474	1.487	1.036	1.304	1.187	5.538	5.013	8.679
Weighted Average	SMAPE	**11.658**	11.698	11.807	11.829	11.927	11.851	14.718	13.525	13.639	12.840	12.780	12.909	16.987	14.086	18.200
	MASE	**1.541**	1.556	1.590	1.585	1.613	1.599	2.408	2.111	2.095	1.701	1.756	1.771	3.265	2.718	4.223
	OWA	**0.832**	0.838	0.851	0.851	0.861	0.855	1.172	1.051	1.051	0.918	0.930	0.939	1.480	1.230	1.775

F. Limitations and Future Work

We note LETO has a few limitations worth acknowledging. First, the use of gradient-based meta in-context updates at test time, while powerful, introduces additional computational overhead compared to traditional non-adaptive sequence models. Although our dual-form implementation and parallel training strategies mitigate some of this cost, the memory and compute requirements may still be prohibitive in resource-constrained settings, particularly for long-horizon forecasting tasks.

Second, while LETO is designed to model both cross-time and cross-variate dependencies, its reliance on Taylor approximations for the variate attention mechanism may limit its capacity to fully capture complex, high-order variate interactions in some datasets. More expressive non-parametric approximators or learned kernel functions could offer improved generalization and efficiency.

Finally, our current formulation assumes access to reasonably stationary statistics at test time for the meta-memorization process to be effective. In highly non-stationary environments or under strong distribution shifts, the learned test-time updates may generalize poorly, leading to suboptimal performance.

Table 7: Complete experiments on long term forecasting tasks over four prediction lengths: {96, 192, 336, 720}. A lower MAE and MSE indicates a better prediction. As a convention for all experimental results, best performance is highlighted in **red**, and the second-best is underlined. We take the average of 5 separate runs for each prediction length.

		LETO (ours)		TimeMixer (2024)		Simba (2024)		TCN (2024)		iTransformer (2024a)		RLinear (2023)		PatchTST (2023)		Crossformer (2023)		TiDE (2023)		TimesNet (2023)		DLinear (2023c)		SCINet (2022a)		FEDformer (2022b)		Stationary (2022c)		Autoformer (2021)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.312	**0.343**	0.320	0.357	0.342	0.360	**0.292**	0.346	0.334	0.368	0.355	0.376	0.329	0.367	0.404	0.426	0.364	0.387	0.338	0.375	0.345	0.372	0.418	0.438	0.379	0.419	0.386	0.398	0.505	0.475
	192	**0.330**	**0.365**	0.361	0.381	0.363	0.382	0.332	0.368	0.377	0.391	0.391	0.392	0.367	0.385	0.450	0.451	0.398	0.404	0.374	0.387	0.380	0.389	0.439	0.450	0.426	0.441	0.459	0.444	0.553	0.496
	336	**0.355**	**0.384**	0.390	0.404	0.395	0.405	0.365	0.391	0.426	0.420	0.424	0.415	0.399	0.410	0.532	0.515	0.428	0.425	0.410	0.411	0.413	0.413	0.490	0.485	0.445	0.459	0.495	0.464	0.621	0.537
	720	**0.391**	**0.408**	0.454	0.441	0.451	0.437	0.416	0.417	0.491	0.459	0.487	0.450	0.454	0.439	0.666	0.589	0.487	0.461	0.478	0.450	0.474	0.453	0.595	0.550	0.543	0.490	0.585	0.516	0.671	0.561
	Avg	**0.347**	**0.375**	0.381	0.395	0.383	0.396	0.351	0.381	0.407	0.410	0.414	0.407	0.387	0.400	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407	0.485	0.481	0.448	0.452	0.481	0.456	0.588	0.517
ETTm2	96	**0.164**	0.248	0.175	0.258	0.177	0.263	0.166	0.256	0.180	0.264	0.182	0.265	0.175	0.259	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.292	0.286	0.377	0.203	0.287	0.192	0.274	0.255	0.339
	192	0.217	0.284	0.237	0.299	0.245	0.306	0.222	0.293	0.250	0.309	0.246	0.304	0.241	0.302	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.362	0.399	0.445	0.269	0.328	0.280	0.339	0.281	0.340
	336	0.266	0.312	0.298	0.340	0.304	0.343	0.272	0.324	0.311	0.348	0.307	0.342	0.305	0.343	0.597	0.542	0.377	0.422	0.321	0.351	0.369	0.427	0.637	0.591	0.325	0.366	0.334	0.361	0.339	0.372
	720	0.349	0.363	0.391	0.396	0.400	0.399	0.351	0.381	0.412	0.407	0.407	0.398	0.402	0.400	1.730	1.042	0.558	0.524	0.408	0.403	0.554	0.522	0.960	0.735	0.421	0.415	0.417	0.413	0.433	0.432
	Avg	**0.249**	**0.302**	0.275	0.323	0.271	0.327	0.253	0.314	0.288	0.332	0.286	0.327	0.281	0.326	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401	0.571	0.537	0.305	0.349	0.306	0.347	0.327	0.371
ETTh1	96	0.365	0.383	0.375	0.400	0.379	0.395	0.368	0.394	0.386	0.405	0.386	0.395	0.414	0.419	0.423	0.448	0.479	0.464	0.384	0.402	0.386	0.400	0.654	0.599	0.376	0.419	0.513	0.491	0.449	0.459
	192	0.396	0.400	0.429	0.421	0.432	0.424	0.405	0.413	0.441	0.436	0.437	0.424	0.460	0.445	0.471	0.474	0.525	0.492	0.436	0.429	0.437	0.432	0.719	0.631	0.420	0.448	0.534	0.504	0.500	0.482
	336	0.461	0.462	0.484	0.458	0.473	0.443	0.391	0.412	0.487	0.458	0.479	0.446	0.501	0.466	0.570	0.546	0.565	0.515	0.491	0.469	0.481	0.459	0.778	0.659	0.459	0.465	0.588	0.535	0.521	0.496
	720	0.427	0.428	0.498	0.482	0.483	0.469	0.450	0.461	0.503	0.491	0.481	0.470	0.500	0.488	0.653	0.621	0.594	0.558	0.521	0.500	0.519	0.516	0.836	0.699	0.506	0.507	0.643	0.616	0.514	0.512
	Avg	**0.393**	**0.401**	0.447	0.440	0.441	0.432	0.404	0.420	0.454	0.447	0.446	0.434	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.747	0.647	0.440	0.460	0.570	0.537	0.496	0.487
ETTh2	96	0.258	0.337	0.289	0.341	0.290	0.339	0.263	0.332	0.297	0.349	0.288	0.338	0.302	0.348	0.745	0.584	0.400	0.440	0.340	0.374	0.333	0.387	0.707	0.621	0.358	0.397	0.476	0.458	0.346	0.388
	192	0.316	0.379	0.372	0.392	0.373	0.390	0.320	0.374	0.380	0.400	0.374	0.390	0.388	0.400	0.877	0.656	0.528	0.509	0.402	0.414	0.477	0.476	0.860	0.689	0.429	0.439	0.512	0.493	0.456	0.452
	336	0.309	0.379	0.386	0.414	0.376	0.406	0.313	0.376	0.428	0.432	0.415	0.426	0.426	0.433	1.043	0.731	0.643	0.571	0.452	0.452	0.594	0.541	1.000	0.744	0.496	0.487	0.552	0.551	0.482	0.486
	720	0.389	0.430	0.412	0.434	0.407	0.431	0.392	0.433	0.427	0.445	0.420	0.440	0.431	0.446	1.104	0.763	0.874	0.679	0.462	0.468	0.831	0.657	1.249	0.838	0.463	0.474	0.562	0.560	0.515	0.511
	Avg	**0.318**	0.381	0.364	0.395	0.361	**0.377**	0.322	0.379	0.383	0.407	0.374	0.398	0.387	0.407	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515	0.954	0.723	0.437	0.449	0.526	0.516	0.450	0.459
Exchange	96	0.079	0.208	0.090	0.235	-	-	0.080	0.196	0.086	0.206	0.093	0.217	0.088	0.205	0.256	0.367	0.094	0.218	0.107	0.234	0.088	0.218	0.267	0.396	0.148	0.278	0.111	0.237	0.197	0.323
	192	0.164	0.298	0.187	0.343	-	-	0.166	0.288	0.177	0.299	0.184	0.307	0.176	0.299	0.470	0.509	0.184	0.307	0.226	0.344	0.176	0.315	0.351	0.459	0.271	0.315	0.219	0.335	0.300	0.369
	336	0.308	0.329	0.353	0.473	-	-	0.307	0.398	0.331	0.417	0.351	0.432	0.301	0.397	1.268	0.883	0.349	0.431	0.367	0.448	0.313	0.427	1.324	0.853	0.460	0.427	0.421	0.476	0.509	0.524
	720	0.637	0.621	0.934	0.761	-	-	0.656	0.582	0.847	0.691	0.886	0.714	0.901	0.714	1.767	1.068	0.852	0.698	0.964	0.746	0.839	0.695	1.058	0.797	1.195	0.695	1.092	0.769	1.447	0.941
	Avg	**0.297**	0.364	0.391	0.453	-	-	0.302	0.366	0.360	0.403	0.378	0.417	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414	0.750	0.626	0.519	0.429	0.461	0.454	0.613	0.539
Traffic	96	0.380	0.247	0.462	0.285	0.468	0.268	0.368	0.253	0.395	0.268	0.649	0.389	0.462	0.295	0.522	0.290	0.805	0.493	0.593	0.321	0.650	0.396	0.788	0.499	0.587	0.366	0.612	0.338	0.613	0.388
	192	0.391	0.258	0.473	0.296	0.413	0.317	0.379	0.261	0.417	0.276	0.601	0.366	0.466	0.296	0.530	0.293	0.756	0.474	0.617	0.336	0.598	0.370	0.789	0.505	0.604	0.373	0.613	0.340	0.616	0.382
	336	0.409	0.266	0.498	0.296	0.529	0.284	0.397	0.270	0.433	0.283	0.609	0.369	0.482	0.304	0.558	0.305	0.762	0.477	0.629	0.336	0.605	0.373	0.797	0.508	0.621	0.383	0.618	0.328	0.622	0.337
	720	0.452	0.297	0.506	0.313	0.564	0.297	0.440	0.296	0.467	0.302	0.647	0.387	0.514	0.322	0.589	0.328	0.719	0.449	0.640	0.350	0.645	0.394	0.841	0.523	0.626	0.382	0.653	0.355	0.660	0.408
	Avg	0.408	**0.267**	0.484	0.297	0.493	0.291	**0.398**	0.270	0.428	0.282	0.626	0.378	0.481	0.304	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383	0.804	0.509	0.610	0.376	0.624	0.340	0.628	0.379
Weather	96	0.155	0.203	0.163	0.209	0.176	0.219	0.149	0.200	0.174	0.214	0.192	0.232	0.177	0.218	0.158	0.230	0.202	0.261	0.172	0.220	0.196	0.255	0.221	0.306	0.217	0.296	0.173	0.223	0.266	0.336
	192	0.173	0.240	0.222	0.260	0.222	0.260	0.196	0.245	0.221	0.254	0.240	0.271	0.225	0.259	0.206	0.277	0.242	0.298	0.219	0.261	0.237	0.296	0.261	0.340	0.276	0.336	0.245	0.285	0.307	0.367
	336	0.232	0.260	0.251	0.287	0.275	0.297	0.238	0.277	0.278	0.296	0.292	0.307	0.278	0.297	0.272	0.335	0.287	0.335	0.280	0.306	0.283	0.335	0.309	0.378	0.339	0.380	0.321	0.338	0.359	0.395
	720	0.307	0.309	0.350	0.349	0.350	0.349	0.314	0.334	0.358	0.347	0.364	0.353	0.354	0.348	0.398	0.418	0.351	0.366	0.365	0.359	0.345	0.381	0.377	0.427	0.403	0.428	0.414	0.410	0.419	0.428
	Avg	**0.216**	**0.253**	0.240	0.271	0.255	0.280	0.224	0.264	0.258	0.278	0.272	0.291	0.259	0.281	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.292	0.363	0.309	0.360	0.288	0.314	0.338	0.382
ECL	96	0.136	0.233	0.153	0.247	0.165	0.253	0.129	0.226	0.148	0.240	0.201	0.281	0.181	0.270	0.219	0.314	0.237	0.329	0.168	0.272	0.197	0.282	0.247	0.345	0.193	0.308	0.169	0.273	0.201	0.317
	192	0.144	0.221	0.166	0.256	0.173	0.262	0.143	0.239	0.162	0.253	0.201	0.283	0.188	0.274	0.231	0.322	0.236	0.330	0.184	0.289	0.196	0.285	0.257	0.355	0.201	0.315	0.182	0.286	0.222	0.334
	336	0.154	0.253	0.185	0.277	0.188	0.277	0.161	0.259	0.178	0.269	0.215	0.298	0.204	0.293	0.246	0.337	0.249	0.344	0.198	0.300	0.209	0.301	0.269	0.369	0.214	0.329	0.200	0.304	0.231	0.338
	720	0.162	0.261	0.225	0.310	0.214	0.305	0.191	0.286	0.225	0.317	0.257	0.331	0.246	0.324	0.280	0.363	0.284	0.373	0.220	0.320	0.245	0.333	0.299	0.390	0.246	0.355	0.222	0.321	0.254	0.361
	Avg	**0.149**	**0.247**	0.182	0.272	0.185	0.274	0.156	0.253	0.178	0.270	0.219	0.298	0.205	0.290	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300	0.268	0.365	0.214	0.327	0.193	0.296	0.227	0.338

Table 8: Standard deviation and statistical tests for **LETO** vs. the strongest baseline **ModernTCN** on long-term forecasting (lower is better). Confidence levels derive from a paired two-tailed t-test over five seeds.

Dataset	LETO (Ours) MSE	LETO (Ours) MAE	ModernTCN (2024) MSE	ModernTCN (2024) MAE	Confidence
ETTm1	0.347 ± 0.010	0.375 ± 0.012	0.351 ± 0.011	0.381 ± 0.013	99%
ETTm2	0.249 ± 0.009	0.302 ± 0.011	0.253 ± 0.010	0.314 ± 0.013	95%
ETTh1	0.393 ± 0.012	0.401 ± 0.014	0.404 ± 0.013	0.420 ± 0.015	99%
ETTh2	0.318 ± 0.010	0.381 ± 0.012	0.322 ± 0.011	0.379 ± 0.013	95%
Exchange	0.297 ± 0.016	0.364 ± 0.018	0.302 ± 0.017	0.366 ± 0.019	95%
Traffic	0.408 ± 0.020	0.267 ± 0.012	0.398 ± 0.019	0.270 ± 0.013	90%
Weather	0.216 ± 0.009	0.253 ± 0.011	0.224 ± 0.010	0.264 ± 0.012	95%
ECL	0.149 ± 0.007	0.247 ± 0.009	0.156 ± 0.008	0.253 ± 0.010	99%

Table 9: Full results for the anomaly detection task. The P, R and F1 represent the precision, recall and F1-score in percentage respectively. A higher value of P, R and F1 indicates a better performance. Best performance is highlighted in **red**, and the second-best is underlined. We take the average of 5 separate runs for each dataset.

Datasets		SMD			MSL			SMAP			SWaT			PSM			Avg F1
Metrics		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	(%)
LSTM	(1997)	78.52	65.47	71.41	78.04	86.22	81.93	91.06	57.49	70.48	78.06	91.72	84.34	69.24	99.53	81.67	77.97
Transformer	(2017)	83.58	76.13	79.56	71.57	87.37	78.68	89.37	57.12	69.70	68.84	96.53	80.37	62.75	96.56	76.07	76.88
LogTrans	(2019)	83.46	70.13	76.21	73.05	87.37	79.57	89.15	57.59	69.97	68.67	97.32	80.52	63.06	98.00	76.74	76.60
TCN	(2019)	84.06	79.07	81.49	75.11	82.44	78.60	86.90	**59.23**	70.45	76.59	95.71	85.09	54.59	**99.77**	70.57	77.24
Reformer	(2020)	82.58	69.24	75.32	**85.51**	83.31	84.40	90.91	57.44	70.40	72.50	96.53	82.80	59.93	95.38	73.61	77.31
Informer	(2021)	86.60	77.23	81.65	81.77	86.48	84.06	90.11	57.13	69.92	70.29	96.75	81.43	64.27	96.33	77.10	78.83
Anomaly*	(2021)	**88.91**	82.23	85.49	79.61	87.37	83.31	91.85	58.11	71.18	72.51	97.32	83.10	68.35	94.72	79.40	80.50
Pyraformer	(2021)	85.61	80.61	83.04	83.81	85.93	84.86	92.54	57.71	71.09	87.92	96.00	91.78	71.67	96.02	82.08	82.57
Autoformer	(2021)	88.06	82.35	85.11	77.27	80.92	79.05	90.40	58.62	71.12	89.85	95.81	92.74	99.08	88.15	93.29	84.26
LSSL	(2021)	78.51	65.32	71.31	77.55	88.18	82.53	89.43	53.43	66.90	79.05	93.72	85.76	66.02	92.93	77.20	76.74
Stationary	(2022b)	88.33	81.21	84.62	68.55	89.14	77.50	89.37	59.02	71.09	68.03	96.75	79.88	97.82	96.76	97.29	82.08
DLinear	(2023a)	83.62	**71.52**	77.10	84.34	85.42	84.88	92.32	55.41	69.26	80.91	95.30	87.52	98.28	89.26	93.55	82.46
ETSformer	(2022)	87.44	79.23	83.13	85.13	84.93	85.03	92.25	55.75	69.50	90.02	**80.36**	84.91	**99.31**	85.28	91.76	82.87
LightTS	(2022a)	87.10	78.42	82.53	82.40	75.78	78.95	92.58	55.27	69.21	91.98	94.72	93.33	98.37	95.97	97.15	84.23
FEDformer	(2022b)	87.95	82.39	85.08	77.14	80.07	78.57	90.47	58.10	70.76	90.17	96.42	93.19	97.31	97.16	97.23	84.97
TimesNet (I)	(2023)	87.76	82.63	85.12	82.97	85.42	84.18	91.50	57.80	70.85	88.31	96.24	92.10	98.22	92.21	95.21	85.49
TimesNet (R)	(2023)	88.66	83.14	85.81	83.92	86.42	85.15	92.52	58.29	71.52	86.76	97.32	91.74	98.19	96.76	**97.47**	86.34
CrossFormer	(2023)	83.6	76.61	79.70	84.68	83.71	84.19	92.04	55.37	69.14	88.49	93.48	90.92	97.16	89.73	93.30	83.45
PatchTST	(2023)	87.42	81.65	84.44	84.07	86.23	85.14	92.43	57.51	70.91	80.70	94.93	87.24	98.87	93.99	96.37	84.82
ModernTCN	(2024)	87.86	83.85	85.81	83.94	85.93	84.92	93.17	57.69	71.26	91.83	95.98	93.86	98.09	96.38	97.23	86.62
LETO	(ours)	**88.20**	**85.52**	**86.84**	83.50	**89.27**	**86.29**	**93.20**	57.10	70.81	**92.00**	96.73	**94.31**	99.20	94.61	96.85	**87.02**

G. Broader Impact

LETO has demonstrated strong performance as a general-purpose model for time series pattern recognition, achieving competitive results across a wide range of tasks including forecasting, classification, and anomaly detection. Its versatility makes it well-suited for deployment in diverse real-world scenarios, such as energy and power demand forecasting with pronounced seasonal trends, weather prediction under complex and dynamic conditions, financial market modeling in rapidly shifting environments, and demand forecasting within supply chains. Furthermore, LETO has shown particular promise in industrial anomaly detection tasks, which often require robustness to noise and structural variability. These capabilities highlight LETO's potential as a foundational model for advancing time series analysis across multiple applied domains.

H. Compute Resources

For experiments, we utilized up to 4 NVIDIA A6000 and A6000 ADA GPUs.

Table 10: Full results for the classification task (accuracy %). We omit "former" from the names of Transformer-based methods. For all methods, the standard deviation is less than 0.1%. A higher average accuracy indicates a better prediction. Best performance is highlighted in **red**, and the second-best is underlined. We take the average of 5 separate runs for each dataset.

Datasets / Models	LSTM (1997)	LSTNet (2018)	LSSL	Trans. (2017)	Re. (2020)	In. (2021)	Pyra. (2021)	Auto. (2021)	Station. (2022b)	FED. (2022b)	/ETS. (2022)	/Flow. (2022a)	/DLinear (2023a)	/LightTS. (2022a)	/TimesNet (2023)	/PatchTST (2023)	/MTCN (2024)	/LETO (ours)
EthanolConcentration	32.3	39.9	31.1	32.7	31.9	31.6	30.8	31.6	32.7	31.2	28.1	33.8	32.6	29.7	35.7	32.8	<u>36.3</u>	**38.8**
FaceDetection	57.7	65.7	66.7	67.3	68.6	67.0	65.7	68.4	68.0	66.0	66.3	67.6	68.0	67.5	68.6	68.3	<u>70.8</u>	**71.3**
Handwriting	15.2	25.8	24.6	32.0	27.4	32.8	29.4	36.7	31.6	28.0	32.5	33.8	27.0	26.1	32.1	29.6	<u>30.6</u>	**32.9**
Heartbeat	72.2	77.1	72.7	76.1	77.1	80.5	75.6	74.6	73.7	73.7	71.2	77.6	75.1	75.1	78.0	74.9	<u>77.2</u>	**78.3**
JapaneseVowels	79.7	98.1	98.4	98.7	97.8	98.9	98.4	96.2	99.2	98.4	95.9	98.9	96.2	96.2	98.4	97.5	**98.8**	<u>98.5</u>
PEMS-SF	39.9	86.7	86.1	82.1	82.7	81.5	83.2	82.7	87.3	80.9	86.0	83.8	75.1	88.4	89.6	89.3	<u>89.1</u>	**89.6**
SelfRegulationSCP1	68.9	84.0	90.8	92.2	90.4	90.1	88.1	84.0	89.4	88.7	89.6	92.5	87.3	89.8	91.8	90.7	<u>93.4</u>	**94.4**
SelfRegulationSCP2	46.6	52.8	52.2	53.9	56.7	53.3	53.3	50.6	57.2	54.4	55.0	56.1	50.5	51.1	57.2	57.8	<u>60.3</u>	**61.1**
SpokenArabicDigits	31.9	100.0	100.0	98.4	97.0	100.0	<u>99.6</u>	100.0	100.0	100.0	100.0	98.8	81.4	**100.0**	99.0	98.3	98.7	98.7
UWaveGestureLibrary	41.2	87.8	85.9	85.6	85.6	85.6	83.4	85.9	87.5	85.3	85.0	86.6	82.1	80.3	85.3	85.8	<u>86.7</u>	**87.1**
Average Accuracy	48.6	71.8	70.9	71.9	71.5	72.1	70.8	71.1	72.7	70.7	71.0	73.0	67.5	70.4	73.6	72.5	<u>74.2</u>	**75.07**

I. Linear Recurrent Expressiveness

We show that our LETO can recover the 2D linear recurrent models that are proven to model full-rank matrices (Behrouz et al., 2024d; Baron et al., 2024). To this end, we show that a special instance of our LETO is equivalent to these linear 2D recurrent models. We let the chunk size to be the size of the sequence length. Therefore, for every $1 \leq t \leq T$, we have:

$$\nabla \ell(\mathcal{M}_0^{(1)}; \mathbf{k}_t, \mathbf{v}_t) = (\mathcal{M}_0^{(1)} \mathbf{k}_t - \mathbf{v}_t)\mathbf{k}_t^\top, \tag{11}$$

where $\mathcal{M}_0^{(1)}$ is the initial state of the memory, which we let $\mathcal{M}_0^{(1)} = \mathbf{I}$ for the simplicity. Replacing this gradient in Equation Variant 2, we have:

$$\mathcal{M}_{t,v}^{(1)} = \alpha_{t,v}\mathcal{M}_{t-1,v}^{(1)} - \eta_{t,v}\left(\underbrace{(\mathbf{k}_t - \mathbf{v}_t)}_{\mathbf{u}_t}\mathbf{k}_t^\top\right) + \beta_{t,v}\mathcal{M}_{t-1,v}^{(2)} - \gamma_{t,v}\left(\mathcal{M}_t^{(2)}\mathbf{k}_t\mathbf{k}_t^\top - \mathbf{v}_t\mathbf{k}_t^\top\right), \tag{12}$$

where we let $\eta_{t,v} = \gamma_{t,v} = 1$. Also, for the attention module, we use polynomials with degree 1 to approximate the softmax attention (which is the special instance and the weaker version of our design, i.e., considering only the first two terms of the Taylor series). The resulting formula can be written as:

$$\mathcal{M}_{t,v}^{(1)} = \alpha_{t,v}\mathcal{M}_{t-1,v}^{(1)} - \eta_{t,v}\mathbf{u}_t\mathbf{k}_t^\top + \beta_{t,v}\mathcal{M}_{t-1,v}^{(2)} - \gamma_{t,v}\mathcal{M}_t^{(2)} + \gamma_{t,v}\mathbf{u}_t\mathbf{k}_t^\top, \tag{13}$$

which is equivalent to the 2-dimensional linear recurrence with diagonal transition matrix. Therefore, as proven by Baron et al. (2024), the recurrence can model full-rank matrix.

On the other hand, the univariate version of this recurrence (i.e., $\gamma_{t,v} = 0$) results in linear attention formulation, which is limited and cannot express full-rank matrices.

J. Visualizations

J.1. Long Term Forecasting

J.2. Ultra Long Term Forecasting

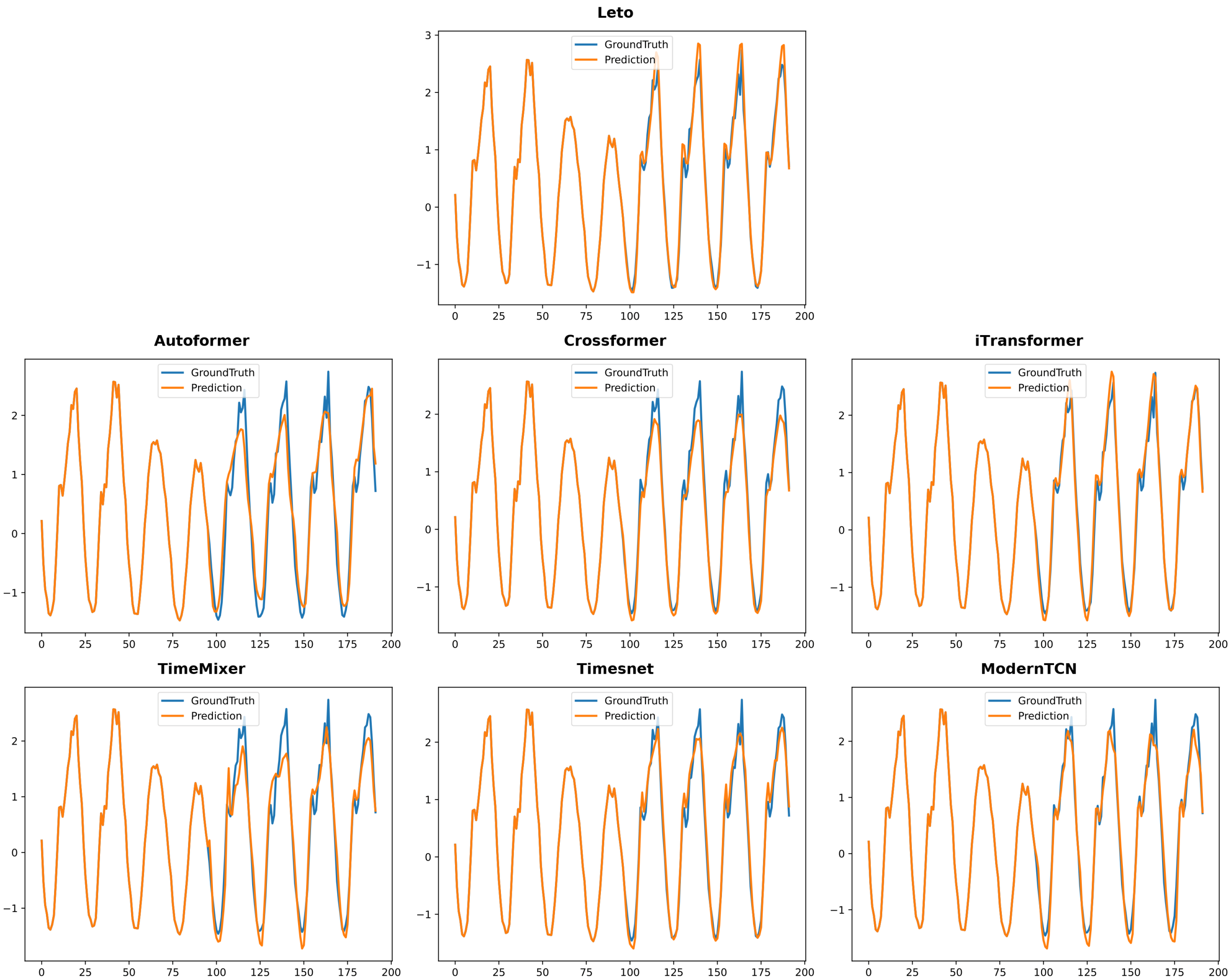
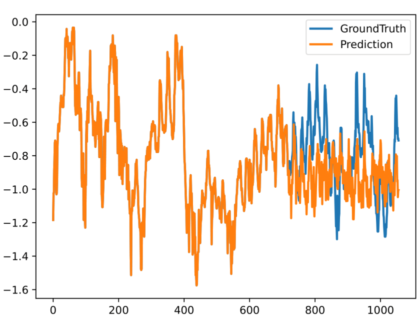
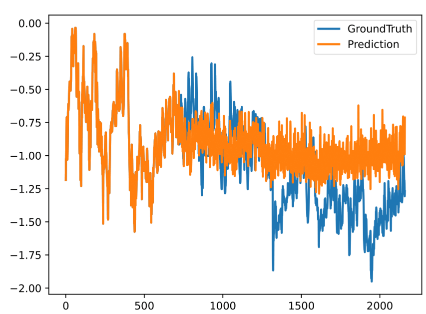


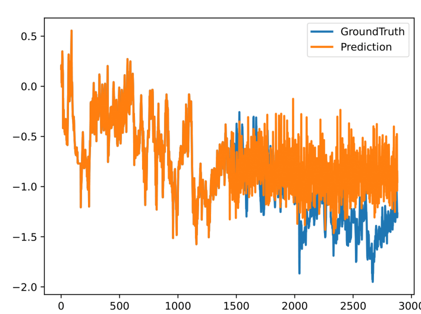
Figure 3: Visualization of Traffic Long Term Forecasting results given by models under the input-96-predict-96 setting. The blue lines stand for the ground truth and the orange lines stand for predicted values.



(a) Input-720, Predict-720



(b) Input-720, Predict-1440



(c) Input-1440 –Predict-1440

Figure 4: Ultra-long-horizon forecasting examples on **ETTh1**. Blue=Ground Truth, Orange=Prediction.