ORTCL: Towards Continual Learning of Time Series Foundation Models on Streaming Data via Orthogonal Rotation

Li Lin[1], **Xinrui Zhang**[1], **Qi Zhang**[1], **Shuai Wang**[1], **Kaiwen Xia**[1,2*]

[1]Southeast University, Nanjing, China
[2]JD Logistics, Beijing, China
{linli321, xinrui_zhang, 230228520, shuaiwang, kevinxia}@seu.edu.cn

Abstract

Time Series Foundation Models (TSFMs) have emerged as a promising approach in time series analysis. Due to the large-scale parameters of TSFMs and pretraining cost, how to adapt TDFMs in streaming data is always the key factor constraining their application effectiveness. Because streaming data often experiences data distribution and task drifts, which cannot be learnt by offline training. Existing methods typically address streaming data modeling with continuous learning through model fine-tuning or model editing. However, fine-tuning incurs significant computational costs, while editing methods can lead to shifts in the original feature space during streaming updates. To address these limitations, we propose a novel **O**rthogonal **R**otation **T**ransformation-based **C**ontinuous **L**earning method, called ORTCL, for TSFMs. Our key insight is to apply orthogonal matrix rotations to the input and output feature spaces of the TSFMs during model editing. This preserves the metric structure of the original feature space and enables new data to be directly mapped into the existing feature space of the TSFMs. Specifically, we obtain the orthogonal matrix for the input layer via singular value decomposition and derive the corresponding transformation matrix for the output layer through least squares optimization. Extensive experimental results demonstrate that ORTCL outperforms existing methods in both single-domain and cross-domain streaming time series forecasting tasks, effectively mitigating catastrophic forgetting.

Code — https://github.com/kaiwxai/ORTCL

Introduction

Time Series Foundational Models (TSFMs) have rapidly emerged as a new paradigm in time series analysis (Liang et al. 2024; Goswami et al. 2024; Liu et al. 2025d), thanks to their efficient modeling capabilities for large-scale cross-domain data, demonstrating unified and excellent performance in downstream tasks, such as forecasting (Liu et al. 2025a,c; Xia et al. 2025b,a; Lin et al. 2024) and decision making (Xia et al. 2023) in urban systems. However, in real-world applications, time series data often arrives in a streaming format and is frequently subject to dynamic shifts in data distribution and task objectives, necessitating that TSFMs continuously adapt to evolving data distributions and emerging domains (Yang et al. 2024). It makes continual learning in streaming data on TSFMs a critical problem to achieve efficient and practical deployment.

Existing methods for continual learning with foundational models are primarily divided into two categories (Wang et al. 2024; Zhou et al. 2024): *model fine-tuning* and *model editing*. Fine-tuning is a direct and effective strategy for updating model knowledge, achieving good performance on target data. However, it typically requires substantial computational resources (Ding et al. 2023; Liu et al. 2022), and when dealing with sequentially arriving streaming data, fine-tuning methods are prone to catastrophic forgetting (Wang et al. 2024). In response, model editing methods have emerged, aiming to update target knowledge through localized modifications of the model structure or parameters while maximizing the retention of existing knowledge. Model editing methods (Yao et al. 2024) have two main types as follows:

The first type is parameter-preserving methods, which introduce additional modules without altering the original model parameters, such as the incorporation of adapter modules (Wan et al. 2025; Wu et al. 2025; Han et al. 2025; Hartvigsen et al. 2023). However, as the number of modules increases, these methods consume more storage resources, limiting their scalability in streaming data scenarios. Another commonly used continuous learning method is experience replay (Mitchell et al. 2022), which alleviates forgetting by storing historical data but may incur significant storage overhead. In contrast, the second type encompasses parameter-modifying methods, which determine the parameters that need to be adjusted (denoted as Δ) through strategies such as "localization-editing" (Fang et al. 2024; Meng et al. 2022a). For example, given original model parameters $\mathbf{W}$, the updated parameters are expressed as $\mathbf{W}^* = \mathbf{W} + \Delta$, where Δ is selected to achieve knowledge updates with minimal disturbance to the distribution of the model's hidden layer representations. This careful adjustment seeks to balance integrating new knowledge and retaining prior knowledge. Notably, recent methods (Fang et al. 2024) leverage the mathematical properties of matrix projection and null space to ensure that the distribution of hidden representations remains unchanged after model editing. As the updates Δ accumulate through nonlinear operations, the continuous

*Corresponding author

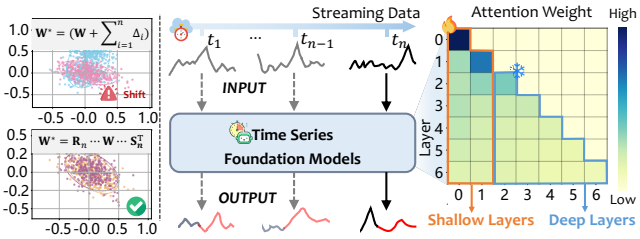

Figure 1: Attention weight distribution and the effect of orthogonal transformation in time series foundation models.

influx of data and the increasing number of accumulations cause the model parameters $\mathbf{W}$ to gradually deviate from the original feature space (Grech 2025; Liu and Chang 2025). This leads to systematic drift in the time series feature representations of the TSFMs, resulting in a decline in performance over time, as evidenced in Figure 6.

To intuitively demonstrate the phenomena of continuous learning on streaming data, we use a TSFMs with a 7-layer Transformer architecture as an example (Liu et al. 2024). We input a batch of time series data into the model and visualize its attention weights (see Figure 1). The results reveal that significant attention weights are predominantly concentrated in the first two layers, indicating their critical role in core feature extraction, while the subsequent layers primarily facilitate feature transmission (Meng et al. 2022a; Zhang et al. 2024). Based on this observation, we propose that during continuous updates, it suffices to adjust the parameters of the shallow layers (e.g., the first two layers) to accommodate new features while freezing the deeper layers to preserve the learned knowledge. To ensure compatibility between the new features in the shallow layers and the frozen deep layers, we introduce orthogonal matrices. First, we apply an orthogonal matrix at the input of the shallow layers to map the new data into their feature space. Subsequently, we apply another orthogonal matrix at the output to align the updated features with the existing structure of the deep layers. Importantly, orthogonal matrices are norm-preserving linear operators by definition. Regardless of how many times these matrices are applied during continual adaptation, they consistently maintain the metric structure of the original model feature space, thereby ensuring the stability and effective retention of knowledge.

Based on the aforementioned insights, we propose an **O**rthogonal **R**otation **T**ransformation-based **C**ontinuous **L**earning method (ORTCL) for the continual learning of TSFMs in streaming data settings. This method unlocks TSFMs' potential for continual learning without altering the distribution of the original model parameter feature space. Technically, we utilize orthogonal rotation transformations to effectively map streaming data features while preserving historical knowledge. First, we perform Singular Value Decomposition (SVD) on the original model parameter matrix to extract the singular vector matrix, which is used to construct the orthogonal rotation matrix for the input layer, facilitating the mapping of new data features. Second, we model the output mapping correction process as a solution to the Procrustes least squares problem to achieve an output orthogonal matrix that satisfies the theoretical optimal solution. Finally, we employ truncated singular value decomposition to quickly obtain a low-rank subspace of the model parameters, enhancing the computational efficiency of the orthogonal matrices. In summary, our contributions include:

- Motivated by the catastrophic forgetting of continual learning in streaming data, we empower the TSFMs with continual learning capabilities through norm-preserving rotations using orthogonal matrices, all while maintaining the integrity of the original feature space.
- We introduce a simple yet general ORTCL, which employs low-rank orthogonal rotation transformations to map input features to the original model parameter feature space and correct the output space, effectively integrating and preserving new and old data features without disrupting the metric of the original feature space.
- Extensive experiments across four datasets demonstrate that ORTCL outperforms various baselines in continual time series forecasting under streaming data conditions, both in single-domain and cross-domain settings.

Preliminary and Problem Definition

Transformer-based Time Series Foundational Models, such as Timer (Liu et al. 2024), the next token $\mathbf{x}$ in the sequence is predicted based on the preceding tokens. The hidden state $\mathbf{h}^l$ at layer l is computed as follows:

$$\mathbf{h}^l = \mathbf{h}^{l-1} + \mathbf{a}^l + \mathbf{m}^l, \mathbf{m}^l = \mathbf{W}^l_{out}\sigma(\mathbf{W}^l_{in}\gamma(\mathbf{h}^{l-1} + \mathbf{a}^l)) \quad (1)$$

where $\mathbf{a}^l$ and $\mathbf{m}^l$ represent the outputs of the attention block and the feedforward network (FFN) layer, respectively. $\mathbf{W}^l_{in}$ and $\mathbf{W}^l_{out}$ are the weight matrices of the FFN layer. σ is activation function, and γ denotes layer normalization. Following (Meng et al. 2022a), in a specific layer, $\mathbf{W}^l_{out}$ (subsequently referred to as $\mathbf{W}$) associates the input key $\mathbf{K} = \sigma(\mathbf{W}^l_{in}\gamma(\mathbf{h}^{l-1} + \mathbf{a}^l))$ with the corresponding output value $\mathbf{V} = \mathbf{m}^l$. This interpretation has inspired many model editing methods aimed at modifying the FFN layer for knowledge updates (Hu et al. 2024; Hase et al. 2023).

Problem Definition. Given a sequentially arriving data stream $\mathcal{T} = \{\mathcal{D}^1, \ldots, \mathcal{D}^T\}$ of time series, where the dataset for the t-th time window is defined as segment $\mathcal{D}^t = \{\mathbf{x}^t, \mathbf{y}^t\}$, with the condition that the data between tasks do not overlap: $\mathcal{D}^t \cap \mathcal{D}^{t'} = \emptyset$ for $t \neq t'$. The learning objective is to make a time series foundational model f_θ that maximizes performance on continuously arriving data stream while retaining historical knowledge, specifically by minimizing the cumulative loss:

$$\min_\theta \sum_{t=1}^{T} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}^t}\Big[\ell\big(f_\theta(\mathbf{x}), \mathbf{y}\big)\Big] \quad (2)$$

where ℓ is the loss function that quantifies the difference between the predicted outputs and the ground truth.

Methodology

In this section, we propose the ORTCL method to enhance the adaptability of TSFMs on streaming data. Specifically,

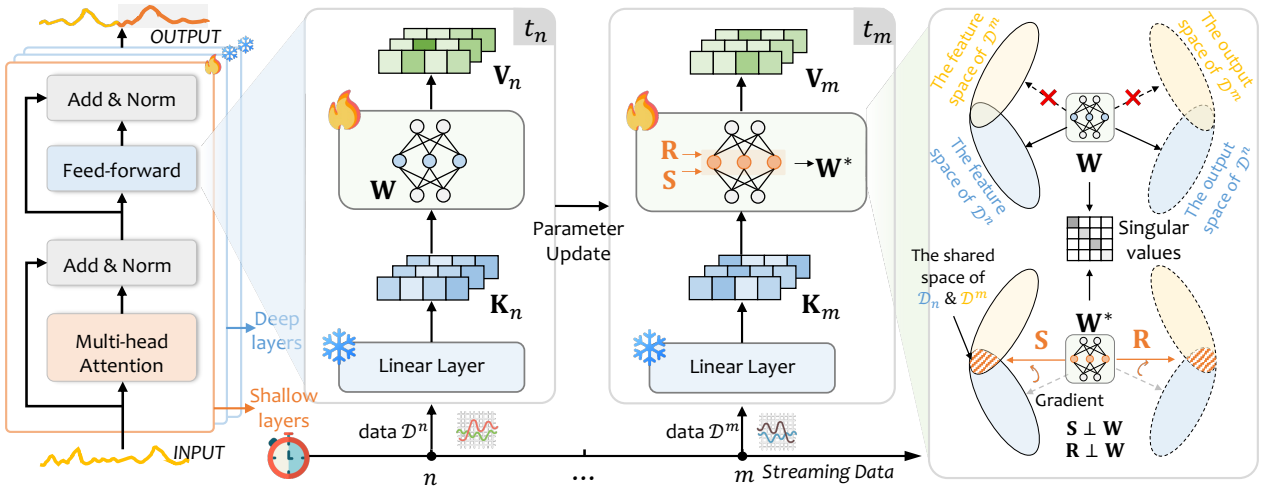

Figure 2: The Overall of the proposed ORTCL framework.

we apply Orthogonal Rotation Transformations (ORT) to the input features of the shallow network to ensure that the model feature space for parameter updates remains orthogonal. Furthermore, we align the shallow output feature space of the TSFMs with the input feature space of deep layers by obtaining the output orthogonal matrix, thereby ensuring the stability of the model's outputs. The overall of ORTCL is shown in Figure 2.

ORT-based Continual Learning Paradigm

In streaming data scenarios, the feature distribution of newly arriving data often exhibits significant differences from that of previously learned data, resulting in the distribution shift. This phenomenon makes it difficult for the backbone model of TSFMs to effectively adapt to the new feature distribution. To address this issue, we propose a continual learning approach that aims to effectively capture the key characteristics of new data while maximizing the retention of previously acquired general knowledge, such as the periodic patterns commonly found in time series, thereby alleviating the problem of catastrophic forgetting. Specifically, let the backbone model f_θ be represented by the parameter matrix $\mathbf{W}$. When processing the segment $\mathcal{D}^t$ at time t, consisting of input keys $\mathbf{K}_t$ and corresponding target outputs $\mathbf{V}_t$. Our goal is to find an updated parameter matrix $\mathbf{W}_t^*$ such that it accurately maps $\mathbf{K}_t \mapsto \mathbf{V}_t$, while preserving the previously learned mappings $\{\mathbf{K}_1, \cdots, \mathbf{K}_{t-1}\} \mapsto \{\mathbf{V}_1, \cdots, \mathbf{V}_{t-1}\}$. Inspired by the orthogonal projection operator (Lezama et al. 2018), we introduce orthogonal matrices $\mathbf{R}_t$ and $\mathbf{S}_t$, which perform rotations on the input and output spaces associated with $\mathbf{W}$, respectively. The update rule is as follows:

$$\mathbf{W}_t^* = \mathbf{R}_t \mathbf{W} \mathbf{S}_t^\top. \tag{3}$$

Here, the left multiplication by $\mathbf{R}_t$ acts on the input space, aiming to ensure that the direction of parameter updates is as orthogonal as possible to the input subspace corresponding to historical features, thereby avoiding interference with the critical weight subspace that preserves previously learned features. The right multiplication by $\mathbf{S}_t^\top$ operates on the output space, rotating the output feature directions to stabilize, ensuring that the shallow outputs of f_θ remain consistent with the inputs of frozen deep layers. Furthermore, applying orthogonal constraints to both the input and output spaces allows the backbone model f_θ to adapt to new data streams while retaining historical knowledge mappings, significantly reducing the risk of forgetting. Additionally, the process of updating the model parameters from time $t-1$ to t can be formalized as the following optimization problem:

$$\min_{\mathbf{R}_t, \mathbf{S}_t} \left(\|\mathbf{W}_t^* \mathbf{K}_t - \mathbf{V}_t\|^2 + \|\mathbf{W}_t^* \mathbf{K}_{t-1} - \mathbf{V}_{t-1}\|^2 \right) \tag{4}$$

where the second term serves as a regularizer, ensuring the model maintains consistent outputs for the previous data at time $t-1$. The rationale for employing orthogonal matrices is based on their geometric properties: orthogonal rotation transformations preserve the lengths and angles of vectors.

Computation of Orthogonal Rotation Matrices

Orthogonal Matrix S. The orthogonal matrix $\mathbf{S}_t$ for input space alignment primarily serves to map the new input key matrix $\mathbf{K}_t$ at the current time step onto the row space of the parameter matrix $\mathbf{W}$, thereby ensuring that the features of the input data lie within the subspace represented by $\mathbf{W}$. The procedure is as follows. First, the SVD is performed on the parameter matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, i.e., $\mathbf{W} = \mathbf{U}_w \mathbf{\Sigma}_w \mathbf{V}_w^\top$, where the columns of $\mathbf{V}_w \in \mathbb{R}^{n \times r}$ form an orthonormal basis of the row space of $\mathbf{W}$, and r is the rank of $\mathbf{W}$. To ensure that the feature mapping of the input data lies within the row space of $\mathbf{W}$ and to prevent new inputs from introducing feature directions unrelated to historical inputs, the historical input key matrix $\mathbf{K}_{t-1}$ and the current input key matrix $\mathbf{K}_t$ are concatenated column-wise as $\mathbf{K} = [\mathbf{K}_{t-1} \mid \mathbf{K}_t]$. Subsequently, we apply to project $\mathbf{K}$ onto the row space of $\mathbf{W}$ by the orthogonal projection operator $\mathbf{V}_w \mathbf{V}_w^\top$, resulting in the projected matrix, and then the SVD is performed:

$$\mathbf{K}_{\text{proj}} = \mathbf{V}_w \mathbf{V}_w^\top \mathbf{K}, \quad \mathbf{K}_{\text{proj}}^\top = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{X}^\top, \tag{5}$$

where the columns of the matrix $\mathbf{X}$ form an orthonormal basis of $\texttt{Col}(\mathbf{K}_{\texttt{proj}})$, with $\texttt{Col}(\cdot)$ denoting the column space of a matrix. We define the input space alignment orthogonal matrix as $\mathbf{S}_t = \mathbf{X}$. After transformation by $\mathbf{S}_t^\top$, the features of both historical and current input in streaming data are confined within the effective operational subspace of the parameter matrix $\mathbf{W}$, satisfying:

$$\texttt{Col}(\mathbf{S}_t^\top \mathbf{K}_{t-1}) \subseteq \texttt{Row}(\mathbf{W}), \texttt{Col}(\mathbf{S}_t^\top \mathbf{K}_t) \subseteq \texttt{Row}(\mathbf{W}) \quad (6)$$

where $\texttt{Row}(\cdot)$ denotes the row space of a matrix. This process unifies constraints on the features of both new and historical data, maintaining consistency of the input feature space.

Orthogonal Matrix R. For the orthogonal matrix $\mathbf{R}_t$ used for output space correction, its objective is to ensure that the transformed weight matrix $\mathbf{W}_t^* = \mathbf{R}_t \mathbf{W} \mathbf{S}_t^\top$ accurately matches the output requirements of both new and old feature mappings, thereby achieving correction in the output space. Specifically, we first define the intermediate output matrices $\mathbf{A} = \mathbf{W}\mathbf{S}_t^\top \mathbf{K}_{t-1}$ and $\mathbf{B} = \mathbf{W}\mathbf{S}_t^\top \mathbf{K}_t$. Due to the effect of the input space alignment matrix $\mathbf{S}_t$, both $\mathbf{A}$ and $\mathbf{B}$ lie within the column space of the weight matrix $\mathbf{W}$. According to the optimization objective in Eq.(4), the output space correction matrix $\mathbf{R}_t$ needs to satisfy $\mathbf{R}_t[\mathbf{A} \mid \mathbf{B}] \approx [\mathbf{V}_{t-1} \mid \mathbf{V}_t]$. Thus, the output space correction reduces to a Procrustes problem (Schönemann 1966), minimizing the Frobenius norm to align the orthogonally transformed output with the target output:

$$\min_{\mathbf{R}_t} \|\mathbf{R}_t[\mathbf{A} \mid \mathbf{B}] - [\mathbf{V}_{t-1} \mid \mathbf{V}_t]\|_F^2, \text{s.t.} \quad \mathbf{R}_t^\top \mathbf{R}_t = \mathbf{I} \quad (7)$$

Since the optimization is a minimization problem under orthogonality constraints, it is not straightforward to solve directly. To address this Procrustes problem, we construct the matrix $\mathbf{M} = \mathbf{V}_{t-1}\mathbf{A}^\top + \mathbf{V}_t \mathbf{B}^\top$, which captures the covariance between intermediate outputs and target outputs. By rewriting the objective in terms of the trace, the problem above is equivalent to maximizing $\text{tr}(\mathbf{R}_t \mathbf{M}^\top)$ over the space of orthogonal matrices. Performing SVD on $\mathbf{M}$, i.e., $\mathbf{M} = \mathbf{U}_M \mathbf{\Sigma}_M \mathbf{V}_M^\top$, allows us to construct the orthogonal matrix $\mathbf{R}_t$ that maximizes the objective while satisfying the orthogonality constraint. The detailed proof is provided in Appendix D. According to the closed-form solution to the Procrustes problem, the optimal solution is as follows:

$$\mathbf{R}_t = \mathbf{U}_M \mathbf{V}_M^\top \quad (8)$$

which maximizes the covariance between intermediate and target outputs and thus achieves the theoretically optimal alignment of the output space.

Model Parameter Continuous Update Process

In the context of continual learning for streaming data, when the backbone model f_θ needs to adapt to the data distribution within the current time t, the first step is to obtain the corresponding set of key-value pairs $\{\mathbf{K}_t, \mathbf{V}_t\}$. Since using the entire dataset incurs significant computational and storage overhead, we adopt a random sampling strategy under the assumption of Lipschitz continuity. Specifically, a subset $\mathcal{D}_{sub}^t = \{\mathbf{X}_t, \mathbf{Y}_t\}$ is sampled with ratio α from $\mathcal{D}^t$ to reduce the computational burden. The statistical bias (mean and variance) of the sampled subset, compared to the full dataset, is defined as $O(1/\sqrt{\alpha N})$ (Boucheron, Lugosi, and Massart 2013), where N is the total number of samples in $\mathcal{D}^t$. Subsequently, a forward propagation is performed for $\mathbf{x}_t \in \mathcal{D}_{sub}^t$ to generate predictions $\hat{\mathbf{y}}_t = f_\theta(\mathbf{x}_t)$, and model parameters are updated via gradient descent as follows:

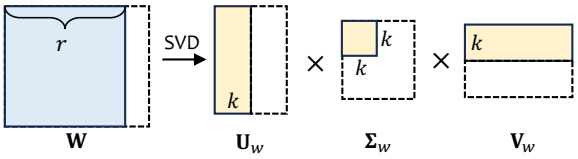

Figure 3: Illustration of truncated SVD.

$$\theta \leftarrow \theta - \eta \nabla_\theta \left(\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|_2^2 \right). \quad (9)$$

After fine-tuning, the same subset $\mathcal{D}_{sub}^t$ is utilized for model inference to obtain the updated key-value pairs $\{\mathbf{K}_t, \mathbf{V}_t\}$, which assist in subsequent model updates. It is important to note that if no rank constraint is imposed on the parameter matrix $\mathbf{W}$, the dimensions of the associated orthogonal matrices $\mathbf{R}_t$ and $\mathbf{S}_t$ grow linearly with the rank of $\mathbf{W}$. For example, if the rank of $\mathbf{W}$ is r, storing the orthogonal matrices requires $O(r^2)$ parameters, leading to excessive memory consumption. To address this issue, we leverage the Eckart–Young–Mirsky theorem (Golub, Hoffman, and Stewart 1987) on low-rank matrix approximation and perform truncated SVD on $\mathbf{W}$ to obtain its best low-rank approximation in the least-squares sense. Specifically, as shown in Figure 3, when constructing the input alignment matrix $\mathbf{S}_t$ and output correction matrix $\mathbf{R}_t$, we rely on the truncated SVD of $\mathbf{W}$ at rank k, retaining only the top k principal components or singular vectors, reducing spatial complexity while maintaining a approximation error.

Once $\mathbf{R}_t$ and $\mathbf{S}_t$ are obtained, we can update the model parameters according to Eq.(3), which can then be applied to downstream tasks. Furthermore, as new streaming data continues to arrive, there is no need to retrain the entire model network from scratch; instead, efficient continual model updates can be achieved simply by computing the corresponding orthogonal matrices $\mathbf{R}$ and $\mathbf{S}$.

Experiment

Experimental Setup

We begin with a brief overview of the baselines, datasets, evaluation metrics, and task settings. For more detailed experimental configurations, please refer to Appendix A.

Base Models and Baseline Methods. Our experiments are conducted on three base models: two transformer-based time Series foundation models (Timer (Liu et al. 2024) and Moirai (Woo et al. 2024)) and one conventional transformer-based time series model (PatchTST (Nie et al. 2022)). We compare our proposed `ORTCL` against two categories of continuous learning baselines: 1) Model fine-tuning methods, including **TrafficStream** (Chen, Wang, and Xie 2021), **URCL** (Miao et al. 2024), **Storm** (Luo et al. 2025), and

Method	ECL								Weather							
	$\mathcal{D}^1$		$\mathcal{D}^2$		$\mathcal{D}^3$		$\mathcal{D}^4$		$\mathcal{D}^1$		$\mathcal{D}^2$		$\mathcal{D}^3$		$\mathcal{D}^4$	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Timer	0.2229	0.3131	0.2258	0.3391	0.2611	0.3430	0.3650	0.3868	0.8067	0.5463	0.6406	0.5529	0.4792	0.3694	0.6549	0.4337
TrafficStream	0.1414	0.2201	0.1326	0.2382	0.2028	0.2784	0.2714	0.3096	0.7148	0.4817	0.4523	0.4395	0.4452	0.3259	0.6002	0.4461
URCL	0.1399	0.2282	0.1399	0.2466	0.2060	0.2703	0.2747	0.3053	0.6928	0.4631	0.4455	0.4375	0.4228	0.3274	0.6176	0.4321
Storm	0.1398	0.2169	0.1351	0.2508	0.1967	0.2734	0.2711	0.2983	0.6912	0.4783	0.4383	0.4160	0.4295	0.3221	0.5916	0.4112
SD-LoRA	0.1421	0.2278	0.1372	0.2470	0.1961	0.2667	0.2661	0.3167	0.7073	0.4821	0.4591	0.4234	0.4296	0.3329	0.6150	0.4397
ROME	0.1527	0.2395	0.1395	0.2713	0.2253	0.2933	0.2901	0.3438	0.7644	0.5050	0.5089	0.4777	0.4604	0.3785	0.6675	0.4630
MEMIT	0.1524	0.2432	0.1453	0.2495	0.2051	0.2905	0.2740	0.3175	0.7459	0.4939	0.4894	0.4434	0.4386	0.3373	0.6415	0.4558
AlphaEdit	0.1423	0.2352	0.1382	0.2602	0.2027	0.2941	0.2794	0.3073	0.6866	0.4536	0.4604	0.4413	0.4420	0.3555	0.6503	0.4582
ORTCL	**0.1332**	**0.2139**	**0.1256**	**0.2334**	**0.1930**	**0.2613**	**0.2602**	**0.2880**	**0.6786**	**0.4456**	**0.4255**	**0.4070**	**0.4070**	**0.3191**	**0.5880**	**0.4087**

Method	Traffic								ETT (Avg)							
	$\mathcal{D}^1$		$\mathcal{D}^2$		$\mathcal{D}^3$		$\mathcal{D}^4$		$\mathcal{D}^1$		$\mathcal{D}^2$		$\mathcal{D}^3$		$\mathcal{D}^4$	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Timer	0.3150	0.3070	0.3047	0.3019	0.4259	0.2885	0.3854	0.3427	0.8452	0.6490	1.0428	0.6340	1.4313	0.7957	0.4709	0.5098
TrafficStream	0.2852	0.2417	0.2692	0.2418	0.3825	0.2393	0.3622	0.3122	0.6363	0.5508	0.8458	0.5890	1.2767	0.7494	0.3478	0.4323
URCL	0.2792	0.2337	0.2609	0.2419	0.3969	0.2447	0.3650	0.3156	0.6370	0.5732	0.8531	0.5887	1.2179	0.7777	0.3461	0.4383
Storm	0.2624	0.2276	0.2525	0.2348	0.3745	0.2303	0.3494	0.3053	0.6099	0.5788	0.8258	0.5520	1.1895	0.7125	0.3379	0.4206
SD-LoRA	0.2772	0.2390	0.2557	0.2461	0.3797	0.2317	0.3866	0.3152	0.6065	0.5613	0.8212	0.5625	1.2437	0.7678	0.3455	0.4209
ROME	0.2890	0.2623	0.2745	0.2736	0.4032	0.2701	0.4054	0.3419	0.6571	0.6398	0.9655	0.6148	1.3579	0.8095	0.3893	0.4553
MEMIT	0.2722	0.2444	0.2757	0.2505	0.4131	0.2400	0.3955	0.3374	0.6429	0.5932	0.8753	0.5959	1.3258	0.8125	0.3591	0.4415
AlphaEdit	0.2623	0.2211	0.2575	0.2400	0.4074	0.2385	0.3754	0.3342	0.5978	0.5659	0.8132	0.5575	1.3306	0.8177	0.3523	0.4583
ORTCL	**0.2517**	**0.2191**	**0.2432**	**0.2292**	**0.3605**	**0.2220**	**0.3445**	**0.2873**	**0.5841**	**0.5349**	**0.8098**	**0.5425**	**1.1888**	**0.7096**	**0.3188**	**0.4124**

Table 1: The experimental results on the ECL, Weather, Traffic, and ETT datasets in a single-domain streaming data setting, **Bold** and underline denote the best and second-best results. The Timer is the zero-shot setting.

SD-LoRA (Wu et al. 2025). 2) Model editing methods, including **ROME** (Meng et al. 2022a), **MEMIT** (Meng et al. 2022b), and **AlphaEdit** (Fang et al. 2024).

Datasets and Evaluation Metrics. During the pre-training phase, we follow previous work (Liu et al. 2024) by pre-training the base models on the UTSD-12G dataset. We then comprehensively evaluate the proposed method using four benchmark datasets: ETT (including ETTh1, ETTh2, ETTm1, ETTm2) (Zhou et al. 2021), ECL (Asuncion, Newman et al. 2007), Traffic, and Weather. To prevent data leakage, all downstream datasets are excluded from the pre-training stage. Model performance is evaluated using two metrics: 1) Mean Squared Error (MSE) and 2) Mean Absolute Error (MAE). Detailed implementation specifics and model configurations can be found in Appendix B.

Streaming Data Settings. To evaluate the effectiveness of the proposed method, we design two experimental setups: 1) **Single-Domain Streaming Data.** To simulate the gradual evolution of data distributions over time, we create a sequence of streaming data derived from a single dataset. For instance, using the ETT dataset, we divide the entire timeline into four consecutive, non-overlapping segments, denoted as $\{\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3, \mathcal{D}^4\}$, each representing an equal time window. Within each segment, the data is further partitioned into training, validation, and test subsets. At each time step t, we randomly sample 1% of the training sequences (i.e., $\alpha = 1\%$) from segment $\mathcal{D}^t$ to construct the knowledge key-value pairs. The model is then evaluated on the corresponding test set for that segment. 2) **Cross-Domain Streaming Data.** To evaluate adaptability under domain shifts, we use four benchmark datasets (ETT, ECL, Traffic, Weather) as distinct domains, presenting data sequentially to simulate a cross-domain streaming environment.

Method	ETTh1		ECL		Weather		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Sota-B	0.3757	0.4013	0.1546	0.2479	0.1695	0.2295	0.4234	0.2881
Sota-B-f_α	0.3812	0.4082	0.1684	0.2578	0.1783	0.2412	/	/
↑	1.46%	1.72%	8.93%	3.99%	5.20%	5.11%	/	/
FFT	0.3659	0.3955	0.1400	0.2384	0.1659	0.2144	0.3812	0.2690
FFT-f_α	0.3787	0.4015	0.1628	0.2582	0.1796	0.2961	/	/
↑	3.50%	1.51%	16.25%	8.35%	8.27%	38.10%	/	/
ORTCL	0.3689	0.3960	0.1394	0.2380	0.1678	0.2164	0.3852	0.2718
ORTCL-f_α	0.3729	0.4001	0.1454	0.2443	0.1696	0.2169	/	/
↑	**1.08%**	**1.03%**	**4.25%**	**2.63%**	**1.07%**	**0.22%**	/	/

Table 2: The results on the cross-domain streaming data setting. The Sota-B is the best result in all baselines.

Overall Performance

We continually update the Timer model using ORTCL and several baselines (additional Moirai and PatchTST experiments in Appendix C). Tables 1 and 2 show results under two streaming-data settings. The main findings are as follows:

1) ORTCL outperforms baselines on most metrics. In the single-domain setting, for the first segment $\mathcal{D}^1$, ORTCL achieves average reductions of 2.4% (MSE) and 1.7% (MAE) versus the best baseline across four datasets. By the fourth segment $\mathcal{D}^4$, these improve to 2.8% and 2.4%. With no historical retention, we also apply Full Fine-Tuning (FFT) separately on each segment. Figure 4 shows ORTCL

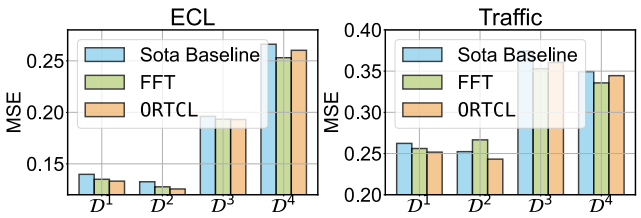

Figure 4: Comparison results with the FFT.

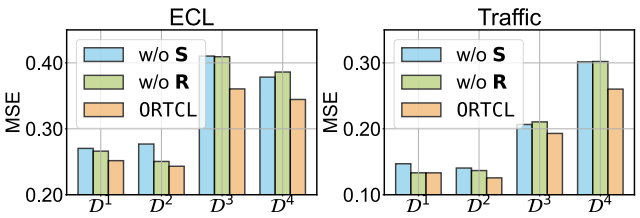

Figure 5: The ablation study on ECL and Traffic datasets.

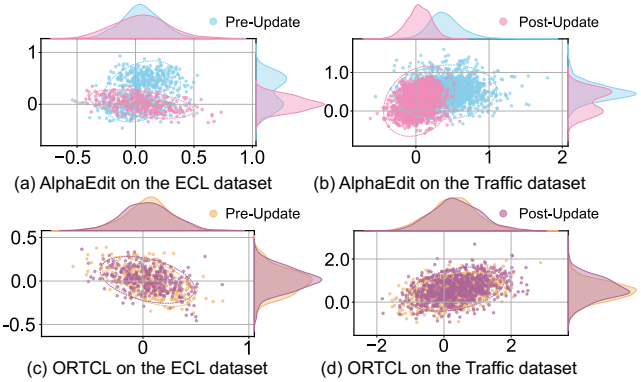

Figure 6: The results of the knowledge retention study.

performs better than FFT in the first three segments, while other baselines perform worse than FFT. This validates the effectiveness of orthogonal rotation transformation in input alignment and output calibration.

2) In the cross-domain streaming scenario, we sequentially input four datasets (ETTh1, ECL, Weather, Traffic) into FFT and ORTCL to update the model parameters. The resulting models are denoted as FFT-f_α and ORTCL-f_α, respectively, then evaluate both on ETTh1, ECL, and Weather. ORTCL-f_α attains average improvements of about 2.14% (MSE) and 1.30% (MAE) on those datasets, versus 9.34% and 15.99% for FFT-f_α. This indicates traditional fine-tuning is prone to catastrophic forgetting in transfer.

Ablation Study

To further analyze the impact of orthogonal rotation transformations on model performance, we designed two ablation variants: one that removes the input-side rotation matrix $\mathbf{S}$, denoted as w/o $\mathbf{S}$, and another that removes the output-side rotation matrix $\mathbf{R}$, denoted as w/o $\mathbf{R}$. As shown in Figure 5, we conducted experiments on the ECL and Traffic datasets, and the results indicate: 1) The performance of w/o $\mathbf{R}$ significantly declines across segments $\mathcal{D}^1$ to $\mathcal{D}^4$. Compared to the ORTCL model, the average MAE for ECL increases by 11.30%, and for Traffic, it increases by 11.22%. This highlights the critical role of rotation transformations in effectively mapping and extracting features from the input data. 2) The performance degradation experienced by w/o $\mathbf{R}$ in $\mathcal{D}^1$ and $\mathcal{D}^2$ is limited, indicating that the rotated input features still provide effective data characteristics, allowing the model's output to remain aligned with the frozen deep components. However, in the later segments ($\mathcal{D}^3$ and $\mathcal{D}^4$), the performance noticeably deteriorates, suggesting that without the correct transformation of the output feature space, the output features become difficult to separate effectively, thereby limiting the model's ability to represent and adapt to changes in the time series.

Knowledge Retention Study

As previously mentioned, existing model editing methods often lead to a gradual deviation of model parameters from the original feature space when handling streaming data, due to the accumulation of parameter increments Δ. This deviation can cause shifts in the distribution of hidden layer representations in historical data, ultimately resulting in catastrophic forgetting. To verify whether ORTCL can effectively mitigate this representation drift, we conduct experimental analyses on the ECL and Traffic datasets. The specific steps of the experiment are as follows:

First, we input the data from the first segment D^1 into the model to extract and visualize the corresponding hidden layer representations, labeled as "Pre-Update". Next, we sequentially perform streaming updates on the model using data from $\mathcal{D}^2 \rightarrow \mathcal{D}^3 \rightarrow \mathcal{D}^4$. After completing the updates, we input the data from $\mathcal{D}^1$ into the updated model to re-extract and visualize the hidden layer representations, labeled as "Post-Update". The visualization results are obtained by using t-SNE in Figure 6. The experimental results indicate that after continuously updating the model parameters with data from $\mathcal{D}^2$, $\mathcal{D}^3$, and $\mathcal{D}^4$, the hidden layer representations of ORTCL maintain a high degree of consistency when processing data from $\mathcal{D}^1$. This suggests that the continual learning process under ORTCL does not cause significant shifts in the distribution of hidden layer representations. In contrast, under the same experimental conditions, AlphaEdit (Fang et al. 2024) leads to a noticeable shift in the distribution of hidden layer representations. This comparative result underscores the importance of our orthogonal rotation transformation in maintaining model stability and preserving historical knowledge.

Feature Visualization Analysis

To further validate that the proposed ORTCL achieves prediction performance comparable to the FFT methods specifically designed for each dataset, we conduct experiments on the ECL and Traffic datasets. By comparing the prediction results of ORTCL and FFT, along with their corresponding attention weight matrices, we find that the prediction curves of ORTCL closely align with those of FFT and the ground

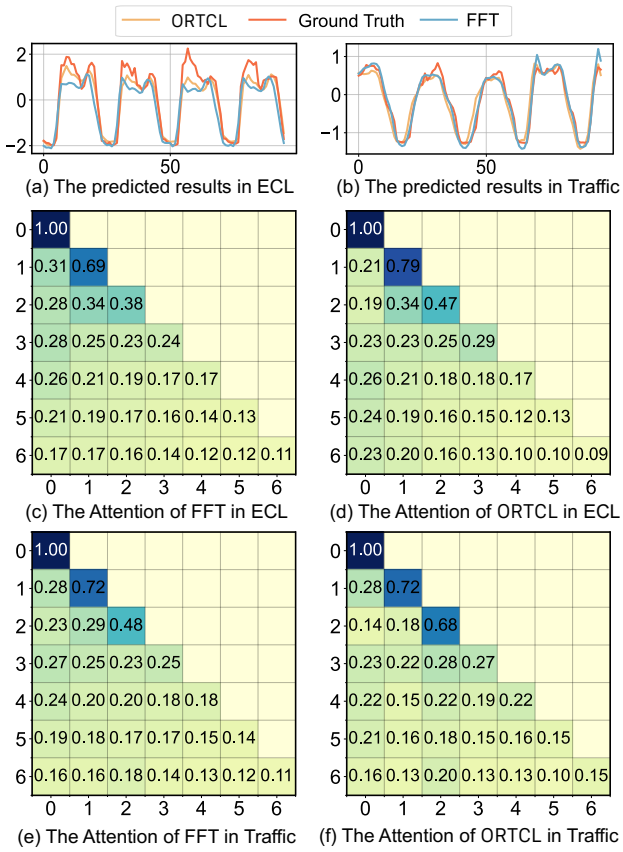

Figure 7: The attention weight visualization analysis.

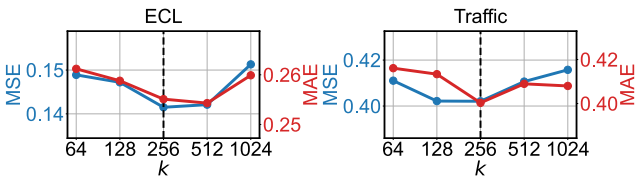

Figure 8: Rank k hyperparameter sensitivity.

truth, as shown in Figure 7. This indicates that ORTCL maintains a prediction accuracy similar to that of FFT. Additionally, the distribution of the attention weight matrices for both methods shows a high degree of similarity. This suggests that ORTCL effectively focuses on updating local parameters within the model, thereby avoiding global parameter overfitting during fine-tuning and mitigating the forgetting of historical knowledge.

Hypeparameter Study

To investigate the impact of the hyperparameter k during the SVD decomposition of the model parameter $\mathbf{W}$, we set k to $\{64, 128, 256, 512, 1024\}$ and tested the model's average performance separately on the datasets $\{\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3, \mathcal{D}^4\}$ constructed from both the ECL and Traffic datasets. The results shown in Figure 8 indicate that: (1) As k varies, metrics MSE and MAE exhibit significant fluctuations, demonstrating that the model is robust to a range of k values. (2) As the value of k increases, the model's performance does not improve accordingly and may instead introduce redundant parameters and increase computational overhead.

Related Work

Continual Learning for Streaming Data. Continual learning involves progressively learning a series of tasks or data by knowledge transfer (Wang et al. 2024). In streaming data scenarios, the data distribution changes dynamically over time, requiring models to continuously adapt to new knowledge without the need to store all historical knowledge, while also avoiding knowledge forgetting (Qiao et al. 2024). Currently, methods for continual learning primarily focus on two strategies: (1) Regularization-based methods (Aljundi et al. 2019) impose additional constraints on important parameters to protect old knowledge, effectively reducing the interference of new data learning on previously acquired knowledge (Chen, Wang, and Xie 2021; Ahn et al. 2019; Zhao et al. 2024). (2) Memory replay-based methods (Kirkpatrick et al. 2017) maintain a capacity-limited episodic memory to store historical data or generate samples for experience replay during new data training (Miao et al. 2024; Luo et al. 2025; Kiyasseh, Zhu, and Clifton 2021; Zhao and Shen 2025; Liu et al. 2025b). However, these methods often face challenges related to computational and storage overhead when applied to large foundational models.

Continual Learning with Foundational Models. In the era of foundational models, continual learning is undergoing a transition from a fine-tuning paradigm to an editing paradigm (Yao et al. 2024). Early adapter methods achieved incremental knowledge acquisition by introducing low-rank matrices or learnable prompts, with typical representatives including LoRA (Hu et al. 2022) and its extensions (Wan et al. 2025; Wu et al. 2025; Han et al. 2025; Hartvigsen et al. 2023). Due to the computational overhead associated with fine-tuning the parameters of foundational models (Yang et al. 2025), existing advanced works primarily focus on parameter modification strategies, such as ROME (Meng et al. 2022a), MEMIT (Meng et al. 2022b), and AlphaEdit (Fang et al. 2024). These methods identify key parameters in the model that need modification and edit them to achieve global effects. However, in streaming data scenarios, parameter updates usually optimize for the current distribution without constraints on past data, causing gradual drift from the original feature space and leading to knowledge forgetting.

Conclusion

In this work, we focus on the streaming data setting for time series, aiming to prevent forgetting and enable adaptation of the time series foundational model in continuous streams. We propose the ORTCL, which aligns new and old features at the shallow input-output layer of the model using low-rank orthogonal rotation transformations. We conduct extensive experiments demonstrating that ORTCL achieves better performance in streaming forecasting tasks. Furthermore, we hope future work will validate its long-term stability and scalability in resource-constrained online deployment.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.62402105 and Grant No.62272098, the Major Project of Fundamental Research on Frontier Leading Technology of Jiangsu Province under Grant BK20222006, the Natural Science Foundation of Jiangsu Province under Grant BK20230815, and the Central University Basic Research Fund of China under Grant No. 2242025RCB0040. Additional support was provided by the Big Data Computing Center of Southeast University and the Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China.

References

Ahn, H.; Cha, S.; Lee, D.; and Moon, T. 2019. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32.

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32.

Asuncion, A.; Newman, D.; et al. 2007. UCI machine learning repository.

Boucheron, S.; Lugosi, G.; and Massart, P. 2013. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. ISBN 9780199535255.

Chen, X.; Wang, J.; and Xie, K. 2021. TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. *arXiv preprint arXiv:2106.06273*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3): 220–235.

Fang, J.; Jiang, H.; Wang, K.; Ma, Y.; Jie, S.; Wang, X.; He, X.; and Chua, T.-S. 2024. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*.

Golub, G. H.; Hoffman, A.; and Stewart, G. W. 1987. A generalization of the Eckart-Young-Mirsky matrix approximation theorem. *Linear Algebra and its applications*, 88: 317–327.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

Grech, D. 2025. *The Dynamics and Role of Representational Drift*. Ph.D. thesis.

Han, X.; Wang, Y.; Feng, J.; Hu, Q.; Deng, C.; et al. 2025. LOIRE: LifelOng learning on Incremental data via pre-trained language model gRowth Efficiently. In *The Thirteenth International Conference on Learning Representations*.

Hartvigsen, T.; Sankaranarayanan, S.; Palangi, H.; Kim, Y.; and Ghassemi, M. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36: 47934–47959.

Hase, P.; Bansal, M.; Kim, B.; and Ghandeharioun, A. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36: 17643–17668.

Hu, C.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2024. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. *arXiv preprint arXiv:2402.10987*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Kiyasseh, D.; Zhu, T.; and Clifton, D. 2021. A clinical deep learning framework for continually learning from cardiac signals across diseases, time, modalities, and institutions. *Nature Communications*, 12(1): 4221.

Lezama, J.; Qiu, Q.; Musé, P.; and Sapiro, G. 2018. Ole: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8109–8118.

Liang, Y.; Wen, H.; Nie, Y.; Jiang, Y.; Jin, M.; Song, D.; Pan, S.; and Wen, Q. 2024. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 6555–6565.

Lin, L.; Xia, K.; Zheng, A.; Hu, S.; and Wang, S. 2024. Hierarchical Spatio-Temporal Graph Learning Based on Metapath Aggregation for Emergency Supply Forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 1410–1419.

Liu, C.; Hettige, K. H.; Xu, Q.; Long, C.; Xiang, S.; Cong, G.; Li, Z.; and Zhao, R. 2025a. ST-LLM+: Graph Enhanced Spatio-Temporal Large Language Models for Traffic Prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, C.; Miao, H.; Xu, Q.; Zhou, S.; Long, C.; Zhao, Y.; Li, Z.; and Zhao, R. 2025b. Efficient Multivariate Time Series Forecasting via Calibrated Language Models with Privileged Knowledge Distillation. In *41th IEEE International Conference on Data Engineering*.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025c. TimeCMA: Towards LLM-Empowered Multivariate Time Series Forecasting via Cross-Modality Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18780–18788.

Liu, C.; Zhou, S.; Xu, Q.; Miao, H.; Long, C.; Li, Z.; and Zhao, R. 2025d. Towards Cross-Modality Modeling for Time Series Analytics: A Survey in the LLM Era. In *IJCAI*.

Liu, X.; and Chang, X. 2025. LoRA Subtraction for Drift-Resistant Space in Exemplar-Free Continual Learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 15308–15318.

Liu, Y.; Zhang, H.; Li, C.; Huang, X.; Wang, J.; and Long, M. 2024. Timer: Generative pre-trained transformers are large time series models. *arXiv preprint arXiv:2402.02368*.

Liu, Z.; Xu, Y.; Xu, Y.; Qian, Q.; Li, H.; Ji, X.; Chan, A.; and Jin, R. 2022. Improved fine-tuning by better leveraging pre-training data. *Advances in Neural Information Processing Systems*, 35: 32568–32581.

Luo, T.; Fang, Z.; Duan, K.; Chen, L.; Feng, P.; and Lu, M. 2025. Towards Online Spatio-Temporal Prediction: A Knowledge Distillation Driven Continual Learning Approach. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, 2642–2655. IEEE Computer Society.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35: 17359–17372.

Meng, K.; Sharma, A. S.; Andonian, A.; Belinkov, Y.; and Bau, D. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Miao, H.; Zhao, Y.; Guo, C.; Yang, B.; Zheng, K.; Huang, F.; Xie, J.; and Jensen, C. S. 2024. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 1050–1062. IEEE.

Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C. D.; and Finn, C. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, 15817–15831. PMLR.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Qiao, Z.; Pham, Q.; Cao, Z.; Le, H. H.; Suganthan, P. N.; Jiang, X.; and Ramasamy, S. 2024. Class-incremental learning for time series: Benchmark and evaluation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5613–5624.

Schönemann, P. H. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1): 1–10.

Wan, Z.; Du, W.; Li, L.; Pan, M.; and Qin, X. 2025. Budget-Adaptive Adapter Tuning in Orthogonal Subspaces for Continual Learning in LLMs. *arXiv preprint arXiv:2505.22358*.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8): 5362–5383.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified training of universal time series forecasting transformers.

Wu, Y.; Piao, H.; Huang, L.-K.; Wang, R.; Li, W.; Pfister, H.; Meng, D.; Ma, K.; and Wei, Y. 2025. Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. *arXiv preprint arXiv:2501.13198*.

Xia, K.; Lin, L.; Wang, S.; Wang, H.; Zhang, D.; and He, T. 2023. A predict-then-optimize couriers allocation framework for emergency last-mile logistics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5237–5248.

Xia, K.; Lin, L.; Wang, S.; Zhang, Q.; Wang, S.; and He, T. 2025a. ProST: Prompt future snapshot on dynamic graphs for spatio-temporal prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 1645–1656.

Xia, K.; Lin, L.; Zhang, X.; Wang, H.; Wang, S.; and He, T. 2025b. A Transferable Spatio-temporal Learning Framework for Cross-city Logistics Demand Prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 5071–5082.

Yang, L.; Luo, Z.; Zhang, S.; Teng, F.; and Li, T. 2024. Continual learning for smart City: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Yang, Y.; Zhou, J.; Ding, X.; Huai, T.; Liu, S.; Chen, Q.; Xie, Y.; and He, L. 2025. Recent advances of foundation language models-based continual learning: A survey. *ACM Computing Surveys*, 57(5): 1–38.

Yao, Y.; Wang, P.; Tian, B.; Cheng, S.; Li, Z.; Deng, S.; Chen, H.; and Zhang, N. 2024. Editing Large Language Models: Problems, Methods, and Opportunities. The 2023 Conference on Empirical Methods in Natural Language Processing.

Zhang, N.; Yao, Y.; Tian, B.; Wang, P.; Deng, S.; Wang, M.; Xi, Z.; Mao, S.; Zhang, J.; Ni, Y.; et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Zhao, L.; and Shen, Y. 2025. Proactive model adaptation against concept drift for online time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2020–2031.

Zhao, X.; Wang, H.; Huang, W.; and Lin, W. 2024. A statistical theory of regularization-based continual learning. *arXiv preprint arXiv:2406.06213*.

Zhou, D.-W.; Sun, H.-L.; Ning, J.; Ye, H.-J.; and Zhan, D.-C. 2024. Continual learning with pre-trained models: a survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8363–8371.

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