STRUCTURAL KNOWLEDGE INFORMED CONTINUAL LEARNING FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies in multivariate time series (MTS) forecasting reveal that explicitly modeling the hidden dependencies among different time series can yield promising forecasting performance and reliable explanations. However, modeling variable dependencies remains underexplored when MTS is continuously accumulated under different regimes (stages). Due to the potential distribution and dependency disparities, the underlying model may encounter the catastrophic forgetting problem, *i.e.*, it is challenging to memorize and infer different types of variable dependencies across different regimes while maintaining forecasting performance. To address this issue, we propose a novel Structural Knowledge Informed Continual Learning (SKI-CL) framework to perform MTS forecasting within a continual learning paradigm, which leverages structural knowledge to steer the forecasting model toward identifying and adapting to different regimes, and selects representative MTS samples from each regime for memory replay. Specifically, we develop a forecasting model based on graph structure learning, where a consistency regularization scheme is imposed between the learned variable dependencies and the structural knowledge (*e.g.*, physical constraints, domain knowledge, feature similarity, which provides regime characterization) while optimizing the forecasting objective over the MTS data. As such, MTS representations learned in each regime are associated with distinct structural knowledge, which helps the model memorize a variety of conceivable scenarios and results in accurate forecasts in the continual learning context. Meanwhile, we develop a representation-matching memory replay scheme that maximizes the temporal coverage of MTS data to efficiently preserve the underlying temporal dynamics and dependency structures of each regime. Thorough empirical studies on synthetic and real-world benchmarks validate SKI-CL's efficacy and advantages over the state-of-the-art for continual MTS forecasting tasks. SKI-CL can also infer faithful dependency structures that closely align to structural knowledge in the test stage.

1 INTRODUCTION

Multivariate time series (MTS) forecasting aims to predict future samples from multiple time series based on their historical values and has shown its importance in various applications, *e.g.*, healthcare, traffic control, energy management, and finance Jin et al. (2018); Gonzalez-Vidal et al. (2019); Guo et al. (2019); Zhang et al. (2017). Accurate MTS forecasting not only relies on capturing temporal dynamics of the historical time series data van den Oord et al. (2016); Bai et al. (2018); Borovykh et al. (2017); Lai et al. (2018); Wu et al. (2021); Zhou et al. (2022); Nie et al. (2022), but also relies on modeling dependency structures among different variables Lai et al. (2018); Shih et al. (2019); Wu et al. (2020); Shang & Chen (2021); Liu et al. (2023).

Despite the promising forecasting accuracy and explainable structure characterization, capability of these MTS forecasters is limited to one regime (stage) of MTS data characterized by a set of similar dependency patterns. In real-world applications, different regimes of MTS data are often continuously collected under different operational logic of the target system. In this learning scenario, where regimes arrive sequentially, the major challenge in MTS forecasting is to keep track of the latest regime while maintaining forecasting capability on the past ones. For example, in the context of solar energy, a model needs to maintain accurate and robust forecasts across regimes spanned by seasons or locations with different sunlight patterns (*e.g.*, summer and winter, northern and southern areas) to ensure reliable energy storage and supply. While an intuitive and efficient solution is to retrain the forecaster periodically over the newly collected regime, this will inevitably lead to the catastrophic forgetting issue, *i.e.*, the learned dependency structures cannot be maintained over existing regimes and the forecasting performance will deteriorate accordingly, as shown in Figure 1. On the other hand, joint training may be infeasible due to the need to store all historical data (different regimes) and the computational complexity of handling an ever-increasing number of diverse scenarios.

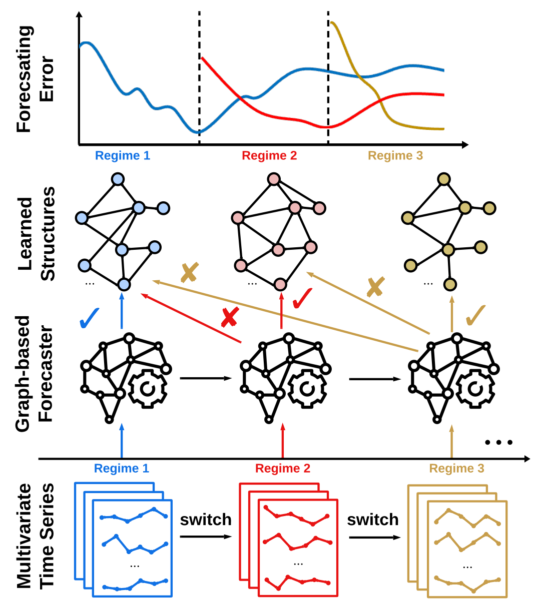


Figure 1: An illustration depicting the catastrophic forgetting of learned dependency structures (*i.e.*, the interactions of variables) in multivariate time series forecasting across regimes. Each regime is characterized by a distinct operational logic of the system.

We resort to memory replay to tackle the aforementioned challenge, where the key idea is to replay a subset of samples from previous regimes while the model is learning the new one Rolnick et al. (2019); Zhou & Cao (2021). Motivated by the principle of maximum entropy Guiasu & Shenitzer (1985); Du et al. (2021), we aim to maximize the coverage of each regime by selecting MTS samples that represent the most diverse modes. To deal with the multivariate nature with an emphasis on dependencies modeling, we seek to enhance this strategy by further incorporating external structural knowledge into the replay process. Structural knowledge provides universal and task-independent insights that characterize the dependency patterns from a specific regime. It helps the model better identify and adapt to regime-specific patterns, letting the model memorize varieties of conceivable scenarios and thus mitigating the catastrophic forgetting issue. The structural knowledge can be available in different formats, *e.g.*, physical constraints such as traffic networks, power grids, and sensor networks Li et al. (2017); Luo et al. (2021); Khodayar & Wang (2018); Yan et al. (2018), or dependencies (correlations) that are inferred or derived from raw data by leveraging either domain knowledge or traditional statistical methods Chen et al. (2022); Duan et al. (2022); Lin et al. (2021); Cao et al. (2020); Shang & Chen (2021).

In this paper, we present a novel Structural Knowledge Informed Continual Learning (SKI-CL) framework that sequentially learns and preserves meaningful dependency structures for MTS forecasting under different regimes. As shown in Figure 2, we first exploit structural knowledge to characterize the variable dependencies within each regime. In our forecasting model, we build a graph structure learning module that encodes the temporal patterns and dynamically infers dependency structures (in the form of graphs) based on different MTS input windows to cope with the dependencies variations within each regime. We jointly optimize the forecasting objective and a consistency regularizer that enforces the inferred structure toward the existing structural knowledge. MTS data from each regime is associated with the distinct structure knowledge via the model, which steers the model to identify and adapt across regimes in continual learning. Note that different scenarios regarding the description of dependencies and the availability of structural knowledge is considered, *i.e.*, the discrete/continuous edge description, and fully/partially observed structural knowledge. We present a novel representation-matching memory replay scheme to select samples that maximize the temporal coverage of MTS data, to efficiently preserve the underlying temporal dynamics and dependency structure of each regime (Figure 2(middle)). We first partition the MTS representations into the most diverse distribution modes along the temporal dimension. Subsequently, we deal with each mode individually for sample selection. Given the memory budget, we select a subset of MTS samples whose representations are the most similar to that of the entire mode, measured by CORAL Sun & Saenko (2016). Thereby the coverage of each regime is efficiently preserved by the union of representative MTS samples from diverse modes. By jointly learning from the current regime and the constructed memory of structural knowledge and representative MTS samples, the obtained model is able to maintain accurate forecasts and infer the learned structures from the existing regimes.

In summary, our work makes the following four contributions. (1) We present a novel Structural Knowledge Informed Continual Learning (SKI-CL) framework to perform MTS forecasting and infer dependency structures in the continual learning setting. (2) We develop a graph-based forecaster that contains a structure learning module to capture temporal dependencies and dynamically infer dependency structures, and employ a consistency regularization scheme that exploits structural knowledge to facilitate continual forecasting. (3) We propose a novel representation-matching memory replay scheme to maximize the temporal coverage of MTS data and preserve the underlying temporal dynamics as well as dependency structures within each regime. (4) Thorough experiments on one synthetic dataset and three benchmark datasets demonstrate the superiority of SKI-CL over the state-of-the-art in continual MTS forecasting and dependency structure inference.

2 Related Work

2.1 Modeling Dependencies in Multivariate Time Series Forecasting

In recent years, modeling variable dependencies of MTS has received increasing attention for forecasting tasks. Early methods apply linear or convolution transformations to capture variable dependencies in an implicit recurrent process Lai et al. (2018); Graves (2013); Shih et al. (2019), which fall short of modeling the non-Euclidean interactions due to the underlying fully-connected or translation-invariant assumptions. The advent of Graph Neural Networks (GNNs) has inspired the formulation of variable dependencies as a given or learnable graph, with variables being nodes and pairwise relationships being edges. Existing literature models the dependency structures based on different topological perspectives and temporal granularity (*e.g.*, undirected Yu et al. (2018) and directed graph Li et al. (2017), static Bai et al. (2020); Wu et al. (2020) and dynamic graphs Ye et al. (2022); Cao et al. (2020), single or multiple layers Lin et al. (2021)). On the other hand, structural knowledge has been an important component in GNN-based forecasting methods. In many tasks such as the traffic Guo et al. (2019); Li et al. (2017) and skeleton-based action prediction Yan et al. (2018), structural knowledge is explicitly presented as spatial connections. In other cases without an explicit topological structure, the structural knowledge can be drawn from either domain knowledge (*e.g.*, transfer entropy Duan et al. (2022), Mel-frequency cepstral coefficients Lin et al. (2021)) or feature similarity (*e.g.*, the correlations of decomposed time series Ng et al. (2022), kNN graph Shang & Chen (2021)). Their promising results suggest the capability of structural knowledge to convey meaningful dependency information. Our proposed method enforces the consistency between the learned graph structures and the structural knowledge so as to characterize the underlying relation-temporal dependencies and improve the continual MTS forecasting performance.

2.2 Continual Learning in Multivariate Time Series Forecasting

Deep learning models use continual learning to address the catastrophic forgetting issue when sequentially adapting to new tasks. Existing literature in continual learning can be roughly classified into three categories: experience-replay methods Rolnick et al. (2019), parameter-isolation methods Rusu et al., and regularization-based methods Kirkpatrick et al. (2017); Li & Hoiem (2017). Current continual learning works have been extensively studied on images Wang et al. (2022), texts Ke & Liu (2022), and graph data Zhang et al. (2022a;b). However, much less attention is drawn to time series data, and the focus has primarily been on classification and forecasting tasks without explicitly addressing complex variable dependencies Gupta et al. (2021); He & Sick (2021). How to maintain the meaningful dependency structures and forecasting performance over different regimes is underexplored. Our proposed SKI-CL tackles this issue by jointly optimizing the inferred structure toward the structural knowledge and forecasting objectives based on samples drawn from the current regime and a representative memory.

Recent progress has been made in forecasting Pham et al. (2022); Zhang et al. (2023) and topology identification Money et al. (2021); Natali et al. (2022); Isufi et al. (2019); Zaman et al. (2020) from MTS data in an online learning setting, which focuses on adapting the forecasting model and dependency structure from the historical MTS data to future unseen data. In contrast, our study aims to maintain the forecasting performance and infer the learned structures from the existing regimes

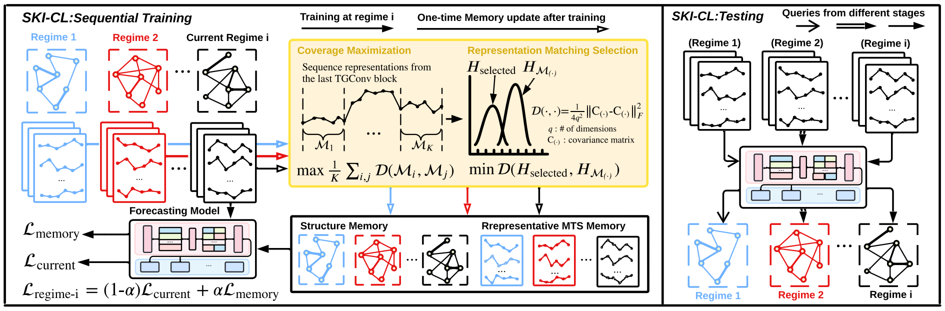


Figure 2: The proposed SKI-CL framework for continual MTS forecasting. The training objectives for each regime contains the current training data and the memory buffer. After training at each regime, the structural knowledge and samples selected by our representation-matching scheme are added to the current memory. At testing phase, SKI-CL is able to dynamically infer faithful dependency structures for different regimes without accessing the memory buffer.

while continuously updating the model over the latest regime. Without forgetting the structural knowledge that has been acquired previously, the model can easily cope with similar regimes that may be encountered in the future. We further discuss other related topics in Appendix B.

3 METHODOLOGY

In this section, we present the proposed SKI-CL framework for continual MTS forecasting as shown in Figure 2. We first formally state the continual MTS forecasting problem with dependency structure learning. Then, we introduce the structural knowledge-informed graph learning model (the detailed model design is shown in Figure 3) and the representation-matching sample selection scheme for continual MTS forecasting.

3.1 PROBLEM STATEMENT

We first introduce the MTS forecasting task in a single regime (stage). Let $X \in \mathbb{R}^{N\times T}$ denote the MTS data containing N variables and T total time steps, where $X_{:,t} \in \mathbb{R}^{N\times 1}$ denotes t-th time step across all variables and $X_{i,:} \in \mathbb{R}^{1\times T}$ denotes i-th variable. Our target is to learn a model that includes a dynamic graph inference module $\mathcal{G}(\cdot)$ summarizing a historical τ-step window of MTS as a graph to encode the dependency structure, as well as a forecasting module $\mathcal{F}(\cdot)$ predicting the next τ' time steps based on the input window and inferred graph. Mathematically, at a starting time step t, the corresponding forecast is defined as: $\hat{X}_{:,t:t+\tau'-1} = \mathcal{F}(X_{:,t-\tau:t-1}, \mathcal{G}(X_{:,t-\tau:t-1}))$.

We further extend the forecasting task to a continual learning setting. In continual learning, there exist S distinct regimes of MTS data with different dependencies and temporal dynamics. The model can only access MTS data of the current regime. Denoting the data of s-th regime as $X^{(s)}$, the objective is to learn a model to minimize the forecasting error across all seen regimes: $\mathcal{F}^*, \mathcal{G}^* = \arg\min_{\mathcal{F},\mathcal{G}} \sum_{s=1}^{\mathcal{S}} L\left(\hat{X}^{(s)}_{:,t:t+\tau'-1}, X^{(s)}_{:,t:t+\tau'-1}\right)$ with L being the loss function. We assume there is a readily available or extracted structural knowledge $A \in \mathbb{R}^{N\times N}$(either partial or completed) at each regime that serves as a reference to characterize the underlying dependencies.

3.2 STRUCTURAL KNOWLEDGE INFORMED GRAPH LEARNING FOR MTS FORECASTING

Dynamic Graph Inference Different from the existing works that generate a static graph at the regime level Wu et al. (2020); Bai et al. (2020); Shang & Chen (2021), we aim to model the variable dependencies of MTS as a dynamic graph at the granularity of an input window. Therefore, as shown in Figure 3 (left), we construct a dynamic graph inference module that more precisely reveals the relation-temporal dynamics in a single regime and has the capacity to handle dependencies change when the regime shifts. Following Shang & Chen (2021); Cini et al. (2023), we explicitly model and parameterized each edge for all node pairs. For a possible edge connecting node i and j, we use a temporal encoding function as a feature extractor to yield node embedding z_i and z_j (*i.e.*, $z_* = \Phi(X_{*,t-\tau:t-1})$) which are concatenated as the edge embedding. Next, we use another

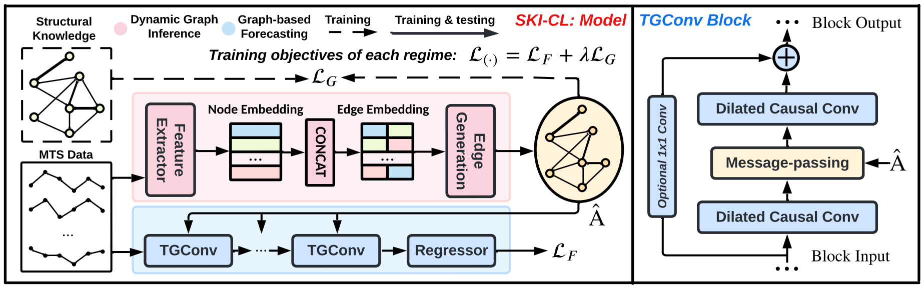


Figure 3: The proposed SKI-CL Model for dependencies modeling and MTS forecasting.

generic mapping to finalize the edge generation as $\hat{A}_{ij} = \Psi(z_i \| z_j)$. The output graph $\hat{A} \in \mathbb{R}^{N \times N}$ summarizes the variable interactions from the temporal dynamics within a sequence, and can be further used to generate the forecasts. Note that there are multiple choices to parameterize Φ and Ψ, where we use stacked convolution layers and a multilayer perceptron (MLP), respectively.

Dependencies Characterization with Structural Knowledge To learn a faithful dynamic graph structure that characterizes the underlying dependencies of each regime, we incorporate structural knowledge as a reference in learning objectives. In real-world MTS modeling, the edges can be either continuous, if we can quantify the strength in the context, or binary if we are more confident of the connection in a qualitative sense (*e.g.*, physical connections). It also interleaves with the fact that if the structural knowledge can be fully observed, as in many cases we are only confident in the existence of certain relationships.

To fully leverage different forms of structural knowledge in dependency structure learning, we design an adaptive scheme that imposes different constraints on the parameterized graph in the objective function, which is denoted as $\mathcal{L}_{\mathrm{G}}$. If an edge is treated as a binary variable, we activate the parameterized edge with a sigmoid function to approximate the Bernoulli distribution $\hat{A}_{ij} \sim \mathrm{Bern}(\theta_{ij})$, where $P(\hat{A}_{ij} = 1) = \theta_{ij}$ is the probability that an edge is forming between node i and node j. Then, we encourage the probability of edge to be consistent with the prior, which essentially minimizes the binary cross entropy: $\mathcal{L}_{\mathrm{G}} = \sum_{i,j} -A_{ij} \log \theta_{ij} - (1 - A_{ij}) \log (1 - \theta_{ij})$.

If an edge is treated as a continuous variable, we activate the parameterized edge with a ReLU function to remove the weak connections, and enforce the consistency between the numerical values of the parameterized edge and the prior, representing a similar interaction strength. This is achieved by minimizing the MSE objective $\mathcal{L}_{\mathrm{G}} = \frac{1}{N^2} \sum_{i,j} \|A_{ij} - \hat{A}_{ij}\|^2$. So far, we have discussed the cases when structural knowledge is readily available/fully observed. For partially observed structural knowledge, we only enforce the consistency between the known entries in the structural knowledge and corresponding parameterized edges, as the dynamic graph inference module is still able to capture and infer the underlying dynamic dependencies via optimization based on the existing structural knowledge and the forecasting objective.

Graph-based Forecasting To further exploit the structural and temporal dependencies and produce forecasts, we design a graph-based forecasting module consisting of multiple Temporal Graph Convolution (TGConv) blocks, as shown in Figure 3 (right). In each block, we leverage a dilated causal convolution Bai et al. (2018); van den Oord et al. (2016) to effectively capture forward dynamics of time series. The dilated causal convolution operation on a 1D sequence input $\mathbf{h}$ is expressed as $r_t = \sum_{k=0}^{\mathcal{K}-1} f(k) \cdot h_{t-d \cdot k}$, where r_t denotes the t-th step of obtained representation $\mathbf{r}$, d represents dilation factor, $f(k)$ is the convolution kernel with size k. Since dilated causal convolution exclusively processes univariate time series, we facilitate the modeling of dependencies in MTS by exchanging and aggregating information through the learned structure, denoted as $\hat{A}$. This is achieved by a simple yet effective message-passing neural operation Morris et al. (2019), given the collection of univariate representations $(\mathbf{r}_1, \cdots, \mathbf{r}_N)$:

$$\text{MessagePassing}_{\hat{A}}(\mathbf{r}_i) = \mathbf{W}_1 \mathbf{r}_i + \mathbf{W}_2 \sum_{j \in \mathcal{N}(i)} e_{j,i} \cdot \mathbf{r}_j, \tag{1}$$

where $\mathcal{N}(i)$ represents the neighbors of variable i, $e_{j,i} \in \hat{A}$ denotes an edge weight, $\mathbf{W}_{(\cdot)}$ denotes the learnable weights, and the bias term is omitted for simplicity.

We stack both operations to construct a TGConv block, with an optional 1×1 convolution tackling the possibly different dimensions between the residual input and output. Finally, a fully connected layer serves as a regressor projecting sequence representations onto the forecasts. We adopt the mean squared error between the forecasts and the ground truths as the main learning objective $\mathcal{L}_F$. The total learning objective function for each regime consists of the forecasting objective and the consistency regularization weighted by a hyperparameter λ.

$$\mathcal{L}_{total} = \mathcal{L}_F + \lambda \mathcal{L}_G = \frac{1}{\tau'} \sum_{t'=t}^{t+\tau'-1} \|\hat{X}_{:,t'} - X_{:,t'}\|^2 + \lambda \mathcal{L}_G \tag{2}$$

3.3 Representation-matching Sample Selection for Continual MTS Forecasting

To tackle the forgetting of variable dependencies and temporal dynamics in sequential training, we store a small subset of MTS samples and the structural knowledge from the previous regimes for memory replay when adapting the model to the current regime. Specifically, we propose an efficient sample selection scheme that maximizes the temporal and dependencies coverage of each regime given a limited memory budget. According to the principle of maximum entropy Guiasu & Shenitzer (1985), we can best represent the underlying knowledge of MTS in each regime with the largest entropy, namely, with the most diverse partitions/modes of relational and temporal patterns. Inspired by this principle and its success in characterizing temporal distribution Du et al. (2021), we perform a distribution characterization by splitting the MTS data to the most diverse modes on the representation space (*i.e.*, the representation of all time-consecutive samples that encode variable dependencies and temporal dynamics), which is a constrained optimization problem:

Algorithm 1 Representation-Matching Sample Selection

1: **Input:** Sample representation H, hyperparameters Δ_1, Δ_2, K_0, memory budget for a single regime N_{m}, the number of training samples in a single regime n
2: Split H into modes $\mathcal{M}_1, \ldots, \mathcal{M}_K$ by optimizing (3)
3: Initialize an empty memory buffer S for a regime
4: **for** $k \leftarrow 1$ to K **do**
5: $\quad$ $\mathrm{n}_{\text{sample}} \leftarrow 0$; $\mathrm{n}_{\text{select}} \leftarrow N_{\mathrm{m}} \times \frac{|\mathcal{M}_k|}{n}$; $\mathrm{s} \leftarrow \{\}$
6: $\quad$ **while** $\mathrm{n}_{\text{sample}} \leq \mathrm{n}_{\text{select}}$ **do**
7: $\quad\quad$ $i_{\text{selected}} \leftarrow \min_{\mathrm{i}} \mathcal{D}\left(H_{\mathrm{s}\cup i}, H_{\mathcal{M}_k}\right)$ $\quad \triangleright\ i \notin \mathrm{s}$
8: $\quad\quad$ $\mathrm{s} = \mathrm{s} \cup i_{\text{selected}}$; $\mathrm{n}_{\text{sample}} \mathrel{+}= 1$
9: $\quad$ $S = S \cup \mathrm{s}$
10: **Output:** The memory buffer S

$$\begin{gathered}\max_{0<K\leq K_0} \max_{n_1,\cdots,n_K} \frac{1}{K} \sum_{1\leq i\neq j\leq K} \mathcal{D}\left(\mathcal{M}_i, \mathcal{M}_j\right) \\ \text{s.t. } \forall i, \Delta_1 < |\mathcal{M}_i| < \Delta_2; \sum_i |\mathcal{M}_i| = n,\end{gathered} \tag{3}$$

where $\mathcal{D}(\cdot,\cdot)$ can be any distribution-related distance metric, n is the number of training samples in a single regime, $\Delta 1$, $\Delta 2$ and K_0 are hyperparameters to avoid trivial partitions and over-splitting, $\mathcal{M}$ denotes the subset of representations that corresponds to contiguous samples. Specifically, we choose the Deep Correlation Alignment (CORAL) Sun & Saenko (2016) to measure the temporal distribution similarity, *i.e.*, $D(\cdot,\cdot) = \frac{1}{4q^2}\left\|\mathrm{C}_{(\cdot)} - \mathrm{C}_{(\cdot)}\right\|_F^2$, where q is the number of dimensions for each hidden state, $\mathrm{C}(\cdot)$ denotes the second-order statistics (covariance matrix). The detailed optimization procedure is explained in Appendix A.

After the most diverse modes are obtained, the distribution of a regime can be efficiently preserved by selecting a small number of the most representative samples of each mode. The selection algorithm is shown in Algorithm 1, where we select samples that minimize CORAL to ensure that the selected small number of samples are well aligned/matched to each mode of MTS. By iterating all modes, we update the memory buffer as the union of all selected sample sets.

3.4 Sequential Training and Testing with SKI-CL

We briefly introduce the pipeline of continual MTS forecasting, as shown in Figure 2. At the training phase of i-th regime, the training objectives $\mathcal{L}_{\text{regime-i}}$ contains the objective for the current training samples, denoted as $\mathcal{L}_{\text{current}}$ and that for the memory of previous i-1 regimes, denoted as $\mathcal{L}_{\text{memory}}$, where $\mathcal{L}_{\text{regime-i}} = \mathcal{L}_{\text{current}} + \alpha\mathcal{L}_{\text{memory}}$, with α being the weight of memory loss.

After training, the structural knowledge of the current regime is saved in the structural memory and the training samples are selected to enrich the MTS memory as aforementioned. At the testing phase, SKI-CL is able to maintain the forecasting performance on the queries of testing samples from all regimes up to the current one, and accordingly recover the learned dependency structures informed by the structural knowledge of each regime (as shown in Figure 2(right)). Unlike other graph structure learning methods for continual MTS forecasting, our method is able to dynamically infer faithful dependency structures for existing and current regimes without accessing the memory buffer, which is more practical in real world applications.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets To evaluate the performance of SKI-CL on continual MTS forecasting, we conduct experiments on three public benchmark MTS datasets including the traffic (Traffic-CL), solar energy (Solar-CL), and human activity recognition (HAR-CL), as well as one synthetic dataset based on Non-repeating Random Walk Denton (2005) that is used in Liu et al. (2022b) under continual learning setting. The statistics of these datasets are summarized in Table 4 in Appendix C. For Traffic-CL and Solar-CL, the structural knowledge is the spatial proximity of the sensor/station. For HAR-CL, the partial structural knowledge is drawn from the domain-specific motion dynamics. For synthetic data, structural knowledge is the feature similarity of different variables. The edges from the structural knowledge are binary for Traffic-CL and HAR-CL, and continuous for Solar-CL and Synthetic-CL.

Baselines We compare SKI-CL with a number of dependency-modeling-based forecasting methods and commonly used continual learning methods to resolve the catastrophic forgetting issue in sequential training. The forecasting methods include statistical model VAR Lütkepohl (2005), ARIMA Box et al. (2015), and deep learning models including TCN Bai et al. (2018), LSTNet Lai et al. (2018), STGCN Yu et al. (2018), MTGNN Wu et al. (2020), AGCRN Bai et al. (2020), GTS Shang & Chen (2021), ESG Ye et al. (2022), StemGNN Cao et al. (2020), Autoformer Wu et al. (2021), PatchTST Nie et al. (2022), Dlinear Zeng et al. (2023), TimesNet Wu et al. (2022), iTransformer Liu et al. (2023), and OFA Tian Zhou (2023) where STGCN and GTS use structural knowledge and ESG learns a dynamic graph in MTS modeling. All forecasting methods are first evaluated on sequential training without any countermeasures (denoted as seq). The continual learning methods employ the memory-replay-based methods including the herding method (denoted as herd) Rebuffi et al. (2016), the randomly replay training samples (denoted as er), a DER++ method Buzzega et al. (2020) that enforces a L_2 knowledge distillation loss on the previous logits (denoted as der++) and a MIR method that selects samples with highest forgetting for experience replay (denoted as mir). For the proposed SKI-CL, we also evaluate our proposed representation-matching sample selection scheme.

Evaluation Metrics We adopt two metrics based on Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the performance on continual MTS forecasting, *i.e.*, the Average Performance (AP) and Average Forgetting (AF) Lopez-Paz & Ranzato (2017); Zhang et al. (2022a). The AP at i-th regime is defined as $AP = \sum_{j=1}^{i} \mathrm{P}_{i,j}/i$ for $i \geq 1$, where $P_{i,j}$ denotes the performance on regime j after the model has been sequentially trained from stage 1 to i. Similarly, the Average Forgetting is defined as $AF = \sum_{j=1}^{i-1}(\mathrm{P}_{i,j} - \mathrm{P}_{j,j})/(i-1)$ for $i \geq 2$. Besides the forecasting performance, we also evaluate AP and AF on the learned dependency structures, where the average precision (Prec.) and average recall (Rec.) are used for a binary graph, MAE and RMSE are used for a continuous graph. More details of datasets, baselines, and evaluations are provided in Appendix C.

4.2 PERFORMANCE EVALUATION FOR CONTINUAL MTS FORECASTING

In this paper, we focus on a multi-horizon continual forecasting task. Table 1 summarizes the comparison results of selected baselines (full experiment results are shown in Appendix Table 7) versus our proposed SKI-CL method and its variants for 12 horizon predictions. The corresponding standard deviations and results for different horizons are provided in Table 9 and 8 in Appendix F.

Based on the experiment results in Table 1, we observe that statistical methods, such as VAR and ARIMA, cannot perform well under continual learning settings and exhibit obvious performance

Table 1: Experiment Results for 12 Horizon Prediction. (Lower MAE and RMSE for AP mean better; When AP is comparable, lower MAE and RMSE for AF mean better.)

Model	Traffic-CL				Solar-CL				HAR-CL ($\times 10^{-2}$)				Synthetic-CL($\times 10^{-2}$)			
	AP ↓		AF		AP ↓		AF		AP ↓		AF		AP ↓		AF	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
VAR$_{seq}$	88.19	126.01	58.38	80.58	167.30	534.42	205.27	658.80	19.59	28.38	1.93	2.19	22.34	32.70	9.18	13.54
ARIMA$_{seq}$	141.75	159.89	77.61	77.40	14.97	18.92	4.75	2.92	40.68	52.87	2.35	2.38	42.24	43.51	13.39	12.98
TCN$_{seq}$	16.88	28.67	3.77	6.83	2.03	4.84	0.06	0.24	14.85	23.42	3.60	5.06	4.30	4.90	0.66	0.99
TCN$_{mir}$	15.70	26.53	1.70	3.22	1.99	4.79	0.10	0.19	13.91	22.15	2.64	2.93	3.79	4.63	0.46	0.73
TCN$_{herd}$	15.55	26.21	1.49	2.81	2.01	4.82	0.13	0.23	13.87	22.08	2.05	2.88	3.72	4.61	0.35	0.67
TCN$_{er}$	15.51	26.23	1.46	2.80	1.98	4.73	-0.05	0.02	13.66	21.78	1.82	2.59	3.29	4.30	0.34	0.61
TCN$_{der++}$	15.46	25.68	1.33	2.49	1.95	4.69	-0.07	-0.02	13.56	21.55	1.69	2.28	**3.00**	**4.00**	0.28	0.32
ESG$_{seq}$	18.77	30.02	6.46	10.07	2.80	5.77	1.28	1.74	17.63	26.84	7.28	9.41	8.98	14.19	1.32	1.98
ESG$_{mir}$	18.24	29.83	5.02	8.25	2.03	4.83	0.25	0.49	17.25	26.63	4.01	5.21	8.95	13.91	1.21	1.81
ESG$_{herd}$	17.49	28.64	4.82	7.45	1.92	4.72	0.13	0.53	17.22	26.59	3.99	5.13	8.94	13.88	1.11	1.74
ESG$_{er}$	16.40	27.50	3.05	5.34	2.01	4.82	0.24	0.44	17.15	25.84	4.63	5.33	8.84	13.86	1.21	1.62
ESG$_{der++}$	17.40	29.21	4.01	6.97	1.91	4.57	0.09	0.21	16.20	24.32	5.18	6.00	8.81	13.77	1.02	1.42
GTS$_{seq}$	17.26	29.11	2.33	3.48	2.19	5.20	0.27	0.59	16.44	25.41	3.68	5.10	6.51	8.89	1.88	3.39
GTS$_{mir}$	17.17	29.08	2.13	3.31	2.15	5.16	0.14	0.68	15.83	24.85	3.27	4.91	6.44	8.57	1.31	2.86
GTS$_{herd}$	17.00	29.01	2.17	2.98	2.11	5.06	0.13	0.30	15.65	24.33	1.99	2.86	6.34	8.23	1.18	1.81
GTS$_{er}$	15.83	26.20	1.12	2.52	2.01	4.75	0.12	0.05	15.06	23.52	2.00	2.73	5.59	6.32	0.44	0.69
GTS$_{der++}$	15.84	26.05	1.15	2.33	1.94	4.57	-0.25	-0.19	14.80	23.01	1.52	1.88	5.43	6.67	0.23	0.30
MTGNN$_{seq}$	19.88	32.94	7.83	12.68	2.12	4.75	0.38	0.44	14.86	22.58	2.59	3.61	10.26	14.92	1.16	1.81
MTGNN$_{mir}$	18.01	31.84	5.03	8.97	2.00	4.73	0.21	0.40	14.59	22.52	2.24	3.53	8.92	12.91	1.07	1.33
MTGNN$_{herd}$	17.93	30.70	4.90	8.40	1.89	4.68	0.13	0.35	14.09	22.50	1.13	1.62	8.11	12.88	1.03	1.27
MTGNN$_{er}$	15.79	26.52	2.76	4.87	1.94	4.62	0.14	0.25	13.59	21.85	1.91	2.79	8.70	13.69	0.61	1.21
MTGNN$_{der++}$	15.40	25.99	2.22	4.10	$\underline{1.90}$	$\underline{4.57}$	0.06	0.14	13.57	21.75	1.63	2.40	8.63	13.51	0.50	0.92
PatchTST$_{seq}$	19.11	32.50	2.34	2.97	2.64	5.32	0.72	0.43	17.91	27.13	7.18	6.88	4.85	5.93	1.59	1.78
PatchTST$_{mir}$	19.04	32.23	2.28	2.79	2.61	5.30	0.70	0.40	17.82	26.89	6.82	4.81	4.83	5.86	1.55	1.72
PatchTST$_{herd}$	18.96	32.10	2.21	2.67	2.60	5.30	0.68	0.35	17.73	26.84	6.62	4.79	4.80	5.79	1.43	1.68
PatchTST$_{er}$	18.77	31.50	1.98	2.01	2.57	5.27	0.47	0.30	17.57	26.40	6.02	4.69	4.72	5.26	1.03	1.54
PatchTST$_{der++}$	18.53	31.34	1.75	1.98	2.53	5.17	0.43	0.28	17.12	26.13	5.79	4.32	4.64	5.13	0.83	0.88
DLinear$_{seq}$	19.69	32.75	2.91	2.83	3.47	6.56	1.17	1.12	17.32	26.31	2.71	3.43	4.81	5.81	1.64	1.57
DLinear$_{mir}$	19.37	32.25	2.17	2.59	3.45	6.51	1.02	1.01	16.87	26.12	2.67	3.01	4.79	5.73	1.47	1.40
DLinear$_{herd}$	19.53	32.40	2.25	2.68	3.41	6.50	1.03	1.00	16.83	25.81	2.57	2.91	4.77	5.70	1.59	1.46
DLinear$_{er}$	19.19	32.30	1.73	2.14	3.37	6.43	0.93	0.98	16.71	25.75	2.13	2.85	4.74	5.20	1.23	1.43
DLinear$_{der++}$	19.02	31.97	1.75	1.93	3.25	6.37	0.83	0.79	16.58	25.47	1.92	2.77	4.21	4.88	1.12	1.13
TimesNet$_{seq}$	17.77	29.91	3.13	6.93	3.92	7.18	1.46	2.51	18.38	27.61	4.33	5.15	5.18	6.13	1.72	2.03
TimesNet$_{mir}$	17.53	29.61	2.44	5.32	3.77	7.15	1.22	1.57	18.27	27.59	4.01	5.08	5.12	6.05	1.69	1.97
TimesNet$_{herd}$	17.38	29.53	2.56	5.83	3.83	7.10	1.03	1.44	18.01	27.53	3.46	5.03	5.10	6.03	1.68	1.95
TimesNet$_{er}$	17.25	29.33	1.97	4.19	3.55	7.02	0.42	0.91	17.84	27.07	3.28	4.01	4.93	5.90	1.42	1.90
TimesNet$_{der++}$	17.13	29.28	1.56	4.02	3.45	6.55	0.37	0.90	17.73	26.86	3.11	3.87	4.81	5.88	1.32	1.78
iTransformer$_{seq}$	16.23	27.83	2.33	3.41	2.87	5.84	1.23	1.31	16.03	25.08	4.87	5.35	6.28	7.72	1.52	1.92
iTransformer$_{mir}$	16.19	27.62	1.98	3.01	2.23	4.90	1.12	1.14	15.89	24.90	4.73	4.92	6.13	7.55	1.47	1.81
iTransformer$_{herd}$	16.11	27.50	1.84	2.95	2.01	4.73	0.88	0.92	15.33	23.88	3.54	4.17	6.09	7.31	1.30	1.59
iTransformer$_{er}$	16.06	27.28	1.78	2.93	1.95	4.67	0.53	0.94	15.11	23.71	3.23	3.93	5.92	7.09	1.06	1.23
iTransformer$_{der++}$	15.98	27.18	1.65	2.88	1.88	4.53	0.43	0.86	14.86	22.93	2.93	3.03	5.77	7.03	0.97	1.03
OFA$_{seq}$	19.10	32.48	2.21	2.43	3.04	6.33	1.26	1.57	17.40	26.20	5.32	3.69	4.72	5.22	1.63	1.85
OFA$_{mir}$	19.03	32.27	2.30	2.21	2.97	5.93	1.07	1.32	17.32	26.17	4.86	3.59	4.63	5.15	1.58	1.81
OFA$_{herd}$	18.91	32.20	1.99	2.13	2.83	5.73	0.91	0.75	17.35	26.19	4.51	3.36	4.45	4.91	1.55	1.62
OFA$_{er}$	18.83	32.12	1.83	1.97	2.53	5.25	0.50	0.38	17.32	26.17	4.33	3.17	4.17	4.80	1.53	1.47
OFA$_{der++}$	18.50	31.33	1.70	1.84	2.47	5.13	0.40	0.28	17.25	26.15	4.11	3.07	4.03	4.71	1.20	1.14
SKI-CL$_{seq}$	17.30	29.38	4.38	7.80	2.02	4.73	0.30	0.50	14.73	23.31	3.91	5.07	4.70	5.85	1.97	3.46
SKI-CL$_{mir}$	15.77	26.32	1.95	3.47	1.98	4.69	0.35	0.56	13.65	21.77	2.61	3.82	4.53	5.01	1.67	2.21
SKI-CL$_{herd}$	15.45	25.73	1.82	3.28	2.00	4.70	0.33	0.52	13.71	21.82	2.53	3.67	4.44	4.82	1.23	2.02
SKI-CL$_{er}$	15.43	25.60	1.69	2.92	1.95	4.67	0.11	0.23	13.58	21.57	1.82	2.50	3.39	4.52	0.30	0.38
SKI-CL$_{der++}$	$\underline{15.39}$	$\underline{25.57}$	1.63	2.87	1.91	4.60	0.10	0.21	$\underline{13.50}$	$\underline{21.47}$	1.74	2.42	3.33	4.43	0.28	0.30
SKI-CL	**15.23**	**25.32**	1.51	2.72	**1.75**	**4.46**	0.09	0.06	**13.41**	**21.30**	1.64	2.08	$\underline{3.24}$	$\underline{4.24}$	0.15	0.23

degradation (AF) in sequential training (*i.e.*, seq). All deep learning based baseline methods, including state-of-art forecasting models, such as PatchTST Nie et al. (2022), TimesNet Wu et al. (2022), iTransformer Liu et al. (2023), and even OFA Tian Zhou (2023) that equipped with the large language model (LLM), also suffer from obvious performance degradation (AF) in sequential training (*i.e.*, seq), suggesting the existence of catastrophic forgetting phenomenons when regime shifts. Moreover, we notice that the memory-replay-based methods (*i.e.*, baselines plus herd, er, and der++) generally alleviate the forgetting issues with better APs (lower RMSE and MAE) and smaller relative AFs compared to sequential training (*i.e.*, seq). Finally, SKI-CL and its variants (SKI-CL$_{er}$ and SKI-CL$_{der++}$) consistently achieve the best or the second-best APs, showing advantages over other baseline models equipped with memory-replay-based methods (*e.g.*, MTGNN$_{der++}$,TCN$_{er}$, GTS$_{der++}$, iTransformer$_{der++}$). These observations demonstrate that learning a dynamic structure is beneficial for MTS modeling, and the structural knowledge helps to characterize the general variable behaviors in each regime. Even partially observed structural knowledge can serve as a valid reference to learn the dependency structures. Finally, we observe that SKI-CL consistently outperforms its variants

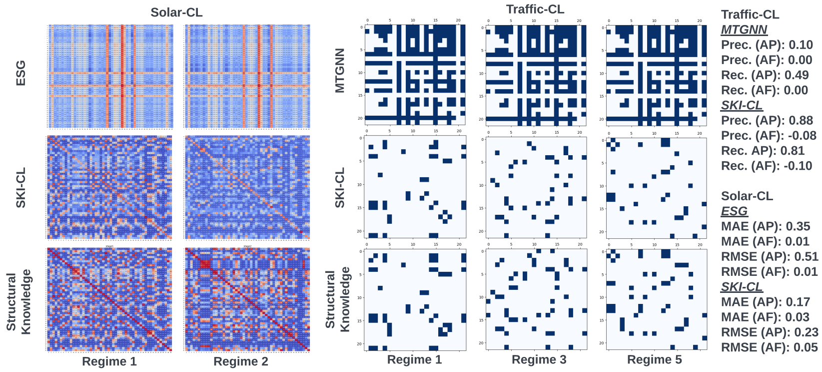


Figure 4: The Structure Visualizations on Traffic-CL and Solar-CL datasets.

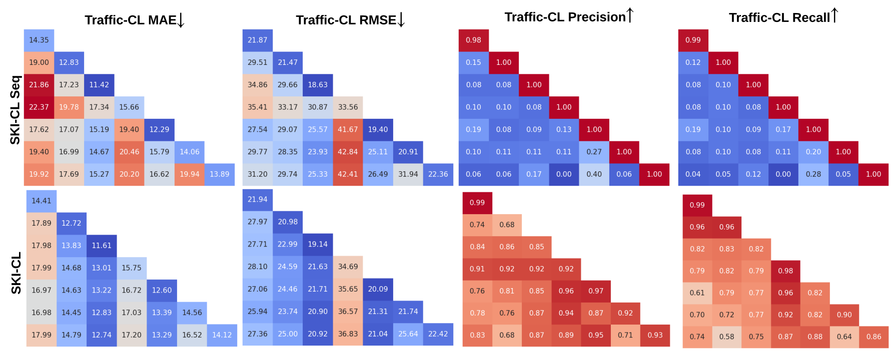


Figure 5: Model Performance with and without Memory Replay (Lower MAE and RMSE indicate better forecasting performance; higher Precision and Recall indicate higher structure similarity.

(*i.e.*, SKI-CL$_{\text{mir}}$, SKI-CL$_{\text{herd}}$, SKI-CL$_{\text{er}}$, and SKI-CL$_{\text{der++}}$) with better APs, suggesting the superiority of our proposed representation-matching scheme to maximize the coverage of dependencies and temporal dynamics.

We visualize the performance matrices of SKI-CL$_{\text{seq}}$ (without memory replay) and SKI-CL (with memory replay and representation-matching scheme) based on the Traffic-CL dataset as shown in Figure 5. Each cell corresponds to the aforementioned $P_{i,j}$, *i.e.*, the performance on regime j after the model has been sequentially trained from stage 1 to i, where i and j denote the row number and column number, respectively. We observe the forecasting accuracy, measured by MAE and RMSE, significantly decreases when no samples are replayed during sequential training. On the contrary, SKI-CL utilizes the proposed representation-matching scheme based memory replay and can maintain reasonably well forecasting performance and infer the dependency structures accurately.

4.3 PRESERVING FAITHFUL DEPENDENCY STRUCTURES

We also evaluate how the baselines and our method preserve the learned structures that are highly correlated to the structural knowledge. Specifically, we compare SKI-CL with GTS on Traffic-CL dataset, and ESG on Solar-CL dataset to investigate the binary edges and continuous edges, respectively. The results of the learned structures and structural knowledge are shown in Figure 4, where the average performance and average forgetting are also annotated. It is clear that SKI-CL is able to alleviate the forgetting of the learned structures on both datasets at the testing phase, while the baselines fail to model the dependencies of MTS in different regimes. This observation also suggests the importance of dynamic structure learning and the incorporation of structural knowledge, to maintain a faithful structure at each regime. We further visualize the similarity matrices between the structures inferred by SKI-CL and structural knowledge, as shown in Figure 5. We observe that the inferred structures at the testing phases of each regime still reveal similarities to the structural knowledge, by comparing the values of each row with the diagonal ones.

We emphasize that we don't intend to use structural knowledge as a ground truth. We have demonstrated that exploiting structural knowledge helps to reduce the performance degradation between

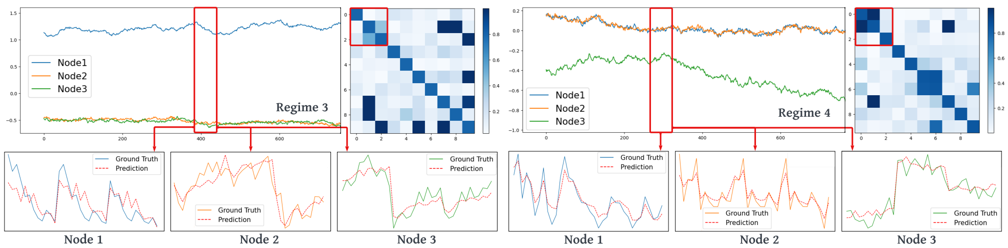


Figure 6: A Case Study of SKI-CL on Synthetic Dataset.

Table 2: Analysis of λ on Traffic-CL

λ	Forecasting Performance				Structure Similarity			
	AP		AF		AP		AF	
	MAE	RMSE	MAE	RMSE	Prec.	Recall	Prec.	Recall
0.0	15.79	26.33	2.71	4.23	0.09	0.13	-0.01	-0.01
0.01	15.42	26.08	2.30	4.13	0.52	0.46	-0.52	-0.57
0.1	15.39	25.97	2.12	4.07	0.84	0.75	-0.11	-0.19
0.5	15.33	25.68	1.68	3.07	0.85	0.78	-0.08	-0.11
1.0	15.23	25.32	1.51	2.72	0.84	0.80	-0.10	-0.12
2.0	15.27	25.51	1.70	3.10	0.85	0.79	-0.08	-0.14
5.0	15.31	25.63	2.04	3.56	0.83	0.72	-0.16	-0.24

Table 3: Analysis of Memory Budget on Traffic-CL

Ratio	Forecasting Performance				Structure Similarity			
	AP		AF		AP		AF	
	MAE	RMSE	MAE	RMSE	Prec.	Recall	Prec.	Recall
0.01	15.23	25.32	1.51	2.72	0.84	0.80	-0.10	-0.12
0.05	14.49	24.35	1.16	2.03	0.86	0.79	-0.10	-0.13
0.1	14.44	24.03	0.77	1.23	0.84	0.78	-0.07	-0.13
0.2	14.25	23.74	0.85	1.34	0.89	0.80	-0.06	-0.14
0.5	14.18	23.06	0.55	0.79	0.90	0.81	-0.05	-0.09

consecutive regimes. Besides, it is beneficial to have the model aligned with the structural knowledge for a better interpretation of each regime. More visualization results are provided in Appendix D.

4.4 Case Study: Inferred Structures and Forecasts across Different Regimes

We provide a case study on the Synthetic-CL dataset to further illustrate the efficacy of SKI-CL, as shown in Figure 6. Our analysis is based on the final SKI-CL model that has been sequentially trained over all regimes. We select three variables (nodes) and visualize the testing data of regime 3 and 4 with different temporal dynamics (The full visualization of all regimes is provided in Appendix G). It is clear that SKI-CL can render a faithful dependency structure that well aligns the variables interactions in each regime (*e.g.*, only nodes 2 and node 3 are similar in regime 3; node 1 and node 2 are highly similar in regime 4). Moreover, SKI-CL gives relatively accurate forecasts that capture each variable's temporal dynamics of ground truths.

4.5 Hyperparameter Analysis

We perform experiments on the Traffic-CL dataset to validate the effectiveness and sensitivity of two key hyperparameters in SKI-CL, the weight of structure regularizer λ (1 by default) and the memory budget (sampling ratio) at each regime (0.01 by default). As shown in Table 2, within a small range, the model is relatively stable in terms of λ, resulting in similar average performance and average forgetting on forecasting performance as well as the inferred structure similarity. We also study the model performance when the memory budget varies in Table 3. We can observe that a larger memory budget can achieve better forecasting performance and less forgetting. Meanwhile, the structure similarity is relatively stable regarding the memory budget. The exploration of other hyperparameter settings is provided in Appendix E.

5 Conclusion

In this paper, we propose a novel Structural Knowledge Informed Continual Learning (SKI-CL) framework to perform MTS forecasting and infer dependency structures inference in the continual learning setting. We develop a forecasting model based on dynamic graph learning and impose a consistency regularization that exploits structural knowledge to facilitate continual learning. We further alleviate the catastrophic forgetting by proposing a novel representation-matching memory replay scheme, which maximizes the temporal coverage of MTS data to efficiently preserve each regime's underlying temporal dynamics and dependency structure. Experiments on one synthetic dataset and three real-world benchmark datasets demonstrate the effectiveness and advantages of the proposed SKI-CL on continual MTS forecasting tasks.

REFERENCES

Davide Anguita, Alessandro Ghio, Luca Oneto, Francesc Xavier Llanas Parra, and Jorge Luis Reyes Ortiz. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. *Journal of universal computer science*, 19(9):1295–1314, 2013a.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3, 2013b.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, 33:17804–17815, 2020.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, 33:17766–17778, 2020.

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1): 96–102, 2001.

Weijun Chen, Yanze Wang, Chengshuo Du, Zhenglong Jia, Feng Liu, and Ran Chen. Balanced graph structure learning for multivariate time series forecasting, 2022.

Xu Chen, Junshan Wang, and Kunqing Xie. Trafficstream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning.

Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. Graph deep learning for time series forecasting. *arXiv preprint arXiv:2310.15978*, 2023.

Anne Denton. Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 8–pp. IEEE, 2005.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 402–411, 2021.

Ziheng Duan, Haoyan Xu, Yida Huang, Jie Feng, and Yueyang Wang. Multivariate time series forecasting with transfer entropy graph. *Tsinghua Science and Technology*, 28(1):141–149, 2022.

Aurora Gonzalez-Vidal, Fernando Jimenez, and Antonio F Gomez-Skarmeta. A methodology for energy multivariate time series forecasting in smart buildings based on feature selection. *Energy and Buildings*, 196:71–82, 2019.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Silviu Guiasu and Abe Shenitzer. The principle of maximum entropy. *The mathematical intelligencer*, 7:42–48, 1985.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 922–929, 2019.

Vibhor Gupta, Jyoti Narwariya, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. Continual learning for multivariate time series tasks with variable input dimensions. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 161–170. IEEE, 2021.

Yujiang He and Bernhard Sick. Clear: An adaptive continual learning framework for regression tasks. *AI Perspectives*, 3(1):1–16, 2021.

Elvin Isufi, Andreas Loukas, Nathanael Perraudin, and Geert Leus. Forecasting time series with varma recursions on graphs. *IEEE Transactions on Signal Processing*, 67(18):4870–4885, 2019.

Bo Jin, Haoyu Yang, Leilei Sun, Chuanren Liu, Yue Qu, and Jianing Tong. A treatment engine by predicting next-period prescriptions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1608–1616, 2018.

Zixuan Ke and Bing Liu. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*, 2022.

Mahdi Khodayar and Jianhui Wang. Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Transactions on Sustainable Energy*, 10(2):670–681, 2018.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

Jaehoon Lee, Chan Kim, Gyumin Lee, Haksoo Lim, Jeongwhan Choi, Kookjin Lee, Dongeun Lee, Sanghyun Hong, and Noseong Park. Time series forecasting with hypernetworks generating parameters in advance. *arXiv preprint arXiv:2211.12034*, 2022.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Haozhe Lin, Yushun Fan, Jia Zhang, and Bing Bai. Rest: Reciprocal framework for spatiotemporal-coupled predictions. In *Proceedings of the Web Conference 2021*, pp. 3136–3145, 2021.

Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. Parameter-free dynamic graph embedding for link prediction. *Advances in Neural Information Processing Systems*, 35: 27623–27635, 2022a.

Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. Multivariate time-series forecasting with temporal polynomial graph neural networks. In *Advances in Neural Information Processing Systems*, 2022b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

Yonghong Luo, Chao Lu, Lipeng Zhu, and Jie Song. Data-driven short-term voltage stability assessment based on spatial-temporal graph convolutional network. *International Journal of Electrical Power & Energy Systems*, 130:106753, 2021.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

Rohan Money, Joshin Krishnan, and Baltasar Beferull-Lozano. Online non-linear topology identification from graph-connected time series. In *2021 IEEE Data Science and Learning Workshop (DSLW)*, pp. 1–6. IEEE, 2021.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Alberto Natali, Elvin Isufi, Mario Coutino, and Geert Leus. Learning time-varying graphs from online data. *IEEE Open Journal of Signal Processing*, 3:212–228, 2022.

William T Ng, K Siu, Albert C Cheung, and Michael K Ng. Expressing multivariate time series as graphs with time series attention transformer. *arXiv preprint arXiv:2208.09300*, 2022.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Namyong Park, Ryan Rossi, Eunyee Koh, Iftikhar Ahamath Burhanuddin, Sungchul Kim, Fan Du, Nesreen Ahmed, and Christos Faloutsos. Cgc: Contrastive graph clustering forcommunity detection and tracking. In *Proceedings of the ACM Web Conference 2022*, pp. 1115–1126, 2022.

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven CH Hoi. Learning fast and slow for online time series forecasting. *arXiv preprint arXiv:2202.11672*, 2022.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2016.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks.

Chao Shang and Jie Chen. Discrete graph structure learning for forecasting multiple time series. In *Proceedings of International Conference on Learning Representations*, 2021.

Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8):1421–1441, 2019.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

Xue Wang Liang Sun Rong Jin Tian Zhou, Peisong Niu. One Fits All: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. 2016.

Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 119:3–11, 2019.

Yiwei Wang, Yujun Cai, Yuxuan Liang, Henghui Ding, Changhu Wang, Siddharth Bhatia, and Bryan Hooi. Adaptive data augmentation on temporal graphs. *Advances in Neural Information Processing Systems*, 34:1440–1452, 2021.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149, June 2022.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 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.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Junchen Ye, Zihan Liu, Bowen Du, Leilei Sun, Weimiao Li, Yanjie Fu, and Hui Xiong. Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 2296–2306, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539274. URL https://doi.org/10.1145/3534678.3539274.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

Bakht Zaman, Luis Miguel Lopez Ramos, Daniel Romero, and Baltasar Beferull-Lozano. Online topology identification from vector autoregressive time series. *IEEE Transactions on Signal Processing*, 69:210–225, 2020.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2141–2149, 2017.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Cglb: Benchmark tasks for continual graph learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022a.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Hierarchical prototype networks for continual graph representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022b.

YiFan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Fan Zhou and Chengtai Cao. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4714–4722, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

A Representation-based Distribution Characterization

A.1 Optimizing the Objective for Coverage Maximization

Inspired by the principle of maximum entropy, we perform a distribution characterization by splitting the MTS data into the most diverse modes on the representation space, which incorporates variable dependence and temporal information. The distribution characterization splits the hidden representation by solving a constrained optimization problem whose objective is formulated as:

$$\begin{aligned} &\max_{0<K\leq K_0} \max_{n_1,\cdots,n_K} \frac{1}{K} \sum_{1\leq i\neq j\leq K} \mathcal{D}\left(\mathcal{M}_i, \mathcal{M}_j\right) \\ &\text{s.t. } \forall i, \Delta_1 < |\mathcal{M}_i| < \Delta_2; \sum_i |\mathcal{M}_i| = n, \end{aligned} \tag{4}$$

where $\mathcal{D}(\cdot,\cdot)$ can be any distribution-related distance metric, n is the number of training samples in a single regime, $\Delta 1$, $\Delta 2$ and K_0 are hyperparameters to avoid trivial partitions and over-splitting, $\mathcal{M}$ denotes the subset of representations that correspond to contiguous samples. The optimization problem can be computationally intractable and the closed-form solution may not exist. We adopt the greedy algorithm proposed by Du et al. (2021). First, we obtain the ordered representation set from the trained model for single regime data. Then, for efficient computation, we evenly split the representation into N parts and randomly search the value of K in $\{2, 3, 4 \cdots N-1\}$ Denote the start and the end index of the representation by A and B respectively. We first consider K = 2 by choosing the first splitting point C from all candidate splitting points via maximizing the distance metric $\mathcal{D}\left(\mathcal{M}_{AC}, \mathcal{M}_{CB}\right)$. After C is determined, we then consider K = 3 and use the same strategy to select another point D. A similar strategy is applied until the number of representation modes is obtained.

B Relation to Other Research Topics

Our method characterizes dynamic variable dependencies of MTS data within each regime to perform continual MTS forecasting tasks. The dynamic variable dependencies modeling and continual adaptation nature relate our method to two research topics, *i.e.*, the dynamic graph learning and learning with temporal drift of MTS. Dynamic graph learning focuses on adapting to the changes in the explicit graph structure and possibly features over time, and leverage it to facilitate downstream tasks (*e.g.,* node classification Liu et al. (2022a), link prediction Wang et al. (2021), community detection Park et al. (2022)). In contrast, our proposed continual multivariate time series forecasting method needs to discover underlying dependencies structure of variables over different regimes (stages), by capturing the temporal dynamics of MTS data. On the other hand, temporal drift of MTS represents the phenomenon where the data distribution of the target MTS changes over time in unforeseen ways, where the approaches are often more reactive and focused on adapting to changes in data distribution for a specific task Du et al. (2021); Kim et al. (2021); Lee et al. (2022). Based on the notion, it is important to emphasize the continual adaptation nature of continual learning setting regarding the temporal drifts, where the catastrophic forgetting happens due to the shifts over multiple regimes. That being said, continual MTS forecasters need to handle a variety of regimes, not just adapt to changes in data distribution for a single one, which is to some degree more realistic and challenging. Temporal drift represents a specific challenge within this broader spectrum of adaptation.

Moreover, while the aforementioned topics as well as our study deal with the evolving data, the learning objectives can be very different. For the temporal drift methods, the objective is to adapt the forecaster to these new MTS patterns by effectively detecting, responding to, and learning from these changes. Similarly, the focus of dynamic graph learning is to adapt its model to cope with the structural and possible feature changes. However, the main focus this paper is to prevent the model from forgetting previously learned knowledge (coupled temporal dynamics and variable dependencies in our scope) when adapting to new MTS distributions.

Table 4: Summary of Datasets

Dataset	Traffic-CL	Solar-CL	HAR-CL	Synthetic-CL
# of nodes	22	50	9	10
# of all time steps	106,848	52,560	600,576	24,000
# of regimes	7	5	4	4
Regime	year	state	activity	adjacency
Structure avail.	Completed	Completed	Partial	Completed

C Datasets, Baselines and Evaluations for Continual MTS Forecasting

C.1 Datasets

Traffic-CL Following the fashion in PEMSD3-Stream Chen et al., we construct the Traffic-CL dataset based on the PEMSD3 benchmark for continual MTS forecasting tasks. The PEMSD3 benchmark data was collected by the Performance Measurement System in California Chen et al. (2001) in real-time by every 30 seconds and further aggregated to 5-min granularity. The PEMSD3-Stream dataset contains traffic data from 2011 to 2017. Specifically, data within a month period from July 10th to August 9th from every year was selected, where the traffic network keeps expanding from year to year. The adjacency matrix for τ-th year is extracted by applying a Gaussian kernel to the spatial all pairwise distances between two traffic sensors, as shown in Equation 5.

$$A_\tau[i,j] = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma_d^2}\right), i \neq j \quad \text{and} \quad d_{ij} \leq \epsilon \\ 0, \text{ otherwise} \end{cases} \tag{5}$$

where d_{ij} denotes the spatial distance between sensor i and j. σ_d and ϵ are the standard deviation and a predefined threshold (controlling the sparsity of the adjacency matrix, set as 1), respectively.

Based on the constructed PEMSD3-Stream dataset, we make the following modifications to further simulate distinct regimes in the setting of continual forecasting. For each year, we rank and select the top 100 traffic sensors with the largest node degrees, based on which we randomly select 22 sensors as a set representing a part of the major traffic. Next, we transform the continuous adjacency weights to binary ones by a threshold, and use it as the structural knowledge. As such, each regime is represented by a different portion of a temporally expanding traffic network from different years, with a binarized structural prior and MTS data defined accordingly. The input horizon for the Traffic-CL dataset is 12.

Solar-CL We build our continual MTS forecasting dataset based on the database for NREL's Solar Power Data for Integration Studies[1], which contains 5-minute solar power data for near 6,000 simulated photovoltaic power plants in the United States for the year 2006. Note that the data in Alabama with a 10-minute granularity is also known as the commonly used Solar dataset in many existing MTS studies Wu et al. (2020); Lai et al. (2018); Cao et al. (2020); Liu et al. (2022b).

We construct different regimes by states with different average annual sunlight levels (measured by $\mathrm{kJ/m^2}$). Based on the statistics[2] and the aggregated MTS data at 10-minute, we select five states, Massachusetts/MA (3944 $\mathrm{kJ/m^2}$) - Arizona/AZ (5755 $\mathrm{kJ/m^2}$) - North Carolina/NC (4456 $\mathrm{kJ/m^2}$) - Texas/TX (5137 $\mathrm{kJ/m^2}$) - Washington/WA (3467 $\mathrm{kJ/m^2}$) as five regimes representing different sunlight patterns in different spatial locations. For each state, 50 photovoltaic power plants are randomly selected, where the spatial information is also used as a valid structural prior, as plants that are geographically close share similar weather and sunlight conditions at a local level. Specially, we first extract the longitude and latitude for each plant and calculate the pairwise geographic distances among all plant pairs. Based on all pairwise distances, we generate the adjacency matrix by applying a Gaussian kernel in Equation 3, where we set $\epsilon = \infty$, indicating a fully connected continuous graph. The input horizon for the Solar-CL dataset is 24.

[1] https://www.nrel.gov/grid/solar-power-data.html
[2] https://worldpopulationreview.com/state-rankings/sunniest-states

HAR-CL We leverage the class boundaries in MTS classification data to construct different regimes in forecasting tasks. Specifically, we build our continual forecasting dataset based on a commonly used MTS Classification benchmark, the Human Activity Recognition (HAR) dataset Anguita et al. (2013b;a) in the UCI database[3], where the data is collected from a group of 30 volunteers from 19-48 years, wearing a Samsung smartphone on the waist and performing six activities of daily living (Walking, Walking upstairs, Walking downstairs, Standing, Sitting, Lying). Each MTS sequence contains 128 time steps and 9 variables that are recorded based on the accelerometer and gyroscope, including the linear accelerations, the angular velocities, and total accelerations along the X-Y-Z axis. The detailed setup for data collection can be found in Figure 1 of Anguita et al. (2013a).

We notice that different human activities naturally form distinct regimes with unique temporal dynamics, where deep learning methods easily achieve over 94% classification accuracy Wang et al. (2019). Motivated by this fact, we construct the continual HAR forecasting dataset considering the following details. Firstly, we select Walking, Walking upstairs, Walking downstairs, and Lying as four different regimes in our task. Secondly, each regime contains over one thousand sequences that are not always temporally connected due to different volunteers. As such, we iterate all sequences and construct the input-output MTS windows in each 128-step sequence, after which all windows are stacked as the training/validation/testing data in one regime. Secondly, we try to build the structural knowledge for each regime based on the domain knowledge. We've already known that linear accelerations are highly correlated to total accelerations along each axis, regardless of the activities, but the dependencies among other variables are not very clear. Therefore, we examine the Pearson correlations of all training sequences for each regime, where the mean of correlation matrices demonstrate distinct patterns, and validate the strong correlations between linear and total accelerations. However, the standard deviations are only small at the diagonal and the aforementioned entries, suggesting a varying and uncertain structure for other variable pairs. To this end, we check a small (15-th) percentile of each entry in the absolute correlation matrix, and apply a threshold to get a binary mask representing the variable dependencies that we are confident of. Note that we only regularize the learned structures to be consistent with the partially observed structural knowledge at the masked entries. As such, we simulate the structured scenario when we are not aware of the completed structural knowledge of MTS. The input for the HAR-CL dataset has 12 steps.

Synthetic-CL Lastly, we generate the synthetic data based on Non-repeating Random Walk Denton (2005), which is used in Liu et al. (2022b) for the evaluation of the learned graph structure in a single regime. Next, we introduce how to generate the MTS data for continual forecasting tasks.

Firstly, we describe how to generate the MTS data step by step. At time step t, given a dynamic weighted adjacency matrix $\mathbf{W}^{(t)}$, $\mathbf{X}^{(t)} = \mathcal{N}\left(\mathbf{W}^{(t-1)}\mathbf{X}^{(t-1)}, \sigma\right) \in \mathbb{R}^{N\times 1}$, where $\mathcal{N}$ denotes the Gaussian distribution, $\sigma \in \mathbb{R}$ controls the variance, N denotes the number of variables, and $\mathbf{X}^{(0)}$ is randomly initialized from the set $[-1, -0.5, 0.5, 1]$. Secondly, we describe how the $\mathbf{W}^{(t)}$ is generated and how to construct different regimes based on $\mathbf{W}^{(\cdot)}$. Assuming there are L total time steps, we define $\mathbf{W}^{(t)}$ as one of S constant matrices $\left(\mathbf{G}^{(1)}, \dots, \mathbf{G}^{(S)}\right)$, where $\mathbf{G}^{(\cdot)}$ is a Laplacian of sparsified random adjacency matrix with sparsity δ, spanning $\lfloor L/S \rfloor$ time steps ($\lfloor \cdot \rfloor$ denotes the floor function). As such, each regime is represented by MTS data with time steps from $(i-1) \times \lfloor L/S \rfloor$ to $i \times \lfloor L/S \rfloor$, with the corresponding weighted adjacency matrix $\mathbf{W}^{(t)} = \mathbf{G}^{(i)}$. Specifically, we set the number of variable $N = 10$, the total time steps $L = 24,000$, the number of regimes $S = 4$, the standard deviation of noise $\sigma = 0.01$, and the matrix sparsity $\delta = 0.1$.

There are two main differences between the synthetic setting of Liu et al. (2022b) and our work despite the setting of continual learning. Firstly, we don't reinitialize the value of each variable when the dynamic weighted adjacency matrix transits to a different one in order to preserve the temporal continuity of MTS data across different regimes. Secondly, the evaluation of graph structure learning is also different due to the forecasting setting. In this particular synthetic dataset, the dynamic weighted adjacency matrix $\mathbf{W}^{(t)}$ describes the data generation process at the single-step level, which can be treated as the ground truth if the non-linear part of $\mathbf{W}^{(t)}$ in the model learns an identity mapping with Gaussian noise. As the graph learned in Liu et al. (2022b) is under the setting of single-step prediction, the $\mathbf{W}^{(t)}$ itself is a reasonable reference for evaluation. In our cases of performing multi-horizon forecasting, the matrix $\mathbf{W}^{(t)}$ raised to a higher power can also demonstrate how the dynamic is propagated in a sequence. In our exploration, we don't assume $\mathbf{W}^{(t)}$ is explicitly

[3] https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

given as the structural prior. Instead, we exploit the Pearson correlation of the generated MTS data and formulate a binary structural prior based on strong absolute correlations, where we will examine if the learned graph structure is able to reveal the variable interactions in $\mathbf{W}^{(t)}$. The input horizon for the Synthetic-CL dataset is 12.

C.2 Baselines

In this part, we introduce the state-of-the-art baseline methods evaluated in our paper and compare the number of parameters for each baseline model in Table 5

- TCN Bai et al. (2018): Temporal convolution networks (TCN) models the temporal causality using causal convolution and do not involve structural dependence modeling.
- LSTNet Lai et al. (2018): LSTNet leverages the Convolution Neural Network (CNN) with a kernel spanning the variable dimension to extract short-term local variable dependencies, and the Recurrent Neural Network (RNN) to discover long-term patterns based on the extracted dependency patterns for MTS forecasting.
- STGCN Yu et al. (2018): STGCN jointly captures the spatial-temporal patterns by stacking spatial graph convolution layers that perform graph convolution using continuous structural knowledge and temporal-gated convolution layers that capture temporal dynamics based on the yielded spatial representations.
- MTGNN Wu et al. (2020): MTGNN learns a parameterized graph with top-k connections for each node, and performs mix-hop propagation for graph convolution and dilated inception for temporal convolution. The parameterized graph is purely optimized based on the forecasting objective. At the testing stage, the inferred graph is static due to the fixed parameters.
- AGCRN Bai et al. (2020): AGCRN models the dependencies graph structure as a product of trainable node embedding and performs graph convolution in the recurrent convolution layer for MTS forecasting. The node embedding and yielded graph are purely optimized based on the forecasting objective. At the testing stage, the inferred graph is static due to the fixed parameters.
- GTS Shang & Chen (2021): GTS infers steady node representations and global node relations from entire training MTS data. The learned dependency structure is used in Diffusion Convolution Recurrent Neural Networks (DCRNN) for MTS forecasting. The parameterized graph is optimized based on the forecasting objective as well as the regularization based on binary structure priors. At the testing stage, the graph is sampled from learned binary edge distributions.
- ESG Ye et al. (2022): ESG learns evolving and scale-specific node relations from features extracted from MTS data. A series of dynamic graphs representing dynamic correlations are utilized in sequential graph convolution and temporal convolution. The dynamic graphs are learned via the optimization of feature extraction layers based on the forecasting objective. At the testing stage, the graphs are dynamics inferred based on each MTS input window.
- StemGNN Cao et al. (2020): The Spectral Temporal Graph Neural Network (StemGNN) is a Graph-based multivariate time-series forecasting model, which jointly learns temporal dependencies and inter-series correlations in the spectral domain, by combining Graph Fourier Transform (GFT) and Discrete Fourier Transform (DFT).
- Autoformer Wu et al. (2021): Autoformer is a Transformer-based model using decomposition architecture with an Auto-Correlation mechanism to capture cross-time dependency for forecasting.
- PatchTST Nie et al. (2022): PatchTST model uses channel-independent and patch techniques to tokenize input time series and perform time series forecasting by utilizing the vanilla Transformer encoders.
- Dlinear Zeng et al. (2023): DLinear adopts trend-seasonal components decomposition techniques for time series data and applies MLP-based architectures for time series forecasting.
- TimesNet Wu et al. (2022): TimesNet model leverages intricate temporal patterns by exploring time series' multi-periodicity and capturing the temporal 2D-variations in 2D space using transformer-based backbones.

Table 5: Baseline Model Parameter Comparison

Model	Number of Parameters	Rank
LSTNet	53253	14
STGCN	96606	13
MTGNN	139990	12
AGCRN	252130	10
GTS	14647763	3
ESG	5999516	6
TCN	170886	11
StemGNN	1060802	8
Autoformer	10612758	4
PatchTST	3226124	7
Dlinear	49168	15
TimesNet	36849590	2
iTransformer	6331916	5
OFA	82033932	1
SKI-CL	614731	9

- iTransformer Liu et al. (2023):iTransformer applies the attention and feed-forward network on the inverted dimensions of the time series data to capture multivariate correlations for time series forecasting.
- OFA Tian Zhou (2023): OFA represents time series data into patched tokens to fine-tune the pre-trained GPT2 (Radford et al., 2019) for various time series analysis tasks

Compared with the state-of-the-art method, our proposed SKI-CL backbone model learns a dynamic graph for MTS modeling, which is also capable of incorporating structural knowledge with different forms and availability scenarios to characterize the dependency structure and temporal dynamics of each regime.

Training Details The dynamic graph inference module consists of 3 stacked 2D convolutional layers. Using C_{in}, C_{out} to denote the number of channels coming in and out, the parameters of these convolutional layers are [C_{in} = 1, C_{out} = 8, kernel size = (1,2), stride = 1, dilation = 2], [C_{in} = 8, C_{out} = 16, kernel size = (1,3), stride = 1, dilation = 2] and [C_{in} = 16, C_{out} = 32, kernel size = (1,3), stride = 1, dilation = 2] respectively. Each batched output is normalized using the Batch Norm2d layer. The hidden dimension for the node feature project is set at 128. For optimization, we train SKI-CL with 100 epochs for every stage under Adam optimizer with a linear scheduler. For the learning rate schedule, we use a linear scheduler, which drops the linear rate from 0.0001 to the factor of 0.8 for every 20 epochs. The data split is 6/2/2 for training/validation/testing. We use a batch size of 32/64/64 for the Traffic-CL, Solar-CL and HAR-CL datasets and use a batch size of 8 for our synthetic dataset. Considering the sizes of datasets, the default memory for each regime is 1% for Traffic-CL and Synthetic-CL and 0.1% for Solar-CL and HAR-CL, respectively. We also weigh the examples in the current stage and in memory differently. We apply a weighted loss regarding the sizes of memory and training data, as stated in the manuscript, and we also construct a data loader that guarantees the balance between training data and memory data. For the setting of our distribution characterization scheme, the default values are N = 10 and K = 7. We implement our models in Pytorch. All experiments are run on one server with four NVIDIA RTX A6000 GPUs. **We will release our code upon paper acceptance.**

C.3 Evaluations

Multi-horizon MTS Forecasting We use two common evaluation metrics for multi-horizon MTS forecasting, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are given as:

$$\text{MAE}(Y, \hat{Y}) = \frac{1}{\tau}\sum_{t=1}^{\tau}|y_i - \hat{y}_i| \tag{6}$$

$$\text{RMSE}(Y, \hat{Y}) = \sqrt{\frac{1}{\tau}\sum_{t=1}^{\tau}(y_i - \hat{y}_i)^2} \tag{7}$$

where τ is the number of time steps, $\hat{y}_t$ is the forecasting results at t-th time step and y_t is the corresponding ground truth. Besides, $\bar{y}$ and $\bar{\hat{y}}$ denote the mean values of ground truth and forecasting results, respectively.

Learning Faithful Dependency Structures For continuous edge variables, we still use MAE and RMSE to measure the similarity between the learned weighted graphs and continuous structural knowledge in an average sense, where the τ in Equations 6 and 7 denotes the entry of adjacency matrix. For binary edge variables, we use the average precision (Prec.) and recall (Rec.) to measure the similarity between the learned dependency structures over all testing MTS input windows and the binary structural prior at each regime, which are given as:

$$\text{Prec.} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Rec.} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

where TP denotes the number of identified edges that exist in the structural prior, TN denotes the number of non-identified that do not exist in the structural prior, FP denotes the number of identified edges that do not exist in the structural prior, and FN denotes the number of non-identified edges that exist in the structural prior.

Continual MTS Forecasting and Dependency Structures Preserving We adopt two widely used metrics to evaluate the performance on continual MTS forecasting and dependency structures preserving, *i.e.*, the Average Performance (AP) and Average Forgetting (AF) Lopez-Paz & Ranzato (2017); Zhang et al. (2022a), where the AP and AF at i-th regime are defined as:

$$\text{AP} = \sum_{j=1}^{i}\frac{\text{P}_{i,j}}{i}, \forall i \geq 1 \tag{10}$$

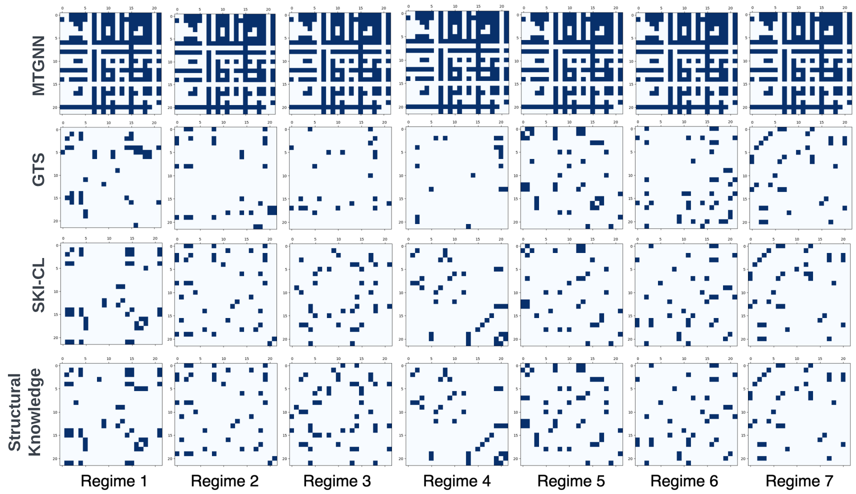
$$\text{AF} = \sum_{j=1}^{i-1}\frac{(\text{P}_{i,j} - \text{P}_{j,j})}{i-1}, \forall i \geq 2 \tag{11}$$

where $\text{P}_{i,j}$ denotes the performance on regime j (including the forecasting performance and structure similarity) after the model has been sequentially trained from stage 1 to i.

Even if we provide both metrics for performance evaluation, we need to emphasize the superiority of average performance over average forgetting in continual learning. Average performance provides a direct measure of how well a learning system is performing on a task or set of tasks. It reflects the system's ability to retain previously learned knowledge over past regimes while adapting to new information. While average forgetting is a relevant metric in assessing the memory capabilities of a learning system, it does not provide a complete picture of the learning system's retention abilities. The average performance takes into account both the retention of old knowledge and the acquisition of new knowledge, providing a more comprehensive evaluation of the learning system's performance. Therefore, we use average performance as the main evaluation metric and average forgetting as an auxiliary metric to measure knowledge retention and model adaptivity.

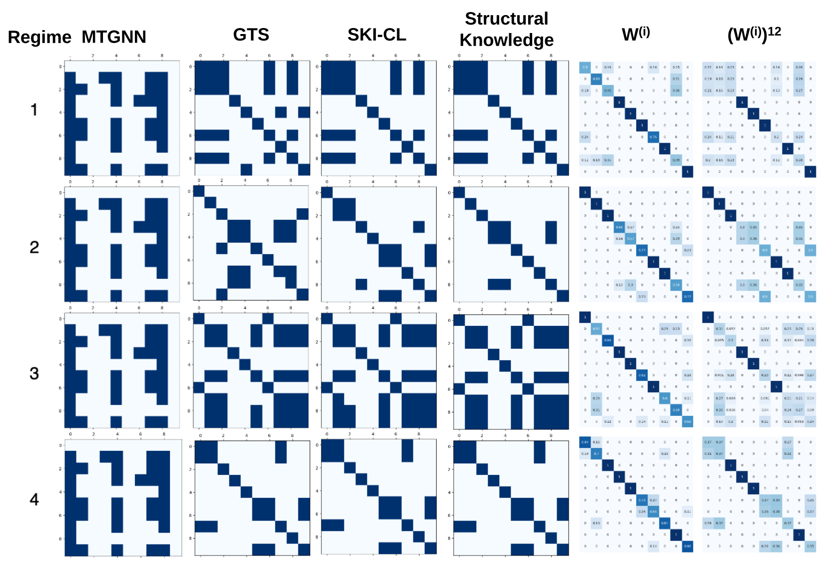
D Visualization of Learned Dependency Structures

In this section, we provide case studies of the learned dependency structures for continual MTS forecasting on all datasets, as shown in Figure 7,8,9, and 10 . For binary edge scenarios, we compare



MTGNN **Prec. (AP): 0.10 Prec. (AF): 0.00 Rec. (AP): 0.49 Rec. (AF): 0.00**
GTS **Prec. (AP): 0.64 Prec. (AF): -0.028 Rec. (AP): 0.53 Rec. (AF): -0.37**
SKI-CL **Prec. (AP): 0.88 Prec. (AF): -0.08 Rec. (AP): 0.81 Rec. (AF): -0.10**

Figure 7: Visualization of Learned Structures on Traffic-CL Dataset.

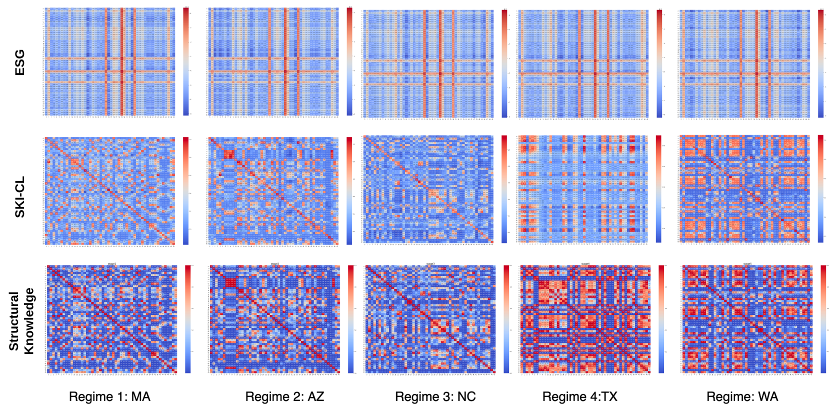


MTGNN **Prec. (AP): 0.53 Prec. (AF): 0.00 Rec. (AP): 0.42 Rec. (AF): 0.00**
GTS **Prec. (AP): 0.91 Prec. (AF): -0.01 Rec. (AP): 0.88 Rec. (AF): -0.01**
SKI-CL **Prec. (AP): 0.88 Prec. (AF): -0.01 Rec. (AP): 0.87 Rec. (AF): -0.004**

Figure 8: Visualization of Learned Structures on Synthetic-CL Dataset.

our proposed SKI-CL with MTGNN$_{\text{er}}$ and GTS$_{\text{er}}$, which only support discrete edge formulation and yield better performance over other baselines. For continuous edge scenarios, we compare our SKI-CL with ESG$_{\text{er}}$ that only support continuous edge formulation. For ESG$_{\text{er}}$, we visualize the generated graph at the last step representing the temporal dynamics of the whole sequence. All the results are plotted after the model is trained at the last regime, where the average structure similarities over all testing windows are also annotated under the case visualizations.

We first discuss the results on the Traffic-CL dataset as shown in Figure 7. As MTGNN directly learns a parameterized graph, it can only infer one fixed dependency structure reflecting the model at the latest regime. That being said, even if the model with experience replay is able to maintain the forecasting performance over the past regimes, the inferred graph fails to preserve the learned unique



ESG **MAE (AP): 0.35 MAE (AF): 0.01 RMSE (AP): 0.51 RMSE (AF): 0.01**
SKI-CL **MAE (AP): 0.17 MAE (AF): 0.03 RMSE (AP): 0.23 RMSE (AF): 0.05**

Figure 9: Visualization of Learned Structures on Solar-CL Dataset.

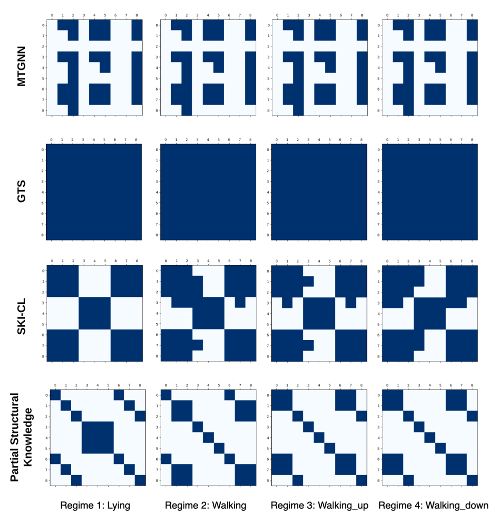


Figure 10: Visualization of Learned Structures on HAR-CL Dataset.

variable dependencies for each regime. Similarly, GTS infers a graph based on parameterized edge distributions, which also reveals the dependencies in a static sense. The incorporation of structural knowledge helps GTS characterize each regime in continual learning, which renders more relevant dependency structure for each regime. Nevertheless, learning a static edge distribution over regimes falls short of handling varying relational and temporal dynamics in complex environments. Another inherent drawback of GTS is that *GTS has to access the training data and memory buffer when inferring graphs at the testing stage, which is not realistic for practical model deployment in real-world applications*. Compared to these baselines, our model yields better forecasting performance (as shown in the manuscripts, as well as Table 8, and a faithful dependency structure that is more consistent with structure knowledge for each regime. These observations demonstrate the importance of the joint design of the dynamic graph inference module and the regularizer based on a structural prior.

For results on Synthetic-CL dataset as shown in Figure 8, the aforementioned observations still hold despite the comparative performance of GTS due to the simplicity of this dataset (Here we use the binary structural knowledge by thresholding the correlations for fair comparisons with GTS).

Table 6: Hyperparameter analysis for distribution characterization

N	K	Forecasting Performance ($\times 10^{-2}$)				Structure Similarity			
		AP		AF		AP		AF	
		MAE	RMSE	MAE	RMSE	Prec.	Recall	Prec.	Recall
5	3	3.54	4.35	0.30	0.38	0.69	0.60	-0.11	-0.17
10	3	3.34	4.27	0.20	0.27	0.65	0.69	-0.04	-0.11
10	5	3.27	4.25	0.18	0.25	0.80	0.78	-0.01	-0.06
10	7	3.24	4.24	0.15	0.23	0.76	0.76	-0.06	-0.08
15	5	3.45	4.18	0.27	0.33	0.75	0.71	-0.06	-0.12
15	10	3.26	4.25	0.15	0.21	0.86	0.75	-0.06	-0.09

Moreover, our model and GTS regularized with strong absolute correlations are able to render binary structures that capture the single-step and multi-step variable interactions of the 'ground truth' dynamic adjacency matrix W^i and its 12th power (the input window size) for each regime.

We next discuss the results on the Solar-CL dataset as shown in Figure 9, where the edge is formulated as a continuous variable. It is clear that our proposed SKI-CL still gains advantages in terms of preserving a faithful continuous dependency structure for each regime. Instead, ESG that learns dynamic graphs still falls short of capturing a relevant structure due to the lack of regime characterizations.

Finally, we investigate our method on the HAR-CL dataset (as shown in Figure 10) when the structural knowledge is partially observed. Here, we do not evaluate the structure similarity due to the incompleteness of the prior as a referencing graph. Instead, we focus on how our model leverages the limited but confident knowledge in dependency structure learning. It can be seen that MTGNN fails to capture the important relationships that reveal in the structural knowledge. Besides, GTS consistently inferred a fully connected graph at the testing stages (even if we have tuned the temperature in the Gumbel-Softmax module), which renders a less meaningful dependency structure. Instead, the proposed SKI-CL exploits partial knowledge and renders faithful structures. For example, SKI-CL is able to capture the correlations of linear accelerations, angular velocities, and total accelerations within three axes, and the irrelevance between accelerations and angular velocities, which is reasonable in a binary sense for lying down behavior. Besides, even if the partial structural knowledge is the same for walking upstairs and walking downstairs, the SKI-CL is able to identify different dependencies patterns for different activities. It demonstrates the effectiveness of the SKI-CL on partially observed structural knowledge, suggesting a certain generalizability of our proposed framework in MTS modeling.

E Hyperparameters Analysis

In this section, we supplement additional analysis of distribution characterization hyperparameters, namely the granularity N (10 by default) and the mode number K (7 by default), using Synthetic-CL dataset. As shown in Table 6, for a fixed N, a relatively large K facilitates inferred structure-preserving and mitigated the forgetting in time series forecasting. When N and K are close, average performance and average forgetting behavior are insensitive to the choice of these hyperparameters. However, a small N degrades the structure-preserving ability as the average forgetting on precision and recall increases.

F Experimental Results for Different Horizon Forecasting

Table 8 summarize the experiment results of baselines and our proposed SKI-CL method with variants for different horizon forecasting performance based on three rounds of experiments. We intentionally select 3-horizon prediction on Traffic-CL and Solar-CL datasets as the settings in Wu et al. (2020) and Cao et al. (2020). While for HAR-CL and synthetic-CL dataset, 6-horizon prediction performance is reported. Based on the results, the memory-replay-based methods generally alleviate the forgetting issues with better APs and smaller relative AFs. Under different horizon prediction settings, SKI-CL and its variants (SKI-CL$_{\text{er}}$ and SKI-CL$_{\text{der++}}$) still consistently achieve the best or the second-best APs, demonstrating its advantages over other baseline methods.

We also provide the standard deviation of AP and AF for the above continual forecasting performance, as shown in Table 9. It can be seen that the standard deviation of SKI-CL is generally comparative or lower to other variants except the Synthetic-CL dataset.

G A Case Study with Inferred Structures and Prediction Visualizations

We provide a case study on the Synthetic-CL dataset to illustrate the efficacy of SKI-CL, as shown in Figure 11. Our analysis is based on the final SKI-CL model that has been sequentially trained over all regimes. We select three variables (nodes) and visualize the testing data, where the temporal dynamics obviously differ across four regimes. It is clear that SKI-CL can render a faithful dependency structure that well aligns the similarity of variables in each regime. Moreover, SKI-CL gives relatively accurate forecasts that capture each variable's temporal dynamics of ground truths.

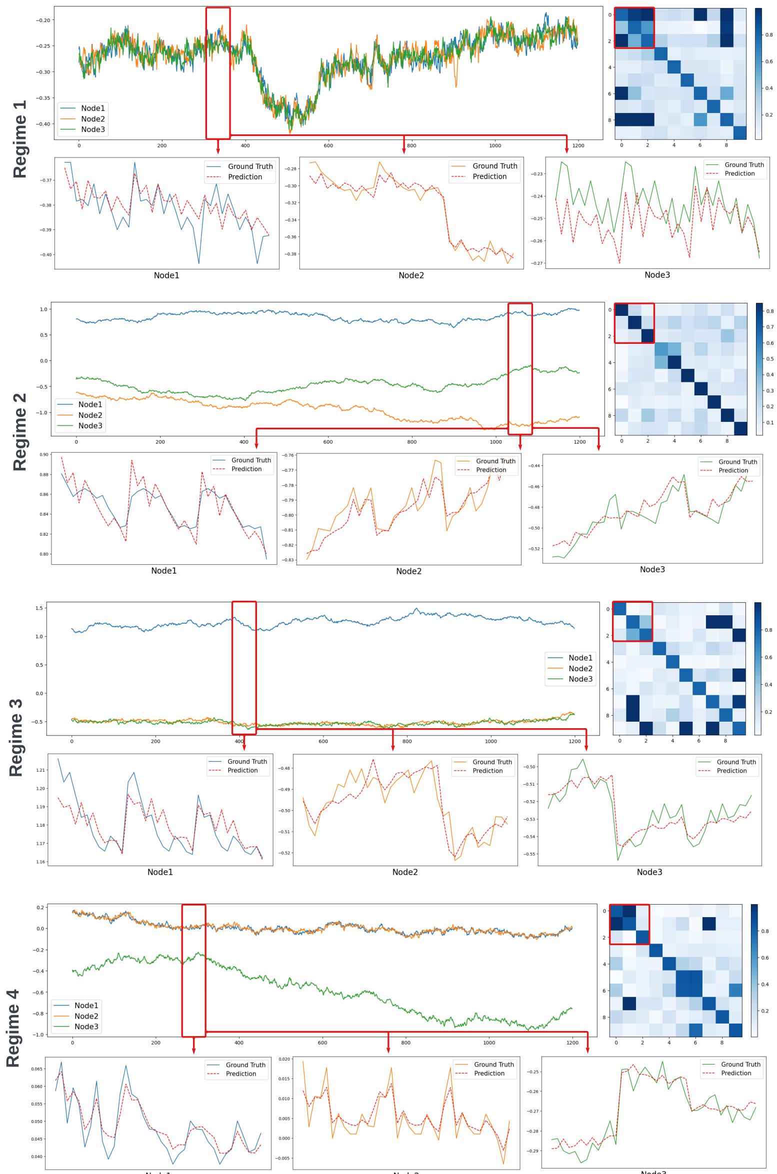


Figure 11: A case study of SKI-CL on Synthetic-CL dataset. In each regime, red rectangles indicate the correspondence between the ground truth time series (top-left) and the inferred variable dependencies (top-right), red arrows indicate the comparisons between these ground truth values and corresponding predictions (bottoms).

Table 7: Experiment Results for 12 Horizon Prediction. (Lower MAE and RMSE for AP mean better; When AP is comparable, lower MAE and RMSE for AF mean better.)

Model	Traffic-CL				Solar-CL				HAR-CL ($\times 10^{-2}$)				Synthetic-CL($\times 10^{-2}$)			
	AP ↓		AF		AP ↓		AF		AP ↓		AF		AP ↓		AF	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
VAR$_{\text{seq}}$	88.19	126.01	58.38	80.58	167.30	534.42	205.27	658.80	19.59	28.38	1.93	2.19	22.34	32.70	9.18	13.54
ARIMA$_{\text{seq}}$	141.75	159.89	77.61	77.40	14.97	18.92	4.75	2.92	40.68	52.87	2.35	2.38	42.24	43.51	13.39	12.98
LSTNet$_{\text{seq}}$	27.01	42.87	11.19	16.72	3.15	5.76	1.12	1.14	16.00	23.96	2.78	3.77	24.13	31.35	5.46	7.32
LSTNet$_{\text{mir}}$	21.79	35.37	4.39	5.21	2.73	5.51	0.93	0.84	15.73	23.14	1.45	1.67	21.44	27.23	0.57	0.72
LSTNet$_{\text{herd}}$	20.86	33.56	3.24	4.58	3.09	5.71	1.08	0.99	15.06	22.98	1.37	1.59	20.31	26.22	0.36	0.43
LSTNet$_{\text{er}}$	20.67	33.38	3.21	4.83	2.49	5.18	0.25	0.32	14.79	22.44	0.79	0.86	20.12	26.17	1.01	1.08
LSTNet$_{\text{der++}}$	20.05	32.22	2.66	3.96	2.44	5.13	0.16	0.20	14.96	21.83	0.70	0.83	20.14	26.08	0.33	0.76
STGCN$_{\text{seq}}$	30.45	49.51	13.91	20.77	2.99	5.75	0.81	0.99	17.36	26.38	1.79	2.25	8.89	13.67	4.55	7.36
STGCN$_{\text{mir}}$	25.63	43.07	8.51	9.27	2.86	5.69	0.73	0.90	17.25	25.80	1.40	2.11	8.57	13.58	4.20	6.82
STGCN$_{\text{herd}}$	24.89	41.42	5.04	8.51	2.83	5.62	0.67	0.74	17.30	25.84	1.49	2.16	8.50	13.21	3.80	6.46
STGCN$_{\text{er}}$	28.53	47.63	9.38	15.79	2.81	5.57	0.81	0.95	16.19	24.81	1.15	1.96	7.77	12.10	3.56	5.99
STGCN$_{\text{der++}}$	27.06	45.77	7.78	13.88	2.71	5.60	0.56	0.53	16.05	23.53	0.94	1.16	7.64	11.37	2.77	5.06
AGCRN$_{\text{seq}}$	21.84	35.93	7.87	12.06	4.02	7.32	-0.41	-0.39	18.01	25.91	1.10	1.62	14.58	23.68	1.37	1.75
AGCRN$_{\text{mir}}$	20.03	34.52	3.13	7.01	3.81	6.92	-0.33	-0.35	15.51	23.57	0.83	1.03	14.40	22.56	1.33	1.65
AGCRN$_{\text{herd}}$	18.63	31.32	2.39	4.19	3.14	5.78	-0.23	-0.13	15.65	23.92	0.21	1.93	14.42	22.58	1.31	1.67
AGCRN$_{\text{er}}$	18.58	31.00	2.82	5.17	2.11	4.07	-1.77	-2.53	15.26	23.25	0.32	0.51	13.67	20.98	0.56	1.11
AGCRN$_{\text{der++}}$	18.29	30.41	2.58	4.56	4.36	7.63	-0.65	-0.86	15.23	23.23	0.28	0.49	13.14	20.62	1.25	1.54
StemGNN$_{\text{seq}}$	18.53	31.23	4.48	7.44	2.79	5.53	0.42	0.52	16.19	24.81	2.21	3.01	13.83	20.01	1.91	1.88
StemGNN$_{\text{mir}}$	17.73	29.75	2.21	5.17	2.75	5.51	0.15	0.39	16.11	23.93	1.53	1.77	12.75	19.23	1.13	1.45
StemGNN$_{\text{herd}}$	17.55	29.63	1.73	3.29	2.78	5.50	1.05	1.16	16.07	23.77	1.04	1.53	12.69	18.46	1.10	1.30
StemGNN$_{\text{er}}$	17.01	28.68	2.07	3.69	2.73	5.52	0.04	0.16	15.95	23.32	1.21	1.25	12.13	18.18	0.65	0.59
StemGNN$_{\text{der++}}$	17.26	29.21	1.63	3.29	2.20	4.88	-0.04	0.02	15.78	23.12	1.01	0.92	12.19	17.98	0.26	0.61
TCN$_{\text{seq}}$	16.88	28.67	3.77	6.83	2.03	4.84	0.06	0.24	14.85	23.42	3.60	5.06	4.30	4.90	0.66	0.99
TCN$_{\text{mir}}$	15.70	26.53	1.70	3.22	1.99	4.79	0.10	0.19	13.91	22.15	2.64	2.93	3.79	4.63	0.46	0.73
TCN$_{\text{herd}}$	15.55	26.21	1.49	2.81	2.01	4.82	0.13	0.23	13.87	22.08	2.05	2.88	3.72	4.61	0.35	0.67
TCN$_{\text{er}}$	15.51	26.23	1.46	2.80	1.98	4.73	-0.05	0.02	13.66	21.78	1.82	2.59	3.29	4.30	0.34	0.61
TCN$_{\text{der++}}$	15.46	25.68	1.33	2.49	1.95	4.69	-0.07	-0.02	13.56	21.55	1.69	2.28	**3.00**	**4.00**	0.28	0.32
ESG$_{\text{seq}}$	18.77	30.02	6.46	10.07	2.80	5.77	1.28	1.74	17.63	26.84	7.28	9.41	8.98	14.19	1.32	1.98
ESG$_{\text{mir}}$	18.24	29.83	5.02	8.25	2.03	4.83	0.25	0.49	17.25	26.63	4.01	5.21	8.95	13.91	1.21	1.81
ESG$_{\text{herd}}$	17.49	28.64	4.82	7.45	1.92	4.72	0.13	0.53	17.22	26.59	3.99	5.13	8.94	13.88	1.11	1.74
ESG$_{\text{er}}$	16.40	27.50	3.05	5.34	2.01	4.82	0.24	0.44	17.15	25.84	4.63	5.33	8.84	13.86	1.21	1.62
ESG$_{\text{der++}}$	17.40	29.21	4.01	6.97	1.91	4.57	0.09	0.21	16.20	24.32	5.18	6.00	8.81	13.77	1.02	1.42
GTS$_{\text{seq}}$	17.26	29.11	2.33	3.48	2.19	5.20	0.27	0.59	16.44	25.41	3.68	5.10	6.51	8.89	1.88	3.39
GTS$_{\text{mir}}$	17.17	29.08	2.13	3.31	2.15	5.16	0.14	0.68	15.83	24.85	3.27	4.91	6.44	8.57	1.31	2.86
GTS$_{\text{herd}}$	17.00	29.01	2.17	2.98	2.11	5.06	0.13	0.30	15.65	24.33	1.99	2.86	6.34	8.23	1.18	1.81
GTS$_{\text{er}}$	15.83	26.20	1.12	2.52	2.01	4.75	0.12	0.05	15.06	23.52	2.00	2.73	5.59	6.32	0.44	0.69
GTS$_{\text{der++}}$	15.84	26.05	1.15	2.33	1.94	4.57	-0.25	-0.19	14.80	23.01	1.52	1.88	5.43	6.67	0.23	0.30
MTGNN$_{\text{seq}}$	19.88	32.94	7.83	12.68	2.12	4.75	0.38	0.44	14.86	22.58	2.59	3.61	10.26	14.92	1.16	1.81
MTGNN$_{\text{mir}}$	18.01	31.84	5.03	8.97	2.00	4.73	0.21	0.40	14.59	22.52	2.24	3.53	8.92	12.91	1.07	1.33
MTGNN$_{\text{herd}}$	17.93	30.70	4.90	8.40	1.89	4.68	0.13	0.35	14.09	22.50	1.13	1.62	8.11	12.88	1.03	1.27
MTGNN$_{\text{er}}$	15.79	26.52	2.76	4.87	1.94	4.62	0.14	0.25	13.59	21.85	1.91	2.79	8.70	13.69	0.61	1.21
MTGNN$_{\text{der++}}$	15.40	25.99	2.22	4.10	<u>1.90</u>	<u>4.57</u>	0.06	0.14	13.57	21.75	1.63	2.40	8.63	13.51	0.50	0.92
Autoformer$_{\text{seq}}$	23.92	40.40	2.58	4.42	6.04	11.41	0.53	1.56	19.97	28.76	2.81	3.08	5.15	6.87	0.53	0.46
Autoformer$_{\text{mir}}$	23.85	40.13	2.21	4.07	5.95	11.18	0.87	1.14	18.57	27.93	1.45	2.88	5.12	6.73	0.32	0.37
Autoformer$_{\text{herd}}$	23.90	40.21	2.43	4.29	5.91	11.12	0.90	1.20	18.78	28.03	1.42	2.66	5.00	6.65	0.23	0.24
Autoformer$_{\text{er}}$	23.26	38.98	1.28	2.00	5.83	10.89	0.49	1.32	18.42	27.05	1.29	2.01	5.02	6.66	0.24	0.25
Autoformer$_{\text{der++}}$	22.11	37.34	1.08	1.92	5.34	9.29	0.21	1.14	18.12	26.91	1.12	1.85	4.97	6.61	0.18	0.23
PatchTST$_{\text{seq}}$	19.11	32.50	2.34	2.97	2.64	5.32	0.72	0.43	17.91	27.13	7.18	6.88	4.85	5.93	1.59	1.78
PatchTST$_{\text{mir}}$	19.04	32.23	2.28	2.79	2.61	5.30	0.70	0.40	17.82	26.89	6.82	4.81	4.83	5.86	1.55	1.72
PatchTST$_{\text{herd}}$	18.96	32.10	2.21	2.67	2.60	5.30	0.68	0.35	17.73	26.84	6.62	4.79	4.80	5.79	1.43	1.68
PatchTST$_{\text{er}}$	18.77	31.50	1.98	2.01	2.57	5.27	0.47	0.30	17.57	26.40	6.02	4.69	4.72	5.26	1.03	1.54
PatchTST$_{\text{der++}}$	18.53	31.34	1.75	1.98	2.53	5.17	0.43	0.28	17.12	26.13	5.79	4.32	4.64	5.13	0.83	0.88
DLinear$_{\text{seq}}$	19.69	32.75	2.91	2.83	3.47	6.56	1.17	1.12	17.32	26.31	2.71	3.43	4.81	5.81	1.64	1.57
DLinear$_{\text{mir}}$	19.37	32.25	2.17	2.59	3.45	6.51	1.02	1.01	16.87	26.12	2.67	3.01	4.79	5.73	1.47	1.40
DLinear$_{\text{herd}}$	19.53	32.40	2.25	2.68	3.41	6.50	1.03	1.00	16.83	25.81	2.57	2.91	4.77	5.70	1.59	1.46
DLinear$_{\text{er}}$	19.19	32.30	1.73	2.14	3.37	6.43	0.93	0.98	16.71	25.75	2.13	2.85	4.74	5.20	1.23	1.43
DLinear$_{\text{der++}}$	19.02	31.97	1.75	1.93	3.25	6.37	0.83	0.79	16.58	25.47	1.92	2.77	4.21	4.88	1.12	1.13
TimesNet$_{\text{seq}}$	17.77	29.91	3.13	6.93	3.92	7.18	1.46	2.51	18.38	27.61	4.33	5.15	5.18	6.13	1.72	2.03
TimesNet$_{\text{mir}}$	17.53	29.61	2.44	5.32	3.77	7.15	1.22	1.57	18.27	27.59	4.01	5.08	5.12	6.05	1.69	1.97
TimesNet$_{\text{herd}}$	17.38	29.53	2.56	5.83	3.83	7.10	1.03	1.44	18.01	27.53	3.46	5.03	5.10	6.03	1.68	1.95
TimesNet$_{\text{er}}$	17.25	29.33	1.97	4.19	3.55	7.02	0.42	0.91	17.84	27.07	3.28	4.01	4.93	5.90	1.42	1.90
TimesNet$_{\text{der++}}$	17.13	29.28	1.56	4.02	3.45	6.55	0.37	0.90	17.73	26.86	3.11	3.87	4.81	5.88	1.32	1.78
iTransformer$_{\text{seq}}$	16.23	27.83	2.33	3.41	2.87	5.84	1.23	1.31	16.03	25.08	4.87	5.35	6.28	7.72	1.52	1.92
iTransformer$_{\text{mir}}$	16.19	27.62	1.98	3.01	2.23	4.90	1.12	1.14	15.89	24.90	4.73	4.92	6.13	7.55	1.47	1.81
iTransformer$_{\text{herd}}$	16.11	27.50	1.84	2.95	2.01	4.73	0.88	0.92	15.33	23.88	3.54	4.17	6.09	7.31	1.30	1.59
iTransformer$_{\text{er}}$	16.06	27.28	1.78	2.93	1.95	4.67	0.53	0.94	15.11	23.71	3.23	3.93	5.92	7.09	1.06	1.23
iTransformer$_{\text{der++}}$	15.98	27.18	1.65	2.88	1.88	4.53	0.43	0.86	14.86	22.93	2.93	3.03	5.77	7.03	0.97	1.03
OFA$_{\text{seq}}$	19.10	32.48	2.21	2.43	3.04	6.33	1.26	1.57	17.40	26.20	5.32	3.69	4.72	5.22	1.63	1.85
OFA$_{\text{mir}}$	19.03	32.27	2.30	2.21	2.97	5.93	1.07	1.32	17.32	26.17	4.86	3.59	4.63	5.15	1.58	1.81
OFA$_{\text{herd}}$	18.91	32.20	1.99	2.13	2.83	5.73	0.91	0.75	17.35	26.19	4.51	3.36	4.45	4.91	1.55	1.62
OFA$_{\text{er}}$	18.83	32.12	1.83	1.97	2.53	5.25	0.50	0.38	17.32	26.17	4.33	3.17	4.17	4.80	1.53	1.47
OFA$_{\text{der++}}$	18.50	31.33	1.70	1.84	2.47	5.13	0.40	0.28	17.25	26.15	4.11	3.07	4.03	4.71	1.20	1.14
SKI-CL$_{\text{seq}}$	17.30	29.38	4.38	7.80	2.02	4.73	0.30	0.50	14.73	23.31	3.91	5.07	4.70	5.85	1.97	3.46
SKI-CL$_{\text{mir}}$	15.77	26.32	1.95	3.47	1.98	4.69	0.35	0.56	13.65	21.77	2.61	3.82	4.53	5.01	1.67	2.21
SKI-CL$_{\text{herd}}$	15.45	25.73	1.82	3.28	2.00	4.70	0.33	0.52	13.71	21.82	2.53	3.67	4.44	4.82	1.23	2.02
SKI-CL$_{\text{er}}$	15.43	25.60	1.69	2.92	1.95	4.67	0.11	0.23	13.58	21.57	1.82	2.50	3.39	4.52	0.30	0.38
SKI-CL$_{\text{der++}}$	<u>15.39</u>	<u>25.57</u>	1.63	2.87	1.91	4.60	0.10	0.21	<u>13.50</u>	<u>21.47</u>	1.74	2.42	3.33	4.43	0.28	0.30
SKI-CL	**15.23**	**25.32**	1.51	2.72	**1.75**	**4.46**	0.09	0.06	**13.41**	**21.30**	1.64	2.08	<u>3.24</u>	<u>4.24</u>	0.15	0.23

Table 8: Experiment Results for Different Horizon Prediction (3-step Horizon Prediction for Traffic-CL and Solar-CL and 6-step Horizon Prediction for HAR-CL and Synthetic-CL) (Lower MAE and RMSE for AP mean better; When AP is comparable, lower MAE and RMSE for AF mean better.)

Model	Traffic-CL (3)				Solar-CL (3)				HAR-CL (6) ($\times 10^{-2}$)				Synthetic-CL (6) ($\times 10^{-2}$)			
	AP ↓		AF		AP ↓		AF		AP ↓		AF		AP ↓		AF	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
VAR$_{seq}$	73.36	104.97	70.26	97.19	59.30	196.75	72.88	242.41	18.04	27.37	1.01	1.28	18.61	27.33	7.70	11.40
ARIMA$_{seq}$	141.53	159.81	77.65	78.19	14.93	19.12	5.84	3.71	41.02	53.11	2.42	2.11	42.25	43.53	13.41	13.01
LSTNet$_{seq}$	24.89	39.49	10.50	15.63	2.45	4.77	1.09	1.45	14.13	21.90	4.05	5.49	25.31	33.71	7.44	10.61
LSTNet$_{mir}$	20.07	31.86	4.18	4.91	2.27	4.45	0.82	0.96	13.20	20.85	2.73	3.96	19.86	27.67	1.71	2.41
LSTNet$_{herd}$	19.11	30.68	3.22	4.73	2.26	4.43	0.81	0.89	13.12	20.66	2.64	3.66	19.75	26.28	0.60	0.90
LSTNet$_{er}$	18.29	29.42	2.55	3.95	1.87	4.10	0.37	0.55	12.65	20.06	2.08	2.94	19.31	24.76	0.31	0.77
LSTNet$_{der++}$	17.94	28.96	2.03	3.18	1.78	3.98	0.18	0.28	12.55	19.94	1.85	2.86	18.92	24.26	0.77	0.57
STGCN$_{seq}$	26.80	42.22	11.90	14.48	3.14	6.30	1.86	3.16	16.52	25.04	5.21	6.52	8.21	12.90	2.27	4.63
STGCN$_{mir}$	26.56	42.31	9.79	14.48	2.67	6.03	1.27	2.75	15.54	24.06	4.22	5.83	5.62	8.82	1.58	3.48
STGCN$_{herd}$	26.53	43.56	9.36	14.79	2.57	5.99	1.15	2.65	15.41	23.97	4.12	5.69	5.53	8.65	1.42	3.37
STGCN$_{er}$	25.54	42.43	8.29	11.71	2.41	4.88	1.10	1.52	15.11	23.54	3.72	4.25	5.12	8.49	1.15	1.76
STGCN$_{der++}$	25.53	42.43	8.12	10.03	2.26	4.41	0.66	0.63	15.51	23.64	3.52	4.13	6.95	11.12	1.86	2.89
AGCRN$_{seq}$	16.13	26.21	3.65	5.29	1.44	3.21	0.31	0.60	13.82	21.88	2.97	4.06	13.00	21.16	0.99	0.47
AGCRN$_{mir}$	15.20	25.39	1.77	2.94	1.32	3.15	0.11	0.34	12.39	20.85	2.16	3.97	12.23	19.48	0.29	0.21
AGCRN$_{herd}$	15.17	25.32	1.59	2.71	1.30	3.14	0.11	0.33	12.32	20.63	2.03	3.96	12.18	19.27	0.12	0.16
AGCRN$_{er}$	14.88	24.72	1.47	2.43	1.27	3.04	0.10	0.04	12.28	20.54	1.98	3.03	10.84	16.82	0.10	0.17
AGCRN$_{der++}$	15.02	25.15	1.41	2.66	1.17	3.01	0.03	0.03	12.01	20.18	1.76	2.95	12.40	19.56	0.25	0.23
StemGNN$_{seq}$	14.10	23.51	1.92	1.63	1.30	3.10	0.24	0.28	17.31	25.43	6.31	7.23	11.71	17.77	1.54	1.01
StemGNN$_{mir}$	13.81	23.04	1.05	1.07	1.19	2.92	0.22	0.25	16.66	25.05	5.21	4.37	9.41	14.45	1.08	0.69
StemGNN$_{herd}$	13.78	23.01	0.98	1.03	1.17	2.92	0.22	0.25	16.51	25.03	5.21	4.23	9.17	13.97	1.03	0.65
StemGNN$_{er}$	13.49	22.59	0.36	0.42	1.00	2.73	0.04	0.05	16.43	24.83	5.19	4.13	8.38	12.95	0.96	0.97
StemGNN$_{der++}$	13.37	22.42	0.31	0.37	1.03	2.87	0.04	0.07	16.25	24.51	5.10	4.01	9.02	13.65	0.89	0.85
TCN$_{seq}$	13.89	22.12	2.57	4.77	0.93	2.71	0.12	0.13	12.71	20.46	2.98	4.35	3.81	3.37	0.26	0.33
TCN$_{mir}$	13.66	21.99	1.74	2.02	0.91	2.64	0.05	0.04	11.89	19.51	1.92	2.63	2.81	3.35	0.24	0.32
TCN$_{herd}$	13.62	21.97	1.63	1.89	0.90	2.64	0.04	0.03	11.85	19.34	1.70	2.53	2.77	3.31	0.23	0.31
TCN$_{er}$	13.21	21.85	1.58	1.63	0.92	2.64	0.03	0.02	11.68	18.98	1.54	2.29	2.67	3.18	0.18	0.22
TCN$_{der++}$	13.50	21.90	1.48	1.57	0.87	2.63	0.03	0.05	11.63	18.97	1.46	2.11	2.73	3.29	0.15	0.17
ESG$_{seq}$	14.62	23.83	3.72	5.39	1.75	2.93	0.92	0.83	16.57	25.07	5.34	6.21	8.68	13.95	1.06	1.57
ESG$_{mir}$	14.54	23.79	3.27	4.78	1.02	2.80	0.27	0.28	14.32	23.28	4.61	5.48	7.69	12.54	0.75	1.20
ESG$_{herd}$	14.53	23.79	3.26	4.71	1.01	2.78	0.14	0.27	14.31	23.17	4.48	5.39	7.61	12.25	0.71	1.15
ESG$_{er}$	13.07	22.11	1.62	2.93	1.15	2.71	0.32	0.25	14.21	23.01	4.37	5.21	7.21	11.65	0.61	0.89
ESG$_{der++}$	13.92	23.17	2.52	3.98	0.98	2.67	0.16	0.21	14.10	22.87	4.13	4.83	8.08	12.95	0.35	0.52
GTS$_{seq}$	15.20	27.09	3.23	7.32	1.20	3.19	0.19	0.48	14.44	23.41	3.51	4.63	4.90	6.71	0.56	0.61
GTS$_{mir}$	14.40	24.05	2.87	3.83	1.05	2.83	0.10	0.16	13.87	22.99	3.15	4.09	4.72	6.66	0.19	0.27
GTS$_{herd}$	14.33	23.55	2.82	3.78	1.03	2.69	0.05	0.10	13.78	22.89	3.08	4.05	4.71	6.66	0.13	0.20
GTS$_{er}$	14.54	24.65	1.96	3.58	0.94	2.67	0.08	0.05	13.38	22.45	2.95	3.97	4.41	6.40	0.13	0.19
GTS$_{der++}$	14.07	24.12	1.25	3.39	0.96	2.67	0.05	0.15	13.34	22.41	2.93	3.89	4.39	6.39	0.12	0.18
MTGNN$_{seq}$	14.42	23.94	3.34	5.13	1.14	2.76	0.32	0.30	12.87	20.66	3.51	4.80	7.63	12.44	0.59	0.67
MTGNN$_{mir}$	13.66	22.68	2.26	3.43	0.97	2.70	0.14	0.24	12.10	19.85	2.48	3.68	7.43	12.00	0.46	0.61
MTGNN$_{herd}$	13.55	22.55	2.13	3.35	0.96	2.69	0.13	0.23	11.98	19.78	2.42	3.61	7.43	11.99	0.45	0.61
MTGNN$_{er}$	13.95	22.29	2.03	3.02	0.95	2.74	0.05	0.16	11.30	18.67	1.51	2.15	6.38	10.13	0.35	0.53
MTGNN$_{der++}$	12.99	21.86	1.24	2.24	0.91	2.72	0.03	0.10	$\underline{11.27}$	$\underline{18.64}$	1.22	1.81	6.31	10.10	0.18	0.41
Autoformer$_{seq}$	18.19	30.22	2.37	4.23	3.19	6.47	0.82	1.69	16.79	24.78	1.83	1.98	3.83	5.38	0.35	0.24
Autoformer$_{mir}$	17.75	27.15	1.95	3.15	2.91	5.51	0.68	1.30	16.12	24.23	1.61	1.82	3.71	5.27	0.31	0.20
Autoformer$_{herd}$	17.72	26.93	1.89	3.02	2.89	5.34	0.67	1.23	16.01	24.12	1.58	1.78	3.71	5.26	0.31	0.20
Autoformer$_{er}$	16.07	26.82	1.36	2.54	2.54	5.02	0.13	0.23	15.66	23.88	1.57	1.58	3.65	5.26	0.24	0.17
Autoformer$_{der++}$	16.17	27.01	1.74	2.93	2.47	4.79	0.12	0.33	16.14	24.28	1.59	1.56	3.52	5.01	0.17	0.13
PatchTST$_{seq}$	13.78	23.21	2.39	4.33	1.05	3.02	0.17	0.27	14.13	22.93	1.50	1.57	3.46	4.93	0.43	0.35
PatchTST$_{mir}$	13.60	22.93	1.87	3.67	1.03	2.98	0.15	0.23	14.01	22.87	1.45	1.50	3.41	4.84	0.37	0.29
PatchTST$_{herd}$	13.53	22.75	1.54	2.77	0.99	2.90	0.13	0.19	13.94	22.51	1.38	1.47	3.38	4.79	0.31	0.26
PatchTST$_{er}$	13.20	22.26	1.13	1.73	0.95	2.84	0.10	0.15	13.90	21.99	1.31	1.44	3.35	4.75	0.25	0.20
PatchTST$_{der++}$	13.11	22.03	1.05	1.66	0.89	2.65	0.10	0.13	13.65	21.67	1.25	1.37	3.26	4.68	0.19	0.17
DLinear$_{seq}$	13.76	23.05	2.37	4.63	1.65	3.44	0.57	0.64	14.85	23.43	1.81	3.19	3.47	4.90	0.51	0.73
DLinear$_{mir}$	13.71	23.00	2.30	4.22	1.55	3.40	0.45	0.51	14.64	23.37	1.75	2.86	3.41	4.80	0.50	0.72
DLinear$_{herd}$	13.70	22.99	2.28	4.06	1.53	3.35	0.39	0.47	14.28	23.15	1.72	2.83	3.23	4.11	0.47	0.69
DLinear$_{er}$	13.64	22.97	2.15	3.87	1.50	3.30	0.33	0.41	14.26	22.85	1.71	2.82	2.96	3.92	0.38	0.59
DLinear$_{der++}$	13.50	22.74	2.06	3.73	1.42	3.20	0.28	0.33	14.12	22.15	1.65	2.73	2.93	3.88	0.33	0.52
TimesNet$_{seq}$	12.99	21.71	2.01	2.55	0.98	2.69	0.18	0.24	14.10	22.85	2.93	4.59	4.43	6.24	1.18	1.62
TimesNet$_{mir}$	12.92	21.67	1.75	2.21	0.95	2.66	0.16	0.20	12.92	21.06	1.92	2.48	4.35	5.98	1.06	1.23
TimesNet$_{herd}$	12.67	21.30	1.72	2.12	0.93	2.60	0.14	0.16	12.85	20.47	1.72	2.31	4.30	5.90	1.05	1.12
TimesNet$_{er}$	12.63	21.23	1.64	2.06	0.91	2.58	0.11	0.14	12.66	20.44	1.56	2.23	4.29	5.87	1.02	1.05
TimesNet$_{der++}$	12.50	21.00	1.51	1.85	0.90	2.55	0.09	0.13	12.53	20.24	1.54	2.21	4.17	5.93	1.01	1.04
iTransformer$_{seq}$	12.86	21.58	1.99	2.58	0.99	2.70	0.18	0.24	13.96	22.62	2.96	4.54	4.36	6.30	1.17	1.60
iTransformer$_{mir}$	12.80	21.50	1.77	2.23	0.95	2.67	0.16	0.20	13.05	20.85	1.90	2.46	4.31	6.04	1.07	1.24
iTransformer$_{herd}$	12.79	21.48	1.70	2.10	0.92	2.62	0.14	0.16	12.98	20.68	1.74	2.33	4.30	5.99	1.04	1.11
iTransformer$_{er}$	12.76	21.42	1.66	2.08	0.90	2.60	0.11	0.14	12.79	20.65	1.54	2.21	4.25	5.93	1.01	1.06
iTransformer$_{der++}$	12.63	21.21	1.53	1.83	0.89	2.57	0.09	0.13	12.66	20.44	1.52	2.19	4.21	5.87	1.00	1.05
OFA$_{seq}$	13.74	23.16	2.41	4.29	1.04	3.05	0.17	0.27	14.27	23.16	1.49	1.59	3.43	4.88	0.43	0.35
OFA$_{mir}$	13.67	22.98	1.56	2.80	0.98	2.93	0.13	0.19	13.80	22.28	1.39	1.49	3.41	4.83	0.31	0.26
OFA$_{herd}$	13.50	22.88	1.30	2.03	0.96	2.90	0.12	0.18	13.79	22.03	1.35	1.46	3.35	4.80	0.25	0.20
OFA$_{er}$	13.33	22.74	1.14	1.75	0.96	2.87	0.10	0.15	13.76	21.77	1.32	1.45	3.32	4.76	0.27	0.23
OFA$_{der++}$	12.98	22.25	1.04	1.68	0.88	2.68	0.10	0.13	13.51	21.89	1.26	1.36	3.29	4.63	0.19	0.17
SKI-CL$_{seq}$	13.81	23.16	2.52	4.25	0.90	2.63	0.09	0.14	12.41	19.37	2.54	3.78	3.72	5.26	2.02	3.10
SKI-CL$_{mir}$	12.94	21.94	1.55	2.76	0.89	2.64	0.08	0.12	11.88	19.09	1.93	3.08	3.20	4.33	0.52	0.46
SKI-CL$_{herd}$	12.91	21.80	1.43	2.55	0.87	2.65	0.07	0.12	11.78	19.04	1.89	2.95	3.11	4.18	0.23	0.45
SKI-CL$_{er}$	12.66	21.21	1.26	2.27	0.88	2.64	0.05	0.06	11.67	18.95	1.54	2.35	2.85	3.60	0.17	0.16
SKI-CL$_{der++}$	$\underline{12.61}$	$\underline{21.20}$	1.10	1.72	$\underline{0.85}$	$\underline{2.56}$	0.04	0.05	11.55	18.84	1.40	2.06	$\underline{2.33}$	$\underline{3.05}$	0.10	0.13
SKI-CL	**12.49**	**21.01**	0.93	1.60	**0.82**	**2.54**	0.02	0.02	**11.01**	**18.25**	1.21	1.78	**2.01**	**2.85**	0.05	0.03

Table 9: Standard Deviations for 12 Horizon Prediction.

Model	Traffic-CL				Solar-CL				HAR-CL ($\times 10^{-2}$)				Synthetic-CL($\times 10^{-2}$)			
	AP		AF		AP		AF		AP		AF		AP		AF	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTNet$_{\text{seq}}$	0.42	0.65	0.55	0.74	0.08	0.03	0.09	0.03	0.41	0.51	0.45	0.61	0.48	0.64	0.81	0.17
LSTNet$_{\text{mir}}$	0.21	0.27	0.42	0.64	0.05	0.05	0.08	0.04	0.32	0.43	0.39	0.58	0.47	0.41	0.42	0.11
LSTNet$_{\text{herd}}$	0.17	0.21	0.40	0.60	0.04	0.03	0.07	0.03	0.28	0.42	0.44	0.57	0.43	0.45	0.39	0.12
LSTNet$_{\text{rnd}}$	0.19	0.25	0.39	0.65	0.02	0.07	0.05	0.05	0.34	0.48	0.67	0.52	0.40	0.40	0.41	0.11
LSTNet$_{\text{der++}}$	0.07	0.13	0.15	0.21	0.03	0.05	0.04	0.04	0.28	0.37	0.29	0.45	0.30	0.35	0.49	0.61
STGCN$_{\text{seq}}$	1.09	1.91	1.30	2.25	0.07	0.09	0.11	0.12	0.35	0.47	0.37	0.31	1.02	1.62	0.60	0.90
STGCN$_{\text{mir}}$	1.71	1.49	1.90	2.71	0.08	0.22	0.03	0.14	0.25	0.43	0.31	0.23	0.27	0.60	0.39	0.57
STGCN$_{\text{herd}}$	1.54	2.10	1.72	2.50	0.11	0.30	0.07	0.31	0.28	0.27	0.20	0.32	0.38	0.59	0.36	0.63
STGCN$_{\text{rnd}}$	1.65	1.41	1.87	2.94	0.07	0.16	0.05	0.19	0.27	0.41	0.25	0.27	0.25	0.63	0.42	0.59
STGCN$_{\text{der++}}$	1.58	2.18	1.77	2.54	0.10	0.27	0.15	0.35	0.23	0.27	0.19	0.25	0.38	0.59	0.42	0.60
AGCRN$_{\text{seq}}$	0.44	0.58	0.45	0.88	0.33	0.26	0.24	0.35	0.61	0.51	0.41	0.37	0.26	0.79	0.13	0.35
AGCRN$_{\text{mir}}$	0.37	0.41	0.37	0.70	0.27	0.32	0.33	0.20	0.53	0.41	0.35	0.44	0.35	0.83	0.31	0.51
AGCRN$_{\text{herd}}$	0.49	0.99	0.38	0.92	0.36	0.26	0.43	0.17	0.48	0.52	0.55	0.59	0.28	0.95	0.36	0.47
AGCRN$_{\text{rnd}}$	0.31	0.36	0.34	0.64	0.15	0.35	0.31	0.13	0.58	0.47	0.39	0.40	0.30	0.89	0.25	0.50
AGCRN$_{\text{der++}}$	0.56	1.15	0.43	0.85	0.29	0.27	0.37	0.18	0.45	0.62	0.49	0.56	0.30	0.95	0.32	0.21
StemGNN$_{\text{seq}}$	0.22	0.18	0.77	1.24	0.16	0.13	0.18	0.19	0.25	0.47	0.23	0.31	0.95	0.44	0.38	0.58
StemGNN$_{\text{mir}}$	0.20	0.23	0.32	0.43	0.19	0.20	0.14	0.26	0.29	0.35	0.31	0.22	0.24	0.27	0.42	0.28
StemGNN$_{\text{herd}}$	0.25	0.43	0.72	0.85	0.15	0.19	0.15	0.13	0.28	0.33	0.26	0.24	0.45	0.70	0.40	0.43
StemGNN$_{\text{rnd}}$	0.16	0.16	0.21	0.22	0.17	0.18	0.14	0.22	0.23	0.39	0.27	0.21	0.26	0.26	0.49	0.32
StemGNN$_{\text{der++}}$	0.29	0.40	0.65	0.89	0.12	0.11	0.19	0.07	0.28	0.41	0.29	0.31	0.54	0.71	0.55	0.47
TCN$_{\text{seq}}$	0.11	0.15	0.13	0.17	0.05	0.07	0.08	0.13	0.01	0.03	0.05	0.08	0.05	0.05	0.25	0.44
TCN$_{\text{mir}}$	0.18	0.10	0.12	0.21	0.15	0.05	0.12	0.09	0.09	0.11	0.09	0.07	0.15	0.35	0.34	0.55
TCN$_{\text{herd}}$	0.09	0.08	0.09	0.10	0.06	0.03	0.05	0.13	0.08	0.13	0.07	0.12	0.17	0.27	0.16	0.25
TCN$_{\text{rnd}}$	0.10	0.11	0.11	0.13	0.04	0.04	0.10	0.10	0.04	0.16	0.02	0.02	0.21	0.37	0.29	0.51
TCN$_{\text{der++}}$	0.10	0.10	0.08	0.09	0.01	0.08	0.06	0.10	0.04	0.09	0.03	0.07	0.25	0.37	0.14	0.23
ESG$_{\text{seq}}$	0.48	0.70	0.65	0.87	0.62	0.67	0.76	0.83	0.31	0.30	0.35	0.31	0.10	0.65	0.35	0.33
ESG$_{\text{mir}}$	0.43	0.37	0.19	0.45	0.18	0.18	0.13	0.27	0.82	0.57	0.59	0.54	0.88	1.12	0.97	1.08
ESG$_{\text{herd}}$	0.45	0.52	0.41	0.48	0.16	0.30	0.20	0.36	0.73	1.32	1.01	1.33	0.51	0.56	0.55	0.41
ESG$_{\text{rnd}}$	0.43	0.49	0.44	0.35	0.19	0.29	0.25	0.42	1.05	1.88	1.43	2.60	0.45	0.63	0.53	0.43
ESG$_{\text{der++}}$	0.37	0.47	0.39	0.54	0.06	0.02	0.06	0.08	0.89	0.67	0.59	0.45	0.85	1.21	0.95	1.15
GTS$_{\text{seq}}$	0.11	0.27	0.10	0.21	0.14	0.43	0.19	0.55	0.29	0.34	0.38	0.51	0.57	0.80	0.56	0.63
GTS$_{\text{mir}}$	0.38	0.29	0.07	0.20	0.22	0.20	0.12	0.11	0.32	0.50	0.02	0.21	0.67	0.94	0.34	0.83
GTS$_{\text{herd}}$	0.20	0.52	0.28	0.70	0.10	0.23	0.07	0.10	0.55	0.75	0.40	0.70	0.54	0.57	0.21	0.40
GTS$_{\text{rnd}}$	0.18	0.29	0.20	0.39	0.03	0.14	0.06	0.08	0.58	0.81	0.51	0.65	0.43	0.45	0.28	0.42
GTS$_{\text{der++}}$	0.23	0.28	0.16	0.19	0.04	0.15	0.07	0.21	0.30	0.45	0.11	0.20	0.61	0.95	0.48	0.85
MTGNN$_{\text{seq}}$	0.31	0.35	0.15	0.30	0.03	0.06	0.05	0.07	0.10	0.15	0.10	0.24	0.41	0.64	0.24	0.10
MTGNN$_{\text{mir}}$	0.35	0.12	0.17	0.15	0.05	0.12	0.10	0.18	0.11	0.23	0.31	0.31	0.53	0.52	0.25	0.19
MTGNN$_{\text{herd}}$	0.11	0.29	0.33	0.31	0.13	0.15	0.11	0.22	0.09	0.39	0.26	0.36	0.60	0.58	0.17	0.27
MTGNN$_{\text{rnd}}$	0.10	0.12	0.12	0.17	0.03	0.05	0.11	0.17	0.13	0.19	0.14	0.27	0.34	0.57	0.23	0.13
MTGNN$_{\text{der++}}$	0.15	0.25	0.35	0.54	0.06	0.02	0.08	0.07	0.17	0.16	0.18	0.23	0.48	0.56	0.21	0.17
Autoformer$_{\text{seq}}$	0.58	1.11	1.51	2.88	0.11	0.27	0.20	0.23	0.12	0.31	0.27	0.24	0.65	0.41	0.32	0.51
Autoformer$_{\text{mir}}$	0.69	1.23	0.75	1.33	0.10	0.24	0.21	0.37	0.22	0.39	0.18	0.25	0.40	0.31	0.19	0.30
Autoformer$_{\text{herd}}$	0.77	1.43	0.67	1.20	0.36	0.36	0.23	0.41	0.12	0.13	0.14	0.12	0.90	0.61	0.71	0.70
Autoformer$_{\text{rnd}}$	0.86	1.42	0.74	1.18	0.18	0.31	0.24	0.31	0.17	0.27	0.35	0.31	0.46	0.31	0.25	0.43
Autoformer$_{\text{der++}}$	0.84	1.54	0.63	1.34	0.26	0.43	0.22	0.40	0.18	0.30	0.21	0.23	0.72	0.71	0.54	0.47
PatchTST$_{\text{seq}}$	0.27	0.21	0.25	0.46	0.24	0.17	0.12	0.21	0.24	0.10	0.26	0.15	0.16	0.54	0.40	0.54
PatchTST$_{\text{mir}}$	0.13	0.11	0.17	0.21	0.29	0.12	0.21	0.23	0.21	0.11	0.09	0.10	0.32	0.42	0.49	0.57
PatchTST$_{\text{herd}}$	0.17	0.13	0.19	0.17	0.12	0.08	0.12	0.14	0.24	0.10	0.27	0.13	0.10	0.54	0.40	0.49
PatchTST$_{\text{er}}$	0.146	0.25	0.19	0.18	0.16	0.09	0.09	0.17	0.07	0.09	0.08	0.09	0.14	0.11	0.26	0.35
PatchTST$_{\text{der++}}$	0.20	0.15	0.04	0.16	0.14	0.10	0.15	0.31	0.15	0.27	0.17	0.19	0.26	0.23	0.21	0.48
Dlinear$_{\text{seq}}$	0.21	0.19	0.23	0.36	0.18	0.21	0.17	0.18	0.16	0.11	0.26	0.26	0.14	0.25	0.39	0.53
Dlinear$_{\text{mir}}$	0.25	0.24	0.23	0.22	0.19	0.13	0.15	0.12	0.15	0.29	0.11	0.14	0.32	0.53	0.36	0.49
Dlinear$_{\text{herd}}$	0.15	0.17	0.17	0.25	0.14	0.22	0.13	0.26	0.12	0.37	0.18	0.27	0.25	0.14	0.21	0.43
Dlinear$_{\text{er}}$	0.23	0.17	0.18	0.12	0.11	0.14	0.16	0.23	0.14	0.31	0.27	0.23	0.22	0.46	0.56	0.62
Dlinear$_{\text{der++}}$	0.16	0.08	0.08	0.14	0.13	0.14	0.08	0.28	0.01	0.16	0.13	0.12	0.29	0.49	0.23	0.32
TimesNet$_{\text{seq}}$	0.17	0.14	0.23	0.22	0.09	0.13	0.15	0.08	0.16	0.21	0.11	0.14	0.32	0.53	0.36	0.49
TimesNet$_{\text{mir}}$	0.12	0.16	0.18	0.20	0.10	0.08	0.14	0.12	0.14	0.12	0.27	0.21	0.14	0.25	0.31	0.35
TimesNet$_{\text{herd}}$	0.14	0.19	0.17	0.25	0.14	0.20	0.09	0.26	0.15	0.37	0.09	0.32	0.23	0.14	0.21	0.43
TimesNet$_{\text{er}}$	0.08	0.10	0.12	0.20	0.18	0.13	0.12	0.20	0.09	0.15	0.07	0.24	0.11	0.21	0.18	0.36
TimesNet$_{\text{der++}}$	0.07	0.13	0.11	0.15	0.09	0.11	0.08	0.26	0.07	0.13	0.06	0.15	0.18	0.31	0.21	0.33
iTransformer$_{\text{seq}}$	0.18	0.17	0.20	0.34	0.19	0.19	0.17	0.22	0.12	0.11	0.25	0.24	0.14	0.28	0.43	0.57
iTransformer$_{\text{mir}}$	0.23	0.29	0.22	0.22	0.23	0.09	0.12	0.07	0.11	0.28	0.09	0.13	0.32	0.57	0.40	0.46
iTransformer$_{\text{herd}}$	0.11	0.12	0.17	0.27	0.17	0.22	0.16	0.21	0.15	0.33	0.20	0.28	0.29	0.16	0.25	0.45
iTransformer$_{\text{er}}$	0.21	0.20	0.18	0.16	0.08	0.09	0.19	0.25	0.19	0.29	0.25	0.26	0.19	0.46	0.58	0.65
iTransformer$_{\text{der++}}$	0.14	0.08	0.10	0.12	0.12	0.13	0.11	0.30	0.02	0.20	0.15	0.16	0.30	0.50	0.20	0.21
OFA$_{\text{seq}}$	0.14	0.10	0.27	0.23	0.13	0.09	0.18	0.13	0.12	0.19	0.13	0.14	0.29	0.51	0.40	0.52
OFA$_{\text{mir}}$	0.10	0.17	0.21	0.17	0.11	0.11	0.14	0.14	0.15	0.15	0.23	0.25	0.19	0.23	0.35	0.33
OFA$_{\text{herd}}$	0.19	0.24	0.19	0.22	0.17	0.18	0.11	0.21	0.13	0.41	0.08	0.34	0.21	0.17	0.25	0.45
OFA$_{\text{er}}$	0.10	0.12	0.15	0.16	0.21	0.13	0.09	0.20	0.05	0.18	0.03	0.23	0.10	0.17	0.13	0.38
OFA$_{\text{der++}}$	0.08	0.17	0.13	0.18	0.09	0.14	0.06	0.28	0.09	0.16	0.01	0.11	0.22	0.28	0.18	0.37
SKI-CL$_{\text{seq}}$	0.30	0.40	0.39	0.54	0.03	0.03	0.04	0.04	0.03	0.02	0.04	0.17	0.17	0.26	0.17	0.23
SKI-CL$_{\text{mir}}$	0.14	0.31	0.22	0.20	0.01	0.03	0.08	0.02	0.01	0.22	0.22	0.29	0.36	0.31	0.11	0.21
SKI-CL$_{\text{herd}}$	0.15	0.21	0.14	0.20	0.05	0.07	0.07	0.09	0.05	0.12	0.15	0.19	0.18	0.25	0.25	0.36
SKI-CL$_{\text{rnd}}$	0.04	0.10	0.04	0.16	0.06	0.06	0.05	0.10	0.02	0.09	0.23	0.21	0.16	0.22	0.35	0.49
SKI-CL$_{\text{der++}}$	0.04	0.07	0.12	0.17	0.04	0.05	0.08	0.11	0.04	0.12	0.15	0.13	0.25	0.29	0.15	0.11
SKI-CL	0.04	0.06	0.03	0.03	0.02	0.05	0.03	0.08	0.04	0.09	0.08	0.09	0.27	0.27	0.13	0.09