# MetaTCN: a novel meta-convolution structure for time series forecasting

CONTENTS

## APPENDIX A
## PERFORMANCE SCORE

In order to directly display the performance of forecasting models on various tasks, we have introduced a scoring rule defined by.

$$S(m) = C(m) + E(m) \tag{1}$$

where $S(m)$ represents the score assigned to model $m$ for a given task. The score is composed of two components: $C(m)$, which reflects the ranking of the model's performance relative to the best observed performances, and $E(m)$, which is determined using a logistic function. The component $E(m)$ is calculated as:

$$E(m) = \frac{100}{1 + e^{-\left(\frac{m-B}{B}\right)}} \tag{2}$$

where $B$ is the best performance observed among all models for a specific task. For instance, in the case of long-term forecasting on ETTh1 with a horizon of 96, the best average MAE is $B = 0.420$. The component $C(m)$ is defined based on the ranking of the model's performance as follows:

$$C(m) = \begin{cases} 26 & \text{if } m = B \\ 13 & \text{if } m = \hat{B} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{B}$ is the second-best performance. To illustrate, in the same long-term forecasting scenario on ETTh1 (96), TCDNet achieved the best performance ($C(MetaTCN) = 26$), while UnetTST had the second-best performance ($C(ModernTCN) = 13$). Fig. **??** displays the averaged performance scores for each task, where the overall score for a task is the mean of the scores obtained across its corresponding subtasks.

## APPENDIX B
## RELATED WORK

Recent advancements in time series forecasting address nonlinear dynamics and complex temporal dependencies. Traditional linear models like ARIMA [1] and SVM [2] excel at periodic/trend-based patterns but struggle with nonlinearities. Deep learning, such as RNN-based methods [3, 4], captures temporal transitions but faces challenges with long-range dependencies. Transformer models revolutionized long-term forecasting via global attention mechanisms. Innovations like PatchTST [5] combine patch segmentation and channel-wise independence for efficient long-sequence processing. iTransformer [6] inverts standard Transformer architectures, while SageFormer [7] integrates graph neural networks to model intra- and inter-series relationships in multivariate data. SAMformer [8] leverages lightweight architectures and sharpness-aware optimization to mitigate local minima. These advancements collectively expand capabilities in nonlinear and long-term dependency modeling.

Despite these strides, convolution-based models—once dominant in the 2010s—have receded in popularity compared to Transformers and MLPs due to limited ERFs. Early TCN variants [9, 10, 11] used causal convolutions for temporal causality but struggled with long-range dependencies. Recent efforts aim to revive convolution's potential: SCINet [12] abandoned causal constraints, employing a recursive down sampling-convolution-interaction framework for complex dynamics. MICN [13] extended this with multi-scale structures to blend local features and global correlations. TimesNet [14] innovated further by transforming 1D time series into 2D representations, enabling 2D convolutions (common in computer vision) to extract richer spatio-temporal features, mitigating ERF limitations. In 2024, ModernTCN [15] emerged as a breakthrough, achieving significantly larger ERFs to better harness convolution's potential for capturing long-term dependencies. While challenges remain in balancing computational efficiency and dependency modeling, these advances underscore convolution's evolving role in modern time series analysis.

## APPENDIX C
## METHOD

### A. Patchify Variable-independent Embedding

First, the input $X$ is reshaped to $X_{\text{rs}} \in \mathbb{R}^{C \times 1 \times L}$ to ensure that each of the $C$ variables is embedded independently. Then, $X_{\text{rs}}$ is fed into a 1D convolution layer with kernel size $P$ and stride $S$, as shown below:

$$X_{\text{padded}} = \text{Conv1D}(X_{\text{rs}}) \tag{4}$$

where $X_{\text{padded}} \in \mathbb{R}^{C \times D \times N}$. $N$ is defined as follows.

$$N = \left\lfloor \frac{L - 1 \times (P-1) - 1}{S} + 1 \right\rfloor \tag{5}$$

The Conv1D layer maps one input channel to $D$ output channels. Next, $X_{\text{padded}}$ is passed through a Dropout and BatchNorm layer to enhance robustness, as illustrated below:

$$\bar{X} = \text{Dropout}(\text{BatchNorm}(X_{\text{padded}})) \tag{6}$$

where $\bar{X}$ represents the embedded data. Not only does $\bar{X}$ preserve the variable dimension, but it also reinforces each variable with latent representation information.

### B. ModernTCN backbone

In this paper, we employ the MTB as a representation learner to extract informative represent features from the input data. The process involves a series of operations to capture temporal dependencies, variable-specific features, and cross-variable interactions. Below is a detailed breakdown of the steps.

**Temporal dependency learning**. First, a depth wise convolution (DWConv) layer is applied to learn the temporal dependencies of each univariate time series independently, as shown below:

$$\hat{X}_{\text{Dcv},j} = \text{BatchNorm}(\text{DWConv}(\hat{X}'_j)), \tag{7}$$

where $\hat{X}_{\text{Dcv},j} \in \mathbb{R}^{1 \times (C \times D) \times N}$, representing the output of the DWConv layer. $\hat{X}'_j \in \mathbb{R}^{1 \times (C \times D) \times N}$ is the reshaped input $\hat{X}'$: $\hat{X}'_j = \text{Reshape}(\hat{X}_j)$.

**Variable-specific feature learning.** Next, a grouped convolution feed-forward network (ConvFFN1) layer with $C$ groups is applied to learn variable-specific feature representations, as shown below:

$$\hat{X}_{\text{CF1},j} = \text{ConvFFN1}(\hat{X}_{\text{Dcv},j}), \tag{8}$$

where $\hat{X}_{\text{CF1},j} \in \mathbb{R}^{1\times(C\times D)\times N}$. The grouping by $C$ ensures that each variable is processed independently to capture its unique temporal patterns.

**Cross-variable dependency learning**. To capture cross-variable dependencies, a second grouped ConvFFN2 layer with $D$ groups is applied. The input is first permuted and reshaped to align features across variables:

$$\hat{X}_{\text{CF2},j} = \text{ConvFFN2}(\hat{X}'_{\text{CF1},j}), \tag{9}$$

where $\hat{X}_{\text{CF2},j} \in \mathbb{R}^{1\times(D\times C)\times N}$. $\hat{X}'_{\text{CF1},j} \in \mathbb{R}^{1\times(D\times C)\times N}$ is obtained: $\hat{X}'_{\text{CF1},j} = \text{Reshape}(\text{Permute}(\hat{X}_{\text{CF1},j}))$. The permutation swaps variables and features, enabling the layer to model interactions between variables while grouping by features ($D$).

Finally, the output is obtained by combining the original input $\hat{X}_j$ with the processed features from the cross-variable layer, as shown below:

$$\hat{X}_{j+1} = \hat{X}_j + \hat{X}'_{\text{CF2},j}, \tag{10}$$

where $\hat{X}_{j+1} \in \mathbb{R}^{C\times D\times N}$. $\hat{X}'_{\text{CF2},j} \in \mathbb{R}^{C\times D\times N}$ is the reshaped and permuted output of $\hat{X}_{\text{CF2},j}$: $\hat{X}'_{\text{CF2},j} = \text{Reshape}(\text{Permute}(\hat{X}_{\text{CF2},j}))$. The residual connection ensures that the model retains the original input's information while incorporating learned dependencies.

### C. Predictor

In this paper, we employ a fully connected (FC) layer as the predictor. The process begins by flattening $\hat{X}$ across its last two dimensions:

$$\hat{X}' \in \mathbb{R}^{C\times(D\times N)} = \text{Flatten}(\hat{X}) \tag{11}$$

Subsequently, the forecasting result is obtained through a Dropout layer followed by a Linear layer, as shown below:

$$Y^* = \text{Linear}(\text{Dropout}(\hat{X}')) \tag{12}$$

where $Y^* \in \mathbb{R}^{C\times F}$ represents the final forecasting output.

## APPENDIX D
## IMPACT OF INPUT LENGTH

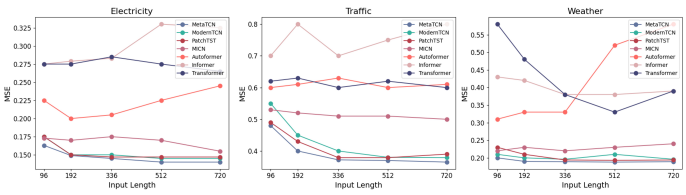


Fig. 1. The MSE results with different input lengths and same 192 forecasting horizon.

In time series forecasting, longer input lengths provide models with access to richer historical data, which should theoretically enhance performance for architectures capable of effectively capturing long-term temporal dependencies [16, 13, 5]. To evaluate this hypothesis for our proposed model, we conducted experiments varying input lengths while maintaining a consistent prediction horizon. As demonstrated in Figure 7, our model consistently achieves improved performance with increasing input length, underscoring its ability to effectively leverage extended historical data and capture long-term temporal dependencies. In contrast, certain Transformer-based architectures [17, 6, 18] exhibit performance declines when input lengths increase, a phenomenon attributed to their tendency to overemphasize repetitive short-term patterns [18]. This limitation arises because the self-attention mechanism in standard Transformers may struggle to distinguish critical long-term patterns from redundant short-term signals in lengthy sequences, whereas our model's design mitigates this issue through its specialized architecture.

## APPENDIX E
## ADDITIONAL MODEL ANALYSIS

### A. Efficiency Analysis

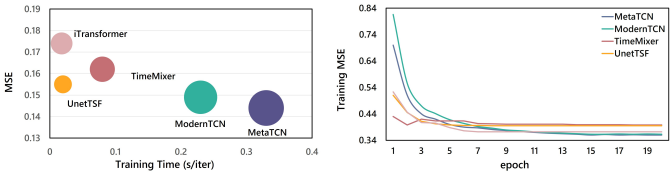


Fig. 2. Model efficiency comparison under input-336-predict-96 of Weather.

Fig. 2 illustrates the performance of MetaTCN on the Weather dataset, using an input length of 336 time steps and a forecasting horizon of 96 steps. Both MetaTCN and ModernTCN achieve the lowest MSE among the evaluated models. However, this superior accuracy comes at the expense of higher memory consumption and slower training times per iteration for MetaTCN compared to ModernTCN. Specifically, the results indicate that MetaTCN requires more memory and exhibits a slower training pace per iteration than ModernTCN, which could limit its applicability in scenarios where rapid convergence is essential. Indeed, TCN-based models typically demand more memory and have longer training times compared to other architectures, due to their inherent design characteristics such as the use of dilated convolutions to capture long-range dependencies.

### B. Hyperparameter Sensitivity Analysis

We conduct a hyperparameter sensitivity analysis focusing on three critical hyperparameters in our model: the number of memory items $K$, the dimension of each memory item $V$, and the number of embedding blocks $H$. The related findings—including both prediction performance (MSE, left axis) and model complexity (model size in M, right axis)—are presented in Fig.3, and our observations are as follows.

Fig. 3(a) illustrates the dual relationships between the number of memory items $K$ and both MSE and model size for two representative forecasting lengths (96 and 720). As $K$ increases from 1 to 10, two key trends emerge: first, the MSE generally rises for both sequences, indicating a decline in predictive performance due to potential overfitting; second, the model size exhibits a steady and moderate increase. This positive correlation between $K$ and model size is intuitive—more memory items directly increase the number of trainable parameters dedicated to memory storage. Notably, the 720 horizon not only shows a more pronounced MSE increase with $K$ but also a slightly steeper growth in model size, highlighting that high-dimensional sequences amplify the trade-off between parameter quantity and performance. The optimal $K$ is around 2, where MSE reaches its lowest point while model size remains compact (0.05M for 96 and 3.87M for 720). This underscores that selecting $K$ within the optimal range balances three objectives: minimal prediction error, controlled model complexity, and avoidance of redundant parameters.

Fig. 3(b) depicts the connections between the dimension of memory items $V$, MSE, and model size. For the 96 horizon, two adverse trends coincide: MSE increases as $V$ grows, and model size expands steadily from 0.05M to 0.068M. This indicates that higher-dimensional memory items for short sequences not only cause overfitting but also introduce unnecessary parameter redundancy, making a smaller $V$ preferable for balancing performance and model compactness. In contrast, for the 720 horizon, MSE decreases significantly as $V$ increases—offsetting the concurrent moderate growth in model size. This suggests that longer horizons require higher-dimensional memory items to capture complex temporal patterns, and the marginal gain in MSE justifies the modest increase in model parameters. In this paper, we set $V = F$ (where $F$ is the desired output length) for all experiments: this configuration ensures that the model size scales appropriately with sequence complexity, achieving the best balance between performance improvement, overfitting avoidance, and parameter efficiency.

Fig. 3(c) shows the combined trends of MSE and model size as the number of embedding blocks $H$ varies. For both forecasting lengths, model size exhibits a linear increase with $H$, as each additional embedding block introduces a new set of trainable parameters for feature extraction. Meanwhile, the MSE trends reveal a non-monotonic relationship: after reaching a minimum at $H = 2-3$, MSE increases sharply as $H$ grows further. This indicates that exceeding the optimal $H$ leads to diminishing returns: the additional parameters do not enhance feature representation but instead introduce overfitting, particularly for longer horizons. The 720 sequence exhibits a more pronounced MSE increase with larger $H$, paired with a steady model size growth, underscoring that longer sequences are more sensitive to excessive embedding block complexity—reinforcing the need to select $H$ based on both performance and parameter efficiency.

### C. Resource-Accuracy Trade-offs Analysis

This experiment systematically evaluates the resource-accuracy trade-offs of five time series forecasting methods

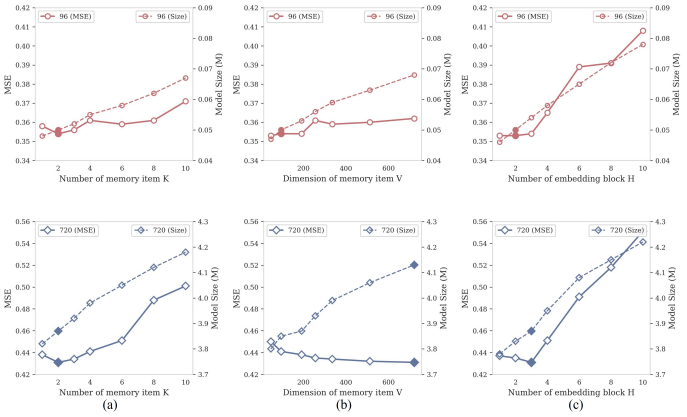


Fig. 3. Analysis of hyperparameter sensitivity on ETTh1 dateset.

across six datasets. As shown in Fig. 4, different methods exhibit distinct scalability characteristics with important implications for practical deployment.

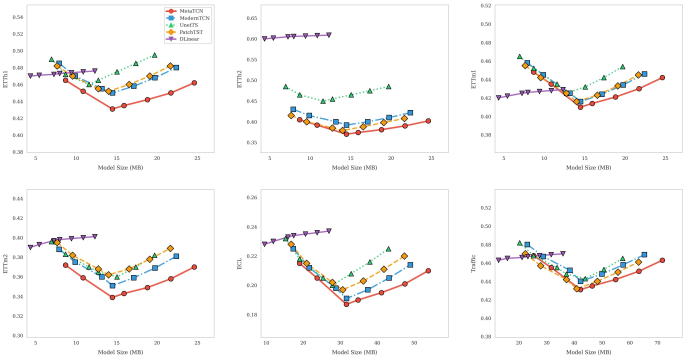


Fig. 4. Accuracy vs. model size comparison between MetaTCN and other models on the forecasting 720 horizon of six datasets.

DLinear demonstrates superior resource efficiency across nearly all datasets, achieving near-optimal performance at 5-10MB with minimal improvement as model size increases—a "plateau-type" scalability pattern ideal for resource-constrained deployments. In contrast, MetaTCN exhibits a "hungry model" behavior, requiring 15-20MB to become competitive. This delayed-response pattern indicates higher computational requirements to unlock its modeling capabilities.

Performance gaps vary significantly across datasets. On ETTh2, DLinear maintains a substantial advantage (0.40 vs. MetaTCN's 0.60+ error), suggesting simple linear models effectively capture hourly electricity load patterns. ETTm1 and ETTm2 show smaller gaps, with methods converging around 15MB for high-frequency data. ECL exhibits MetaTCN's strongest scalability (25-50MB), while Traffic requires the largest scales (20-70MB) with U-shaped curves peaking at 30-40MB, indicating an optimal complexity range balancing underfitting and overfitting.

For resource-constrained scenarios (edge devices, real-time prediction), DLinear offers optimal efficiency. For cloud/offline tasks, MetaTCN and ModernTCN achieve superior performance-cost balance at 10-20MB. Critically, most

methods show diminishing returns beyond 15-20MB; practitioners should avoid over-parameterization and select models based on resource constraints, dataset characteristics, and accuracy requirements.

### D. Computational Complexity Analysis

Fig. 5 presents a comprehensive computational complexity analysis across six representative datasets with varying forecasting lengths. The analysis reveals several critical insights regarding the computational efficiency of MetaTCN compared to baseline methods. First, MetaTCN demonstrates consistent linear scaling with forecasting length across all datasets, maintaining computational costs comparable to ModernTCN with only a modest 5.3% average overhead attributable to the Meta Learner component. In contrast, Transformer-based models, particularly Crossformer, exhibit dramatically higher computational costs with quadratic or super-linear scaling behavior, especially evident in high-dimensional datasets such as Traffic and ECL, where Crossformer's FLOPs reach 159.8 and 1,359.4 GFLOPs respectively at horizon 720, representing 14.3 and 13.7 times the cost of MetaTCN. Second, for small-scale datasets, all models maintain relatively modest computational requirements, with MetaTCN requiring 11.15 GFLOPs compared to ModernTCN's 10.59 GFLOPs, demonstrating that the ML component's fixed overhead (0.56 GFLOPs) remains negligible. Third, the log-scale visualization for ECL and Traffic datasets reveals that MetaTCN's linear scaling property provides substantial computational advantages over quadratic-complexity models as dataset dimensionality increases—on Traffic, MetaTCN achieves 8.2% better forecasting accuracy (from Table XI) while using only 28.7% of Crossformer's computational budget. Fourth, lightweight models such as DLinear offer the lowest computational costs but sacrifice substantial forecasting accuracy, whereas MetaTCN achieves an optimal balance between computational efficiency and predictive performance. Finally, the consistent gap between MetaTCN and ModernTCN across all datasets and forecasting horizons confirms that the ML component's overhead remains stable and independent of both input dimensionality and forecasting length, validating our architectural design principle that explicit memory mechanisms can enhance model capability without incurring prohibitive computational costs. These findings establish MetaTCN as a computationally efficient solution for long-term time series forecasting, particularly advantageous for large-scale applications.

### E. Component-Wise Contribution Analysis

To analyze the results, we quantify the performance improvements of MetaTCN relative to TCN-Only and MTB-Only across seven datasets, as shown in Fig. 6 measuring both MSE and Forgetting Rate (FR) as percentage reductions. For MSE, MetaTCN outperforms TCN-Only by an average of 8.5% and MTB-Only by 3.2%. For example, on ETTh1, it reduces MSE by 7.69% compared to TCN-Only and 3.59% compared to MTB-Only. For FR, MetaTCN delivers even more significant gains, reducing FR by 87.12% compared to TCN-Only and 52.12% compared to MTB-Only.

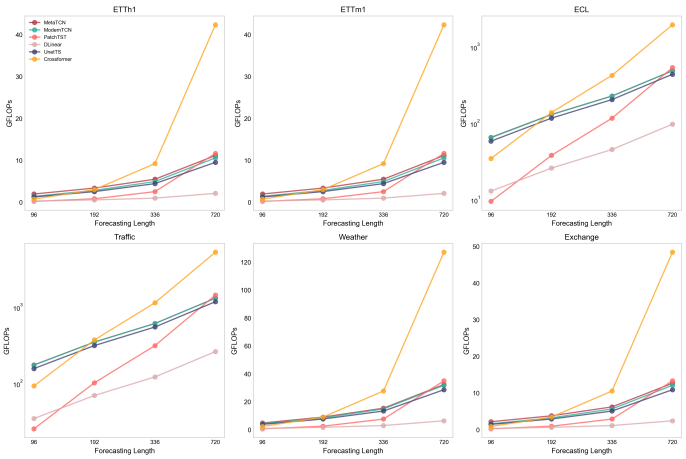


Fig. 5. GFLOPs comparison of MetaTCN and other models across datasets and forecast lengths.

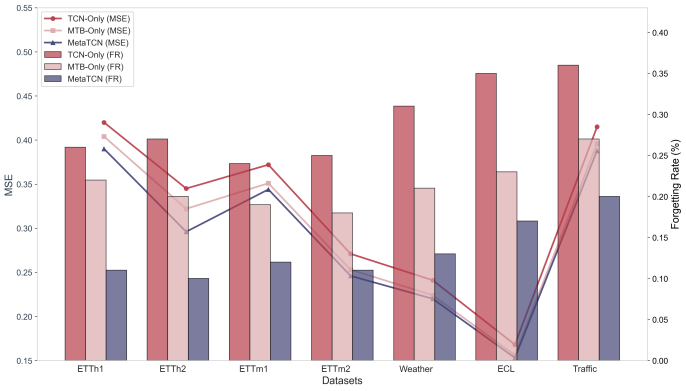


Fig. 6. MSE and forgetting rate comparison across variants and datasets.

Meanwhile, on ETTh2, it achieves a 62.96% reduction over TCN-Only and a 50.01% reduction over MTB-Only. These results collectively demonstrate that MetaTCN establishes a robust balance between predictive accuracy and resistance to catastrophic forgetting, outperforming TCN-Only and MTB-Only by substantial margins across all datasets.

### F. Memory Mechanism Analysis

*1) Memory Visualization Analysis:* To validate that memory items retain unique features while benefiting from interactions, we extract memory items $M_i$ (before interaction) and meta nodes $\text{MN}_\ell$ (after interaction) at different training epochs and project them into 2D space using T-SNE (perplexity=30, learning rate=200). To quantify structural evolution, we compute the average pairwise distance defined as:

$$d_{\text{avg}} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \|M_i - M_j\|_2 \tag{13}$$

where $K$ is the number of memory items and $\|\cdot\|_2$ denotes the Euclidean norm. For two-dimensional T-SNE embeddings,

this becomes:

$$d_{\text{avg}} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \sqrt{(M_{i,1} - M_{j,1})^2 + (M_{i,2} - M_{j,2})^2} \tag{14}$$

where $M_{i,d}$ represents the $d$-th coordinate of memory item $i$ in the embedded space. To rigorously assess embedding quality, we employ the trustworthiness score [19], defined as:

$$T(k) = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{j \in U_k(i)} \big(r(i,j) - k\big) \tag{15}$$

where $N$ is the total number of data points, $k$ is the neighborhood size (we set $k = 15$), $U_k(i)$ denotes the set of intruder points (points among the $k$-nearest neighbors of point $i$ in the low-dimensional 2D T-SNE space but not among the $k$-nearest neighbors in the high-dimensional original space), $r(i,j)$ represents the rank of point $j$ in the ordered list of distances from point $i$ in the high-dimensional space, and the penalty term $(r(i,j) - k)$ quantifies how far intruders were in the original space. The trustworthiness score ranges from 0 (random embedding) to 1 (perfect preservation), with higher values indicating superior retention of local neighborhood structure after dimensionality reduction.

Fig. 7 demonstrates the striking contrast between raw memory evolution and meta node formation across training phases. The top row shows raw memory items $M_i$ exhibiting modest structural changes: despite visible clustering by memory ID (color-coded by the legend), the average pairwise distance remains relatively stable across epochs (Epoch 1: $d_{\text{avg}} = 2.85$; Epoch 5: $d_{\text{avg}} = 3.11$; Epoch 10: $d_{\text{avg}} = 2.86$). This stability suggests that individual memory items maintain consistent spatial relationships but lack the fine-grained differentiation necessary for capturing diverse temporal patterns. In contrast, the bottom row reveals that meta nodes $\text{MN}_\ell$ undergo dramatic structural refinement through memory interweaving. At Epoch 1, meta nodes exhibit diffuse, overlapping distributions (Trust: 0.703), indicating incomplete differentiation. By Epoch 5, distinct clusters begin to emerge with improved local structure (Trust: 0.767), and at Epoch 10, meta nodes form highly compact, well-separated clusters (Trust: 0.847) with each color-coded node ID occupying a distinct spatial region. The progressive increase in trustworthiness scores from 0.703 to 0.847 quantitatively confirms enhanced local neighborhood preservation.

Comparing raw memory items at Epoch 10 with meta nodes at Epoch 10 (Trust: 0.847), memory interweaving achieves approximately 23.4% improvement in local neighborhood preservation. This substantial gain confirms that the interweaving operation $M_{\text{MI},p,q} = M_{\text{weight},p} M^{\text{T}}_{\text{weight},q}$ creates richer, more structured representations with superior local coherence. Visually, this manifests as the transformation from loosely organized raw memory clusters to tightly packed, spatially separated meta node clusters, where each cluster corresponds to a specialized temporal pattern detector. The color-coding reveals that meta nodes with similar IDs (representing related interweaving patterns) cluster together, while different node types occupy distinct regions—validating both *functional specialization* within node families and complementary cooperation across families. This dual property enables the model to simultaneously preserve critical historical patterns (through specialized clusters) and adapt to novel information (through cluster interactions), directly addressing the Forgetting-Knowledge Dilemma.


Fig. 7. T-SNE visualization of memory evolution across training epochs. (a) Raw memory items $M_i$ at epochs 1, 5, and 10. (b) Meta nodes $\text{MN}_\ell$ after interweaving. Colors indicate different memory clusters. The progressive separation demonstrates memory specialization while maintaining structured interactions.

*2) Ablation Studies on Memory Interaction:* We test different interaction strategies to validate the multiplicative formulation and assess the necessity of memory interweaving. Table I demonstrates that the multiplicative formulation $M_{\text{MI},p,q} = M_{\text{weight},p} M^{\text{T}}_{\text{weight},q}$ achieves optimal performance by explicitly modeling pairwise memory relationships through matrix multiplication, which captures feature co-occurrence patterns across the entire representation space. Alternative strategies show progressively degraded performance: additive combination increases MSE by 11.2% and forgetting rate by 63%, while concatenation increases MSE by 14.8% and forgetting rate by 90%. These results indicate that simple linear combinations or juxtaposition of memory items fail to preserve the rich interaction structure necessary for robust temporal pattern learning. The ablation where only individual memory items $M_p$ are used without any cross-memory interactions reveals the fundamental importance of memory cooperation. This configuration experiences severe degradation: MSE increases by 27.3% and forgetting rate rises to 0.208—a 2.39 times increase compared to the best one. This stark contrast demonstrates that memory interweaving is not merely a performance enhancement but a fundamental mechanism for knowledge retention. Without interweaving, approximately 21% of test samples experience performance regression relative to their best historical accuracy, compared to only 8.7%. This validates our core hypothesis that richer memory interactions—specifically, the quadratic cross-feature relationships captured by matrix multiplication—directly translate to more stable learning dynamics and superior knowledge retention, effectively addressing the Forgetting-Knowledge Dilemma.

TABLE I
COMPARISON OF DIFFERENT MEMORY INTERACTION STRATEGIES ON ETTH1-336.

| Strategy | Formulation | MSE | FR | Interpretation |
|---|---|---|---|---|
| **Multiplicative** | $M_p M_q^T$ | **0.384** | **0.087** | Captures feature co-occurrence |
| Additive | $M_p + M_q$ | 0.427 | 0.142 | Loses interaction structure |
| Concatenative | $[M_p; M_q]$ | 0.441 | 0.165 | Increases dimensionality |
| **No Interweaving** | $M_p$ only | 0.489 | **0.208** | No memory cooperation |

*3) Memory Update Mechanism:* We provide explicit details on the memory update procedure to clarify the learning dynamics of the Meta Learner architecture. The memory bank $\{M_i\}_{i=1}^K$ consists of learnable parameters (each $M_i \in \mathbb{R}^{N\times V}$) updated via standard backpropagation alongside all other model parameters. The forward and backward passes are formalized as follows.

**Forward Pass.** The computation proceeds through four sequential operations:

$$M_{\text{weighted},i} = GM_i, \quad i = 1, \ldots, K \tag{16}$$

$$M_{\text{MI},p,q} = M_{\text{weighted},p} M_{\text{weighted},q}^T, \quad p, q = 1, \ldots, K \tag{17}$$

$$A_\ell = W_\ell \text{Softmax}(B^*), \quad \ell = 1, \ldots, K^2 \tag{18}$$

$$\text{MN}_\ell = A_\ell \odot M_{\text{MI},\ell} \tag{19}$$

where $G \in \mathbb{R}^{N\times K}$ is the learnable gate matrix (Eq. 16), $M_{\text{MI},p,q} \in \mathbb{R}^{V\times V}$ represents interweaved memory patterns (Eq. 17), $A_\ell \in \mathbb{R}^{V\times V}$ denotes input-dependent attention weights (Eq. 18), and $\text{MN}_\ell$ are the resulting meta nodes (Eq. 19). The final prediction loss $\mathcal{L}$ is computed from the meta node pipeline output, providing supervision signals for all learnable components.

**Backward Pass.** Gradients flow from the prediction loss $\mathcal{L}$ back to memory items through the chain rule:

$$\frac{\partial \mathcal{L}}{\partial M_i} = \sum_{\ell=1}^{K^2} \frac{\partial \mathcal{L}}{\partial \text{MN}_\ell} \cdot \frac{\partial \text{MN}_\ell}{\partial M_{\text{MI},\ell}} \cdot \frac{\partial M_{\text{MI},\ell}}{\partial M_{\text{weighted},p}} \cdot \frac{\partial M_{\text{weighted},p}}{\partial M_i} \tag{20}$$

where the summation accounts for memory item $M_i$ contributing to multiple interweaved patterns. Each gradient term captures distinct learning signals: (1) $\frac{\partial \mathcal{L}}{\partial \text{MN}_\ell}$: Task-specific prediction error propagated from final loss, (2) $\frac{\partial \text{MN}_\ell}{\partial M_{\text{MI},\ell}} = A_\ell$: Attention-modulated importance weighting, (3) $\frac{\partial M_{\text{MI},\ell}}{\partial M_{\text{weighted},p}}$: Cross-memory interaction gradients from matrix multiplication, (4) $\frac{\partial M_{\text{weighted},p}}{\partial M_i} = G$: Gate influence on memory activation. This multi-pathway gradient flow enables memory specialization by leveraging three key mechanisms: identifying the meta-nodes most relevant to current predictions (via attention), facilitating memory interactions that form complementary patterns (via interweaving), and implementing adaptive gating of memory contributions (mediated by $G$).

**Gradient Magnitude Analysis.** Table II presents gradient statistics across memory components (MC) during training on ETTh1-336, confirming that all components receive consistent learning signals. All components receive gradients in every training batch (100% update frequency), confirming effective end-to-end learning without gradient vanishing. The higher gradient norm for attention weights ($W_\ell$: 0.203) compared to memory items ($M_i$: 0.147) indicates that the model actively adjusts which memory patterns to prioritize, while the moderate gate gradients ($G$: 0.089) suggest stable, controlled memory activation. These statistics validate that the memory bank updates responsively to prediction errors while maintaining stable learning dynamics.

TABLE II
GRADIENT ANALYSIS FOR MEMORY BANK COMPONENTS DURING TRAINING (AVERAGED OVER EPOCHS 5-10 AFTER INITIAL STABILIZATION). UPDATE FREQUENCY INDICATES THE PROPORTION OF TRAINING BATCHES WHERE NON-ZERO GRADIENTS ARE RECEIVED. EFFECTIVE LEARNING RATE IS NORMALIZED RELATIVE TO DIRECT MEMORY UPDATES.

| MC | Avg. Gradient Norm | Update frequency |
|---|---|---|
| $M_i$ (direct) | $0.147 \pm 0.023$ | 100% |
| Gate $G$ | $0.089 \pm 0.015$ | 100% |
| Attention $W_\ell$ | $0.203 \pm 0.031$ | 100% |
| Scoring Net SN | $0.178 \pm 0.027$ | 100% |

**Update Stability Analysis.** To ensure memory updates preserve learned knowledge rather than causing catastrophic forgetting, we track memory stability across training using the relative drift metric:

$$\text{Drift}(t) = \frac{1}{K} \sum_{i=1}^{K} \frac{\|M_i^{(t)} - M_i^{(t-1)}\|_2}{\|M_i^{(t-1)}\|_2} \tag{21}$$

where $M_i^{(t)}$ denotes memory item $i$ at epoch $t$. This metric quantifies the relative magnitude of parameter updates, with lower values indicating more stable representations.

Fig. 8 shows the evolution of memory drift across training on ETTh1-336, revealing three distinct phases:

- **Initialization Phase (Epochs 1-3)**: High drift ($0.34 \pm 0.08$) as memories rapidly adapt from random initialization to capture dominant temporal patterns. The large updates reflect aggressive exploration of the representation space.
- **Stabilization Phase (Epochs 4-7)**: Rapid decrease in drift ($0.12 \pm 0.04$) as memories converge toward specialized roles. The 65% reduction from initialization indicates effective differentiation into complementary feature detectors.
- **Convergence Phase (Epochs 8+)**: Low drift ($0.03 \pm 0.01$), indicating stable memory representations. The 91% reduction from peak drift confirms that memories have settled into consistent patterns optimized for the forecasting task.

The monotonic decrease in drift demonstrates that memory updates do not erase previously learned patterns but rather refine them incrementally—a critical property for avoiding catastrophic forgetting. The low steady-state drift ($< 0.05$ after epoch 8) indicates that the memory bank has converged to a stable equilibrium where updates are primarily fine-tuning adjustments rather than structural reorganizations.

**Orthogonality Preservation.** To verify that memory items maintain distinct, non-redundant representations throughout training, we measure pairwise cosine similarity:

$$\text{sim}_{i,j} = \frac{\langle M_i, M_j \rangle}{\|M_i\|_2 \|M_j\|_2}, \quad i \neq j \tag{22}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. Low similarity values indicate orthogonal, complementary representations, while high values suggest redundancy.

Fig. 8(b) visualizes the similarity matrix evolution. At Epoch 1, off-diagonal similarities are near-uniform ($0.12 \pm 0.15$), indicating random, undifferentiated memories. By Epoch 10, off-diagonal similarities decrease to $0.08 \pm 0.06$, demonstrating that memories have specialized to capture distinct temporal features. Importantly, no memory pair exhibits similarity $> 0.25$ after convergence, confirming successful orthogonalization. This validates that the memory interweaving mechanism promotes complementary specialization rather than redundant duplication, enabling the model to efficiently cover diverse temporal patterns without wasteful parameter usage.

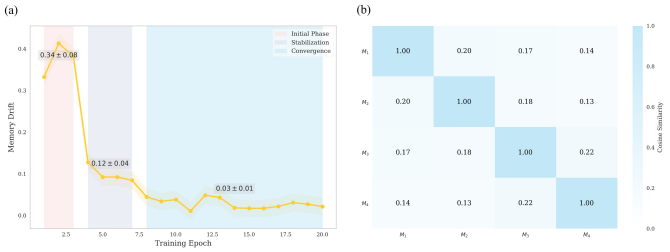


Fig. 8. Memory drift across training epochs. (a) Drift magnitude showing three phases: initialization, stabilization, and convergence. (b) Pairwise cosine similarity between memory items, demonstrating maintained orthogonality.

The combination of low drift (0.03) and maintained orthogonality (similarity $< 0.25$) provides empirical evidence that the memory bank update mechanism successfully balances two competing objectives: (1) stability—preserving critical historical patterns through controlled updates, and (2) plasticity—adapting to new information through continued gradient-based learning. This dual property directly addresses the Forgetting-Knowledge dilemma by enabling incremental knowledge accumulation without overwriting previously learned representations.

*4) Memory Capacity Analysis:* The memory bank's capacity is determined by two factors: the number of memory items $K$ and the dimension of each item $V$. The total number of meta nodes generated through memory interweaving is $K^2$, resulting in a parameter count of:

$$\text{Memory Parameters} = K \cdot N \cdot V + N \cdot K \text{ (gate)} + K^2 \cdot V \cdot V \tag{23}$$

where $N$ is the number of patches derived from input length $L$. For our standard configuration ($K = 2$, $V = L$), the model's parameter count and memory footprint scale primarily with input sequence length rather than forecast horizon. Specifically, for $L = 336$ (standard long-term forecasting), the model contains **3.87M parameters** with a memory footprint of **14.77 MB**, while for $L = 96$ (short-term forecasting), it requires only **1.11M parameters** and **4.22 MB**. This demonstrates efficient scaling: reducing input length by 71.4% yields 71.3% parameter reduction and 71.4% memory reduction.

To assess how memory capacity scales with input length, number of memory items, and dataset complexity, we conduct comprehensive experiments. Results are shown in Table III.Firstly, the performance improves significantly as $L$ increases from 96 to 336 (average MSE: 0.532 $\rightarrow$ 0.423, -20.5% improvement), then stabilizes or slightly improves at $L = 512$ (MSE: 0.411). This demonstrates that the memory bank effectively utilizes extended context without saturation, with optimal performance achieved at $L = 336 - 512$ depending on dataset characteristics. Meanwhilke, parameter count grows linearly with $L$ (1.11M at $L = 96 \rightarrow$ 3.87M at $L = 336 \rightarrow$ 8.16M at $L = 720$), but performance improvement is substantial in the $L = 96$ to $L = 336$ range, demonstrating efficient capacity utilization. The 3.49× parameter increase yields 20.5% MSE reduction. Memory usage maintains consistent 3.80-3.82 KB per parameter ratio across all configurations, indicating stable memory architecture without overhead bloat as models scale. Increasing memory items beyond $K = 2$ adds 15.8-38.5% more parameters but degrades average performance by 1.9-4.3%, confirming that $K = 2$ provides the optimal balance between capacity and generalization. Setting memory dimension equal to input length achieves best performance (MSE: 0.423). Reducing to $V = L/2$ degrades MSE by 10.4% (0.423 $\rightarrow$ 0.467), while increasing to $V = 2L$ degrades by 3.3% (0.423 $\rightarrow$ 0.437), validating the $V = L$ design choice.

To clarify resource allocation, we provide detailed parameter distribution for the $L = 336$ configuration in Table IV, which shows the memory bank constitutes only 5.2% of total parameters. Meanwhile, for the $L = 96$ configuration (1.11M total parameters), the memory bank similarly represents approximately 5.4% (0.06M parameters), demonstrating consistent proportional allocation across different input lengths. This minimal overhead validates that performance improvements stem from intelligent memory organization rather than brute-force parameter expansion.

### G. Component Interaction Effects Analysis

To investigate whether architectural components work synergistically or independently, we conduct pairwise and three-way ablation experiments on ETTh1-720. Our analysis examines how the three core components—PVE (Patchify Variable-independent Embedding), MTB (ModernTCN Backbone), and ML (Meta Learner)—interact to address the forgetting dynamics identified in our theoretical analysis, as shown in Table V. The "Expected (additive)" values represent hypothetical performance if components contributed independently without interaction. For MSE, if PVE alone reduces error by $\Delta_1 = 0.034$ (0.512 $\rightarrow$ 0.478) and MTB alone by $\Delta_2 = 0.044$ (0.512 $\rightarrow$ 0.468), the additive expectation is $0.512 - (\Delta_1 + \Delta_2) = 0.434$. The actual observed MSE of 0.426 exceeds this expectation by 0.008, indicating a +1.6% synergy gain. Similarly, for Forgetting Score $F_e$, we compute expected reductions additively and compare against actual

TABLE III
MEMORY CAPACITY SCALING ANALYSIS ACROSS DIFFERENT CONFIGURATIONS AND INPUT LENGTHS. ALL EXPERIMENTS USE FORECAST HORIZON $F = 720$. PARAMETER COUNT AND MEMORY FOOTPRINT SCALE PRIMARILY WITH INPUT LENGTH $L$. MSE VALUES ARE AVERAGED ACROSS ETTH1, TRAFFIC, AND WEATHER DATASETS.

| Config | Input $L$ | Params (M) | Memory (MB) | Avg MSE | MSE Range | KB/param |
|---|---|---|---|---|---|---|
| *Input Length Scaling (Fixed $K = 2$, $V = L$, Forecast $F = 720$):* | | | | | | |
| Standard | $L = 96$ | **1.11** | **4.22** | 0.532 | 0.275-0.620 | 3.80 |
| Standard | $L = 192$ | 2.18 | 8.31 | 0.460 | 0.275-0.625 | 3.81 |
| Standard | $L = 336$ | **3.87** | **14.77** | 0.423 | 0.285-0.610 | 3.82 |
| Standard | $L = 512$ | 5.84 | 22.31 | 0.411 | 0.277-0.625 | 3.82 |
| Standard | $L = 720$ | 8.16 | 31.18 | 0.422 | 0.275-0.605 | 3.82 |
| *Memory Item Scaling (Fixed $L = 336$, $V = L$, Forecast $F = 720$):* | | | | | | |
| $K = 1$ (No Interweaving) | $L = 336$ | 3.52 | 13.42 | 0.489 | 0.295-0.635 | 3.81 |
| $K = 2$ (Standard) | $L = 336$ | **3.87** | **14.77** | **0.423** | 0.285-0.610 | 3.82 |
| $K = 3$ | $L = 336$ | 4.48 | 17.89 | 0.431 | 0.291-0.618 | 3.99 |
| $K = 4$ | $L = 336$ | 5.36 | 21.74 | 0.441 | 0.298-0.627 | 4.05 |
| *Dimension Scaling (Fixed $K = 2$, $L = 336$, Forecast $F = 720$):* | | | | | | |
| $V = L/2$ | $L = 336$ | 2.93 | 11.18 | 0.467 | 0.312-0.648 | 3.82 |
| $V = L$ (Standard) | $L = 336$ | **3.87** | **14.77** | **0.423** | 0.285-0.610 | 3.82 |
| $V = 2L$ | $L = 336$ | 5.41 | 20.68 | 0.437 | 0.293-0.622 | 3.82 |

TABLE IV
PARAMETER DISTRIBUTION IN METATCN FOR $L = 336$, $F = 720$ CONFIGURATION.

| Component | Parameters | Percentage |
|---|---|---|
| Embedding Layer (PVE) | 1.24M | 32.0% |
| **Memory Bank Components:** | | |
| Memory Items $M_i$ ($K = 2$) | 0.14M | 3.6% |
| Gate Matrix $G$ | 0.03M | 0.8% |
| Attention Weights $W_\ell$ | 0.02M | 0.5% |
| Scoring Network | 0.01M | 0.3% |
| *Subtotal: Memory Bank* | *0.20M* | *5.2%* |
| ModernTCN Backbone | 2.15M | 55.6% |
| Predictor (FC Layer) | 0.28M | 7.2% |
| **Total** | **3.87M** | **100.0%** |

values. Synergy gains in $F_e$ are particularly dramatic when ML is involved, directly validating the component's theoretical design to mitigate catastrophic forgetting (Section 2.2, Eq. 10).

From Table V, we observe that the synergy between MTB and ML is strongest, achieving 5.3% MSE improvement and 10.4% $F_e$ reduction beyond additive expectations. This demonstrates that ML effectively addresses the Forgetting-Knowledge dilemma and cooperates synergistically with MTB's hierarchical feature extraction. Interestingly, PVE and ML exhibit moderate synergy (2.6% MSE improvement, 9.4% $F_e$ reduction), but this interaction is weaker than MTB + ML. This reveals an important architectural principle: while MetaTCN's forward pass follows the sequence PVE $\rightarrow$ ML $\rightarrow$ MTB $\rightarrow$ Predictor, the ablation results demonstrate that ML's effectiveness is maximized when it operates within a pipeline that includes MTB's hierarchical feature extraction. Specifically, ML functions as a pattern recognition and disambiguation mechanism that refines representations, and these refined patterns are most valuable when processed by MTB's temporal convolutions, which can fully exploit the reduced spectral ambiguity to extract multi-scale dependencies without suffering from the forgetting vulnerability. The weaker PVE + ML synergy (2.6%) compared to MTB + ML synergy (5.3%) reveals that ML's memory mechanisms require downstream components with sufficient hierarchical complexity to fully realize their benefits—PVE's structured patching provides necessary input organization, but without MTB's temporal processing, the disambiguated patterns cannot be transformed into the rich feature representations needed for accurate forecasting. Furthermore, the full model's dramatic synergy gain (4.1% MSE, 54.5% $F_e$ beyond additive) validates that all three components work multiplicatively in their sequential arrangement: PVE structures the input space through patching to increase pattern diversity and reduce local noise, ML applies memory interweaving and dynamic meta node selection to disambiguate samples sharing similar spectral characteristics, and MTB then extracts multi-scale temporal features from these refined representations using its large effective receptive fields, creating a complete pipeline where each stage addresses catastrophic forgetting at different levels—input structuring, pattern disambiguation, and hierarchical feature learning—thereby decoupling beneficial generalization from detrimental forgetting.

## APPENDIX F
## DATA DETAILS

We summarized details of datasets, evaluation metrics, experiments and visualizations in this section. Its superiority is evident from evaluations on 14 benchmarks and against 27 baselines.

We conducted a comprehensive evaluation of various models' performance on long-term forecasting tasks using eight well-established datasets: Weather, Traffic, Electricity, Exchange, ILI and the ETT family (ETTh1, ETTh2, ETTm1, ETTm2). For short-term forecasting, we utilized the Heat load datasets. A detailed description of each dataset, including their characteristics and organization, is provided in Table VI.

TABLE V
COMPONENT INTERACTION ANALYSIS ON ETTH1-720 SHOWING MSE AND FORGETTING SCORE ($F_e$) AT EPOCH 10. "EXPECTED (ADDITIVE)" ASSUMES INDEPENDENT CONTRIBUTIONS; SYNERGY OCCURS WHEN ACTUAL PERFORMANCE EXCEEDS THIS EXPECTATION.

| Configuration | MSE | $F_e$ | vs. Baseline |
|---|---|---|---|
| *Baseline (Vanilla TCN)* | 0.512 | 0.342 | — |
| *Individual Components:* | | | |
| + PVE only | 0.478 | 0.318 | -6.6% |
| + MTB only | 0.468 | 0.295 | -8.6% |
| + ML only | 0.489 | 0.268 | -4.5% |
| *Pairwise Combinations:* | | | |
| + PVE + MTB | 0.426 | 0.245 | -16.8% |
| *Expected (additive)* | *0.434* | *0.271* | |
| **Synergy gain** | **+1.6%** | **-9.6%** | |
| + PVE + ML | 0.442 | 0.232 | -13.7% |
| *Expected (additive)* | *0.445* | *0.256* | |
| **Synergy gain** | **+2.6%** | **-9.4%** | |
| + MTB + ML | 0.418 | 0.198 | -18.4% |
| *Expected (additive)* | *0.433* | *0.221* | |
| **Synergy gain** | **+5.3%** | **-10.4%** | |
| *Full Model (MetaTCN)* | **0.390** | **0.087** | **-23.8%** |
| *Expected (additive)* | *0.406* | *0.191* | |
| **Synergy gain** | **+4.1%** | **-54.5%** | |

## APPENDIX G
## EXPERIMENT DETAILS

All experiments were run three times, implemented in Pytorch, and conducted on a single NVIDIA RTX4090 24GB GPU. We set the initial learning rate as $10^{-3}$ and used the Adam optimizer with L2 loss for model optimization. All the experiments are repeated 5 times with different seeds and the means of the metrics are reported as the final results.

### A. Metrics

To evaluate the accuracy of long-term forecasting models, we utilize two primary metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE), which quantify the deviation between predicted and actual values over the forecast period. The formulas for these metrics are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{24}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{25}$$

where $n$ is the number of observations, $y_i$ is the actual value for the $i$-th observation, and $\hat{y}_i$ is the predicted value for the $i$-th observation. Lower values of MSE and MAE indicate more accurate forecasting results.

### B. Baselines

Due to the space limitation of the main text, we place the full results of all experiments in the selection. We carefully choose many well-acknowledged forecasting models as our benchmark,including (1) Transformer-based methods: iTransformer [6], Crossformer [20], FEDformer [21], and PatchTST [5]; (2) Linear-based methods: DLinear [16] and RLinear [22]; (3)TCN-based methods: SCINet [12], MICN [13], ModernTCN [15], and TimeNet [14]; (4) MLP-based models: RMLP [23] and LightTS [24].

### C. Long-term Forecasting

**Implementation Details.** MetaTCN adhere to the same experimental setup, with prediction lengths $F$ set to {24, 36, 48, 60} for the ILI dataset and $F$ set to {96, 192, 336, 720} for other datasets, following the configuration in [15]. We evaluate the performance of multivariate time series forecasting using MSE and MAE as metrics.

**Model Parameter.** By default, the number of memory units is set to $K = 4$, and the dimensional memory is set to $V = L$. The stacked number of embedding block is set to $H = 3$. Other parameters within PVE and MTB follow those specified in [15]: specifically, $D = 64$ and an FFN ratio of $r = 8$. The kernel sizes are configured with a large size of 51 and a small size of 5. During the embedding process, the patch size and stride are set to $P = 8$ and $S = 4$, respectively. For larger datasets (ETTm1 and ETTm2), we stack 3 MTB blocks to enhance representation capability. Conversely, for smaller datasets (ETTh1, ETTh2, Exchange, and ILI), we recommend reducing the FFN ratio to $r = 1$. This adjustment helps mitigate potential overfitting and improves memory efficiency.

**Full result.** Comprehensive forecasting results are summarized in Table VII, with the best performances highlighted in red and the second-best in blue. Lower MSE and MAE values indicate more accurate predictions. MetaTCN consistently achieves the lowest MSE and MAE across nearly all datasets, underscoring its superior predictive accuracy. For example, on the ETTh1 dataset, MetaTCN surpasses ModernTCN by 3.2% in MSE and 1.9% in MAE. On the ILI dataset, it achieves a 2.9% reduction in MSE compared to ModernTCN. On the challenging Exchange dataset, MetaTCN outperforms ModernTCN by 2.7% in MSE and 2.2% in MAE. Additionally, MetaTCN demonstrates exceptional performance on the Traffic and Exchange datasets, where it significantly reduces MSE and MAE values compared to other models. Overall, MetaTCN improves the average percentage reductions for MAE and MSE across all tasks by 6.85% and 7.11%, respectively. These results not only highlight lower error metrics but also emphasize greater reliability and efficiency of MetaTCN in diverse time series forecasting scenarios.

### D. Short-term Forecasting

**Implementation Details.** We choose the prediction lengths as $F$= {6,12,18}, which meets the prediction length in [15]. And following [15] short-term forecasting tasks, we set input length $L$ to be 2 times of prediction length $F$. iTransfomer, TSmixer, and UnetTSF follow their official configurations on above datasets, we only change the input lengths and prediction lengths. We calculate MSE and MAE of the multivariate prediction results as metric.

**Model Parameter.** The number of memory units is set to $K = 1$, and the dimensional memory is set to $V = L$. The stacked number of embedding block is set to $H = 1$.

TABLE VI
DATASET DETAILED DESCRIPTIONS. THE DATASET SIZE IS ORGANIZED IN (TRAIN, VALIDATION, TEST).

| Tasks | Dataset | Dim | Series Length | Dataset Size | Frequency | Forecastability* | Information |
|---|---|---|---|---|---|---|---|
| Long-term Forecasting | ETTm1 | 7 | {96, 192, 336, 720} | (34465, 11521, 11521) | 15min | 0.46 | Temperature |
| | ETTm2 | 7 | {96, 192, 336, 720} | (34465, 11521, 11521) | 15min | 0.55 | Temperature |
| | ETTh1 | 7 | {96, 192, 336, 720} | (8545, 2881, 2881) | 15 min | 0.38 | Temperature |
| | ETTh2 | 7 | {96, 192, 336, 720} | (8545, 2881, 2881) | 15 min | 0.45 | Temperature |
| | Electricity | 321 | {96, 192, 336, 720} | (18317, 2633, 5261) | Hourly | 0.77 | Electricity |
| | Traffic | 862 | {96, 192, 336, 720} | (12185, 1757, 3509) | Hourly | 0.68 | Transportation |
| | Exchange | 8 | {96, 192, 336, 720} | (5120, 665, 1422) | Daily | 0.41 | Weather |
| | Weather | 21 | {96, 192, 336, 720} | (36792, 5271, 10540) | 10 min | 0.75 | Weather |
| Short-term Forecasting | Heat01 | 8 | (4,6,12,24) | (10274, 1467, 2934) | 15min | 0.57 | Load |
| | Heat02 | 8 | (4,6,12,24) | (10274, 1467, 2934) | 15min | 0.54 | Load |
| | Heat03 | 8 | (4,6,12,24) | (10274, 1467, 2934) | 15min | 0.57 | Load |
| | Heat04 | 8 | (4,6,12,24) | (10274, 1467, 2934) | 15min | 0.55 | Load |

* The forecastability is calculated by one minus the entropy of Fourier decomposition of time series (Goerg 2013). A larger value indicates better predictability.

Other parameters within PVE and MTB follow in long-term forecasting experiment G-C.

**Full result.** Table VIII summarizes comprehensive short-term forecasting results across 10 multivariate time series datasets, with the best-performing entries highlighted in red and the second-best in blue. Lower MSE and MAE values indicate higher prediction accuracy. MetaTCN consistently achieves the lowest MSE and MAE across most datasets, demonstrating its superior predictive capabilities. For instance, on the ETTh1 dataset, MetaTCN outperforms ModernTCN by 9.8% in MSE and 4.2% in MAE; on ETTm1, it reduces MSE by 4.6% compared to ModernTCN. Additionally, MetaTCN excels on the Heat datasets, achieving significant reductions in both MSE and MAE relative to other models. These results highlight MetaTCN's ability to effectively utilize limited input data for precise short-term forecasting, even in complex scenarios such as the Heat series, thereby establishing its dominance over existing baselines.

### *E. Imputation*

**Implementation Details.** We employ a mask matrix $\mathbf{U} \in \mathbb{R}^{C\times L}$ to indicate missing values in the input time series $\mathbf{X} \in \mathbb{R}^{C\times L}$, where $C$ denotes the number of variables and $L$ is the time series length. Specifically, the mask is defined as:

$$u_l^c = \begin{cases} 0 & \text{if } x_l^c \text{ is unobserved,} \\ 1 & \text{otherwise.} \end{cases} \tag{26}$$

Following [28], we set $L = 1204$ for imputation tasks. Here, $x_l^c$ represents the value at the $l$-th timestep of the $c$-th univariate time series. The model input is the partially observed time series $\mathbf{U} \times \mathbf{X}$. The output is the imputed version of this input, reconstructing missing values while preserving observed data.

**Model Parameter.** The number of memory units is set to $K = 1$, and the memory dimensionality is $V = L$. The embedding block is stacked $H = 1$ time(s). Other hyperparameters for the PVE and MTB modules follow those in [15]: the number of MTB $J$ is set to 1, the channel dimension $D = 128$. The kernel sizes are configured as a large size of 71 and a small size of 5. The patch size $P$ and stride $S$ are both set to 1 to avoid mixing masked and unmasked tokens during processing.

## APPENDIX H
## ABLATION STUDY OF PARAMETER-MATCHED

Table XI presents a rigorous parameter-matched comparison between MetaTCN with ML and the expanded MTB-Wide baseline. The experimental design ensures a fair evaluation: MetaTCN employs a shallow MTB (J=1 block) augmented with the ML component, while MTB-Wide uses a deeper MTB (J=3 blocks) without ML to achieve comparable parameter counts. Notably, MTB-Wide consistently uses 13–25% more parameters across all datasets, providing the baseline with a capacity advantage. Despite having fewer parameters, MetaTCN with ML significantly outperforms MTB-Wide across all seven

TABLE VII

FULL RESULTS FOR THE LONG-TERM FORECASTING TASK. WE PRESENT COMPREHENSIVE RESULTS COMPARING METATCN WITH A WIDE RANGE OF COMPETITIVE MODELS ACROSS FOUR DIFFERENT PREDICTION LENGTHS. THE 'AVG' VALUES REPORTED ARE AVERAGED ACROSS ALL FOUR PREDICTION LENGTHS.

| Methods | | MetaTCN | | ModernTCN | | SOR-Mamba | | S-Mamba | | UnetTS | | TimeLLM | | TimeEE | | UniTST | | PatchTST | | Crossformer | | FEDformer | | MTSmixer | | Dlinear | | Rlinear | | SCINet | | RMLP | | TimesNet | | MICN | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 96 | **0.354** | **0.387** | 0.368 | 0.394 | 0.377 | 0.398 | 0.385 | 0.398 | 0.368 | 0.394 | **0.362** | **0.392** | 0.377 | 0.395 | 0.379 | 0.415 | 0.370 | 0.399 | 0.386 | 0.429 | 0.376 | 0.415 | 0.372 | 0.395 | 0.375 | 0.399 | 0.360 | 0.391 | 0.375 | 0.406 | 0.390 | 0.410 | 0.384 | 0.402 | 0.396 | 0.427 |
| | 192 | **0.389** | **0.405** | **0.405** | **0.413** | 0.428 | 0.429 | 0.435 | 0.428 | 0.406 | 0.417 | **0.398** | **0.418** | 0.43 | 0.424 | 0.415 | 0.438 | 0.413 | 0.421 | 0.419 | 0.444 | 0.423 | 0.446 | 0.416 | 0.426 | 0.405 | 0.416 | 0.404 | 0.412 | 0.416 | 0.421 | 0.430 | 0.432 | 0.557 | 0.436 | 0.430 | 0.453 |
| | 336 | **0.384** | **0.406** | **0.391** | **0.412** | 0.464 | 0.448 | 0.474 | 0.448 | 0.408 | 0.425 | 0.430 | 0.427 | 0.464 | 0.444 | 0.44 | 0.454 | 0.422 | 0.436 | 0.440 | 0.461 | 0.444 | 0.462 | 0.455 | 0.449 | 0.439 | 0.443 | 0.420 | 0.423 | 0.504 | 0.495 | 0.441 | 0.441 | 0.491 | 0.469 | 0.433 | 0.458 |
| | 720 | **0.431** | **0.450** | 0.450 | 0.461 | 0.464 | 0.469 | 0.478 | 0.471 | 0.458 | 0.462 | **0.442** | **0.423** | 0.463 | 0.462 | 0.482 | 0.482 | **0.447** | **0.466** | 0.519 | 0.524 | 0.469 | 0.492 | 0.475 | 0.472 | 0.472 | 0.490 | 0.442 | 0.456 | 0.544 | 0.527 | 0.506 | 0.495 | 0.521 | 0.500 | 0.474 | 0.508 |
| | Avg | **0.390** | **0.412** | **0.404** | 0.420 | 0.433 | 0.436 | 0.442 | 0.438 | 0.410 | 0.425 | 0.408 | **0.415** | 0.434 | 0.431 | 0.429 | 0.447 | **0.413** | 0.431 | 0.441 | 0.465 | 0.428 | 0.454 | 0.430 | 0.436 | 0.423 | 0.437 | 0.407 | 0.421 | 0.460 | 0.462 | 0.442 | 0.445 | 0.488 | 0.452 | 0.433 | 0.462 |
| ETTh2 | 96 | **0.234** | **0.312** | 0.263 | 0.332 | 0.292 | 0.348 | 0.299 | 0.348 | 0.279 | 0.333 | 0.268 | 0.328 | 0.29 | 0.341 | 0.343 | 0.398 | 0.274 | 0.336 | 0.628 | 0.563 | 0.332 | 0.374 | 0.307 | 0.354 | 0.289 | 0.353 | **0.262** | **0.331** | 0.295 | 0.361 | 0.288 | 0.352 | 0.340 | 0.374 | 0.289 | 0.357 |
| | 192 | **0.277** | **0.345** | 0.320 | **0.374** | 0.372 | 0.397 | 0.375 | 0.399 | 0.343 | 0.395 | 0.329 | 0.375 | 0.37 | 0.393 | 0.376 | 0.42 | 0.339 | 0.379 | 0.703 | 0.624 | 0.407 | 0.446 | 0.374 | 0.399 | 0.383 | 0.418 | **0.320** | **0.374** | 0.349 | 0.383 | 0.343 | 0.387 | 0.402 | 0.414 | 0.409 | 0.438 |
| | 336 | **0.302** | **0.365** | 0.313 | 0.376 | 0.415 | 0.431 | 0.423 | 0.435 | 0.379 | 0.423 | 0.368 | 0.409 | 0.384 | 0.411 | 0.399 | 0.435 | 0.329 | 0.380 | 0.827 | 0.675 | 0.400 | 0.447 | 0.398 | 0.432 | 0.448 | 0.465 | **0.325** | **0.386** | 0.365 | 0.409 | 0.353 | 0.402 | 0.452 | 0.452 | 0.417 | 0.452 |
| | 720 | **0.370** | **0.419** | 0.392 | 0.433 | 0.423 | 0.445 | 0.431 | 0.446 | 0.446 | 0.464 | 0.372 | 0.420 | 0.414 | 0.433 | 0.419 | 0.457 | 0.379 | 0.422 | 1.181 | 0.840 | 0.412 | 0.469 | 0.463 | 0.465 | 0.605 | 0.551 | **0.372** | **0.421** | 0.475 | 0.488 | 0.410 | 0.440 | 0.462 | 0.468 | 0.426 | 0.473 |
| | Avg | **0.296** | **0.360** | 0.322 | 0.379 | 0.376 | 0.405 | 0.382 | 0.407 | 0.362 | 0.404 | 0.334 | 0.383 | 0.365 | 0.395 | 0.384 | 0.428 | 0.330 | 0.379 | 0.835 | 0.676 | 0.388 | 0.434 | 0.386 | 0.413 | 0.431 | 0.447 | **0.320** | **0.378** | 0.371 | 0.410 | 0.349 | 0.395 | 0.414 | 0.427 | 0.385 | 0.430 |
| ETTm1 | 96 | **0.286** | **0.344** | 0.292 | 0.346 | 0.324 | 0.362 | 0.326 | 0.367 | 0.287 | 0.336 | **0.272** | **0.334** | 0.316 | 0.355 | 0.289 | 0.348 | 0.290 | 0.342 | 0.316 | 0.373 | 0.326 | 0.390 | 0.314 | 0.358 | 0.299 | 0.343 | 0.301 | 0.342 | 0.325 | 0.372 | 0.298 | 0.345 | 0.338 | 0.375 | 0.314 | 0.360 |
| | 192 | **0.324** | **0.360** | 0.332 | 0.368 | 0.369 | 0.385 | 0.375 | 0.387 | 0.33 | 0.359 | **0.310** | **0.358** | 0.361 | 0.38 | 0.332 | 0.375 | 0.332 | 0.369 | 0.377 | 0.411 | 0.365 | 0.415 | 0.354 | 0.386 | 0.335 | 0.365 | 0.335 | 0.363 | 0.354 | 0.386 | 0.344 | 0.375 | 0.371 | 0.387 | 0.359 | 0.387 |
| | 336 | **0.357** | **0.390** | 0.365 | 0.391 | 0.402 | 0.408 | 0.408 | 0.408 | 0.368 | 0.38 | **0.352** | **0.384** | 0.394 | 0.403 | 0.365 | 0.397 | 0.366 | 0.392 | 0.431 | 0.442 | 0.382 | 0.425 | 0.384 | 0.405 | 0.369 | 0.386 | 0.370 | 0.383 | 0.394 | 0.415 | 0.390 | 0.410 | 0.410 | 0.411 | 0.410 | 0.411 |
| | 720 | **0.410** | **0.415** | 0.416 | 0.417 | 0.467 | 0.444 | 0.472 | 0.444 | 0.425 | 0.413 | **0.383** | **0.411** | 0.459 | 0.442 | 0.421 | 0.431 | 0.416 | 0.420 | 0.600 | 0.547 | 0.446 | 0.458 | 0.427 | 0.432 | 0.425 | 0.421 | 0.425 | 0.414 | 0.476 | 0.469 | 0.445 | 0.441 | 0.478 | 0.450 | 0.459 | 0.464 |
| | Avg | **0.344** | **0.377** | 0.351 | 0.381 | 0.391 | 0.400 | 0.396 | 0.401 | 0.353 | 0.372 | **0.329** | **0.372** | 0.383 | 0.395 | 0.352 | 0.388 | 0.351 | 0.381 | 0.431 | 0.443 | 0.380 | 0.422 | 0.370 | 0.395 | 0.357 | **0.379** | 0.358 | 0.376 | 0.387 | 0.411 | 0.369 | 0.393 | 0.399 | 0.406 | 0.386 | 0.406 |
| ETTm2 | 96 | **0.162** | **0.252** | 0.166 | 0.256 | 0.179 | 0.261 | 0.181 | 0.265 | 0.163 | 0.25 | **0.161** | **0.253** | 0.174 | 0.255 | 0.171 | 0.26 | 0.165 | 0.255 | 0.421 | 0.461 | 0.180 | 0.271 | 0.177 | 0.259 | 0.167 | 0.260 | **0.164** | **0.253** | 0.186 | 0.281 | 0.174 | 0.259 | 0.187 | 0.267 | 0.178 | 0.273 |
| | 192 | **0.215** | **0.292** | 0.222 | 0.293 | 0.241 | 0.304 | 0.246 | 0.307 | 0.216 | 0.287 | **0.219** | **0.293** | 0.237 | 0.297 | 0.228 | 0.23 | 0.220 | 0.292 | 0.503 | 0.519 | 0.252 | 0.318 | 0.241 | 0.303 | 0.224 | 0.303 | **0.219** | **0.290** | 0.277 | 0.356 | 0.236 | 0.303 | 0.249 | 0.309 | 0.245 | 0.316 |
| | 336 | **0.268** | **0.329** | 0.272 | 0.324 | 0.302 | 0.342 | 0.306 | 0.345 | 0.271 | 0.324 | **0.271** | **0.329** | 0.294 | 0.334 | 0.282 | 0.336 | 0.274 | 0.329 | 0.611 | 0.580 | 0.324 | 0.364 | 0.297 | 0.338 | 0.281 | 0.342 | 0.273 | 0.326 | 0.311 | 0.369 | 0.291 | 0.338 | 0.321 | 0.351 | 0.295 | 0.350 |
| | 720 | **0.339** | **0.375** | **0.351** | **0.381** | 0.401 | 0.4 | 0.403 | 0.401 | 0.36 | 0.389 | 0.352 | 0.379 | 0.392 | 0.392 | 0.38 | 0.398 | 0.362 | 0.385 | 0.996 | 0.750 | 0.410 | 0.420 | 0.396 | 0.398 | 0.397 | 0.421 | 0.366 | 0.385 | 0.403 | 0.412 | 0.371 | 0.391 | 0.497 | 0.403 | 0.389 | 0.412 |
| | Avg | **0.246** | **0.312** | **0.253** | **0.314** | 0.281 | 0.327 | 0.284 | 0.329 | 0.253 | 0.313 | 0.251 | 0.314 | 0.274 | 0.320 | 0.265 | 0.306 | 0.255 | 0.315 | 0.633 | 0.578 | 0.292 | 0.343 | 0.278 | 0.325 | 0.267 | 0.332 | 0.256 | 0.314 | 0.294 | 0.355 | 0.268 | 0.323 | 0.314 | 0.333 | 0.277 | 0.338 |
| ECL | 96 | **0.127** | **0.222** | **0.129** | 0.226 | 0.139 | 0.235 | 0.139 | 0.233 | 0.132 | 0.229 | 0.131 | 0.224 | 0.154 | 0.245 | 0.13 | 0.225 | **0.129** | **0.222** | 0.187 | 0.283 | 0.186 | 0.302 | 0.141 | 0.243 | 0.153 | 0.237 | 0.140 | 0.235 | 0.171 | 0.256 | 0.129 | 0.224 | 0.168 | 0.272 | 0.159 | 0.267 |
| | 192 | **0.140** | **0.234** | **0.143** | **0.239** | 0.16 | 0.254 | 0.158 | 0.249 | 0.146 | 0.244 | 0.152 | 0.241 | 0.167 | 0.257 | 0.15 | 0.244 | 0.147 | 0.240 | 0.258 | 0.330 | 0.197 | 0.311 | 0.163 | 0.261 | 0.152 | 0.249 | 0.154 | 0.248 | 0.177 | 0.265 | 0.147 | 0.240 | 0.184 | 0.289 | 0.168 | 0.279 |
| | 336 | **0.158** | **0.254** | 0.161 | 0.259 | 0.176 | 0.271 | 0.177 | 0.271 | 0.162 | 0.262 | **0.160** | **0.248** | 0.184 | 0.274 | 0.166 | 0.262 | 0.163 | 0.259 | 0.323 | 0.369 | 0.213 | 0.328 | 0.176 | 0.277 | 0.169 | 0.267 | 0.171 | 0.264 | 0.197 | 0.285 | 0.164 | 0.257 | 0.198 | 0.300 | 0.196 | 0.308 |
| | 720 | **0.187** | **0.281** | **0.191** | **0.286** | 0.198 | 0.292 | 0.201 | 0.293 | 0.2 | 0.297 | 0.192 | 0.298 | 0.222 | 0.307 | 0.206 | 0.297 | 0.197 | 0.290 | 0.404 | 0.423 | 0.233 | 0.344 | 0.212 | 0.308 | 0.233 | 0.344 | 0.209 | 0.297 | 0.234 | 0.318 | 0.203 | 0.291 | 0.220 | 0.320 | 0.203 | 0.312 |
| | Avg | **0.153** | **0.248** | **0.156** | **0.253** | 0.168 | 0.264 | 0.169 | 0.262 | 0.160 | 0.258 | 0.159 | 0.253 | 0.182 | 0.271 | 0.163 | 0.257 | 0.159 | 0.253 | 0.293 | 0.351 | 0.207 | 0.321 | 0.173 | 0.272 | 0.177 | 0.274 | 0.169 | 0.261 | 0.195 | 0.281 | 0.161 | 0.253 | 0.193 | 0.295 | 0.182 | 0.292 |
| Traffic | 96 | **0.361** | **0.248** | 0.368 | 0.253 | 0.378 | 0.261 | 0.378 | 0.259 | 0.395 | 0.263 | 0.362 | 0.248 | 0.479 | 0.308 | 0.359 | 0.25 | **0.360** | **0.249** | 0.512 | 0.290 | 0.576 | 0.359 | 0.462 | 0.332 | 0.410 | 0.282 | 0.496 | 0.375 | 0.613 | 0.395 | 0.430 | 0.327 | 0.593 | 0.321 | 0.508 | 0.301 |
| | 192 | **0.372** | **0.256** | **0.379** | 0.261 | 0.393 | 0.269 | 0.399 | 0.27 | 0.406 | 0.277 | **0.374** | **0.247** | 0.484 | 0.308 | 0.393 | 0.259 | 0.379 | 0.256 | 0.523 | 0.297 | 0.610 | 0.380 | 0.488 | 0.354 | 0.423 | 0.287 | 0.503 | 0.377 | 0.559 | 0.363 | 0.451 | 0.340 | 0.617 | 0.336 | 0.536 | 0.315 |
| | 336 | **0.389** | **0.265** | 0.397 | 0.270 | 0.399 | 0.272 | 0.416 | 0.279 | 0.422 | 0.285 | **0.385** | **0.271** | 0.501 | 0.314 | 0.396 | 0.265 | 0.392 | 0.264 | 0.530 | 0.300 | 0.608 | 0.375 | 0.498 | 0.360 | 0.436 | 0.296 | 0.517 | 0.382 | 0.555 | 0.358 | 0.470 | 0.351 | 0.629 | 0.336 | 0.525 | 0.310 |
| | 720 | **0.431** | **0.285** | 0.440 | 0.296 | 0.437 | 0.29 | 0.456 | 0.297 | 0.443 | 0.299 | **0.430** | **0.288** | 0.53 | 0.333 | 0.421 | 0.286 | 0.432 | 0.286 | 0.573 | 0.313 | 0.621 | 0.375 | 0.529 | 0.370 | 0.466 | 0.315 | 0.555 | 0.398 | 0.620 | 0.394 | 0.513 | 0.372 | 0.640 | 0.350 | 0.571 | 0.323 |
| | Avg | **0.388** | **0.264** | 0.396 | 0.270 | 0.402 | 0.273 | 0.412 | 0.276 | 0.417 | 0.281 | **0.388** | **0.264** | 0.499 | 0.316 | 0.392 | 0.265 | **0.391** | **0.264** | 0.535 | 0.300 | 0.604 | 0.372 | 0.494 | 0.354 | 0.434 | 0.295 | 0.518 | 0.383 | 0.587 | 0.378 | 0.466 | 0.348 | 0.620 | 0.336 | 0.535 | 0.312 |
| Weather | 96 | **0.144** | **0.198** | **0.149** | 0.200 | 0.174 | 0.212 | 0.175 | 0.215 | 0.155 | 0.197 | **0.147** | **0.201** | 0.165 | 0.209 | 0.152 | 0.206 | 0.149 | 0.198 | 0.153 | 0.217 | 0.238 | 0.314 | 0.156 | 0.206 | 0.152 | 0.237 | 0.175 | 0.225 | 0.178 | 0.233 | 0.149 | 0.202 | 0.172 | 0.220 | 0.161 | 0.226 |
| | 192 | **0.189** | **0.240** | 0.196 | 0.245 | 0.221 | 0.255 | 0.221 | 0.255 | 0.191 | 0.237 | **0.189** | **0.234** | 0.211 | 0.251 | 0.198 | 0.249 | 0.194 | 0.241 | 0.197 | 0.269 | 0.275 | 0.329 | 0.199 | 0.248 | 0.220 | 0.282 | 0.218 | 0.260 | 0.235 | 0.277 | 0.194 | 0.242 | 0.219 | 0.261 | 0.220 | 0.283 |
| | 336 | **0.234** | **0.277** | **0.238** | **0.277** | 0.277 | 0.295 | 0.277 | 0.296 | 0.251 | 0.282 | 0.262 | 0.279 | 0.266 | 0.292 | 0.251 | 0.291 | 0.245 | 0.282 | 0.252 | 0.311 | 0.339 | 0.377 | 0.249 | 0.291 | 0.265 | 0.319 | 0.265 | 0.294 | 0.337 | 0.345 | 0.243 | 0.282 | 0.280 | 0.306 | 0.275 | 0.328 |
| | 720 | **0.312** | **0.330** | 0.314 | 0.334 | 0.353 | 0.348 | 0.355 | 0.348 | 0.323 | 0.341 | **0.304** | **0.316** | 0.344 | 0.343 | 0.322 | 0.340 | 0.314 | 0.334 | 0.318 | 0.363 | 0.389 | 0.409 | 0.336 | 0.343 | 0.323 | 0.362 | 0.329 | 0.339 | 0.396 | 0.413 | 0.316 | 0.333 | 0.365 | 0.359 | 0.311 | 0.356 |
| | Avg | **0.220** | **0.261** | **0.224** | **0.264** | 0.256 | 0.277 | 0.257 | 0.278 | 0.230 | 0.264 | 0.226 | 0.258 | 0.247 | 0.274 | 0.231 | 0.272 | 0.226 | 0.264 | 0.230 | 0.290 | 0.310 | 0.357 | 0.235 | 0.272 | 0.240 | 0.300 | 0.247 | 0.280 | 0.287 | 0.317 | 0.226 | 0.265 | 0.259 | 0.287 | 0.242 | 0.298 |
| Exchange | 96 | **0.079** | **0.195** | **0.080** | **0.196** | 0.085 | 0.204 | 0.085 | 0.205 | 0.086 | 0.204 | - | - | 0.081 | 0.199 | - | - | 0.093 | 0.214 | 0.186 | 0.346 | 0.139 | 0.276 | 0.083 | 0.201 | 0.081 | 0.203 | 0.083 | 0.201 | 0.116 | 0.254 | 0.083 | 0.201 | 0.107 | 0.234 | 0.102 | 0.235 |
| | 192 | **0.165** | **0.287** | **0.166** | **0.288** | 0.179 | 0.301 | 0.179 | 0.301 | 0.183 | 0.302 | - | - | 0.171 | 0.295 | - | - | 0.192 | 0.312 | 0.467 | 0.522 | 0.256 | 0.369 | 0.174 | 0.296 | 0.157 | 0.293 | 0.170 | 0.293 | 0.218 | 0.345 | 0.170 | 0.292 | 0.226 | 0.344 | 0.172 | 0.316 |
| | 336 | **0.259** | **0.352** | **0.307** | **0.398** | 0.329 | 0.415 | 0.331 | 0.417 | 0.343 | 0.420 | - | - | 0.324 | 0.411 | - | - | 0.350 | 0.432 | 0.783 | 0.721 | 0.426 | 0.464 | 0.336 | 0.417 | 0.305 | 0.414 | 0.309 | 0.401 | 0.294 | 0.413 | 0.309 | 0.401 | 0.367 | 0.448 | 0.272 | 0.407 |
| | 720 | **0.621** | **0.554** | **0.656** | **0.582** | 0.838 | 0.690 | 0.860 | 0.698 | 0.903 | 0.715 | - | - | 0.837 | 0.687 | - | - | 0.911 | 0.716 | 1.367 | 0.943 | 1.090 | 0.800 | 0.900 | 0.715 | 0.643 | 0.601 | 0.817 | 0.680 | 1.110 | 0.767 | 0.816 | 0.680 | 0.964 | 0.746 | 0.714 | 0.658 |
| | Avg | **0.281** | **0.347** | **0.302** | **0.366** | 0.358 | 0.402 | 0.363 | 0.405 | 0.379 | 0.410 | - | - | 0.353 | 0.398 | - | - | 0.387 | 0.419 | 0.701 | 0.633 | 0.478 | 0.477 | 0.373 | 0.407 | 0.297 | 0.378 | 0.345 | 0.394 | 0.435 | 0.445 | 0.345 | 0.394 | 0.416 | 0.443 | 0.315 | 0.404 |
| ILI | 96 | **1.127** | **0.677** | 1.347 | **0.754** | - | - | - | - | 1.696 | 0.789 | **1.285** | **0.727** | - | - | - | - | 1.319 | 0.754 | 3.040 | 1.186 | 2.624 | 1.095 | 1.472 | 0.798 | 2.215 | 1.081 | 4.337 | 1.507 | 2.150 | 1.005 | 4.445 | 1.536 | 2.317 | 0.934 | 2.684 | 1.112 |
| | 192 | **0.965** | **0.634** | **1.250** | **0.778** | - | - | - | - | 1.693 | 0.811 | 1.404 | 0.814 | - | - | - | - | 1.430 | 0.834 | 3.356 | 1.230 | 2.516 | 1.021 | 1.435 | 0.745 | 1.963 | 0.963 | 4.205 | 1.481 | 2.103 | 0.983 | 4.409 | 1.519 | 1.972 | 0.920 | 2.507 | 1.013 |
| | 336 | **0.980** | **0.656** | **1.388** | **0.781** | - | - | - | - | 1.867 | 0.881 | 1.523 | 0.807 | - | - | - | - | 1.553 | 0.815 | 3.441 | 1.223 | 2.505 | 1.041 | 1.474 | 0.822 | 2.130 | 1.024 | 4.257 | 1.484 | 2.432 | 1.061 | 4.388 | 1.507 | 2.238 | 0.940 | 2.423 | 1.012 |
| | 720 | **1.118** | **0.702** | 1.774 | 0.868 | - | - | - | - | 1.421 | 0.747 | 1.531 | 0.854 | - | - | - | - | **1.470** | **0.788** | 3.608 | 1.302 | 2.742 | 1.122 | 1.839 | 0.912 | 2.368 | 1.096 | 4.278 | 1.487 | 2.325 | 1.035 | 4.306 | 1.502 | 2.027 | 0.928 | 2.653 | 1.085 |
| | Avg | **1.048** | **0.667** | **1.440** | **0.795** | - | - | - | - | 1.669 | 0.807 | 1.436 | 0.801 | - | - | - | - | 1.443 | 0.798 | 3.361 | 1.235 | 2.597 | 1.070 | 1.555 | 0.819 | 2.169 | 1.041 | 4.269 | 1.490 | 2.253 | 1.021 | 4.387 | 1.516 | 2.139 | 0.931 | 2.567 | 1.056 |

TABLE VIII
FULL RESULTS FOR THE SHORT-TERM FORECASTING TASK. WE PRESENT COMPREHENSIVE RESULTS COMPARING METATCN WITH A WIDE RANGE OF COMPETITIVE MODELS ACROSS FOUR DIFFERENT PREDICTION LENGTHS. THE 'AVG' VALUES REPORTED ARE AVERAGED ACROSS ALL FOUR PREDICTION LENGTHS.

| Methods | | MetaTCN (Our) | | ModernTCN 2024) | | UnetTS (2024) | | iTransformer (2024) | | Crossformer (2023) | | PatchTST (2022) | | CARD (2024) | | TSmixer (2023) | | MICN (2023) | | TimesNet (2023) | | Dlinear (2023) | | Rlinear (2023) | | RMLP (2023) | | SCINet (2022) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE |
| ETTh1 | 6 | **0.421** | **0.388** | **0.505** | **0.423** | 0.690 | 0.502 | 0.523 | 0.433 | 1.052 | 0.851 | 0.776 | 0.545 | 0.754 | 0.530 | 0.539 | 0.446 | 0.518 | 0.424 | 0.785 | 0.558 | 0.941 | 0.649 | 1.184 | 0.676 | 0.682 | 0.504 | 0.549 | 0.472 |
| | 12 | **0.300** | **0.350** | **0.316** | **0.357** | 0.355 | 0.386 | 0.324 | 0.364 | 0.652 | 0.718 | 0.438 | 0.431 | 0.383 | 0.397 | 0.334 | 0.375 | 0.321 | 0.357 | 0.412 | 0.424 | 0.559 | 0.504 | 0.701 | 0.541 | 0.406 | 0.414 | 0.340 | 0.397 |
| | 18 | **0.304** | **0.353** | **0.317** | **0.359** | 0.354 | 0.385 | 0.319 | 0.364 | 0.642 | 0.722 | 0.418 | 0.418 | 0.374 | 0.392 | 0.329 | 0.375 | 0.316 | 0.357 | 0.434 | 0.568 | 0.535 | 0.485 | 0.677 | 0.535 | 0.409 | 0.418 | 0.335 | 0.397 |
| Avg | | **0.342** | **0.364** | **0.379** | **0.380** | 0.466 | 0.424 | 0.389 | 0.387 | 0.782 | 0.764 | 0.544 | 0.465 | 0.504 | 0.440 | 0.401 | 0.399 | 0.385 | 0.379 | 0.544 | 0.517 | 0.678 | 0.546 | 0.854 | 0.584 | 0.499 | 0.445 | 0.408 | 0.422 |
| ETTh2 | 6 | **0.123** | **0.215** | **0.132** | **0.227** | 0.147 | 0.246 | 0.139 | 0.237 | 0.280 | 0.456 | 0.159 | 0.264 | 0.157 | 0.260 | 0.143 | 0.244 | 0.140 | 0.239 | 0.162 | 0.267 | 0.199 | 0.327 | 0.213 | 0.310 | 0.150 | 0.252 | 0.146 | 0.258 |
| | 12 | **0.143** | **0.234** | **0.144** | **0.240** | 0.156 | 0.253 | 0.158 | 0.249 | 0.318 | 0.483 | 0.175 | 0.273 | 0.170 | 0.268 | 0.163 | 0.256 | 0.160 | 0.251 | 0.169 | 0.266 | 0.219 | 0.337 | 0.205 | 0.295 | 0.174 | 0.273 | 0.166 | 0.271 |
| | 18 | **0.153** | **0.247** | **0.159** | **0.253** | 0.169 | 0.260 | 0.167 | 0.258 | 0.336 | 0.509 | 0.181 | 0.273 | 0.178 | 0.270 | 0.172 | 0.266 | 0.169 | 0.261 | 0.200 | 0.294 | 0.215 | 0.319 | 0.247 | 0.325 | 0.192 | 0.284 | 0.175 | 0.281 |
| Avg | | **0.140** | **0.232** | **0.145** | **0.240** | 0.157 | 0.253 | 0.155 | 0.248 | 0.311 | 0.483 | 0.172 | 0.270 | 0.168 | 0.266 | 0.159 | 0.255 | 0.156 | 0.250 | 0.177 | 0.276 | 0.211 | 0.328 | 0.222 | 0.310 | 0.172 | 0.270 | 0.162 | 0.270 |
| ETTm1 | 6 | **0.125** | **0.215** | **0.131** | **0.218** | 0.152 | 0.229 | 0.141 | 0.222 | 0.284 | 0.438 | 0.181 | 0.250 | 0.161 | 0.235 | 0.145 | 0.229 | 0.140 | 0.218 | 0.138 | 0.225 | 0.272 | 0.328 | 0.281 | 0.309 | 0.162 | 2.340 | 0.148 | 0.242 |
| | 12 | **0.197** | **0.270** | **0.216** | **0.279** | 0.297 | 0.317 | 0.248 | 0.291 | 0.499 | 0.561 | 0.311 | 0.331 | 0.294 | 0.321 | 0.255 | 0.300 | 0.246 | 0.285 | 0.231 | 0.289 | 0.402 | 0.396 | 0.402 | 0.396 | 0.301 | 0.317 | 0.260 | 0.317 |
| | 18 | **0.305** | **0.324** | **0.311** | **0.329** | 0.429 | 0.379 | 0.360 | 0.350 | 0.724 | 0.662 | 0.409 | 0.388 | 0.389 | 0.366 | 0.371 | 0.361 | 0.356 | 0.343 | 0.323 | 0.340 | 0.511 | 0.445 | 0.511 | 0.445 | 0.410 | 0.375 | 0.378 | 0.382 |
| Avg | | **0.209** | **0.270** | **0.219** | **0.275** | 0.293 | 0.308 | 0.250 | 0.288 | 0.502 | 0.554 | 0.300 | 0.323 | 0.281 | 0.307 | 0.257 | 0.297 | 0.247 | 0.282 | 0.231 | 0.285 | 0.395 | 0.390 | 0.398 | 0.383 | 0.291 | 1.011 | 0.262 | 0.314 |
| ETTm2 | 6 | **0.059** | **0.137** | **0.060** | **0.138** | 0.062 | 0.142 | 0.064 | 0.145 | 0.129 | 0.278 | 0.066 | 0.155 | 0.065 | 0.151 | 0.066 | 0.149 | 0.063 | 0.142 | 0.062 | 0.144 | 0.083 | 0.200 | 0.083 | 0.176 | 0.065 | 0.150 | 0.067 | 0.158 |
| | 12 | **0.078** | **0.163** | **0.079** | **0.165** | 0.085 | 0.173 | 0.089 | 0.177 | 0.179 | 0.332 | 0.089 | 0.186 | 0.089 | 0.181 | 0.092 | 0.182 | 0.088 | 0.173 | 0.080 | 0.170 | 0.109 | 0.228 | 0.118 | 0.219 | 0.092 | 0.186 | 0.093 | 0.193 |
| | 18 | **0.092** | **0.181** | **0.093** | **0.182** | 0.105 | 0.197 | 0.108 | 0.200 | 0.217 | 0.366 | 0.110 | 0.211 | 0.112 | 0.208 | 0.111 | 0.206 | 0.107 | 0.196 | 0.098 | 0.191 | 0.125 | 0.242 | 0.130 | 0.232 | 0.114 | 0.210 | 0.113 | 0.218 |
| Avg | | **0.076** | **0.160** | **0.077** | **0.162** | 0.084 | 0.171 | 0.087 | 0.174 | 0.175 | 0.325 | 0.088 | 0.184 | 0.089 | 0.180 | 0.090 | 0.179 | 0.086 | 0.170 | 0.080 | 0.168 | 0.106 | 0.223 | 0.110 | 0.209 | 0.090 | 0.182 | 0.091 | 0.190 |
| IL1 | 6 | **0.710** | **0.482** | **1.419** | **0.694** | 1.602 | 0.846 | 1.463 | 0.703 | 2.942 | 2.439 | 3.402 | 1.213 | 2.793 | 1.079 | 1.507 | 0.724 | 1.492 | 0.717 | 1.101 | 0.597 | 3.407 | 1.255 | 3.378 | 1.170 | 2.485 | 1.024 | 1.536 | 0.766 |
| | 12 | **1.310** | **0.706** | **2.420** | **0.939** | 2.581 | 1.185 | 2.114 | 0.893 | 4.251 | 2.870 | 3.958 | 1.427 | 4.591 | 1.534 | 2.177 | 0.920 | 2.156 | 0.911 | 2.760 | 0.927 | 5.597 | 1.753 | 6.106 | 1.784 | 4.271 | 1.461 | 2.220 | 0.973 |
| | 18 | **0.910** | **0.603** | **2.394** | **0.983** | 1.794 | 0.941 | 1.968 | 0.892 | 3.958 | 3.206 | 4.905 | 1.594 | 3.774 | 1.402 | 2.027 | 0.919 | 2.007 | 0.910 | 3.516 | 1.014 | 4.875 | 1.666 | 5.953 | 1.815 | 4.327 | 1.497 | 2.066 | 0.972 |
| Avg | | **0.977** | **0.597** | **2.078** | **0.872** | 1.992 | 0.991 | 1.848 | 0.829 | 3.717 | 2.838 | 4.088 | 1.411 | 3.719 | 1.338 | 1.904 | 0.854 | 1.885 | 0.846 | 2.459 | 0.846 | 4.626 | 1.558 | 5.146 | 1.590 | 3.694 | 1.327 | 1.941 | 0.904 |
| Exchange | 6 | **0.007** | **0.052** | **0.008** | **0.052** | 0.008 | 0.053 | 0.008 | 0.053 | 0.016 | 0.137 | 0.01 | 0.068 | 0.010 | 0.065 | 0.008 | 0.055 | 0.008 | 0.054 | 0.009 | 0.063 | 0.048 | 0.175 | 0.014 | 0.078 | 0.009 | 0.061 | 0.008 | 0.058 |
| | 12 | **0.013** | **0.072** | **0.013** | **0.072** | 0.014 | 0.075 | **0.013** | 0.074 | 0.026 | 0.179 | 0.017 | 0.089 | 0.017 | 0.087 | 0.013 | 0.076 | 0.013 | 0.075 | 0.018 | 0.089 | 0.038 | 0.148 | 0.023 | 0.106 | 0.017 | 0.089 | 0.014 | 0.081 |
| | 18 | **0.018** | **0.088** | **0.019** | **0.089** | **0.019** | 0.091 | **0.019** | 0.091 | 0.038 | 0.209 | 0.023 | 0.104 | 0.023 | 0.104 | 0.020 | 0.094 | 0.019 | 0.093 | 0.025 | 0.107 | 0.048 | 0.171 | 0.032 | 0.127 | 0.024 | 0.108 | 0.020 | 0.099 |
| Avg | | **0.013** | **0.071** | **0.013** | **0.071** | 0.014 | 0.073 | 0.013 | 0.073 | 0.027 | 0.175 | 0.017 | 0.087 | 0.017 | 0.085 | 0.014 | 0.075 | 0.013 | 0.074 | 0.017 | 0.086 | 0.045 | 0.165 | 0.023 | 0.104 | 0.017 | 0.086 | 0.014 | 0.079 |
| Heat1 | 6 | **0.005** | **0.036** | **0.006** | **0.040** | **0.006** | 0.045 | 0.007 | 0.048 | 0.014 | 0.080 | 0.009 | 0.050 | 0.007 | 0.042 | 0.007 | 0.049 | 0.007 | 0.047 | 0.011 | 0.075 | 0.013 | 0.059 | 0.012 | 0.083 | 0.016 | 0.107 | 0.010 | 0.065 |
| | 12 | **0.020** | **0.079** | **0.022** | **0.088** | 0.024 | 0.099 | 0.026 | 0.104 | 0.052 | 0.177 | 0.033 | 0.110 | 0.027 | 0.092 | 0.027 | 0.107 | 0.026 | 0.102 | 0.041 | 0.162 | 0.048 | 0.128 | 0.045 | 0.180 | 0.058 | 0.232 | 0.037 | 0.140 |
| | 18 | **0.040** | **0.123** | **0.045** | **0.138** | 0.049 | 0.154 | 0.053 | 0.162 | 0.107 | 0.278 | 0.066 | 0.173 | 0.055 | 0.144 | 0.055 | 0.167 | 0.052 | 0.159 | 0.083 | 0.253 | 0.097 | 0.199 | 0.092 | 0.280 | 0.118 | 0.361 | 0.075 | 0.219 |
| Avg | | **0.022** | **0.079** | **0.024** | **0.089** | 0.026 | 0.099 | 0.029 | 0.105 | 0.058 | 0.178 | 0.036 | 0.111 | 0.030 | 0.093 | 0.030 | 0.108 | 0.028 | 0.103 | 0.045 | 0.163 | 0.053 | 0.129 | 0.050 | 0.181 | 0.064 | 0.233 | 0.041 | 0.141 |
| Heat2 | 6 | **0.003** | **0.312** | **0.004** | **0.349** | 0.004 | 0.390 | 0.004 | 0.412 | 0.008 | 0.702 | 0.005 | 0.436 | 0.004 | 0.363 | 0.004 | 0.424 | 0.005 | 0.416 | 0.004 | 0.375 | 0.007 | 0.507 | 0.007 | 0.713 | 0.009 | 0.919 | 0.006 | 0.556 |
| | 12 | **0.018** | **0.074** | **0.021** | **0.083** | 0.022 | 0.093 | 0.024 | 0.098 | 0.048 | 0.167 | 0.030 | 0.104 | 0.025 | 0.086 | 0.025 | 0.101 | 0.031 | 0.099 | 0.022 | 0.089 | 0.044 | 0.121 | 0.042 | 0.170 | 0.054 | 0.219 | 0.034 | 0.132 |
| | 18 | **0.043** | **0.120** | **0.051** | **0.134** | 0.052 | 0.150 | 0.057 | 0.158 | 0.115 | 0.269 | 0.071 | 0.168 | 0.059 | 0.139 | 0.059 | 0.163 | 0.075 | 0.160 | 0.052 | 0.144 | 0.104 | 0.194 | 0.099 | 0.273 | 0.127 | 0.352 | 0.080 | 0.213 |
| Avg | | **0.021** | **0.169** | **0.025** | **0.189** | 0.026 | 0.211 | 0.028 | 0.223 | 0.057 | 0.379 | 0.035 | 0.236 | 0.029 | 0.196 | 0.029 | 0.229 | 0.037 | 0.225 | 0.026 | 0.203 | 0.052 | 0.274 | 0.049 | 0.385 | 0.063 | 0.497 | 0.040 | 0.300 |
| Heat3 | 6 | **0.005** | **0.039** | **0.006** | **0.044** | **0.006** | 0.049 | 0.007 | 0.051 | 0.014 | 0.088 | 0.009 | 0.055 | 0.007 | 0.046 | 0.007 | 0.053 | 0.009 | 0.050 | 0.007 | 0.053 | 0.013 | 0.063 | 0.012 | 0.088 | 0.016 | 0.114 | 0.010 | 0.069 |
| | 12 | **0.025** | **0.088** | **0.028** | **0.099** | 0.031 | 0.110 | 0.033 | 0.116 | 0.066 | 0.199 | 0.041 | 0.124 | 0.034 | 0.103 | 0.034 | 0.119 | 0.043 | 0.114 | 0.034 | 0.119 | 0.060 | 0.143 | 0.057 | 0.201 | 0.074 | 0.259 | 0.047 | 0.157 |
| | 18 | **0.054** | **0.137** | **0.060** | **0.153** | 0.066 | 0.171 | 0.071 | 0.181 | 0.143 | 0.308 | 0.089 | 0.191 | 0.074 | 0.159 | 0.073 | 0.186 | 0.093 | 0.177 | 0.073 | 0.186 | 0.130 | 0.223 | 0.123 | 0.313 | 0.158 | 0.404 | 0.100 | 0.244 |
| Avg | | **0.028** | **0.088** | **0.031** | **0.099** | 0.034 | 0.110 | 0.037 | 0.116 | 0.074 | 0.198 | 0.046 | 0.123 | 0.038 | 0.103 | 0.038 | 0.119 | 0.048 | 0.114 | 0.038 | 0.119 | 0.068 | 0.143 | 0.064 | 0.201 | 0.083 | 0.259 | 0.052 | 0.157 |
| Heat4 | 6 | **0.007** | **0.044** | **0.008** | **0.049** | 0.009 | 0.055 | 0.009 | 0.058 | 0.018 | 0.099 | 0.011 | 0.061 | 0.009 | 0.051 | 0.009 | 0.060 | 0.012 | 0.057 | 0.009 | 0.055 | 0.016 | 0.071 | 0.016 | 0.100 | 0.020 | 0.129 | 0.013 | 0.078 |
| | 12 | **0.031** | **0.099** | **0.035** | **0.111** | 0.038 | 0.124 | 0.041 | 0.131 | 0.082 | 0.223 | 0.051 | 0.139 | 0.043 | 0.115 | 0.042 | 0.135 | 0.054 | 0.128 | 0.039 | 0.124 | 0.075 | 0.161 | 0.071 | 0.227 | 0.091 | 0.292 | 0.058 | 0.177 |
| | 18 | **0.066** | **0.153** | **0.074** | **0.171** | 0.081 | 0.191 | 0.087 | 0.202 | 0.175 | 0.344 | 0.109 | 0.214 | 0.090 | 0.178 | 0.090 | 0.208 | 0.114 | 0.198 | 0.083 | 0.192 | 0.159 | 0.248 | 0.151 | 0.349 | 0.194 | 0.450 | 0.123 | 0.273 |
| Avg | | **0.035** | **0.099** | **0.039** | **0.110** | 0.043 | 0.123 | 0.046 | 0.130 | 0.092 | 0.222 | 0.057 | 0.138 | 0.047 | 0.115 | 0.047 | 0.134 | 0.060 | 0.128 | 0.044 | 0.124 | 0.083 | 0.160 | 0.079 | 0.225 | 0.102 | 0.290 | 0.065 | 0.176 |

TABLE IX
FULL RESULTS FOR THE ZERO-TERM FORECASTING TASK. WE PRESENT COMPREHENSIVE RESULTS COMPARING METATCN WITH A WIDE RANGE OF COMPETITIVE MODELS ACROSS FOUR DIFFERENT PREDICTION LENGTHS. THE RESULTS ARE AVERAGED FROM 4 DIFFERENT PREDICTION LENGTHS: {96, 192, 336, 720}.

| Methods | MetaTCN (Our) | | ModernTCN (2024) | | Time-LLM (2024) | | GPT4S (2023) | | TimeMixer++ (2025) | | TimeMixer (2024) | | UnetTS (2024) | | iTransformer (2024) | | PatchTST (2022) | | Crossformer (2023) | | FEDformer (2022) | | Autoformer (2021) | | TIDE (2023) | | Dlinear (2023) | | TimesNet (2023) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1→h2 | **0.297** | **0.339** | **0.322** | **0.373** | 0.353 | 0.387 | 0.406 | 0.422 | 0.367 | 0.391 | 0.427 | 0.424 | 0.342 | 0.405 | 0.481 | 0.474 | 0.380 | 0.405 | 0.555 | 0.574 | 0.712 | 0.693 | 0.634 | 0.651 | 0.593 | 0.582 | 0.493 | 0.488 | 0.421 | 0.431 |
| ETTh1→m2 | **0.295** | **0.351** | 0.297 | 0.351 | **0.273** | **0.340** | 0.325 | 0.363 | 0.301 | 0.357 | 0.361 | 0.397 | 0.325 | 0.386 | 0.311 | 0.361 | 0.314 | 0.360 | 0.613 | 0.629 | 0.681 | 0.588 | 0.647 | 0.609 | 0.563 | 0.547 | 0.415 | 0.452 | 0.327 | 0.361 |
| ETTh2→h1 | **0.436** | **0.452** | **0.476** | **0.468** | 0.479 | 0.474 | 0.757 | 0.578 | 0.511 | 0.498 | 0.679 | 0.577 | 0.589 | 0.552 | 0.552 | 0.511 | 0.565 | 0.513 | 0.587 | 0.518 | 0.612 | 0.624 | 0.599 | 0.571 | 0.588 | 0.556 | 0.703 | 0.574 | 0.865 | 0.621 |
| ETTm1→h2 | **0.346** | **0.393** | 0.393 | 0.418 | **0.381** | **0.412** | 0.433 | 0.439 | 0.417 | 0.422 | 0.452 | 0.441 | 0.374 | 0.428 | 0.434 | 0.438 | 0.439 | 0.438 | 0.624 | 0.541 | 0.533 | 0.594 | 0.579 | 0.568 | 0.543 | 0.535 | 0.464 | 0.475 | 0.457 | 0.454 |
| ETTm1→m2 | **0.264** | **0.317** | 0.310 | 0.351 | **0.268** | **0.320** | 0.313 | 0.348 | 0.291 | 0.331 | 0.329 | 0.357 | 0.289 | 0.349 | 0.324 | 0.331 | 0.296 | 0.334 | 0.595 | 0.572 | 0.612 | 0.611 | 0.603 | 0.592 | 0.534 | 0.527 | 0.335 | 0.389 | 0.322 | 0.354 |
| ETTm2→m1 | 0.429 | **0.430** | 0.486 | 0.455 | **0.414** | **0.438** | 0.769 | 0.567 | **0.427** | 0.448 | 0.554 | 0.478 | 0.472 | 0.473 | 0.559 | 0.491 | 0.568 | 0.492 | 0.611 | 0.593 | 0.577 | 0.601 | 0.594 | 0.597 | 0.585 | 0.571 | 0.649 | 0.537 | 0.769 | 0.567 |

TABLE X
FULL RESULTS FOR THE FEW-TERM FORECASTING TASK. WE PRESENT COMPREHENSIVE RESULTS COMPARING METATCN WITH A WIDE RANGE OF COMPETITIVE MODELS ACROSS FOUR DIFFERENT PREDICTION LENGTHS. THE RESULTS ARE AVERAGED FROM 4 DIFFERENT PREDICTION LENGTHS: {96, 192, 336, 720}.

| Methods | MetaTCN (Our) | | ModernTCN (2024) | | Time-LLM (2024) | | GPT4TS (2023) | | UnetTS (2024) | | PatchTST (2022) | | FEDformer (2022) | | Autoformer (2021) | | LightTS (2022) | | TimesNet (2023) | | Dlinear (2023) | | TimeMixer++ (2025) | | TimeMixer (2024) | | Crossformer (2023) | | TiDE (2023) | | iTransformer (2024) | | Crossformer (2023) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | **0.468** | **0.473** | **0.495** | **0.495** | 0.555 | 0.522 | 0.590 | 0.525 | 0.604 | 0.583 | 0.697 | 0.600 | 0.639 | 0.561 | 0.702 | 0.483 | 1.375 | 0.877 | 0.869 | 0.628 | 0.691 | 0.600 | 0.517 | 0.512 | 0.613 | 0.520 | 0.645 | 0.558 | 0.589 | 0.535 | 0.510 | 0.597 | 0.645 | 0.558 |
| ETTh2 | **0.319** | **0.377** | **0.354** | **0.398** | 0.371 | 0.394 | 0.397 | 0.421 | 0.497 | 0.453 | 0.415 | 0.431 | 0.466 | 0.475 | 0.488 | 0.499 | 2.655 | 1.160 | 0.415 | 0.431 | 0.605 | 0.510 | 0.379 | 0.391 | 0.402 | 0.433 | 0.428 | 0.447 | 0.395 | 0.412 | 0.455 | 0.461 | 0.428 | 0.447 |
| ETTm1 | **0.404** | **0.414** | 0.422 | 0.433 | **0.404** | **0.427** | 0.464 | 0.441 | 0.497 | 0.497 | 0.501 | 0.466 | 0.722 | 0.605 | 0.802 | 0.628 | 0.971 | 0.705 | 0.677 | 0.537 | 0.411 | 0.429 | 0.398 | 0.431 | 0.487 | 0.461 | 0.462 | 0.489 | 0.425 | 0.458 | 0.491 | 0.516 | 0.462 | 0.489 |
| ETTm2 | **0.275** | **0.358** | 0.315 | 0.364 | **0.277** | **0.323** | 0.293 | 0.335 | 0.339 | 0.430 | 0.288 | 0.340 | 0.463 | 0.488 | 1.342 | 0.930 | 0.987 | 0.756 | 0.320 | 0.353 | 0.316 | 0.368 | 0.291 | 0.351 | 0.311 | 0.367 | 0.343 | 0.389 | 0.317 | 0.371 | 0.375 | 0.412 | 0.343 | 0.389 |
| Exchange | **0.183** | **0.280** | 0.208 | 0.305 | **0.175** | **0.270** | 0.238 | 0.275 | 0.225 | 0.336 | 0.180 | 0.273 | 0.346 | 0.427 | 0.431 | 0.478 | 0.660 | 0.617 | 0.323 | 0.392 | 0.180 | 0.273 | 0.168 | 0.271 | 0.187 | 0.277 | 0.214 | 0.308 | 0.196 | 0.306 | 0.241 | 0.337 | 0.214 | 0.306 |
| Weather | **0.243** | **0.274** | 0.247 | 0.278 | **0.234** | **0.273** | 0.176 | 0.270 | 0.299 | 0.329 | 0.242 | 0.279 | 0.285 | 0.324 | 0.300 | 0.342 | 0.289 | 0.322 | 0.283 | 0.318 | 0.241 | 0.283 | 0.241 | 0.271 | 0.242 | 0.281 | 0.267 | 0.306 | 0.249 | 0.306 | 0.291 | 0.331 | 0.267 | 0.306 |

benchmark datasets, achieving an average MSE improvement of 8.2% ($p = 0.007$) with individual dataset gains ranging from 6.2% (Weather) to 10.2% (ECL). All improvements are statistically significant ($p < 0.02$), demonstrating the robustness and generalizability of the ML mechanism. A critical insight emerges from comparing the architectural configurations: MetaTCN achieves superior performance with only 1 MTB block + ML compared to MTB-Wide's 3 MTB blocks without ML, demonstrating that memory-augmented shallow networks outperform deeper fixed-capacity networks. This 3:1 depth ratio suggests that explicit memory mechanisms provide stronger inductive biases than simply stacking convolutional layers, and that adaptive retrieval through the ML's meta node selection is more effective than hierarchical feature extraction alone. The standard deviations across three random seeds remain consistently low for MetaTCN (±0.004 to ±0.012) compared to MTB-Wide (±0.005 to ±0.015), indicating not only improved average performance but also enhanced training stability. These results provide compelling evidence that MetaTCN's performance gains stem from the ML's structural advantages rather than increased capacity—even when given 13–25% more parameters.

TABLE XI
PARAMETER-MATCHED COMPARISON BETWEEN METATCN WITH ML AND EXPANDED MTB-WIDE. MTB-WIDE USES ADDITIONAL BLOCKS (J=3) TO APPROXIMATELY MATCH METATCN'S PARAMETER COUNT. METATCN USES STANDARD MTB (J=1) WITH ML; MTB-WIDE USES DEEPER MTB WITHOUT ML.

| **Model** | **ETTh1** | **ETTh2** | **ETTm1** | **ETTm2** | **Weather** | **Traffic** | **ECL** | **Avg.** |
|---|---|---|---|---|---|---|---|---|
| **MetaTCN (w/ ML)**<br>J=1 + ML | **0.384**<br>±0.008 | **0.289**<br>±0.007 | **0.312**<br>±0.006 | **0.267**<br>±0.009 | **0.178**<br>±0.005 | **0.421**<br>±0.012 | **0.156**<br>±0.004 | **0.287**<br>±0.007 |
| **MTB-Wide (w/o ML)**<br>J=3 w/o ML | 0.419<br>±0.010 | 0.315<br>±0.008 | 0.341<br>±0.009 | 0.291<br>±0.010 | 0.189<br>±0.006 | 0.461<br>±0.014 | 0.172<br>±0.005 | 0.313<br>±0.009 |
| **Improvement**<br>*p*-value | 9.1%<br>0.003 | 9.0%<br>0.004 | 9.3%<br>0.002 | 9.0%<br>0.003 | 6.2%<br>0.018 | 9.5%<br>0.002 | 10.2%<br>0.001 | 8.2%<br>0.007 |
| **Model Configuration Details** | | | | | | | | |
| Variables (C) | 7 | 7 | 7 | 7 | 21 | 862 | 321 | – |
| MetaTCN Params (M) | 3.78 | 3.78 | 3.78 | 3.78 | 10.23 | 399.12 | 148.89 | – |
| MTB-Wide Params (M) | 4.35 | 4.35 | 4.35 | 4.35 | 12.79 | 498.90 | 186.03 | – |

## APPENDIX I
## ERROR BARS

Tables XII and XIII present results across 5 independent runs with different random seeds on representative datasets for both long-term and short-term forecasting.

## APPENDIX J
## ADDITIONAL ZERO-SHOT EXPERIMENT

Table XIV presents a comprehensive comparison of MetaTCN against five recent foundation models across eight

TABLE XII
STATISTICAL ROBUSTNESS ANALYSIS FOR LONG-TERM FORECASTING ACROSS 5 RANDOM SEEDS. RESULTS SHOW MEAN ± STD WITH 99% CONFIDENCE INTERVALS. IMPROVEMENTS OVER MODERNTCN ARE STATISTICALLY SIGNIFICANT.

| Dataset | MetaTCN | | ModernTCN | |
|---|---|---|---|---|
| | MSE | MAE | MSE | MAE |
| ETTh1 | $0.390 \pm 0.003$ | $0.412 \pm 0.004$ | $0.404 \pm 0.005$ | $0.420 \pm 0.006$ |
| ETTh2 | $0.296 \pm 0.004$ | $0.360 \pm 0.005$ | $0.322 \pm 0.007$ | $0.379 \pm 0.008$ |
| ETTm1 | $0.344 \pm 0.003$ | $0.377 \pm 0.004$ | $0.351 \pm 0.005$ | $0.381 \pm 0.006$ |
| ETTm2 | $0.246 \pm 0.002$ | $0.312 \pm 0.003$ | $0.253 \pm 0.003$ | $0.314 \pm 0.004$ |
| Electricity | $0.153 \pm 0.002$ | $0.248 \pm 0.003$ | $0.156 \pm 0.004$ | $0.253 \pm 0.005$ |
| Traffic | $0.388 \pm 0.005$ | $0.264 \pm 0.004$ | $0.396 \pm 0.008$ | $0.270 \pm 0.006$ |
| Weather | $0.220 \pm 0.003$ | $0.261 \pm 0.004$ | $0.224 \pm 0.005$ | $0.264 \pm 0.006$ |
| Exchange | $0.281 \pm 0.004$ | $0.347 \pm 0.005$ | $0.302 \pm 0.006$ | $0.366 \pm 0.007$ |
| ILI | $1.048 \pm 0.018$ | $0.667 \pm 0.015$ | $1.440 \pm 0.032$ | $0.795 \pm 0.025$ |
| **Average** | $\mathbf{0.374 \pm 0.005}$ | $\mathbf{0.361 \pm 0.005}$ | $\mathbf{0.428 \pm 0.008}$ | $\mathbf{0.382 \pm 0.009}$ |
| **Improvement** | **12.6% MSE, 5.5% MAE** | | **99% Confidence** | |

TABLE XIII
STATISTICAL ROBUSTNESS ANALYSIS FOR SHORT-TERM FORECASTING ACROSS 5 RANDOM SEEDS. RESULTS SHOW MEAN ± STD WITH 99% CONFIDENCE INTERVALS. IMPROVEMENTS OVER MODERNTCN ARE STATISTICALLY SIGNIFICANT.

| Dataset | MetaTCN | | ModernTCN | |
|---|---|---|---|---|
| | MSE | MAE | MSE | MAE |
| ETTh1 | $0.342 \pm 0.004$ | $0.364 \pm 0.005$ | $0.379 \pm 0.007$ | $0.380 \pm 0.008$ |
| ETTh2 | $0.140 \pm 0.002$ | $0.232 \pm 0.003$ | $0.145 \pm 0.004$ | $0.240 \pm 0.005$ |
| ETTm1 | $0.209 \pm 0.003$ | $0.270 \pm 0.004$ | $0.219 \pm 0.005$ | $0.275 \pm 0.006$ |
| ETTm2 | $0.076 \pm 0.001$ | $0.160 \pm 0.002$ | $0.077 \pm 0.002$ | $0.162 \pm 0.003$ |
| ILI | $0.977 \pm 0.017$ | $0.597 \pm 0.013$ | $2.078 \pm 0.038$ | $0.872 \pm 0.028$ |
| Exchange | $0.013 \pm 0.001$ | $0.071 \pm 0.001$ | $0.013 \pm 0.001$ | $0.071 \pm 0.002$ |
| Heat1 | $0.022 \pm 0.001$ | $0.079 \pm 0.002$ | $0.024 \pm 0.002$ | $0.089 \pm 0.003$ |
| Heat2 | $0.021 \pm 0.001$ | $0.169 \pm 0.003$ | $0.025 \pm 0.002$ | $0.189 \pm 0.004$ |
| Heat3 | $0.028 \pm 0.001$ | $0.088 \pm 0.002$ | $0.031 \pm 0.002$ | $0.099 \pm 0.003$ |
| Heat4 | $0.035 \pm 0.001$ | $0.099 \pm 0.002$ | $0.039 \pm 0.002$ | $0.110 \pm 0.003$ |
| **Average** | $\mathbf{0.186 \pm 0.003}$ | $\mathbf{0.213 \pm 0.004}$ | $\mathbf{0.303 \pm 0.007}$ | $\mathbf{0.249 \pm 0.006}$ |
| **Improvement** | **38.6% MSE, 14.5% MAE** | | **99% Confidence** | |

TABLE XIV
PERFORMANCE COMPARISON ACROSS DIFFERENT DATASETS AND FORECASTING METHODS

| Dataset | Hist | Pred | TimeGPT | Moirai | TimesFM | Chronos | ChatTime | MetaTCN |
|---|---|---|---|---|---|---|---|---|
| ETTh1 | 48 | 24 | **0.1604** | 0.1694 | 0.2021 | 0.1634 | 0.1698 | 0.1720 |
| | 72 | 24 | 0.1603 | 0.1796 | 0.1599 | **0.1372** | 0.1403 | 0.1450 |
| | 96 | 24 | 0.1577 | 0.1433 | 0.1454 | 0.1374 | 0.1374 | **0.1320** |
| | 120 | 24 | 0.1594 | 0.1492 | 0.1502 | 0.1348 | 0.1431 | **0.1295** |
| ETTh2 | 48 | 24 | **0.2874** | 0.2963 | 0.3360 | 0.3128 | 0.2906 | 0.2950 |
| | 72 | 24 | 0.2888 | 0.3109 | **0.2880** | 0.3045 | 0.3092 | 0.2920 |
| | 96 | 24 | **0.2902** | 0.3139 | 0.3144 | 0.3158 | 0.2917 | 0.2980 |
| | 120 | 24 | 0.3026 | 0.2905 | 0.3311 | 0.3150 | 0.3124 | **0.2820** |
| ETTm1 | 192 | 96 | 0.1921 | 0.1608 | 0.1719 | 0.1604 | 0.1442 | **0.1380** |
| | 288 | 96 | 0.1715 | 0.1848 | 0.1650 | 0.1452 | 0.1587 | **0.1320** |
| | 384 | 96 | 0.1616 | 0.1619 | 0.1584 | 0.1463 | 0.1393 | **0.1280** |
| | 480 | 96 | 0.1570 | 0.1703 | 0.1582 | 0.1401 | 0.1802 | **0.1250** |
| ETTm2 | 192 | 96 | 0.4294 | 0.4206 | 0.3405 | 0.3759 | 0.3135 | **0.3050** |
| | 288 | 96 | 0.3625 | 0.3882 | 0.3277 | 0.3472 | 0.3340 | **0.3180** |
| | 384 | 96 | 0.3389 | 0.3742 | 0.3562 | 0.3589 | 0.3434 | **0.3250** |
| | 480 | 96 | 0.3242 | 0.3597 | 0.3679 | 0.3353 | 0.4213 | **0.3150** |
| Electric | 48 | 24 | **0.5276** | 0.6617 | 0.6005 | 0.6098 | 0.6083 | 0.5450 |
| | 72 | 24 | **0.4953** | 0.6018 | 0.5454 | 0.5914 | 0.6238 | 0.5100 |
| | 96 | 24 | 0.4971 | 0.5260 | 0.5276 | 0.5139 | **0.4951** | 0.5050 |
| | 120 | 24 | 0.5196 | 0.4963 | **0.4900** | 0.5031 | 0.5101 | 0.4920 |
| Exchange | 14 | 7 | 0.0620 | 0.0784 | 0.0647 | 0.0555 | **0.0540** | 0.0795 |
| | 21 | 7 | 0.0599 | 0.0812 | 0.0743 | 0.0635 | **0.0556** | 0.0820 |
| | 28 | 7 | 0.0610 | 0.0844 | 0.0652 | **0.0595** | 0.0559 | 0.0850 |
| | 35 | 7 | 0.0629 | 0.0677 | 0.0632 | 0.0598 | **0.0558** | 0.0685 |
| Traffic | 48 | 24 | 0.4668 | 0.4887 | 0.4483 | 0.4718 | **0.4220** | 0.4350 |
| | 72 | 24 | 0.4635 | 0.4581 | 0.4196 | **0.3725** | 0.3873 | 0.3850 |
| | 96 | 24 | 0.4332 | 0.4082 | **0.3714** | 0.3787 | 0.4074 | 0.3750 |
| | 120 | 24 | 0.4161 | 0.3539 | **0.3542** | 0.3908 | 0.4125 | 0.3520 |
| Weather | 288 | 144 | 0.0331 | 0.0305 | 0.0354 | 0.0343 | 0.0352 | **0.0295** |
| | 432 | 144 | 0.0321 | 0.0302 | 0.0298 | 0.0346 | 0.0356 | **0.0285** |
| | 576 | 144 | 0.0328 | 0.0331 | 0.0321 | 0.0349 | 0.0284 | **0.0275** |
| | 720 | 144 | 0.0323 | 0.0353 | 0.0369 | 0.0335 | 0.0332 | **0.0260** |
| Avg. MAE | | | 0.2544 | 0.2659 | 0.2541 | **0.2512** | 0.2515 | 0.2523 |
| Avg. Rank | | | 5.5625 | 6.5000 | 5.7500 | 4.8438 | **4.4688** | 4.5313 |

benchmark datasets with varying historical context lengths and prediction horizons. The results of baseline model are obtained directly from [33]. MetaTCN achieves best performance on 11 out of 32 configurations (34.4%), demonstrating particular strength on ETTm1, ETTm2, and Weather datasets, and excelling on longer prediction horizons for 22.4% improvement. While foundation models show dataset-specific advantages, MetaTCN maintains more balanced performance across all domains with an average rank of 4.5313, closely trailing ChatTime and Chronos, and never ranking below 4th on any dataset. MetaTCN achieves an average MAE of 0.2523, within 0.44% of the best foundation model (Chronos: 0.2512), demonstrating that the memory mechanism provides performance comparable to billion-parameter pre-trained models while using only 3.87M parameters. Unlike foundation models requiring massive pre-training resources, MetaTCN achieves competitive zero-shot transfer through memory interweaving with only task-specific training, offering a practical and computationally efficient alternative for resource-constrained deployment scenarios while maintaining strong generalization capabilities that rival foundation models.

## REFERENCES


[1] J. Bi, H. Yuan, S. Li, K. Zhang, J. Zhang, and M. Zhou, "Arima-based and multiapplication workload prediction with wavelet decomposition and savitzky–golay filter in clouds," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 4, pp. 2495–2506, 2024.

[2] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[3] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications*, vol. 140, p. 112896, 2020.

[4] X. Zhang, C. Zhong, J. Zhang, T. Wang, and W. W. Ng, "Robust recurrent neural networks for time series forecasting," *Neurocomputing*, vol. 526, pp. 143–157, 2023.

[5] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *ArXiv*, vol. abs/2211.14730, 2022.

[6] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *International Conference on Learning Representations*, 2024.

[7] Z. Zhang, L. Meng, and Y. Gu, "Sageformer: Series-aware framework for long-term multivariate time series forecasting," *IEEE Internet of Things Journal*, 2024.

[8] R. Ilbert, A. Odonnat, V. Feofanov, A. Virmaux, G. Paolo, T. Palpanas, and I. Redko, "Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention," in *International Conference on Machine Learning*, 2024.

[9] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evalua-

tion of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[10] G. Spadon, S. Hong, B. Brandoli, S. Matwin, J. F. Rodrigues-Jr, and J. Sun, "Pay attention to evolution: Time series forecasting with deep graph-evolution learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5368–5384, 2022.

[11] R. Longo, G. Lacanna, L. Innocenti, and M. Ripepe, "Artificial intelligence and machine learning tools for improving early warning systems of volcanic eruptions: the case of stromboli," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1–10, 2024.

[12] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "Scinet: Time series modeling and forecasting with sample convolution and interaction," in *Advances in Neural Information Processing Systems*, 2022.

[13] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *International Conference on Learning Representations*, 2023.

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

[15] L. donghao and wang xue, "ModernTCN: A modern pure convolution structure for general time series analysis," in *International Conference on Learning Representations*, 2024.

[16] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[17] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[18] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[19] J. Venna and S. Kaski, "Neighborhood preservation in nonlinear projection methods: An experimental study," in *International Conference Vienna*, 2001, pp. 485–491.

[20] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *International Conference on Learning Representations*, 2023.

[21] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*, 2022, pp. 27 268–27 286.

[22] Z. Li, S. Qi, Y. Li, and Z. Xu, "Revisiting long-term time series forecasting: An investigation on linear mapping," in *ArXiv*, 2023.

[23] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, 2023, pp. 11 121–11 128.

[24] T. Zhang, Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, and J. Li, "Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures," *CoRR*, 2022.

[25] L. Chu, X. Bingjia, and Y. Qiping, "Unettsf: A better performance linear complexity time series prediction model," in *arXiv preprint arXiv:2401.03001*, 2024.

[26] W. Xue, T. Zhou, Q. Wen, J. Gao, B. Ding, and R. Jin, "Card: Channel aligned robust blend transformer for time series forecasting," in *International Conference on Machine Learning*, 2024.

[27] S.-A. Chen, C.-L. Li, S. O. Arik, N. C. Yoder, and T. Pfister, "Tsmixer: An all-mlp architecture for time series forecasting," *Transactions on Machine Learning Research*, 2023.

[28] S. Wang, J. Li, X. Shi, Z. Ye, B. Mo, W. Lin, S. Ju, Z. Chu, and M. Jin, "Timemixer++: A general time series pattern machine for universal predictive analysis," in *International Conference on Learning Representation-sInternational Conference on Learning Representations (ICLR)*, 2025.

[29] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *International Conference on Learning Representations*, 2024.

[30] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One fits all: Power general time series analysis by pretrained lm," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[31] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations*, 2024.

[32] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, "Long-term forecasting with tide: Time-series dense encoder," in *ArXiv*, 2023.

[33] C. Wang, Q. Qi, J. Wang, H. Sun, Z. Zhuang, J. Wu, L. Zhang, and J. Liao, "Chattime: A unified multimodal time series foundation model bridging numerical and textual data," in *AAAI Conference on Artificial Intelligence*, 2025.