TimeDistill: Efficient Long-Term Time Series Forecasting with MLP via Cross-Architecture Distillation

Juntong Ni [1] **Zewen Liu** [1] **Shiyu Wang** **Ming Jin** [2] **Wei Jin** [1]

Abstract

Transformer-based and CNN-based methods demonstrate strong performance in long-term time series forecasting. However, their high computational and storage requirements can hinder large-scale deployment. To address this limitation, we propose integrating lightweight MLP with advanced architectures using knowledge distillation (*KD*). Our preliminary study reveals different models can capture complementary patterns, particularly multi-scale and multi-period patterns in the temporal and frequency domains. Based on this observation, we introduce TIMEDISTILL, a cross-architecture *KD* framework that transfers these patterns from teacher models (e.g., Transformers, CNNs) to MLP. Additionally, we provide a theoretical analysis, demonstrating that our *KD* approach can be interpreted as a specialized form of *mixup* data augmentation. TIMEDISTILL improves MLP performance by up to 18.6%, surpassing teacher models on eight datasets. It also achieves up to 7× faster inference and requires 130× fewer parameters. Furthermore, we conduct extensive evaluations to highlight the versatility and effectiveness of TIMEDISTILL.

1. Introduction

Forecasting is a notably critical problem in the time series analysis community, which aims to predict future time series based on historical time series records (Wang et al., 2024b). It has broad practical applications such as climate modeling (Wu et al., 2023), traffic flow management (Yin et al., 2021), healthcare monitoring (Kaushik et al., 2020) and finance analytics (Granger & Newbold, 2014).

Recently, there has been an ongoing debate over which deep learning architecture best suits time series forecasting. The rise of Transformers in various domains (Devlin et al., 2018; Khan et al., 2022) has led to their wide adoption in time series forecasting (Wen et al., 2022; Wu et al., 2021; Zhou et al., 2022; 2021; Nie et al., 2023; Liu et al., 2024), leveraging the
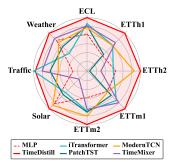

Figure 1. Performance comparison.

strong capabilities of capturing pairwise dependencies and extracting multi-level representations within sequential data. Similarly, CNN architectures have also proven effective by developing convolution blocks for time series (Luo & Wang, 2024; Wang et al., 2023). However, despite the strong performance of Transformer-based and CNN-based models, they face significant challenges in large-scale industrial applications due to their relatively high computational demands, especially in latency-sensitive scenarios like financial prediction and healthcare monitoring (Granger & Newbold, 2014; Kaushik et al., 2020). In contrast, simpler linear or MLP models offer greater efficiency, although with lower performance (Zeng et al., 2023; Lin et al., 2024). These contrasting observations raises an intriguing question:

> *Can we combine MLP with other advanced architectures (e.g., Transformers and CNNs) to create a powerful yet efficient model?*

A promising approach to addressing this question is knowledge distillation (*KD*) (Hinton, 2015), a technique that transfers knowledge from a larger and more complex model (*teacher*) to a smaller and simpler one (*student*) while maintaining comparable performance. In this work, we pioneer *cross-architecture KD* in time series forecasting, with MLP as the *student* and other advanced architectures (e.g., Transformers and CNNs) as the *teacher*. However, designing such a framework is non-trivial, as it remains unclear what "knowledge" should be distilled into MLP.

To investigate this potential, we conduct a comparative analysis of prediction patterns between MLP and other time series models. Our findings reveal that MLP still excels

[1]Department of Computer Science, Emory University, Atlanta, United States [2]School of Information and Communication Technology, Griffith University, Nathan, Australia. Correspondence to: Wei Jin <wei.jin@emory.edu>.

Preliminary work. Do not distribute.

on some data subsets despite its overall lower performance (Sec. 3.1), which highlights the value of harnessing the *complementary capabilities* across different architectures. To further explore the specific properties to distill, we focus on two key time series patterns: multi-scale pattern in temporal domain and multi-period pattern in frequency domain, given that they are vital in capturing the complex structures typical of many time series. **(1) Multi-Scale Pattern:** Real-world time series often show variations at multiple temporal scales. For example, hourly recorded traffic flow data capture changes within each day, while daily sampled data lose fine-grained details but reveal patterns related to holidays (Wang et al., 2024a). We observe that models that perform well on the finest scale also perform accurately on coarser scales, while MLP fails on most scales (Sec. 3.2). **(2) Multi-Period Pattern:** Time series often exhibit multiple periodicities. For instance, weather measurements may have both daily and yearly cycles, while electricity consumption data may show weekly and quarterly cycles (Wu et al., 2022). We find that models that perform well can capture periodicities similar to those in the ground truth, but MLP fails to capture these periodicities (Sec. 3.2). Therefore, enhancing MLP requires distilling and integrating these multi-scale and multi-period patterns from teacher models.

Based on our observations, we propose a cross-architecture *KD* framework named TIMEDISTILL to bridge the performance and efficiency gap between complex teacher models and a simple MLP. Instead of solely matching predictions in conventional *KD*, TIMEDISTILL focuses on aligning multi-scales and multi-period patterns between MLP and the teacher: we downsample the time series for temporal multi-scale alignment and apply Fast Fourier Transform (FFT) to align period distributions in the frequency domain. The *KD* process can be conducted offline, shifting heavy computations from the latency-critical *inference phase*, where millisecond matter, to the less time-sensitive *training phase*, where longer processing time is acceptable. We validate the effectiveness of TIMEDISTILL both theoretically and empirically and summarize our contributions as follows:

(a) We present *the first cross-architecture KD* framework TIMEDISTILL tailored for efficient and effective time series forecasting via MLP, which is supported by our preliminary studies examining multi-scale and multi-period patterns in time series.

(b) We provide theoretical insights into the benefits of TIMEDISTILL, illustrating that the proposed distillation process can be viewed as a form of data augmentation through a special *mixup* strategy.

(c) We show that TIMEDISTILL is both effective and efficient, consistently outperforming standalone MLP by up to **18.6%** and surpassing teacher models in nearly all cases (see Figure 1). Additionally, it achieves up to **7x** faster inference and requires up to **130×** fewer
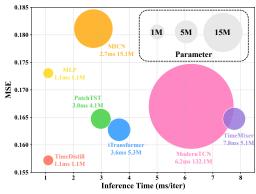

Figure 2. Model efficiency comparison averaged across all prediction lengths (96, 192, 336, 720) for the ECL dataset. Full results on more datasets are listed in Appendix I.

parameters compared to teacher models (see Figure 2).

(d) We conduct deeper analyses of TIMEDISTILL, exploring its adaptability across various teacher/student models and highlighting the distillation impacts it brings to the temporal and frequency domains.

2. Related Work

We present a brief summary of related work, with a detailed version provided in Appendix A. *A Debate in Long-Term Time Series Forecasting:* The trade-off between performance and efficiency has sparked debate between advanced architecture and MLP in long-term time series forecasting (Zeng et al., 2023; Zhang et al., 2022; Lin et al., 2024). Transformers (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2023; Liu et al., 2024) or CNNs (Luo & Wang, 2024; Wu et al., 2022; Wang et al., 2023) excel but face competition from a simple Linear or MLP model, which achieves comparable performance with greater efficiency (Zeng et al., 2023; Oreshkin et al., 2020; Challu et al., 2023; Zhang et al., 2022; Chen et al., 2023). To combine these strengths, we propose cross-architecture knowledge distillation (*KD*) to achieve both high performance and efficiency. *KD in Time Series:* CAKD (Xu et al., 2022) uses adversarial and contrastive learning for feature distillation without specific design for time series, while LightTS (Campos et al., 2023) designs a *KD* framework for ensemble classifiers, limiting its generality. Unlike these, our framework targets time series-specific patterns, such as multi-scale and multi-period, pioneering cross-architecture *KD*.

3. Preliminary Studies

In this section, we explore the reasons behind adopting distillation (***What motivates distillation?***) and investigate the specific time series information to distill into the MLP model (***What should we distill?***). We first introduce key notations. For multivariate long-term time series forecasting, given an input time series $\mathbf{X} \in \mathbb{R}^{T \times C}$, where T represents
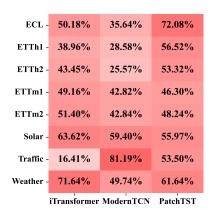
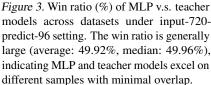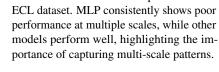*Figure 3.* Win ratio (%) of MLP v.s. teacher models across datasets under input-720-predict-96 setting. The win ratio is generally large (average: 49.92%, median: 49.96%), indicating MLP and teacher models excel on different samples with minimal overlap.

Figure 4. Visualization of model predictions on different downsampled scales of ECL dataset. MLP consistently shows poor performance at multiple scales, while other models perform well, highlighting the importance of capturing multi-scale patterns.
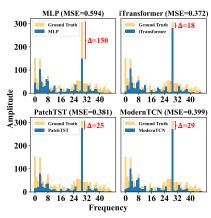

Figure 5. Prediction spectrograms of various models on ECL dataset against the ground truth. MLP fails to match the amplitudes of several main frequencies in the ground truth, with red numbers indicating amplitude differences for the most significant frequency.

the length of the look-back window and C represents the number of variables, the goal is to predict the future S time steps $\mathbf{Y} \in \mathbb{R}^{S \times C}$.

3.1. What Motivates Distillation?

While MLP excel in efficiency, they often lag in performance compared to Transformer and CNN models or achieve only similar results (Figure 1). However, even if MLP underperforms its teachers overall, it may excel on specific samples. To investigate this, we compare prediction errors of MLP's prediction $\hat{\mathbf{Y}}_s$ and teacher model's prediction $\hat{\mathbf{Y}}_t$ for N samples, defined as $e_s = \{\text{MSE}(\hat{\mathbf{Y}}_s^1, \mathbf{Y}^1), \cdots, \text{MSE}(\hat{\mathbf{Y}}_s^N, \mathbf{Y}^N)\}$ and $e_t = \{\text{MSE}(\hat{\mathbf{Y}}_t^1, \mathbf{Y}^1), \cdots, \text{MSE}(\hat{\mathbf{Y}}_t^N, \mathbf{Y}^N)\}$. We calculate the win ratio, indicating where MLP outperforms teachers:

$$\text{Win Ratio} = \sum \mathbb{1}(e_s < e_t)/N, \tag{1}$$

where $\mathbb{1}(\cdot)$ equals 1 if MLP outperforms the teacher, otherwise 0. As shown in Figure 3, the win ratio is high (average: 49.92%, median: 49.96%), despite MLP underperforming teacher models overall. For example, on the Traffic dataset, MLP lags behind ModernTCN but wins on 81.19% of the samples, indicating differing strengths on distinct subsets. **This highlights the potential of distilling the complementary knowledge from the teacher model into MLP.**

An intuitive strategy of *KD* for MLP is to align predictions with teachers, but this faces limitations. **First**, it risks overfitting noise in the teacher's predictions, leading to less stable knowledge (Chen et al., 2017; Takamoto et al., 2020; Gajbhiye et al., 2021). **Second**, MLP may struggle to replicate the complex patterns such as seasonality, trends, and periodicity in teacher predictions directly due to their limited capacity. **Third**, this approach overlooks valuable knowledge

from intermediate features of teacher model. Therefore, the specific knowledge to distill into MLP requires further exploration.

3.2. What Should We Distill?

To investigate which complementary time series patterns to distill into MLP, we analyze prediction patterns in both temporal and frequency domains, considering real-world variations across temporal scales (Wang et al., 2024a) and periodicities (Wu et al., 2022). Thus, we conduct further analysis of these two patterns by presenting illustrative cases. The implementation is detailed in Appendix D. **(1) Multi-Scale Pattern:** By downsampling predictions $\hat{\mathbf{Y}}$ using convolutional operations, we obtain their multi-scale representations. Figure 4 shows the analysis of multi-scale temporal patterns by downsampling the time series (*Scale 0*) to coarser scales (*Scales 1–3*). Models excelling at *Scale 0* generally perform well across all scales. The teacher models align closely with the ground truth at *Scale 3*, capturing underlying trends effectively. In contrast, MLP significantly deviates, showing its limitations in handling multi-scale patterns. **(2) Multi-Period Pattern:** Periodicities in time series are visible in the frequency domain by transforming time series into spectrograms, with the x-axis representing frequency and periodicity calculated as the time series length S divided by frequency. Figure 5 shows spectrograms of model predictions and the ground truth. Models with lower MSE capture periodicities more accurately, closely matching the ground truth at dominant frequencies. In contrast, MLP predictions show larger discrepancies, highlighting its inability to effectively capture periodic patterns.

From our observations, we conjecture that MLP underperforms on certain samples due to their inability to capture essential multi-scale and multi-period patterns, which are

essential for effective time series forecasting. To improve MLP, it is crucial to incorporate these complementary patterns from teacher models during distillation.

4. Methodology

Motivated by our preliminary studies, we propose a novel *KD* framework TIMEDISTILL for time series to transfer the knowledge from a fixed, pretrained teacher model f_t to a student MLP model f_s. The student produces predictions $\hat{\mathbf{Y}}_s \in \mathbb{R}^{S \times C}$ and internal features $\mathbf{H}_s \in \mathbb{R}^{D \times C}$. The teacher model produces predictions $\hat{\mathbf{Y}}_t \in \mathbb{R}^{S \times C}$ and internal features $\mathbf{H}_t \in \mathbb{R}^{D_t \times C}$. Our general objective is:

$$\min_{\theta_s} \mathcal{L}_{sup}(\mathbf{Y}, \hat{\mathbf{Y}}_s) + \mathcal{L}_{\mathrm{KD}}^{\mathbf{Y}}(\hat{\mathbf{Y}}_t, \hat{\mathbf{Y}}_s) + \mathcal{L}_{\mathrm{KD}}^{\mathbf{H}}(\mathbf{H}_t, \mathbf{H}_s), \quad (2)$$

where θ_s is the parameter of the student; $\mathcal{L}_{sup}$ is the supervised loss (e.g., MSE) between predictions and ground truth; $\mathcal{L}_{\mathrm{KD}}^{\mathbf{Y}}$ and $\mathcal{L}_{\mathrm{KD}}^{\mathbf{H}}$ are the distillation loss terms that encourage the student model to learn knowledge from the teacher on both **prediction level** (Hinton, 2015) and **feature level** (Romero et al., 2014), respectively. Unlike conventional approaches that emphasize matching model predictions, TIMEDISTILL integrates key time-series patterns including multi-scale and multi-period knowledge. The overall framework of TIMEDISTILL is shown in Figure 6.

4.1. Multi-Scale Distillation

One key component of TIMEDISTILL is multi-scale distillation, where "multi-scale" refers to representing the same time series at different sampling rates. This enables MLP to effectively capture both coarse-grained and fine-grained patterns. By distilling at both the prediction level and the feature level, we ensure that MLP not only replicates the teacher's multi-scale predictions but also aligns with its internal representations from the intermediate layer.

Prediction Level. At the prediction level, we directly transfer multi-scale signals from the teacher's outputs to guide the MLP's predictions. We first produce multi-scale predictions by downsampling the original predictions from the teacher $\hat{\mathbf{Y}}_t \in \mathbb{R}^{S \times C}$ and the MLP $\hat{\mathbf{Y}}_s \in \mathbb{R}^{S \times C}$, where S is the prediction length and C is the number of variables. The predictions at *Scale 0* are equal to the original predictions, that is, $\hat{\mathbf{Y}}_t^0 = \hat{\mathbf{Y}}_t$ and $\hat{\mathbf{Y}}_s^0 = \hat{\mathbf{Y}}_s$. We then downsample these predictions across M scales using convolutional operations with a stride of 2, generating multi-scale prediction sets $\mathcal{Y}_t = \{\hat{\mathbf{Y}}_t^0, \hat{\mathbf{Y}}_t^1, \cdots, \hat{\mathbf{Y}}_t^M\}$ and $\mathcal{Y}_s = \{\hat{\mathbf{Y}}_s^0, \hat{\mathbf{Y}}_s^1, \cdots, \hat{\mathbf{Y}}_s^M\}$, where $\hat{\mathbf{Y}}_t^M, \hat{\mathbf{Y}}_s^M \in \mathbb{R}^{\lfloor S/2^M \rfloor \times C}$. The downsampling is defined as:

$$\hat{\mathbf{Y}}_x^m = \mathrm{Conv}(\hat{\mathbf{Y}}_x^{m-1}, \mathrm{stride} = 2), \quad (3)$$

where $x \in \{t, s\}$, $m \in \{1, \cdots, M\}$, Conv denotes a 1D-convolutional layer with a temporal stride of 2. The predictions at the lowest level $\hat{\mathbf{Y}}_x^0 = \hat{\mathbf{Y}}_x$ maintain the original

temporal resolution, while the highest-level predictions $\hat{\mathbf{Y}}_x^M$ represent coarser patterns. We define the multi-scale distillation loss at the prediction level as:

$$\mathcal{L}_{scale}^{\mathbf{Y}} = \sum_{m=0}^{M} ||\hat{\mathbf{Y}}_t^m - \hat{\mathbf{Y}}_s^m||^2 / (M + 1). \quad (4)$$

Here we use MSE loss to match the MLP's predictions to the teacher's predictions at multiple scales.

Feature Level. At the feature level, we align MLP's intermediate features with teacher's multi-scale representations, enabling MLP to form richer internal structures that support more accurate forecasts. Let $\mathbf{H}_s \in \mathbb{R}^{D \times C}$ and $\mathbf{H}_t \in \mathbb{R}^{D_t \times C}$ denote MLP and teacher features with feature dimensions D and D_t, respectively. As their dimensions can be different, we first use a parameterized regressor (e.g. MLP) to align their feature dimensions:

$$\mathbf{H}_t' = \mathrm{Regressor}(\mathbf{H}_t), \quad (5)$$

where $\mathbf{H}_t' \in \mathbb{R}^{D \times C}$. Similar to the prediction level, we compute $\mathbf{H}_x^m$ by downsampling $\mathbf{H}_s$ and $\mathbf{H}_t'$ across multiple scales using the same approach as in Equation 3. We define the multi-scale distillation loss at the feature level as:

$$\mathcal{L}_{scale}^{\mathbf{H}} = \sum_{m=0}^{M} ||\mathbf{H}_t^m - \mathbf{H}_s^m||^2 / (M + 1). \quad (6)$$

4.2. Multi-Period Distillation

In addition to multi-scale distillation in the temporal domain, we further propose multi-period distillation to help MLP learn complex periodic patterns in the frequency domain. By aligning periodicity-related signals from the teacher model at both the prediction and feature levels, the MLP can learn richer frequency-domain representations and ultimately improve its forecasting performance.

Prediction Level. For the predictions from the teacher $\hat{\mathbf{Y}}_t \in \mathbb{R}^{S \times C}$ and the MLP $\hat{\mathbf{Y}}_s \in \mathbb{R}^{S \times C}$, we first identify their periodic patterns. We perform this in the frequency domain using the Fast Fourier Transform (FFT):

$$\mathbf{A}_x = \mathrm{Amp}(\mathrm{FFT}(\hat{\mathbf{Y}}_x)), \quad (7)$$

where $x \in \{t, s\}$ and spectrograms $\mathbf{A}_x \in \mathbb{R}^{\frac{S}{2} \times C}$. Here, FFT$(\cdot)$ denotes the FFT operation and Amp$(\cdot)$ calculates the amplitude. We remove the direct current (DC) component from $\mathbf{A}_x$. For certain variable c, the i-th value $\mathbf{A}_x^{i;c}$ indicates the intensity of the frequency-i component, corresponding to a period length $\lceil S/i \rceil$. Larger amplitude values indicate that the associated frequency (period) is more significant.

To reduce the influence of minor frequencies and avoid noise introduced by less meaningful frequencies (Wu et al., 2022; Zhou et al., 2022), we propose a distribution-based
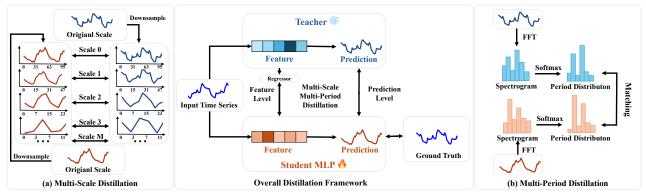
Figure 6. Overall framework of TIMEDISTILL, which distills knowledge from a teacher model to a student MLP using (a) Multi-Scale Distillation and (b) Multi-Period Distillation at both feature and prediction levels. (a) Multi-Scale Distillation involves downsampling the original time series into multiple coarser scales and aligning these scales between the student and teacher. (b) Multi-Period Distillation applies FFT to transform the time series into a spectrogram, followed by matching the period distributions after applying softmax.

matching scheme. We use a softmax function with a colder temperature to highlight the most significant frequencies:

$$\mathbf{Q}_x^{\mathbf{Y}} = \exp\bigl(\mathbf{A}_x^i/\tau\bigr)\Big/\sum\nolimits_{j=1}^{S/2} \exp\bigl(\mathbf{A}_x^j/\tau\bigr), \qquad (8)$$

where $\mathbf{Q}_x^{\mathbf{Y}} \in \mathbb{R}^{\frac{S}{2} \times C}$ and τ is a temperature parameter that controls the sharpness of the distribution. We set $\tau = 0.5$ by default. The period distribution $\mathbf{Q}_x^{\mathbf{Y}}$ represents the multi-period pattern in the prediction time series, which we want the MLP to learn from the teacher. We use KL divergence to match these distributions (Hinton, 2015). We define the multi-period distillation loss at the prediction level as:

$$\mathcal{L}_{period}^{\mathbf{Y}} = \mathrm{KL}\bigl(\mathbf{Q}_t^{\mathbf{Y}}, \mathbf{Q}_s^{\mathbf{Y}}\bigr). \qquad (9)$$

Feature Level. Similar to the prediction level, we apply multi-period distillation at the feature level. For the features $\mathbf{H}_t' \in \mathbb{R}^{D \times C}$ and $\mathbf{H}_s \in \mathbb{R}^{D \times C}$, we compute the spectrograms and the corresponding period distributions $\mathbf{Q}_x^{\mathbf{H}}$ using the same approach as in Equations 7 and 8. Multi-period distillation loss at feature level is then defined as:

$$\mathcal{L}_{period}^{\mathbf{H}} = \mathrm{KL}\bigl(\mathbf{Q}_t^{\mathbf{H}}, \mathbf{Q}_s^{\mathbf{H}}\bigr). \qquad (10)$$

4.3. Overall Optimization and Theoretical Analaysis

During the training of TIMEDISTILL, we jointly optimize both the multi-scale and multi-period distillation losses at both the prediction and feature levels, together with the supervised ground-truth label loss:

$$\mathcal{L}_{sup} = ||\mathbf{Y} - \hat{\mathbf{Y}}_s||^2, \qquad (11)$$

where $\mathcal{L}_{sup}$ is the ground-truth loss (for example, MSE loss) used to train MLP directly. Thus, the overall training loss for the student MLP is defined as:

$$\mathcal{L} = \mathcal{L}_{sup} + \alpha \cdot \bigl(\mathcal{L}_{scale}^{\mathbf{Y}} + \mathcal{L}_{period}^{\mathbf{Y}}\bigr) + \beta \cdot \bigl(\mathcal{L}_{scale}^{\mathbf{H}} + \mathcal{L}_{period}^{\mathbf{H}}\bigr), \qquad (12)$$

where α and β are hyper-parameters that control the contributions of the prediction-level and feature-level distillation loss terms, respectively. The teacher model is pretrained and remains frozen throughout the training process of MLP.

Theoretical Interpretations. We provide a theoretical understanding of multi-scale and multi-period distillation loss from **a novel data augmentation perspective**. We further show that the proposed distillation loss can be interpreted as training with augmented samples derived from a special *mixup* (Zhang, 2017) strategy. The distillation process augments data by blending ground truth with teacher predictions, analogous to label smoothing in classification, and provides several benefits for time series forecasting: *(1) Enhanced Generalization:* It enhances generalization by exposing the student model to richer supervision signals from augmented samples, thus mitigating overfitting, especially with limited or noisy data. *(2) Explicit Integration of Patterns:* The augmented supervision signals explicitly incorporate patterns across multiple scales and periods, offering insights that are not immediately evident in the raw ground truth. *(3) Stabilized Training Dynamics:* The blending of targets softens the supervision signals, which diminishes the model's sensitivity to noise and leads to more stable training phases. This will in turn support smoother optimization dynamics and fosters improved convergence. For clarity, our discussion is centered at the prediction level. We present the following theorem:

Theorem 4.1. *Let (x, y) denote original input data pairs and (x, y^t) represent corresponding teacher data pairs. Consider a data augmentation function $\mathcal{A}(\cdot)$ applied to (x, y), generating augmented samples (x', y'). Define the training loss on these augmented samples as $\mathcal{L}_{aug} = \sum_{(x',y')\in\mathcal{A}(x,y)} |f_s(x') - y'|^2$. Then, the following inequality holds: $\mathcal{L}_{sup} + \eta\mathcal{L}_{scale} \geq \mathcal{L}_{aug}$, when $\mathcal{A}(\cdot)$ is instantiated as a mixup function (Zhang, 2017) that interpolates between the original input data (x, y) and teacher data (x, y^t) with a mixing coefficient $\lambda = \frac{1}{1+\eta}$, i.e. $y' = \lambda y + (1-\lambda)y^t$.*

5

We provide proof of Theorem 4.1 in Appendix B. Theorem 4.1 suggests that optimizing multi-scale distillation loss $\mathcal{L}_{\text{scale}}$ jointly with supervised loss $\mathcal{L}_{\text{sup}}$ is equivalent to minimizing an upper bound on a special *mixup* augmentation loss. In particular, we mix multi-scale teacher predictions $\{\hat{\mathbf{Y}}_t^{(m)}\}_{m=0}^M$ with ground truth $\mathbf{Y}$, thereby allowing MLP to learn more informative time series temporal pattern. Similarly, we present a theorem for understanding $\mathcal{L}_{period}$.

Theorem 4.2. *Define the training loss on the augmented samples using KL divergence as* $\mathcal{L}_{aug} = \sum_{(x',y')\in\mathcal{A}(x,y)} KL(y', \mathcal{X}(f_s(x')))$, *where* $\mathcal{X}(\cdot) = Softmax(FFT(\cdot))$. *Then, the following inequality holds:* $\mathcal{L}_{sup} + \eta\mathcal{L}_{period} \geq \mathcal{L}_{aug}$, *where* $\mathcal{A}(\cdot)$ *is instantiated as a mixup function that interpolates between the period distribution of original input data* $(x, \mathcal{X}(y))$ *and teacher data* $(x, \mathcal{X}(y^t))$ *with a mixing coefficient* $\lambda = \eta$, *i.e.* $y' = \mathcal{X}(y) + \lambda\mathcal{X}(y^t)$.

The proof can be found in Appendix B. Theorem 4.2 shows that optimizing the multi-period distillation loss $\mathcal{L}_{\text{period}}$ jointly with the supervised loss $\mathcal{L}_{\text{sup}}$ is equivalent to minimizing an upper bound on the KL divergence between the student period distribution $\mathcal{X}(f_s(x'))$ (or $\mathbf{Q}_s$) and a *mixed* period distribution y' (or $\mathbf{Q}_y + \lambda\,\mathbf{Q}_t$).

5. Experiment

5.1. Experimental Setup

Datasets and Baselines. We run experiments to evaluate the performance and efficiency of TIMEDISTILL on 8 widely used benchmarks: Electricity (ECL), the ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Solar, Traffic, and Weather, following (Liu et al., 2024; Wang et al., 2024a; Luo & Wang, 2024). To examine the effectiveness of our method across diverse tasks, we compare TIMEDISTILL with 8 baseline models that cover a range of architectures. Specifically, we use *Transformer-based models*: iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022), Autoformer (Wu et al., 2021); *CNN-based models*: ModernTCN (Luo & Wang, 2024), MICN (Wang et al., 2023), TimesNet (Wu et al., 2022); and an *MLPs-based model*: TimeMixer (Wang et al., 2024a). More details can be found in Appendix C.

5.2. Main Results

Table 1 presents the long-term time series forecasting performance of the proposed TIMEDISTILL compared with previous state-of-the-art baselines. By default, TIMEDISTILL uses ModernTCN as the teacher, though results with alternative teachers are provided in Sec. 5.3. Notably, TIMEDISTILL outperforms the baselines on **7 out of 8** datasets on MSE and **all** datasets on MAE. Furthermore, TIMEDISTILL consistently exceeds the performance of its teacher (Mod-

Figure 7. Forecasting performance is evaluated with different look-back lengths of $T \in \{96, 192, 336, 720\}$, and the results are averaged across all prediction lengths $S \in \{96, 192, 336, 720\}$ on eight datasets. The complete results are provided in Appendix N.

ernTCN) by up to **5.37%** and improves over vanilla MLP by up to **13.87%**, as shown in Table 2. These results highlight the effectiveness of our multi-scale and multi-period distillation approach in transferring knowledge for enhanced forecasting performance.

Efficiency Analysis. Beyond its strong predictive performance, another notable advantage of TIMEDISTILL is its extremely lightweight architecture, as it is simply an MLP. Figure 2 in the Introduction section shows the trade-off between inference time, memory footprint, and performance. We can observe that TIMEDISTILL can achieve up to **7×** speedup and up to **130×** fewer parameters compared with baselines. This property makes TIMEDISTILL suitable for deployment on devices with limited computational resources and in latency-sensitive applications that require fast inference. Compared to previous Transformer-based method Autoformer, we achieve **196×** speedup as shown in Table 10. We list full results of efficiency analysis in Appendix I.

5.3. Versatility of TIMEDISTILL

In this subsection, we further demonstrate the versatility of the proposed TIMEDISTILL by evaluating its performance under different configurations, including variations in teacher/student models and look-back window lengths.

Different Teachers. We use ModernTCN as the teacher model for our main results. In Table 2, we present results using other teacher models, such as iTransformer, TimeMixer, and PatchTST. Results with additional teachers are provided in Appendix E. We observe that TIMEDISTILL can effectively learn from various teachers, improving MLP by up to **13.87%**. Furthermore, TIMEDISTILL achieves significant performance improvements over the teachers themselves, with gains of up to **21.41%**. We hypothesize two key reasons for these improvements. First, the student MLP model already demonstrates strong learning capabilities; for instance, on the Solar dataset, MLP outperforms most teach-

Table 1. Long-term time series forecasting results with prediction lengths $S \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates a better prediction. For consistency, we maintain a fixed input length of 720 throughout all the experiments. Results are averaged from all prediction lengths. The best performance is highlighted in **red**, and the second-best is <u>underlined</u>. Full results are listed in Appendix M.

Models	TIMEDISTILL (**Ours**)		iTranformer (2024)		ModernTCN (2024)		TimeMixer (2024a)		PatchTST (2023)		MICN (2023)		FEDformer (2022)		TimesNet (2022)		Autoformer (2021)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	**0.157**	**0.254**	<u>0.163</u>	0.259	0.167	0.262	0.165	<u>0.259</u>	0.165	0.266	0.181	0.293	0.274	0.376	0.250	0.347	0.238	0.347
ETTh1	**0.429**	**0.441**	0.468	0.476	0.469	0.465	<u>0.459</u>	<u>0.465</u>	0.498	0.490	0.739	0.631	0.527	0.524	0.507	0.500	0.731	0.659
ETTh2	**0.345**	**0.395**	0.398	0.426	<u>0.357</u>	<u>0.403</u>	0.422	0.444	0.444	0.443	1.078	0.736	0.456	0.485	0.419	0.446	1.594	0.940
ETTm1	**0.348**	**0.379**	0.372	0.402	0.390	0.410	<u>0.367</u>	<u>0.388</u>	0.383	0.412	0.439	0.461	0.423	0.451	0.398	0.419	0.570	0.526
ETTm2	**0.244**	**0.311**	0.276	0.337	<u>0.267</u>	<u>0.330</u>	0.279	0.339	0.274	0.335	0.348	0.404	0.359	0.401	0.291	0.349	0.420	0.448
Solar	**0.184**	**0.241**	0.214	0.270	<u>0.191</u>	<u>0.243</u>	0.238	0.288	0.210	0.257	0.213	0.277	0.300	0.383	0.196	0.262	1.037	0.742
Traffic	<u>0.387</u>	**0.271**	**0.379**	<u>0.271</u>	0.413	0.284	0.391	0.275	0.402	0.284	0.500	0.316	0.629	0.388	0.693	0.399	0.696	0.427
Weather	**0.220**	**0.269**	0.259	0.290	0.238	0.277	<u>0.230</u>	<u>0.271</u>	0.246	0.283	0.240	0.292	0.355	0.398	0.257	0.294	0.471	0.465

Table 2. Performance improvement by **TIMEDISTILL** with **different teachers**. Δ_{MLP}, $\Delta_{Teacher}$ indicate the improvement of **MLP+TIMEDISTILL** over a trained MLP and Teacher, respectively. We report average MSE across all prediction lengths. Full results are in Appendix E.

Teacher Models		iTranformer (2024)	ModernTCN (2024)	TimeMixer (2024a)	PatchTST (2023)
ECL	Teacher	0.163	0.167	0.165	0.165
	MLP	0.173	0.173	0.173	0.173
	+TIMEDISTILL	**0.157**	**0.157**	**0.159**	**0.159**
	$\Delta_{Teacher}$	3.68%	5.61%	3.31%	3.64%
	Δ_{MLP}	9.27%	9.09%	7.94%	8.11%
ETT(avg)	Teacher	0.379	0.371	0.382	0.400
	MLP	0.397	0.397	0.397	0.397
	+TIMEDISTILL	**0.345**	**0.342**	**0.353**	**0.358**
	$\Delta_{Teacher}$	8.92%	7.94%	7.46%	10.38%
	Δ_{MLP}	13.06%	13.87%	10.91%	9.65%
Solar	Teacher	0.214	0.191	0.288	0.210
	MLP	0.194	0.194	0.194	**0.194**
	+TIMEDISTILL	**0.185**	**0.184**	**0.187**	0.204
	$\Delta_{Teacher}$	13.55%	3.60%	21.41%	2.86%
	Δ_{MLP}	4.80%	5.14%	3.58%	-4.98%
Traffic	Teacher	**0.379**	0.413	0.391	0.402
	MLP	0.434	0.434	0.434	0.434
	+TIMEDISTILL	0.389	**0.387**	**0.391**	**0.390**
	$\Delta_{Teacher}$	-2.64%	6.32%	-0.04%	2.99%
	Δ_{MLP}	10.30%	10.70%	9.76%	10.07%
Weather	Teacher	0.259	0.238	0.230	0.246
	MLP	0.234	0.234	0.234	0.234
	+TIMEDISTILL	**0.220**	**0.220**	**0.219**	**0.220**
	$\Delta_{Teacher}$	15.06%	7.37%	4.82%	10.57%
	Δ_{MLP}	5.83%	5.66%	6.16%	5.83%

Table 3. Performance improvement of **TIMEDISTILL** with **different students** on ETTh1, averaged across all prediction lengths. $\Delta_{Student}$ represents the improvement of **Student+TIMEDISTILL** over the original student. Additional results are in Appendix F.

Student Models	MLP		LightTS (2022)		TSMixer (2023)	
	MSE	MAE	MSE	MAE	MSE	MAE
Student	0.502	0.489	0.465	0.471	0.471	0.474
+TIMEDISTILL	**0.428**	**0.445**	**0.436**	**0.445**	**0.433**	**0.446**
$\Delta_{Student}$	14.74%	9.00%	6.26%	5.57%	8.02%	5.92%

use ModernTCN as the teacher model, consistent with other experiments. As shown in Table 3, TIMEDISTILL consistently improves the performance of TSMixer and LightTS, achieving remarkable MSE reductions of **6.26%** and **8.02%**, respectively. These results demonstrate that TIMEDISTILL is highly adaptable and can effectively enhance other lightweight methods. We also explore the influence of student MLP architecture (e.g. layer number and hidden dimension) on TIMEDISTILL in Appendix F.

Different Look-Back Window Lengths. The length of the look-back window significantly influences forecasting accuracy, as it determines how much historical information can be utilized for learning. Figure 7 presents the average MSE results across all eight datasets. Overall, the performance of all models, particularly MLP, improves as the look-back window size increases. Notably, TIMEDISTILL consistently enhances the performance of MLP and outperforms the teacher models across all look-back window lengths.

5.4. Deeper Analysis into Our Distillation Framework

Ablation study. As our proposed TIMEDISTILL incorporates two distillation strategies, *multi-scale distillation* and *multi-period distillation*, we assess their effectiveness by removing the corresponding losses, $\mathcal{L}_{\text{scale}}$ and $\mathcal{L}_{\text{period}}$, from TIMEDISTILL. Additionally, we evaluate the impact of *prediction-level* and *feature-level distillation* by removing $\mathcal{L}^{\mathbf{H}}$ and $\mathcal{L}^{\mathbf{Y}}$, respectively. Furthermore, we test the model by removing the supervised loss $\mathcal{L}_{\text{sup}}$, using only the distillation losses as the overall loss for TIMEDISTILL.

ers. Second, the multi-scale and multi-period *KD* approach delivers both temporal and frequency knowledge from teachers to MLP, offering additional valuable insights. Similar findings have been reported in recent *KD* studies (Allen-Zhu & Li, 2020; Zhang et al., 2021; Guo et al., 2023), which suggest that integrating diverse views from multiple models can enhance performance.

Different Students. To evaluate whether TIMEDISTILL can enhance the performance of other lightweight models, we select two MLPs-based models, TSMixer (Chen et al., 2023) and LightTS (Zhang et al., 2022), as students. Unlike a simple MLP, these MLPs-based models consist of multiple MLPs and operate in a channel-dependent manner. Consequently, they are generally more complex than a standard MLP, which results in reduced efficiency. We

Table 4. Ablation study measured by MSE on different components of TIMEDISTILL (Teacher: ModernTCN). The best performance is in **red**, and the second-best is <u>underlined</u>. More ablation study results are listed in Appendix O.

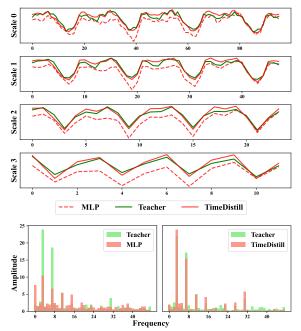| Method	ECL	ETT(avg)	Solar	Traffic	Weather
Teacher	0.167	0.371	0.191	0.413	0.238
MLP	0.173	0.397	0.194	0.434	0.234
TIMEDISTILL	**0.157**	**0.342**	**0.184**	**0.387**	**0.220**
w/o prediction level	<u>0.157</u>	0.373	<u>0.184</u>	0.392	<u>0.221</u>
w/o feature level	0.161	0.349	0.188	0.393	0.224
w/o multi-scale	0.162	0.377	0.187	0.393	0.224
w/o multi-period	0.157	<u>0.342</u>	0.184	<u>0.391</u>	0.221
w/o sup	0.165	0.344	0.192	0.506	0.225

Figure 8. Prediction comparison across temporal scales and spectrograms before and after distillation on ETTh1. MSE for MLP, Teacher (ModernTCN), and TIMEDISTILL are 0.790, 0.365, and 0.366, showing TIMEDISTILL bridges temporal and frequency domain gaps via multi-scale and multi-period distillation. More cases in Appendix K.

Table 4 presents the results of these ablations, compared with the full TIMEDISTILL, a stand-alone MLP, and the teacher models. We draw the following observations: **First**, each loss term at both levels, when used individually, already outperforms the stand-alone MLP. **Second**, when combined, the losses complement each other, enabling TIMEDISTILL to achieve the best performance, surpassing the teacher model. **Third**, notably, TIMEDISTILL maintains superior performance over both the MLP and the teacher even without the supervised loss $\mathcal{L}_{sup}$. This can be attributed to two potential reasons: (1) ground truth may contain noise, making it more challenging to fit, whereas teacher provides simpler and more learnable knowledge; and (2) *multi-scale* and *multi-period* distillation processes effectively transfer complementary knowledge from teacher to MLP.

Table 5. Win ratio (%) of MLP and TIMEDISTILL vs. ModernTCN under input-720-predict-96 setting. U_M and U_T denote samples where MLP and TIMEDISTILL outperform the teacher. *Win Keep*, $\frac{|U_M \cap U_T|}{|U_M|}$, shows TIMEDISTILL's retention of MLP's wins.

Dataset	MLP	TIMEDISTILL	*Win Keep*
ECL	35.64%	59.69%	95.44%
ETTh1	28.58%	58.35%	79.27%
ETTh2	25.57%	57.63%	69.10%
ETTm1	42.82%	63.12%	84.61%
ETTm2	42.84%	57.99%	80.14%
Solar	59.40%	61.02%	91.77%
Traffic	81.19%	91.24%	99.46%
Weather	49.74%	56.43%	76.61%

Does TIMEDISTILL Truly Enhance MLP's Learning from the Teacher? As shown in Table 5, TIMEDISTILL improves the win ratio of MLP over the teacher by an average of 17.46% across eight datasets. To determine whether this improvement is due to TIMEDISTILL succeeding on samples MLP previously failed, we present the *Win Keep* results. A higher *Win Keep* indicates that TIMEDISTILL's improvements come from previously failed samples. *Win Keep* remains consistently high across datasets (above 76.61%, with an average of 84.55%), indicating that TIMEDISTILL not only retains MLP's success on samples where it already outperformed the teacher but also allows MLP to outperform the teacher on many samples it previously struggled with. This demonstrates that TIMEDISTILL effectively transfers knowledge from the teacher to student MLP, enabling MLP to perform better on challenging samples while maintaining its existing strengths.

Does TIMEDISTILL Effectively Bridge the Gap Between Student and Teacher? We present a case visualization from the ETTh1 dataset in Figure 8. The figure shows that TIMEDISTILL reduces the difference between the teacher model (ModernTCN) and the student model (MLP) at multiple scales, and makes the fine-level series prediction more precise by transferring trend patterns at coarser scales from the teacher. In addition, for the same case, we visualize the spectrogram of the prediction and find that TIMEDISTILL also helps reduce the difference between the teacher and MLP in the frequency domain, enabling the MLP to learn more structured multi-period patterns.

6. Conclusion and Future work

We propose TIMEDISTILL, a cross-architecture *KD* framework enabling lightweight MLP to surpass complex teachers. By distilling multi-scale and multi-period patterns, TIMEDISTILL efficiently transfers temporal and frequency-domain knowledge. Theoretical interpretations and experiments confirm its effectiveness. Future work includes distilling from advanced time series models, e.g. time series foundation models, and incorporating multivariate patterns.

7. Impact Statement

Transparency Impact. Transparency is vital to facilitate accountability, competition, and collective understanding (Bommasani et al., 2024). To support these objectives, we will publicly release our code base, data sources, and evaluation pipeline. However, due to the requirements of anonymized submission, these materials are currently included only in the supplementary material.

Environmental Impact. TIMEDISTILL achieves fast inference speed, which significantly reduces its carbon footprint by lowering GPU usage. This reduction also contributes to decreased carbon emissions from the local electricity grid.

Ethical Impact. While TIMEDISTILL performs well in long-term time series forecasting, caution is needed in critical and high-stakes domains like healthcare, where incorrect predictions could have severe consequences.

References

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Bommasani, R., Klyman, K., Kapoor, S., Longpre, S., Xiong, B., Maslej, N., and Liang, P. The foundation model transparency index v1. 1: May 2024. *arXiv preprint arXiv:2407.12929*, 2024.

Campos, D., Zhang, M., Yang, B., Kieu, T., Guo, C., and Jensen, C. S. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2):1–27, 2023.

Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., and Dubrawski, A. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Chen, G., Choi, W., Yu, X., Han, T., and Chandraker, M. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

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

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., and Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

Gajbhiye, A., Fomicheva, M., Alva-Manchego, F., Blain, F., Obamuyide, A., Aletras, N., and Specia, L. Knowledge distillation for quality estimation. *arXiv preprint arXiv:2107.00411*, 2021.

Granger, C. W. J. and Newbold, P. *Forecasting economic time series*. Academic press, 2014.

Guo, Z., Shiao, W., Zhang, S., Liu, Y., Chawla, N. V., Shah, N., and Zhao, T. Linkless link prediction via relational distillation. In *International Conference on Machine Learning*, pp. 12012–12033. PMLR, 2023.

Han, L., Ye, H.-J., and Zhan, D.-C. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Hinton, G. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jensen, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

Kaushik, S., Choudhury, A., Sheron, P. K., Dasgupta, N., Natarajan, S., Pickett, L. A., and Dutt, V. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021a.

Kim, T., Oh, J., Kim, N., Cho, S., and Yun, S.-Y. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*, 2021b.

Kingma, D. P. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lin, S., Lin, W., Wu, W., Chen, H., and Yang, J. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. 2022.

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

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

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

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Takamoto, M., Morishita, Y., and Imaoka, H. An efficient method of training small models for regression problems with knowledge distillation. In *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 67–72. IEEE, 2020.

Thomas, M. and Joy, A. T. *Elements of information theory*. Wiley-Interscience, 2006.

Wang, H., Peng, J., Huang, F., Wang, J., Chen, J., and Xiao, Y. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J. Y., and Zhou, J. Timemixer: Decomposable multi-scale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024a.

Wang, Y., Wu, H., Dong, J., Liu, Y., Long, M., and Wang, J. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024b.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Wu, H., Zhou, H., Long, M., and Wang, J. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 5(6):602–611, 2023.

Xu, Q., Chen, Z., Ragab, M., Wang, C., Wu, M., and Li, X. Contrastive adversarial knowledge distillation for deep model compression in time-series regression tasks. *Neurocomputing*, 485:242–251, 2022.

Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., and Niu, Z. Frequency-domain MLPs are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., and Yin, B. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4927–4943, 2021.

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

Zhang, H. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, S., Liu, Y., Sun, Y., and Shah, N. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727*, 2021.

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

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A. Related Work

A.1. Debate in Long-Term Time Series Forecasting

The trade-off between performance and efficiency has prompted a long-standing debate between Transformer-based models and MLP in long-term time series forecasting (Zeng et al., 2023; Zhang et al., 2022; Lin et al., 2024). Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022) were among the leading Transformer-based methods. However, recent findings show that a simple Linear or MLP model can achieve performance comparable to these complex Transformer models across various benchmarks while offering significantly better efficiency (Zeng et al., 2023). This outcome has raised questions about the necessity of Transformers in time series forecasting. Following this, research has moved in two directions. One direction suggests that the issues with Transformer-based models arise from the way they are applied. For example, PatchTST (Nie et al., 2023) uses patching to preserve local information, and iTransformer (Liu et al., 2024) focuses on capturing multivariate correlations. These approaches surpass the simple one-layer MLP and demonstrate that Transformers can still deliver strong results in time series forecasting if applied effectively. Meanwhile, CNN-based models have also shown strong performance similar to Transformer-based models. TimesNet (Wu et al., 2022) transforms 1D time series into 2D variations and applies 2D CNN kernels, MICN (Wang et al., 2023) adopt multi-scale convolution structures to capture local features and global correlations, and ModernTCN (Luo & Wang, 2024) proposes a framework with much larger receptive fields than prior CNN-based structures. Nevertheless, these powerful CNN-based models also face efficiency issues, which further broadens the scope of the debate between performance and efficiency. The other direction focuses on developing lightweight MLP, such as N-BEATS (Oreshkin et al., 2020), N-hits (Challu et al., 2023), LightTS (Zhang et al., 2022), FreTS (Yi et al., 2023), TSMixer (Chen et al., 2023), TiDE (Das et al., 2023), which offer improved efficiency. However, these models typically only match, rather than surpass, the performance of state-of-the-art Transformer-based methods and CNN-based methods. In summary, while Transformer-based and CNN-based models generally offer better performance, simple MLP is more efficient. Therefore, we managed to combine the performance of Transformer-based and CNN-based models with MLP to produce a powerful and efficient model by cross-architecture *KD*.

A.2. Knolwedge Distillation on Time Series

Knowledge distillation (*KD*) (Hinton, 2015) transfers knowledge from a larger, more complex model (teacher) to a smaller, simpler model (student) while maintaining com-parable performance. By aligning the output distributions of teacher and student models, *KD* provides richer training signals than hard labels alone, enabling the student to capture subtle patterns that the teacher has learned. In the context of time series, CAKD (Xu et al., 2022) introduces a two-stage distillation scheme that distills features using adversarial and contrastive learning and performs prediction-level distillation. LightTS (Campos et al., 2023) designs a *KD* framework specifically for cases where the teacher is an ensemble classifier in time series classification tasks, which limits its applicability to teachers with other architectures. Both of these works do not incorporate time series-specific designs. In contrast, our framework emphasizes extracting essential time series patterns, including multi-scale and multi-period patterns, enabling more effective knowledge distillation. Additionally, we address architectural differences and are the first to explore cross-architecture *KD*.

B. Theoretical Understanding from Data Augmentation Perspective

In this section, we propose a theoretical viewpoint that TIMEDISTILL unifies the idea of augmenting training data with *blended* data that combines teacher outputs and ground truth. This concept is closely related to *mixup* (Zhang, 2017), a well-known data augmentation strategy. *Mixup* creates new training samples by *mixing* two examples' inputs and corresponding targets according to a random interpolation coefficient. Analogously, in multi-scale or multi-period distillation, we create augmented samples that blend teacher predictions and ground truth signals in a manner that provides richer supervision signals. Concretely, in a standard *mixup* procedure, one obtains augmented samples $(\tilde{x}, \tilde{y})$ from two distinct training samples (x_i, y_i) and (x_j, y_j) by drawing a mixing ratio λ and computing:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda)y_j.$$

In our distillation setting, instead of mixing two different input samples, we can think of $\hat{\mathbf{Y}}_t$ (the teacher predictions) and $\mathbf{Y}$ (the ground truth) as two sources to be mixed. Hence, these augmented targets become:

$$\tilde{\mathbf{Y}} = \lambda \mathbf{Y} + (1 - \lambda)\hat{\mathbf{Y}}_t,$$

while keeping the input as it is. By treating $\hat{\mathbf{Y}}_t$ as another "view" of the target, we are effectively performing a *mixup-style* augmentation in the target space. This approach can be especially valuable in time series forecasting, where incorporating teacher signals from multiple horizons or multiple resolutions may help capture nuanced temporal structures. *Next, we present two theorems that formalize multi-scale and multi-period loss as upper bound of matching mixup augmented prediction and student prediction.*

Theorem B.1. *Let (x, y) denote original input data pairs and (x, y^t) represent corresponding teacher data pairs.*

Consider a data augmentation function $\mathcal{A}(\cdot)$ applied to (x, y), generating augmented samples (x', y'). Define the training loss on these augmented samples as $\mathcal{L}_{aug} = \sum_{(x',y') \in \mathcal{A}(x,y)} |f_s(x') - y'|^2$. Then, the following inequality holds:

$$\mathcal{L}_{sup} + \eta \mathcal{L}_{scale} \geq \mathcal{L}_{aug} \tag{13}$$

when $\mathcal{A}(\cdot)$ is instantiated as a mixup function ([Zhang, 2017](#)) that interpolates between the original input data (x, y) and teacher data (x, y^t) with a mixing coefficient $\lambda = \frac{1}{1+\eta}$, i.e. $y' = \lambda y + (1 - \lambda) y^t$.

Proof. Let $\gamma_0 = 1$ and $\gamma_m = \frac{\eta}{M+1}$ for $m = 1, \ldots, M$. Observe that these weights satisfy $\gamma_0 + \sum_{m=1}^{M} \gamma_m = 1 + \eta$. Using these weights, we have:

$$\mathcal{L}_{sup} + \eta \mathcal{L}_{scale}$$
$$= (\hat{Y}_s - Y)^2 + \eta \sum_{m=0}^{M} \frac{(\hat{Y}_s - \hat{Y}_t^{(m)})^2}{M+1}$$
$$= \sum_{m=0}^{M} \gamma_m (\hat{Y}_s - x_m)^2,$$

where $x_0 = Y$ and $x_{m+1} = \hat{Y}_t^{(m)}$ for $m = 0, \ldots, M$. By Jensen's inequality ([Jensen, 1906](#)), for a convex function f, we have:

$$\sum_{m=0}^{M} \gamma_m f(x_m) \geq f\left(\frac{\sum_{m=0}^{M} \gamma_m x_m}{\sum_{m=0}^{M} \gamma_m}\right).$$

We use Jensen's inequality, which applies due to the convexity of the squared loss function $f(x) = (\hat{Y}_s - x)^2$. We obtain:

$$\sum_{m=0}^{M} \gamma_m (\hat{Y}_s - x_m)^2 \geq \left(\hat{Y}_s - \frac{\sum_{m=0}^{M} \gamma_m x_m}{\sum_{m=0}^{M} \gamma_m}\right)^2.$$

Now, compute the weighted sum of x_m using the weights γ_m and we define $\lambda = \frac{1}{1+\eta}$:

$$\frac{\sum_{m=0}^{M} \gamma_m x_m}{\sum_{m=0}^{M} \gamma_m} = \frac{\gamma_0 x_0 + \sum_{m=1}^{M} \gamma_m x_m}{1 + \eta}$$
$$= \frac{1}{1+\eta} Y + \frac{\eta}{1+\eta} \sum_{m=0}^{M} \frac{\hat{Y}_t^{(m)}}{M+1}$$
$$= \lambda Y + (1 - \lambda) \sum_{m=0}^{M} \frac{\hat{Y}_t^{(m)}}{M+1}.$$

Substituting this result back, we conclude that:

$$\mathcal{L}_{sup} + \eta \mathcal{L}_{scale}$$
$$= (\hat{Y}_s - Y)^2 + \eta \sum_{m=0}^{M} \frac{(\hat{Y}_s - \hat{Y}_t^{(m)})^2}{M+1}$$
$$\geq \left[\hat{Y}_s - [\lambda Y + (1 - \lambda) \sum_{m=0}^{M} \frac{\hat{Y}_t^{(m)}}{M+1}]\right]^2$$
$$= \mathcal{L}_{aug}$$

This completes the proof. $\square$

Theorem B.2. *Define the training loss on the augmented samples using KL divergence as $\mathcal{L}_{aug} = \sum_{(x',y') \in \mathcal{A}(x,y)} KL(y', \mathcal{X}(f_s(x')))$, where $\mathcal{X}(\cdot) = Softmax(FFT(\cdot))$. Then, the following inequality holds:*

$$\mathcal{L}_{sup} + \eta \mathcal{L}_{period} \geq \mathcal{L}_{aug} \tag{14}$$

where $\mathcal{A}(\cdot)$ is instantiated as a mixup function that interpolates between the period distribution of original input data $(x, \mathcal{X}(y))$ and teacher data $(x, \mathcal{X}(y^t))$ with a mixing coefficient $\lambda = \eta$, i.e. $y' = \mathcal{X}(y) + \lambda \mathcal{X}(y^t)$.

Proof. We denote $\mathbf{Q}_y = \mathcal{X}(y)$, $\mathbf{Q}_s = \mathcal{X}(f_s(x))$, $\mathbf{Q}_t = \mathcal{X}(y_t)$, $\hat{Y}_s = f_s(x)$ According to Parseval's theorem, we rewrite $\mathcal{L}_{sup}$ as following:

$$\mathcal{L}_{sup} + \eta \mathcal{L}_{period}$$
$$= (\hat{Y}_s - Y)^2 + \eta \, \mathrm{KL}(\mathbf{Q}_t, \mathbf{Q}_s)$$
$$= (FFT(\hat{Y}_s) - FFT(Y))^2 + \eta \, \mathrm{KL}(\mathbf{Q}_t, \mathbf{Q}_s)$$

When the temperature τ goes to ∞, minimizing KL divergence between two distributions is equivalent to minimizing the MSE error ([Kim et al., 2021b](#)). Thus, we have

$$(FFT(\hat{Y}_s) - FFT(Y))^2 + \eta \, \mathrm{KL}(\mathbf{Q}_t, \mathbf{Q}_s)$$
$$= \lim_{\tau \to \infty} \mathrm{KL}(\mathcal{X}(Y), \mathcal{X}(\hat{Y}_s)) + \eta \, \mathrm{KL}(\mathbf{Q}_t, \mathbf{Q}_s)$$
$$= \lim_{\tau \to \infty} \mathrm{KL}(\mathbf{Q}_y, \mathbf{Q}_s) + \eta \, \mathrm{KL}(\mathbf{Q}_t, \mathbf{Q}_s)$$

The KL divergence of P from Q is defined as

$$\mathrm{KL}[P||Q] = \sum_{x \in \mathcal{X}} p(x) \cdot \log \frac{p(x)}{q(x)}$$

and the log sum inequality ([Thomas & Joy, 2006](#)) states that

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i},$$

where $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are non-negative real numbers. For clarity, we omit $\lim_{\tau \to \infty}$ in the following derivation. Continuing with the main derivation, we have:

$$
\begin{aligned}
& \mathrm{KL}\big(\mathbf{Q}_y, \mathbf{Q}_s\big) \;+\; \eta \, \mathrm{KL}\big(\mathbf{Q}_t, \mathbf{Q}_s\big) \\
&= \sum_{x \in \mathcal{X}} \mathbf{Q}_y(x) \cdot \log \frac{\mathbf{Q}_y(x)}{\mathbf{Q}_s(x)} \\
&\qquad\qquad + \eta \sum_{x \in \mathcal{X}} \mathbf{Q}_t(x) \cdot \log \frac{\mathbf{Q}_t(x)}{\mathbf{Q}_s(x)} \\
&= \sum_{x \in \mathcal{X}} [\mathbf{Q}_y(x) \cdot \log \frac{\mathbf{Q}_y(x)}{\mathbf{Q}_s(x)} \\
&\qquad\qquad + \eta \mathbf{Q}_t(x) \cdot \log \frac{\eta \mathbf{Q}_t(x)}{\eta \mathbf{Q}_s(x)}] \\
&\geq \sum_{x \in \mathcal{X}} [[\mathbf{Q}_y(x) + \eta \mathbf{Q}_t(x)] \\
&\qquad\qquad \cdot \log \frac{\mathbf{Q}_y(x) + \eta \mathbf{Q}_t(x)}{\mathbf{Q}_s(x) + \eta \mathbf{Q}_s(x)}] \\
&= \mathrm{KL}\big(\mathbf{Q}_y + \eta \, \mathbf{Q}_t \,,\, \mathbf{Q}_s\big)
\end{aligned}
$$

To align with the mixup formulation defined in our theorem. We denote $\lambda = \eta$, then we have

$$
\begin{aligned}
& \mathrm{KL}\big(\mathbf{Q}_y + \eta \, \mathbf{Q}_t \,,\, \mathbf{Q}_s\big) \\
&= \mathrm{KL}\big(\mathbf{Q}_y + \lambda \, \mathbf{Q}_t \,,\, \mathbf{Q}_s\big) \\
&= \mathrm{KL}\big(\mathcal{X}(Y) + \lambda \, \mathcal{X}(\hat{Y}_t) \,,\, \mathcal{X}(\hat{Y}_s)\big) \\
&= \mathcal{L}_{aug}
\end{aligned}
$$

This completes the proof. $\qquad\qquad\qquad\square$

These two theorems show that the disllation process provides additional augmented samples that smooth the ground truth and the teacher prediction. This is analogous to label smoothing in classification, where labels are adjusted to be less extreme or certain. This data augmentation perspective brings several notable benefits for time series forecasting: **(1) Enhanced Generalization:** By blending teacher predictions (e.g., from a larger or more accurate model, or from multiple horizons/scales) with the ground truth, the student model is exposed to a richer supervision signal. This can mitigate overfitting, especially in scenarios with limited or noisy training data and the teacher predictions might encapsulate trends or patterns not immediately apparent from the ground truth alone, especially in cases of complex or high-dimensional regression tasks. **(2) Explicit Integration of Patterns:** In time series, important temporal structures often manifest at different resolutions or periods. Incorporating teacher predictions at multiple scales can help the student model learn both short-term fluctuations and long-term trends, which might be overlooked when relying solely

on the ground truth. **(3) Stabilized Training Dynamics:** Blending ground truth with teacher outputs naturally "softens" the targets, making the learning process less sensitive to noise and variance in the dataset. This smoothness can also lead to gentler gradients, reducing abrupt directional changes during optimization. Consequently, training is more stable and less prone to exploding or vanishing updates, facilitating better convergence.

C. Implementation Details

Datasets Details. We conduct experiments to evaluate the performance and efficiency of TIMEDISTILL on eight widely used benchmarks: Electricity (ECL), the ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Solar, Traffic, and Weather, following the pipeline in (Liu et al., 2024; Wang et al., 2024a; Luo & Wang, 2024). Detailed descriptions of these datasets are provided in Table 6.

Metric Details. We use Mean Square Error (MSE) and Mean Absolute Error (MAE) as our evaluation metrics, following (Liu et al., 2024; Luo & Wang, 2024; Wang et al., 2024a; 2023; Wu et al., 2022; 2021; Zhou et al., 2021; Zeng et al., 2023):

$$
\mathrm{MSE} = \frac{1}{S \times C} \sum_{i=1}^{S} \sum_{j=1}^{C} (Y_{ij} - \hat{Y}_{ij})^2, \tag{15}
$$

$$
\mathrm{MAE} = \frac{1}{S \times C} \sum_{i=1}^{S} \sum_{j=1}^{C} |Y_{ij} - \hat{Y}_{ij}|. \tag{16}
$$

Here, $Y \in \mathbb{R}^{S \times C}$ represents the ground truth, and $\hat{Y} \in \mathbb{R}^{S \times C}$ represents the predictions. S denotes the future prediction length, C is the number of channels, and Y_{ij} indicates the value at the i-th future time point for the j-th channel.

Experiment Details. All experiments are implemented in PyTorch (Paszke et al., 2019) and conducted on a cluster equipped with NVIDIA A100, V100, and K80 GPUs. The teacher models are trained using their default configurations as reported in their respective papers. For the student MLP model, we employ a combination of a decomposition scheme (Zeng et al., 2023; Wu et al., 2021; Zhou et al., 2022) and a two-layer MLP with hidden dimension D as the architecture. When using TIMEDISTILL for distillation, the teacher model is frozen, and only the student MLP is trained. The initial learning rate is set to 0.01, and the ADAM optimizer (Kingma, 2014) is used with MSE loss for the training of the student model. We apply early stopping with a patience value of 5 epochs. The batch size is set to 32. The temperature for multi-period distillation is set to $\tau = 0.5$. The number of scales is set to $M = 3$. We use two types of normalization methods, i.e. non-stationary (Liu et al., 2022) and revin (Kim et al., 2021a). We perform a

Table 6. Dataset characteristics for time series long-term forecasting task.

Dataset	Dim	Series Length	Dataset Size	Frequency	Forecastability*	Information
ECL	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Hourly	0.77	Electricity
ETTh1	7	{96, 192, 336, 720}	(8545, 2881, 2881)	15min	0.38	Temperature
ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	15min	0.45	Temperature
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
Solar	137	{96, 192, 336, 720}	(36601, 5161, 10417)	10min	0.33	Electricity
Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Hourly	0.68	Transportation
Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10min	0.75	Weather

*The forecastability is borrowed from TimeMixer (Wang et al., 2024a). A larger value indicates better predictability.

Table 7. Experiment configuration of TIMEDISTILL.

Dataset	Model Hyper-parameter		Training Process		
	D	Norm	Epoch	α	β
ECL	512	non-stationary	20	0.1	0.5
ETTh1	512	non-stationary	20	2	2
ETTh2	512	non-stationary	20	2	0.5
ETTm1	512	non-stationary	20	2	0.1
ETTm2	512	non-stationary	20	2	0.5
Solar	512	non-stationary	20	0.1	2
Traffic	1024	revin	10	0.1	0.1
Weather	512	non-stationary	20	0.5	2

hyperparameter search for α and β within the range {0.1, 0.5, 1, 2}. We conduct hyperparameter sensitivity analysis in Appendix G. For consistency, TIMEDISTILL defaults to using ModernTCN as the teacher and fixes the look-back window length to 720. Additional experiments with other teacher models and different look-back window lengths are presented in Section 5.3. Additional detailed model configuration information is presented in Table 7.

Channel Independent MLP. We denote the model as f. The channel-independent strategy predicts each channel independently, defined as:$\hat{Y} = [f(x^1), \ldots, f(x^C)]$, where $\hat{Y} = [\hat{y}^1, \ldots, \hat{y}^C]$ denotes the model prediction. In contrast, the channel-dependent strategy considers the combination of all channels simultaneously, defined as: $\hat{Y} = f(x^1, \ldots, x^C)$. In this paper, for MLP, we adopt the channel-independent strategy because it eliminates the need to model inter-channel relationships (Han et al., 2024), which reduces model complexity and results in a more lightweight model, which is especially advantageous when dealing with a large number of variables.

D. Implementation Details for Preliminary Study.

For the multi-scale pattern, we downsample the predicted time series $\hat{Y}$ using a 1D convolution layer with a tempo-ral stride of 2 to generate time series at M scales. The downsampling is defined as:

$$\hat{Y}^m = \text{Conv}(\hat{Y}^{m-1}, \text{stride} = 2), \tag{17}$$

where $m \in \{1, \cdots, M\}$ and $\hat{Y}^0 = \hat{Y}$. We set the prediction length S to 96 and display only the first 48 time steps due to space constraints, with M set to 3. We compare the MLP's multi-scale prediction performance against two Transformer-based teacher models (i.e., iTransformer, PatchTST) and one CNN-based teacher model (i.e., ModernTCN), using the ground truth as a reference.

For multi-period distillation, we compute the spectrogram of $\hat{Y}$ via Fast Fourier Transform (FFT), which provides $S/2$ frequencies and their corresponding amplitudes. We remove the DC component in the spectrogram. Each frequency corresponds to a period, calculated as $S/\text{frequency}$, with amplitude indicating the period's significance. Thus, the spectrogram can effectively show the multi-period pattern of a time series. The prediction length S is set to 96. Similar to the multi-scale, we compare the MLP's multi-period prediction performance against iTransformer, PatchTST, and ModernTCN, using the ground truth as a reference.

E. Different Teachers Analysis

In this section, we evaluate the ability of TIMEDISTILL to distill knowledge from various teacher models (additional teachers compared to Section 5.3) into a student MLP. The experiment involves eight teacher models: iTransformer, ModernTCN, TimeMixer, PatchTST, MICN, FEDformer, TimesNet, and Autoformer. The student model is an MLP designed as a channel-independent architecture, making it lightweight and computationally efficient. From Table 8, it is evident that iTransformer, ModernTCN, TimeMixer, PatchTST, MICN, FEDformer, TimesNet, and Autoformer improve the performance of MLP on MSE by 10.33%, 10.88%, 9.11%, 7.23%, 6.24%, 8.90%, 8.44%, and 3.24%

Table 8. Performance promotion obtained by our **TIMEDISTILL** framework with different teacher models. Δ_{MLP} ($\Delta_{Teacher}$) represents the performance promotion between **MLP+TIMEDISTILL** and a trained MLP (Teacher). We report the average performance from all prediction lengths.

Teacher Models		iTransformer (2024)		ModernTCN (2024)		TimeMixer (2024a)		PatchTST (2023)		MICN (2023)		FEDformer (2022)		TimesNet (2022)		Autoformer (2021)	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	Teacher	0.163	0.259	0.167	0.262	0.165	0.259	0.165	0.266	0.181	0.293	0.274	0.376	0.250	0.347	0.238	0.347
	MLP	0.173	0.276	0.173	0.276	0.173	0.276	0.173	0.276	0.173	0.276	0.173	0.276	0.173	0.276	0.173	0.276
	+TIMEDISTILL	**0.157**	**0.254**	**0.157**	**0.254**	**0.159**	**0.256**	**0.159**	**0.256**	**0.158**	**0.256**	**0.157**	**0.255**	**0.162**	**0.261**	**0.158**	**0.256**
	$\Delta_{Teacher}$	3.68%	1.93%	5.61%	3.24%	3.31%	1.02%	3.64%	3.76%	12.78%	12.89%	42.70%	32.18%	35.08%	24.82%	33.61%	26.22%
	Δ_{MLP}	9.27%	7.93%	9.09%	8.06%	7.94%	7.18%	8.11%	7.21%	8.70%	7.34%	9.27%	7.57%	6.22%	5.39%	8.69%	7.21%
ETTh1	Teacher	0.468	0.476	0.469	0.465	0.459	0.465	0.498	0.490	0.739	0.631	0.527	0.524	0.507	0.500	0.731	0.659
	MLP	0.502	0.489	0.502	0.489	0.502	0.489	0.502	0.489	0.502	0.489	0.502	0.489	0.502	0.489	0.502	0.489
	+TIMEDISTILL	**0.428**	**0.445**	**0.429**	**0.441**	**0.457**	**0.464**	**0.443**	**0.456**	**0.496**	**0.480**	**0.438**	**0.454**	**0.441**	**0.454**	**0.475**	**0.480**
	$\Delta_{Teacher}$	8.55%	6.51%	8.42%	5.23%	0.42%	0.25%	11.04%	6.94%	32.90%	23.97%	16.89%	13.36%	13.09%	9.22%	35.02%	27.16%
	Δ_{MLP}	14.74%	9.00%	14.47%	9.85%	8.92%	5.13%	11.75%	6.75%	1.20%	1.82%	12.75%	7.16%	12.14%	7.25%	5.38%	1.84%
ETTh2	Teacher	0.398	0.426	0.357	0.403	0.422	0.444	0.444	0.443	1.078	0.736	0.456	0.485	0.419	0.446	1.594	0.940
	MLP	0.393	0.438	0.393	0.438	0.393	0.438	0.393	0.438	0.393	0.438	0.393	0.438	0.393	0.438	0.393	0.438
	+TIMEDISTILL	**0.345**	**0.395**	**0.345**	**0.395**	**0.357**	**0.403**	**0.371**	**0.416**	**0.372**	**0.418**	**0.359**	**0.408**	**0.356**	**0.404**	**0.368**	**0.419**
	$\Delta_{Teacher}$	13.32%	7.28%	3.52%	2.03%	15.49%	9.17%	16.44%	6.09%	65.50%	43.22%	21.27%	15.88%	14.90%	9.33%	76.91%	55.43%
	Δ_{MLP}	12.21%	9.82%	12.27%	9.90%	9.17%	7.91%	5.60%	5.02%	5.41%	4.60%	8.65%	6.85%	9.31%	7.69%	6.36%	4.34%
ETTm1	Teacher	0.372	0.402	0.390	0.410	0.367	0.388	0.383	0.412	0.439	0.461	0.423	0.451	0.398	0.419	0.570	0.526
	MLP	0.391	0.413	0.391	0.413	0.391	0.413	0.391	0.413	0.391	0.413	0.391	0.413	0.391	0.413	0.391	0.413
	+TIMEDISTILL	**0.354**	**0.390**	**0.348**	**0.379**	**0.347**	**0.383**	**0.358**	**0.397**	**0.375**	**0.400**	**0.354**	**0.398**	**0.356**	**0.391**	**0.376**	**0.407**
	$\Delta_{Teacher}$	4.84%	2.99%	10.94%	7.64%	5.34%	1.25%	6.53%	3.64%	14.66%	13.11%	16.31%	11.75%	10.53%	6.64%	34.04%	22.62%
	Δ_{MLP}	9.42%	5.65%	11.07%	8.43%	11.16%	7.34%	8.39%	3.96%	4.13%	3.17%	9.42%	3.72%	8.79%	5.40%	3.79%	1.54%
ETTm2	Teacher	0.276	0.337	0.267	0.330	0.279	0.339	0.274	0.335	0.348	0.404	0.359	0.401	0.291	0.349	0.420	0.448
	MLP	0.300	0.373	0.300	0.373	0.300	0.373	0.300	0.373	0.300	0.373	0.300	0.373	0.300	0.373	0.300	0.373
	+TIMEDISTILL	**0.252**	**0.316**	**0.244**	**0.311**	**0.252**	**0.317**	**0.261**	**0.321**	**0.262**	**0.322**	**0.262**	**0.322**	**0.258**	**0.320**	**0.265**	**0.328**
	$\Delta_{Teacher}$	8.70%	6.23%	8.63%	5.48%	9.67%	6.31%	4.74%	4.18%	24.64%	20.28%	27.02%	19.70%	11.22%	8.23%	36.90%	26.79%
	Δ_{MLP}	16.07%	15.28%	18.61%	16.50%	16.17%	14.91%	13.08%	13.94%	12.64%	13.75%	12.74%	13.67%	13.91%	14.12%	11.74%	12.06%
ETT(avg)	Teacher	0.379	0.410	0.371	0.402	0.382	0.409	0.400	0.420	0.651	0.558	0.441	0.465	0.404	0.428	0.829	0.643
	MLP	0.397	0.428	0.397	0.428	0.397	0.428	0.397	0.428	0.397	0.428	0.397	0.428	0.397	0.428	0.397	0.428
	+TIMEDISTILL	**0.345**	**0.387**	**0.342**	**0.381**	**0.353**	**0.392**	**0.358**	**0.398**	**0.376**	**0.405**	**0.353**	**0.396**	**0.353**	**0.392**	**0.371**	**0.409**
	$\Delta_{Teacher}$	8.92%	5.79%	7.94%	5.09%	7.46%	4.16%	10.38%	5.36%	42.21%	27.41%	19.94%	14.99%	12.59%	8.42%	55.23%	36.49%
	Δ_{MLP}	13.06%	9.77%	13.87%	10.97%	10.91%	8.50%	9.65%	7.20%	5.13%	5.46%	10.91%	7.67%	10.95%	8.41%	6.44%	4.63%
Solar	Teacher	0.214	0.270	0.191	0.243	0.288	0.259	0.210	0.257	0.213	0.277	0.300	0.383	0.196	0.262	1.037	0.742
	MLP	0.194	0.255	0.194	0.255	0.194	0.255	**0.194**	**0.255**	0.194	0.255	0.194	0.255	0.194	0.255	**0.194**	**0.255**
	+TIMEDISTILL	**0.185**	**0.241**	**0.184**	**0.241**	**0.187**	**0.245**	0.204	0.283	**0.186**	**0.242**	**0.184**	**0.241**	**0.186**	**0.243**	0.237	0.329
	$\Delta_{Teacher}$	13.55%	10.74%	3.60%	0.71%	21.41%	14.98%	2.86%	-10.12%	12.99%	12.47%	38.67%	37.08%	5.19%	6.97%	77.15%	55.66%
	Δ_{MLP}	4.80%	5.42%	5.14%	5.24%	3.58%	3.86%	-4.98%	-11.07%	4.49%	4.83%	5.31%	5.42%	4.47%	4.49%	-21.97%	-29.12%
Traffic	Teacher	**0.379**	0.271	0.413	0.284	**0.391**	0.275	0.402	0.284	0.500	0.316	0.629	0.388	0.693	0.399	0.696	0.427
	MLP	0.434	0.318	0.434	0.318	0.434	0.318	0.434	0.318	0.434	0.318	0.434	0.318	0.434	0.318	0.434	0.318
	+TIMEDISTILL	0.389	**0.271**	**0.387**	**0.271**	0.391	**0.274**	**0.390**	**0.272**	**0.391**	**0.274**	**0.395**	**0.278**	**0.400**	**0.282**	**0.396**	**0.280**
	$\Delta_{Teacher}$	-2.64%	0.00%	6.32%	4.74%	-0.04%	0.38%	2.99%	4.23%	21.78%	13.38%	37.20%	28.35%	42.35%	29.39%	43.10%	34.43%
	Δ_{MLP}	10.30%	14.84%	10.70%	14.91%	9.76%	13.79%	10.07%	14.53%	9.78%	13.89%	8.92%	12.64%	7.83%	11.54%	8.69%	12.01%
Weather	Teacher	0.259	0.290	0.238	0.277	0.230	0.271	0.246	0.283	0.240	0.292	0.355	0.398	0.257	0.294	0.471	0.465
	MLP	0.234	0.294	0.234	0.294	0.234	0.294	0.234	0.294	0.234	0.294	0.234	0.294	0.234	0.294	0.234	0.294
	+TIMEDISTILL	**0.220**	**0.270**	**0.220**	**0.269**	**0.219**	**0.266**	**0.220**	**0.267**	**0.225**	**0.272**	**0.224**	**0.274**	**0.222**	**0.271**	**0.226**	**0.278**
	$\Delta_{Teacher}$	15.06%	6.90%	7.37%	2.97%	4.82%	1.77%	10.57%	5.65%	6.10%	6.61%	36.90%	31.16%	13.38%	7.59%	52.02%	40.22%
	Δ_{MLP}	5.83%	8.02%	5.66%	8.48%	6.16%	9.38%	5.83%	9.04%	3.61%	7.26%	4.12%	6.66%	4.82%	7.57%	3.27%	5.30%

on average across all datasets, respectively. These results demonstrate that TIMEDISTILL can effectively distill knowledge from all the teacher models to improve student performance. Notably, ModernTCN improves MLP performance by up to 18.61% on the ETTm2 dataset.

On Traffic dataset, which involves a large number of channels (861 channels) and is particularly challenging for channel-independent models like the student MLP, TIMEDISTILL achieves the second-best performance, improving MSE by 10.30% when the teacher is iTransformer. This is likely due to TIMEDISTILL can implicitly distill iTransformer's knowledge of modeling multivariate correlations to student MLP. We further explore whether TIMEDISTILL can implicitly learn multi-variate correlation via distillation in Appendix J.

However, when the teacher performs poorly on certain datasets, it can negatively impact the student MLP. For example, MLP achieves much better performance than the teacher Autoformer even without distillation. Autoformer performs exceptionally poorly on the Solar dataset, and after distillation, the MLP's performance degrades by -29.12% in terms of MAE. This suggests that poor teacher models may transfer excessive noise to the student MLP, resulting in degraded performance. Furthermore, we also observe that TIMEDISTILL can effectively learn from various teachers, achieving comparable or even better performance than the teachers themselves. TIMEDISTILL achieves significant improvements over the teachers themselves, with average gains of 8.13%, 6.80%, 7.55%, 7.35%, 23.92%, 29.62%, 18.22%, and 48.59% on MSE across all datasets.

F. Different Students Analysis

Analysis of MLP Architecture. To examine the impact of the student MLP architecture on TIMEDISTILL, we vary the number of layers and the hidden dimensions of the MLP. From Table 11, we observe that increasing the number of layers and hidden dimensions generally enhances performance. However, an excessive number of layers (e.g., 4 layers) or overly small hidden dimensions (e.g., 64) can lead to performance degradation. Therefore, to balance efficiency and performance, we select 2L-512 as the default MLP configuration.

Analysis of Other MLP-Based Student Models. From Table 11, we observe that both LightTS and TSMixer outperform the original MLP. Incorporating TIMEDISTILL further enhances their performance across all datasets, with improvements ranging from a minimum of 2.92% to a maximum of 43.42%, highlighting the effectiveness and adaptability of TIMEDISTILL.

G. Hyperparameter Sensitivity Analysis

Sensitivity of Predction Level α and Feature Level β Distillation. In this subsection, we analyze the robustness of TIMEDISTILL by investigating the sensitivity of two key hyperparameters, α and β, in the final objective function of TIMEDISTILL. The parameters α and β regulate the contributions of the prediction-level and feature-level distillation loss terms, respectively. To assess their effects, we vary both α and β over the set $\{0.1, 0.5, 1, 2\}$ while keeping other hyperparameters fixed. We conduct experiments on all 8 datasets. Figure 9 illustrates that TIMEDISTILL generally performs better when a smaller β, such as 0.1 or 0.5, is selected. For α, the results show dataset-specific preferences. On the four ETT datasets, a smaller α usually leads to inferior performance, while increasing α improves the results. This is because α controls the contribution of the loss term at the prediction level. As shown in Table 8, MLP performs poorly on ETT datasets, leading to a larger gap between MLP and the teacher. Matching predictions directly with a larger α helps reduce this performance gap, resulting in better outcomes on ETT datasets. However, for other datasets such as ECL, Solar, Traffic, and Weather, the performance gap between MLP and the teacher is not as significant, or MLP even outperforms the teacher. In such cases, a large α introduces excessive noise to MLP, negatively impacting the performance of TIMEDISTILL. Consequently, on these datasets, a smaller α is more beneficial.

Sensitivity Analysis of the Number of Scales M in Multi-Scale Distillation M determines the number of scales used for matching in multi-scale distillation. To evaluate the robustness of TIMEDISTILL with respect to M, we vary M from 0 to 5 while keeping other hyperparameters fixed. The left panel of Figure 10 illustrates the forecasting per-

formance (measured by MSE) on the ECL dataset as M increases. When $M = 0$, TIMEDISTILL does not downsample the time series and directly matches them. Consequently, TIMEDISTILL relies solely on the noisy guidance signals at the finest scale from the teacher, resulting in suboptimal performance. As M increases, the performance improves, demonstrating that multi-scale signals provide informative hierarchical knowledge to the MLP. To balance performance and computational efficiency, we set $M = 3$.

Sensitivity Analysis of the Temperature τ in Multi-Period Distillation The temperature τ determines the extent to which the frequency distribution is softened. A larger τ results in a softer distribution. As shown in the right panel of Figure 10, there is an optimal value of τ that delivers the best MSE. When $\tau = 1$, the distribution is softened, reducing the significance of high-magnitude frequencies. This process also amplifies the magnitudes of noisy frequencies, which should not be learned by the student MLP, leading to inferior performance. Conversely, when $\tau = 0.1$, only a few high-magnitude frequencies are retained, resulting in a frequency distribution that is less informative. To balance informativeness and noise, we set $\tau = 0.5$.

H. Directly Match the Patterns in Ground Truth Time Series

The ground truth usually contains rich information for model learning. Based on our preliminary analysis, the multi-scale and multi-period patterns should also exist in ground truth time series that can benefit MLP. Thus, in this section, we try to directly distill the multi-scale and multi-period patterns in supervised ground truth signals using the following loss:

$$\mathcal{L}'_{sup} = \mathcal{L}^{sup}_{scale} + \mathcal{L}^{sup}_{period}. \tag{18}$$

We perform several incremental experiments: (1) $\mathcal{L}^{\mathbf{H}} + \mathcal{L}^{\mathbf{Y}} + \mathcal{L}'_{\text{sup}}$ means that we replace $\mathcal{L}_{sup}$ in Equation 12 with $\mathcal{L}'_{\text{sup}}$ defined in Equation 18. (2) $\mathcal{L}^{\mathbf{H}} + \mathcal{L}'_{\text{sup}}$ and $\mathcal{L}^{\mathbf{Y}} + \mathcal{L}'_{\text{sup}}$ denote we further remove multi-scale and multi-period distillation at logit level or at feature level, respectively. (3) $\mathcal{L}'_{\text{sup}}$ means we only use $\mathcal{L}'_{\text{sup}}$ as the overall training loss $\mathcal{L}$ in Equation 12, which only uses the ground truth for model learning and does not use the teacher. From Table 9, we observe that adding $\mathcal{L}'_{\text{sup}}$ to TIMEDISTILL results in an average MSE reduction of **0.8%**, which slightly degrades performance rather than improving it. Notably, when TIMEDISTILL is removed and only $\mathcal{L}'_{\text{sup}}$ is used to train the MLP, the performance deteriorates by approximately **9.5%** on average. This can be attributed to the noise present in the ground truth, making it harder to fit, whereas the teacher model provides simpler and more learnable knowledge. Furthermore, when multi-scale and multi-period distillation at the feature level (i.e., $\mathcal{L}^{\mathbf{Y}} + \mathcal{L}'_{\text{sup}}$) is removed, performance declines on most datasets. This indicates that higher-dimensional features encapsulate more valuable knowledge compared to
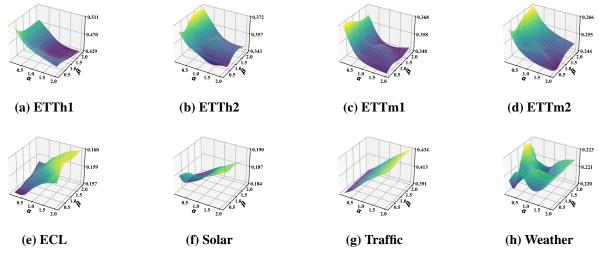
Figure 9. Sensitivity Analysis Result of Predcition Level α and Feature Level β Distillation on Different Datasets.
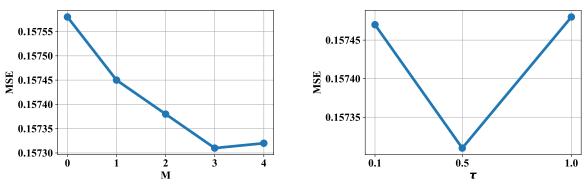

Figure 10. Hyperparameter sensitivity with respect to the number of scales M and temperature τ on ECL dataset. The results are recorded with the lookback window length T = 720 and averaged across all prediction window lengths $S \in \{96, 192, 336, 720\}$.

Table 9. Results of directly match the patterns in ground truth time series on eight datasets. The look-back length is set to be consistent at 720 for fair comparison. The MSE and MAE metrics are averaged from all prediction lengths $S \in \{96, 192, 336, 720\}$.

Method	ECL		ETTh1		ETTh2		ETTm1		ETTm2		Solar		Traffic		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TIMEDISTILL	**0.157**	**0.254**	**0.429**	**0.441**	**0.345**	**0.395**	**0.348**	**0.379**	**0.244**	**0.311**	**0.184**	**0.241**	**0.391**	**0.275**	**0.220**	**0.269**
$\mathcal{L}^{\mathbf{H}} + \mathcal{L}^{\mathbf{Y}} + \mathcal{L}'_{\text{sup}}$	0.159	0.256	0.436	0.449	0.348	0.397	0.349	0.380	0.247	0.314	0.185	0.243	0.393	0.277	0.222	0.270
$\mathcal{L}^{\mathbf{H}} + \mathcal{L}'_{\text{sup}}$	0.159	0.257	0.478	0.479	0.383	0.427	0.374	0.404	0.271	0.332	0.185	0.243	0.394	0.278	0.223	0.275
$\mathcal{L}^{\mathbf{Y}} + \mathcal{L}'_{\text{sup}}$	0.163	0.261	0.440	0.447	0.357	0.404	0.355	0.384	0.251	0.317	0.188	0.251	0.396	0.279	0.224	0.268
$\mathcal{L}'_{\text{sup}}$	0.165	0.265	0.509	0.496	0.398	0.439	0.402	0.418	0.278	0.335	0.188	0.250	0.397	0.281	0.228	0.280

Table 10. Static and runtime metrics of TIMEDISTILL, MLP, and other mainstream models on the ECL dataset. For fair comparison, the look-back length for each model is set to be consistent at 720 and the batch size is set to be 16. The static and runtime metrics are averaged from all prediction lengths $S \in \{96, 192, 336, 720\}$.

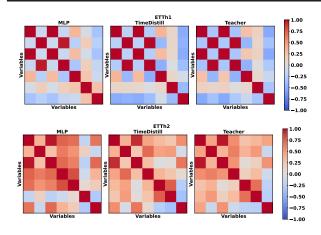| Model	MSE	MAE	Inference Time (ms/batch)	Parameters
TIMEDISTILL	**0.157**	**0.254**	**1.0956**	**1,084,966**
MLP	0.173	0.276	**1.0956**	**1,084,966**
iTransformer (Liu et al., 2024)	0.163	0.258	3.6377	5,276,496
ModernTCN (Luo & Wang, 2024)	0.167	0.262	6.2233	132,075,892
TimeMixer (Wang et al., 2024a)	0.165	0.259	7.7662	5,064,993
PatchTST (Nie et al., 2023)	0.165	0.266	2.9802	24,949,584
MICN (Wang et al., 2023)	0.181	0.293	2.715	60,215,726
Crossformer (Zhang & Yan, 2023)	0.203	0.299	8.7768	2,264,036
Fedformer (Zhou et al., 2022)	0.274	0.376	32.2642	16,827,399
Autoformer (Wu et al., 2021)	0.238	0.347	196.5266	14,914,398

Figure 11. Multi-Variate Person Correlation Visualization on ETTh1 and ETTh2 datasets.

lower-dimensional logits or ground truth, highlighting the importance of feature-level distillation.

I. Efficiency Analysis

In the main text, we have presented the efficiency analysis in Figure 2. Here, we provide the quantitative results in Table 10. Latency-sensitive applications require fast inference, making it crucial for models to minimize inference time. However, most works in the time series domain primarily focus on *less time-sensitive* training time comparisons (Liu et al., 2024; Lin et al., 2024). Additionally, edge-device applications demand models with fewer parameters for deployment. Therefore, in our experiments, we emphasize comparing both inference time and model parameters to evaluate practical efficiency. For a fair comparison, we fix the batch size to 16 during inference and record the average inference time per batch. Notably, TIMEDISTILL (MLP) demonstrates a significant efficiency advantage over Transformer-based models, such as iTransformer, PatchTST, Crossformer, FEDformer, and Autoformer, as well as CNN-based models like ModernTCN and MICN, and the complex MLP-based model TimeMixer, which employs multiple MLPs. In terms of inference time, TIMEDISTILL achieves up to **196×** faster inference than other methods. Additionally, it requires up to **60×** fewer parameters compared to other methods. Beyond the ECL dataset, we also visualize efficiency analyses for other datasets in Figure 14. Across eight datasets, TIMEDISTILL consistently resides in the lower-left corner, achieving the best trade-off between efficiency (inference time) and performance (MSE).

J. Can TIMEDISTILL Learn Multi-Variate Correlation via Distillation?

Some teacher models, such as iTransformer and ModerTCN, are designed in a channel-dependent (CD) manner. While this approach can utilize multi-variate correlation effectively, it incurs a higher computational burden. In contrast, ad-

vanced methods like PatchTST, TimeMixer, and our method TIMEDISTILL are employed in a channel-independent (CI) manner, which offers better computational efficiency. This leads to an intriguing question: *Can TIMEDISTILL enable MLP to learn multi-variate correlation from the teacher without an explicit distillation design?* To explore this, we present a preliminary study using Figure 11, which illustrates two cases from the ETTh1 and ETTh2 datasets, each containing seven variables. Our observations reveal that TIMEDISTILL's multi-variate correlation more closely resembles that of the teacher compared to the MLP model before distillation. This suggests that TIMEDISTILL facilitates learning underlying structures among multi-variates through distillation even though the student MLP is a CI model. These findings highlight a potential benefit of distillation that merits further exploration. We leave this for further work.

K. Additional Cases

In this section, we present additional cases to demonstrate how TIMEDISTILL effectively bridges the gap between the teacher and the student models. Figure 12 and Figure 13 showcase examples from the ETTm1 and ETTm2 datasets, respectively. From these figures, we make the following observations: **First**, in the temporal domain, the MLP struggles to capture the overall trend of the time series at coarser scales, resulting in a significant performance gap compared to the teacher at the finest scale. By effectively capturing the overall trend at coarser scales, TIMEDISTILL narrows this gap, enabling the MLP to approximate the teacher's performance more closely. **Second**, in the frequency domain, the MLP fails to effectively capture the multi-periodic patterns of the time series, leading to inaccuracies in the frequency distribution. In contrast, TIMEDISTILL accurately learns these multi-periodic patterns from the teacher, which helps the MLP improve its overall MSE performance.

These observations highlight the importance of multi-scale and multi-period patterns distillation. In the temporal domain, learning from coarser scales helps the student MLP capture the overall trend in the teacher's predictions without overfitting to noise present at the finest scale. Meanwhile, learning from the finest scale enables the MLP to refine details, resulting in more precise predictions. Consequently, jointly learning across multi-scale achieves a balance between the low accuracy of coarser scales and the high noise sensitivity of finer scales. In the frequency domain, learning from the frequency distribution enables the MLP to effectively capture overlapping periodicities, further enhancing prediction accuracy.

L. Showcases

For clear comparison, we present test set showcases in Appendix P, where TIMEDISTILL shows better performance.

Table 11. Performance promotion obtained by our **TIMEDISTILL** framework with different student models. The notation nL-d represents the structure of an MLP model, where nL specifies the number of layers and d denotes the dimension of the hidden units in each layer. For instance, 2L-512 refers to a model with 2 layers and a hidden dimension of 512. We report the average performance from all prediction lengths. $\Delta_{Student}$ represents the performance promotion between **Student+TIMEDISTILL** and original trained student.

Student Models		MLP (2L-512)		MLP (3L-512)		MLP (4L-512)		MLP (2L-1024)		LightTS (2022)		TSMixer (2023)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Student	0.502	0.489	0.487	0.497	0.468	0.481	0.495	0.500	0.465	0.471	0.471	0.474
	+TIMEDISTILL	**0.428**	**0.445**	**0.442**	**0.448**	**0.443**	**0.453**	**0.442**	**0.446**	**0.436**	**0.445**	**0.433**	**0.446**
	$\Delta_{Student}$	14.74%	9.00%	9.11%	9.86%	5.29%	5.70%	10.79%	10.83%	6.26%	5.57%	8.02%	5.92%
ETTh2	Student	0.393	0.438	0.742	0.601	0.908	0.668	0.639	0.563	0.675	0.582	0.376	0.420
	+TIMEDISTILL	**0.345**	**0.397**	**0.341**	**0.390**	**0.362**	**0.406**	**0.344**	**0.390**	**0.382**	**0.423**	**0.351**	**0.399**
	$\Delta_{Student}$	12.16%	9.46%	53.97%	35.05%	60.12%	39.19%	46.20%	30.72%	43.42%	27.26%	6.83%	5.01%
ETTm1	Student	0.391	0.413	0.378	0.406	0.392	0.414	0.377	0.402	0.376	0.399	0.370	0.395
	+TIMEDISTILL	**0.354**	**0.390**	**0.346**	**0.380**	**0.353**	**0.389**	**0.346**	**0.380**	**0.358**	**0.382**	**0.354**	**0.383**
	$\Delta_{Student}$	9.42%	5.65%	8.54%	6.34%	9.90%	6.12%	8.40%	5.56%	4.83%	4.29%	4.13%	2.92%
ETTm2	Student	0.300	0.373	0.359	0.408	0.346	0.393	0.307	0.374	0.283	0.346	0.295	0.347
	+TIMEDISTILL	**0.252**	**0.316**	**0.259**	**0.321**	**0.254**	**0.317**	**0.245**	**0.312**	**0.258**	**0.324**	**0.253**	**0.319**
	$\Delta_{Student}$	16.07%	15.28%	27.77%	21.31%	26.71%	19.41%	20.03%	16.55%	8.92%	6.38%	14.19%	8.23%
Weather	Student	0.234	0.294	0.232	0.289	0.237	0.291	0.221	0.278	0.235	0.293	0.239	0.278
	+TIMEDISTILL	**0.220**	**0.270**	**0.222**	**0.267**	**0.225**	**0.271**	**0.220**	**0.267**	**0.221**	**0.272**	**0.222**	**0.266**
	$\Delta_{Student}$	5.83%	8.02%	4.53%	7.53%	5.22%	7.04%	0.53%	4.10%	5.95%	7.17%	7.19%	4.09%

Figure 12. Comparison of model predictions across temporal scales and corresponding spectrograms before and after distillation on the ETTm1 dataset. MSE values for MLP, Teacher (ModernTCN (Luo & Wang, 2024)), and TIMEDISTILL are 2.16, 0.73 and 0.74.
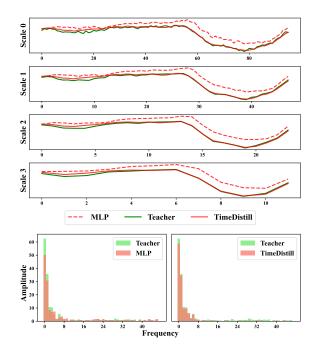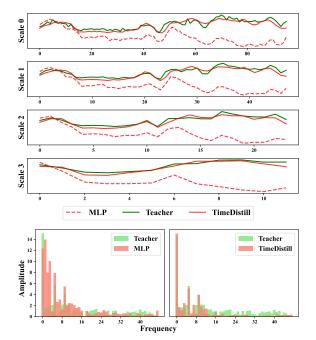
Figure 13. Comparison of model predictions across temporal scales and corresponding spectrograms before and after distillation on the ETTm2 dataset. MSE values for MLP, Teacher (ModernTCN (Luo & Wang, 2024)), and TIMEDISTILL are 3.35, 1.19, 1.28.

M. Full Main Results

Table 12. Long-term time series forecasting results with prediction lengths $S \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates a better prediction. For consistency, we maintain a fixed input length of 720 throughout all the experiments. The best performance is highlighted in **red**, and the second-best is underlined.

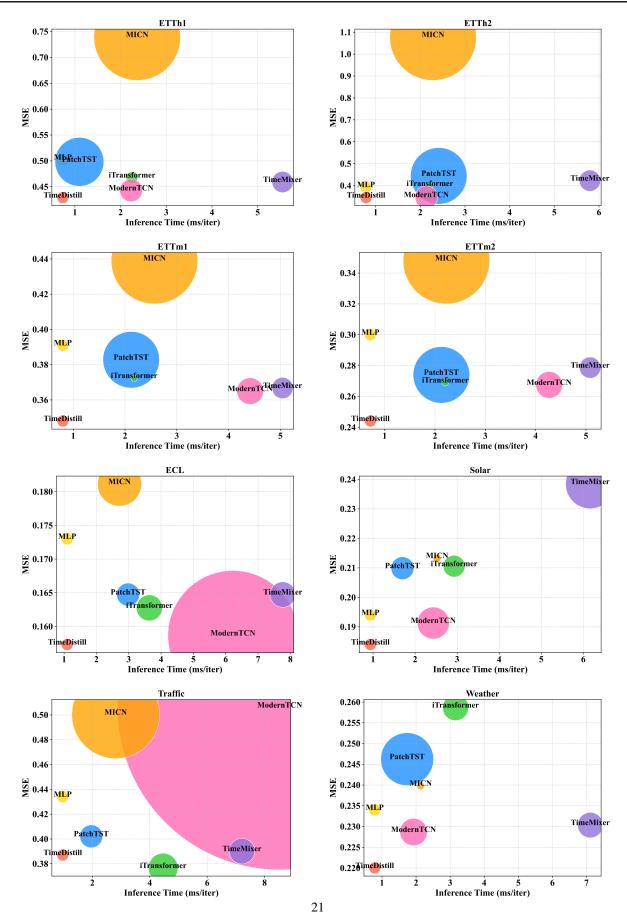| Models		TIMEDISTILL (Ours)		iTransformer (2024)		ModernTCN (2024)		TimeMixer (2024a)		PatchTST (2023)		MICN (2023)		FEDformer (2022)		TimesNet (2022)		Autoformer (2021)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	**0.128**	**0.225**	0.134	0.230	0.140	0.239	0.132	0.227	0.135	0.239	0.169	0.283	0.238	0.347	0.218	0.324	0.224	0.334
	192	**0.145**	**0.241**	0.153	0.248	0.153	0.250	0.156	0.248	0.152	0.255	0.171	0.283	0.239	0.349	0.228	0.331	0.227	0.339
	336	**0.161**	**0.258**	0.171	0.268	0.168	0.264	0.170	0.264	0.168	0.270	0.184	0.296	0.255	0.361	0.240	0.344	0.235	0.344
	720	**0.195**	0.291	0.196	**0.290**	0.206	0.296	0.201	0.295	0.203	0.301	0.201	0.312	0.363	0.447	0.314	0.390	0.267	0.371
	Avg	**0.157**	**0.254**	0.163	0.259	0.167	0.262	0.165	0.259	0.165	0.266	0.181	0.293	0.274	0.376	0.250	0.347	0.238	0.347
ETTh1	96	**0.373**	**0.401**	0.392	0.423	0.389	0.412	0.389	0.417	0.428	0.438	0.746	0.642	0.486	0.502	0.451	0.461	0.590	0.564
	192	**0.411**	**0.426**	0.428	0.448	0.446	0.452	0.443	0.460	0.476	0.476	0.523	0.502	0.483	0.501	0.456	0.469	0.586	0.578
	336	**0.439**	**0.444**	0.461	0.473	0.482	0.469	0.521	0.502	0.543	0.518	0.750	0.647	0.494	0.501	0.494	0.493	0.810	0.700
	720	0.495	0.493	0.590	0.562	0.557	0.528	**0.483**	**0.481**	0.546	0.526	0.939	0.735	0.644	0.594	0.629	0.575	0.941	0.797
	Avg	**0.429**	**0.441**	0.468	0.476	0.469	0.465	0.459	0.465	0.498	0.490	0.739	0.631	0.527	0.524	0.507	0.500	0.731	0.660
ETTh2	96	**0.273**	**0.336**	0.303	0.364	0.288	0.350	0.323	0.384	0.338	0.386	0.395	0.427	0.410	0.457	0.415	0.446	1.173	0.824
	192	**0.334**	**0.381**	0.410	0.423	0.341	0.387	0.506	0.492	0.405	0.423	0.572	0.544	0.420	0.462	0.406	0.435	1.523	0.908
	336	**0.363**	**0.415**	0.440	0.450	0.376	0.419	0.391	0.427	0.488	0.469	1.387	0.896	0.443	0.481	0.411	0.443	2.492	1.217
	720	**0.408**	**0.446**	0.439	0.469	0.424	0.456	0.469	0.473	0.545	0.494	1.956	1.077	0.551	0.539	0.442	0.460	1.190	0.815
	Avg	**0.345**	**0.395**	0.398	0.426	0.357	0.403	0.422	0.444	0.444	0.443	1.078	0.736	0.456	0.485	0.419	0.446	1.594	0.941
ETTm1	96	**0.285**	**0.344**	0.319	0.367	0.325	0.369	0.309	0.357	0.308	0.368	0.356	0.404	0.363	0.422	0.333	0.374	0.475	0.485
	192	**0.331**	**0.368**	0.347	0.388	0.372	0.397	0.339	0.373	0.363	0.395	0.428	0.454	0.401	0.444	0.367	0.398	0.504	0.495
	336	**0.359**	**0.386**	0.387	0.413	0.408	0.424	0.390	0.399	0.414	0.430	0.465	0.483	0.422	0.447	0.417	0.429	0.670	0.559
	720	**0.415**	**0.416**	0.437	0.439	0.456	0.450	0.429	0.423	0.446	0.452	0.507	0.502	0.505	0.492	0.477	0.474	0.635	0.567
	Avg	**0.348**	**0.379**	0.372	0.402	0.390	0.410	0.367	0.388	0.383	0.412	0.439	0.461	0.423	0.451	0.398	0.419	0.571	0.527
ETTm2	96	**0.163**	**0.255**	0.180	0.273	0.180	0.269	0.197	0.292	0.181	0.273	0.215	0.311	0.298	0.362	0.202	0.289	0.309	0.374
	192	**0.220**	**0.294**	0.243	0.316	0.240	0.310	0.240	0.307	0.230	0.308	0.232	0.317	0.322	0.375	0.260	0.332	0.452	0.466
	336	**0.269**	**0.328**	0.299	0.352	0.288	0.344	0.286	0.340	0.306	0.355	0.378	0.436	0.374	0.413	0.321	0.370	0.373	0.416
	720	**0.326**	**0.369**	0.382	0.405	0.363	0.395	0.392	0.416	0.379	0.404	0.567	0.551	0.441	0.455	0.381	0.406	0.550	0.540
	Avg	**0.244**	**0.311**	0.276	0.337	0.267	0.330	0.279	0.339	0.274	0.335	0.348	0.404	0.359	0.401	0.291	0.349	0.421	0.449
Solar	96	**0.166**	**0.229**	0.189	0.255	0.171	0.249	0.283	0.382	0.178	0.232	0.193	0.259	0.275	0.367	0.174	0.251	0.880	0.654
	192	**0.181**	**0.239**	0.237	0.265	0.188	0.236	0.216	0.257	0.201	0.256	0.213	0.275	0.279	0.365	0.189	0.258	1.024	0.742
	336	**0.191**	**0.246**	0.214	0.280	0.192	0.237	0.232	0.267	0.230	0.266	0.209	0.278	0.306	0.384	0.206	0.271	1.258	0.858
	720	**0.199**	**0.252**	0.217	0.280	0.214	0.251	0.223	0.246	0.231	0.275	0.238	0.297	0.341	0.416	0.214	0.267	0.987	0.718
	Avg	**0.184**	**0.241**	0.214	0.270	0.191	0.243	0.238	0.288	0.210	0.257	0.213	0.277	0.300	0.383	0.196	0.262	1.037	0.743
Traffic	96	0.358	0.256	**0.345**	**0.254**	0.392	0.276	0.367	0.272	0.374	0.272	0.485	0.313	0.608	0.388	0.677	0.391	0.691	0.422
	192	0.374	**0.264**	**0.366**	0.266	0.400	0.277	0.377	0.265	0.389	0.278	0.484	0.310	0.616	0.376	0.682	0.392	0.710	0.432
	336	0.389	**0.271**	**0.382**	0.272	0.412	0.283	0.389	0.272	0.403	0.285	0.493	0.311	0.639	0.392	0.695	0.401	0.700	0.433
	720	0.428	**0.292**	**0.422**	0.292	0.450	0.302	0.433	0.293	0.442	0.303	0.539	0.331	0.648	0.397	0.720	0.410	0.687	0.422
	Avg	0.387	**0.271**	**0.379**	0.271	0.413	0.284	0.391	0.275	0.402	0.284	0.500	0.316	0.628	0.388	0.693	0.399	0.697	0.427
Weather	96	**0.145**	**0.204**	0.178	0.229	0.152	0.208	0.147	0.202	0.158	0.214	0.171	0.231	0.333	0.395	0.174	0.234	0.372	0.419
	192	**0.188**	0.247	0.219	0.263	0.198	0.250	0.208	0.258	0.202	0.251	0.213	0.270	0.327	0.375	0.217	0.267	0.403	0.435
	336	**0.238**	**0.286**	0.280	0.307	0.257	0.294	0.251	0.289	0.281	0.312	0.259	0.310	0.360	0.399	0.276	0.311	0.702	0.579
	720	**0.310**	**0.338**	0.358	0.361	0.345	0.355	0.316	0.334	0.345	0.354	0.317	0.355	0.399	0.423	0.360	0.362	0.411	0.429
	Avg	**0.220**	**0.269**	0.259	0.290	0.238	0.277	0.230	0.271	0.246	0.283	0.240	0.292	0.355	0.398	0.257	0.294	0.472	0.466

Figure 14. Model efficiency comparison averaged across all prediction lengths (96, 192, 336, 720) on eight datasets. The size of each bubble represents the number of parameters in the corresponding method, with larger bubbles indicating more parameters.

N. Full Different Look-back Window Results

Table 13. Results with look-back length $S = 96$.

Dataset	MLP		TimeDistill (iTrans)		iTransformer		TimeDistill(Modern)		ModernTCN	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	0.211	0.302	0.188	0.277	**0.180**	**0.270**	**0.188**	**0.277**	0.197	0.282
ETTh1	0.499	0.481	**0.444**	**0.439**	0.453	0.448	**0.443**	**0.433**	0.446	0.433
ETTh2	0.634	0.557	**0.360**	**0.392**	0.383	0.407	**0.365**	**0.396**	0.385	0.406
ETTm1	0.400	0.412	**0.376**	**0.395**	0.407	0.411	**0.376**	**0.395**	0.386	0.400
ETTm2	0.434	0.442	**0.275**	**0.319**	0.291	0.334	**0.274**	**0.319**	0.279	0.322
Solar	0.263	0.321	**0.234**	0.289	0.236	**0.262**	**0.235**	0.293	0.255	**0.278**
Traffic	0.583	0.379	0.488	0.313	**0.421**	**0.282**	**0.499**	0.316	0.645	0.395
Weather	0.246	0.300	**0.245**	0.281	0.260	**0.280**	**0.248**	0.283	0.253	**0.280**
Avg	0.409	0.399	**0.326**	0.338	0.329	**0.337**	**0.328**	0.339	0.356	0.350

Table 14. Results with look-back length $S = 192$.

Dataset	MLP		TimeDistill (iTrans)		iTransformer		TimeDistill(Modern)		ModernTCN	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	0.184	0.284	0.166	0.259	**0.164**	**0.258**	**0.166**	**0.259**	0.171	0.263
ETTh1	0.491	0.484	**0.447**	**0.443**	0.454	0.453	0.432	0.430	**0.429**	**0.427**
ETTh2	0.567	0.522	**0.350**	**0.391**	0.384	0.409	**0.352**	**0.392**	0.372	0.404
ETTm1	0.369	0.392	**0.346**	**0.380**	0.373	0.396	**0.343**	**0.376**	0.365	0.386
ETTm2	0.395	0.424	**0.264**	**0.315**	0.286	0.335	**0.260**	**0.314**	0.265	0.318
Solar	0.213	0.273	**0.199**	**0.258**	0.224	0.260	**0.199**	0.257	0.217	**0.253**
Traffic	0.478	0.338	0.422	0.284	**0.395**	**0.274**	**0.426**	0.286	0.485	0.325
Weather	0.233	0.288	**0.230**	**0.272**	0.246	0.275	**0.231**	**0.273**	0.240	0.273
Avg	0.366	0.376	**0.303**	**0.325**	0.316	0.333	**0.301**	**0.323**	0.318	0.331

Table 15. Results with look-back length $S = 336$.

Dataset	MLP		TimeDistill (iTrans)		iTransformer		TimeDistill(Modern)		ModernTCN	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	0.177	0.279	**0.161**	**0.256**	0.164	0.258	**0.161**	**0.255**	0.166	0.260
ETTh1	0.481	0.487	**0.429**	**0.439**	0.458	0.460	**0.417**	**0.427**	0.418	0.427
ETTh2	0.555	0.521	**0.345**	**0.390**	0.390	0.416	**0.345**	**0.392**	0.351	0.397
ETTm1	0.364	0.390	**0.343**	**0.380**	0.368	0.395	**0.340**	**0.374**	0.363	0.385
ETTm2	0.387	0.416	**0.254**	**0.313**	0.272	0.329	**0.252**	**0.311**	0.266	0.321
Solar	0.203	0.260	**0.194**	**0.248**	0.235	0.270	**0.192**	**0.248**	0.212	0.250
Traffic	0.450	0.326	0.402	0.277	**0.386**	**0.273**	**0.407**	0.278	0.444	0.305
Weather	0.226	0.282	**0.222**	**0.264**	0.239	0.273	**0.223**	**0.267**	0.232	0.269
Avg	0.355	0.370	**0.294**	**0.321**	0.314	0.334	**0.292**	**0.319**	0.307	0.327

Table 16. Results with look-back length $S = 720$.

Dataset	MLP		TimeDistill (iTrans)		iTransformer		TimeDistill(Modern)		ModernTCN	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	0.173	0.276	**0.157**	**0.254**	0.163	0.259	**0.157**	**0.254**	0.167	0.262
ETTh1	0.502	0.489	**0.428**	**0.445**	0.468	0.476	**0.429**	**0.441**	0.469	0.465
ETTh2	0.393	0.438	**0.345**	**0.397**	0.398	0.426	**0.345**	**0.395**	0.357	0.403
ETTm1	0.391	0.413	**0.354**	**0.390**	0.372	0.402	**0.348**	**0.379**	0.390	0.410
ETTm2	0.300	0.373	**0.252**	**0.316**	0.276	0.337	**0.244**	**0.311**	0.267	0.330
Solar	0.194	0.255	**0.185**	**0.241**	0.214	0.270	**0.184**	**0.241**	0.191	0.243
Traffic	0.434	0.318	0.389	**0.271**	0.379	0.271	**0.387**	0.271	0.413	0.284
Weather	0.234	0.294	**0.220**	**0.270**	0.259	0.290	**0.220**	**0.269**	0.238	0.277
Avg	0.328	0.357	**0.291**	**0.323**	0.316	0.341	**0.289**	**0.320**	0.312	0.334

O. Full Ablation Results

Table 17. Ablation study on different datasets (Teacher: **iTransformer** (Liu et al., 2024)).

Method	ECL		ETTh1		ETTh2		ETTm1		ETTm2		Solar		Traffic		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
iTransformer	0.163	0.259	0.468	0.476	0.398	0.426	0.372	0.402	0.276	0.337	0.214	0.270	**0.379**	0.271	0.259	0.290
MLP	0.173	0.276	0.502	0.489	0.393	0.438	0.391	0.413	0.300	0.373	0.194	0.255	0.434	0.318	0.234	0.294
TIMEDISTILL	**0.157**	**0.254**	**0.428**	**0.445**	**0.345**	**0.397**	**0.354**	**0.390**	**0.252**	**0.316**	**0.185**	**0.241**	0.389	**0.271**	**0.220**	**0.270**
w/o prediction level	0.157	0.254	0.480	0.472	0.365	0.413	0.372	0.403	0.261	0.321	0.186	0.242	0.392	0.274	0.221	0.271
w/o feature level	0.163	0.260	0.441	0.452	0.365	0.410	0.373	0.398	0.258	0.320	0.186	0.246	0.393	0.277	0.225	0.277
w/o multi-scale	0.163	0.261	0.483	0.480	0.375	0.423	0.394	0.409	0.268	0.327	0.187	0.248	0.393	0.277	0.223	0.277
w/o multi-period	0.159	0.255	0.507	0.487	0.376	0.422	0.381	0.399	0.268	0.323	0.195	0.256	0.392	0.273	0.222	0.270
w/o sup	0.161	0.258	0.425	0.446	0.353	0.396	0.368	0.396	0.257	0.320	0.205	0.269	0.394	0.277	0.223	0.270

Table 18. Ablation study on different datasets (Teacher: **ModernTCN** (Luo & Wang, 2024)).

Method	ECL		ETTh1		ETTh2		ETTm1		ETTm2		Solar		Traffic		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ModernTCN	0.167	0.262	0.469	0.465	0.357	0.403	0.390	0.410	0.267	0.330	0.191	0.243	0.413	0.284	0.238	0.277
MLP	0.173	0.276	0.502	0.489	0.393	0.438	0.391	0.413	0.300	0.373	0.194	0.255	0.434	0.318	0.234	0.294
TIMEDISTILL	**0.157**	**0.254**	**0.429**	**0.441**	**0.345**	**0.395**	**0.348**	**0.379**	**0.244**	**0.311**	**0.184**	**0.241**	**0.391**	**0.275**	**0.220**	**0.269**
w/o prediction level	0.157	0.254	0.490	0.480	0.370	0.419	0.370	0.401	0.263	0.325	0.184	0.241	0.392	0.275	0.221	0.273
w/o feature level	0.161	0.258	0.442	0.447	0.354	0.401	0.353	0.382	0.248	0.315	0.188	0.250	0.393	0.277	0.224	0.271
w/o multi-scale	0.162	0.260	0.480	0.476	0.380	0.430	0.379	0.402	0.267	0.327	0.187	0.249	0.393	0.278	0.224	0.277
w/o multi-period	0.157	0.254	0.430	0.442	0.346	0.395	0.348	0.379	0.245	0.311	0.184	0.241	0.391	0.274	0.221	0.267
w/o sup	0.165	0.261	0.423	0.438	0.345	0.394	0.356	0.381	0.251	0.317	0.192	0.260	0.506	0.351	0.225	0.269

Table 19. Ablation study measured by MSE on different components of TIMEDISTILL. Teacher is **iTransformer**;

Method	ECL	ETT(avg)	Solar	Traffic	Weather
iTransformer	0.163	0.379	0.214	**0.379**	0.259
MLP	0.173	0.396	0.194	0.434	0.234
TIMEDISTILL	**0.157**	**0.345**	**0.185**	<u>0.389</u>	**0.220**
w/o multi-scale	0.163	0.380	0.187	0.393	0.223
w/o multi-period	0.159	0.383	0.195	0.392	0.222
w/o prediction level	<u>0.157</u>	0.370	0.186	0.392	<u>0.221</u>
w/o feature level	0.163	0.359	<u>0.186</u>	0.393	0.225
w/o sup	0.161	<u>0.351</u>	0.205	0.394	0.223

Table 20. Ablation study measured by MSE on different components of TIMEDISTILL. Teacher is **ModernTCN**.

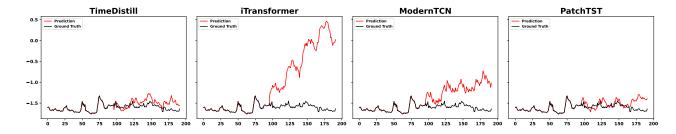| Method	ECL	ETT(avg)	Solar	Traffic	Weather
Teacher	0.167	0.371	0.191	0.413	0.238
MLP	0.173	0.397	0.194	0.434	0.234
TIMEDISTILL	**0.157**	**0.342**	**0.184**	**0.387**	**0.220**
w/o prediction level	<u>0.157</u>	0.373	<u>0.184</u>	0.392	<u>0.221</u>
w/o feature level	0.161	0.349	0.188	0.393	0.224
w/o multi-scale	0.162	0.377	0.187	0.393	0.224
w/o multi-period	0.157	<u>0.342</u>	0.184	<u>0.391</u>	0.221
w/o sup	0.165	0.344	0.192	0.506	0.225

P. Show Cases

Figure 15. Prediction cases from ECL by different models under the input-720-predict-96 settings. **Black** lines are the ground truths and **Red** lines are the model predictions. Due to space constraints, we only retained the last 96 time steps of input for plotting.
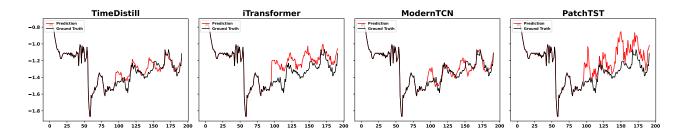

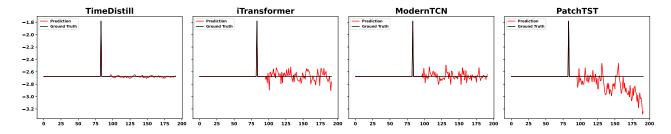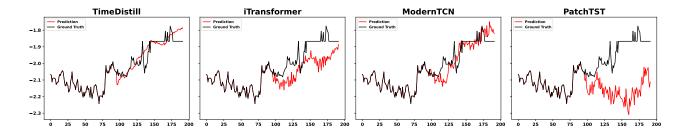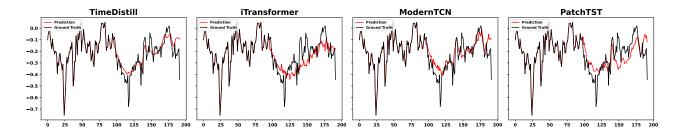*Figure 16.* Prediction cases from ETTh1 by different models under the input-720-predict-96 settings.

Figure 17. Prediction cases from ETTh2 by different models under the input-720-predict-96 settings.

Figure 18. Prediction cases from ETTm1 by different models under the input-720-predict-96 settings.

Figure 19. Prediction cases from ETTm2 by different models under the input-720-predict-96 settings.
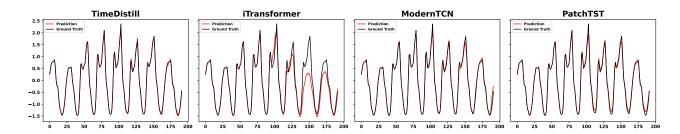

Figure 20. Prediction cases from Traffic by different models under the input-720-predict-96 settings.
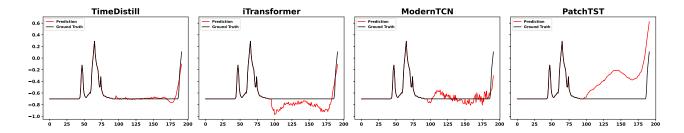

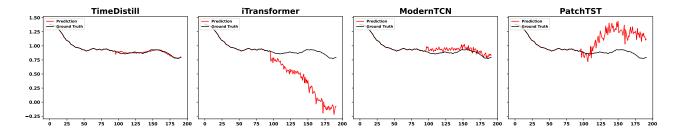*Figure 21.* Prediction cases from Solar by different models under the input-720-predict-96 settings.

Figure 22. Prediction cases from Weather by different models under the input-720-predict-96 settings.