# CoMRes: Semi-Supervised Time Series Forecasting Utilizing Consensus Promotion of Multi-Resolution

**Anonymous authors**
Paper under double-blind review

## Abstract

Long-term time series forecasting poses significant challenges due to the complex dynamics and temporal variations, particularly when dealing with unseen patterns and data scarcity. Traditional supervised learning approaches, which rely on cleaned and labeled data, struggle to capture these unseen characteristics, limiting their effectiveness in real-world applications. In this study, we propose a semi-supervised approach that leverages multi-view setting on augmented data without requiring explicit future values as labels to address these limitations. By introducing a consensus promotion framework, our method enhances agreement among multiple single-view models on unseen augmented data. This approach not only improves forecasting accuracy but also mitigates error accumulation in long-horizon predictions. Furthermore, we explore the impact of autoregressive and non-autoregressive decoding schemes on error propagation, demonstrating the robustness of our model in extending prediction horizons. Experimental results show that our proposed method not only surpasses traditional supervised models in accuracy but also exhibits greater robustness when extending the prediction horizon.

## 1 Introduction

Time series forecasting has seen significant advancements in recent years, with research making substantial progress in capturing the complex dynamics inherent in time-series data (Zhou et al., 2022; Nie et al., 2023; Chen et al., 2024; Liu et al., 2024; Challu et al., 2023; Das et al., 2023; Wang et al., 2024; Zeng et al., 2023). Due to the complex and non-stationary nature of real-world systems, observed time series often exhibit intricate temporal patterns, such as multi-scale dependencies and diverse fluctuations. To capture the multi-scale characteristics effectively, several advanced models structurally incorporate multi-scale features(Chen et al., 2024; Shabani et al., 2023; Wang et al., 2024; Challu et al., 2023). This approach generally increases model capacity, allowing for the learning of complex patterns across different scales through an increase in parameters. However, this increased capacity also increases the risk of overfitting, especially when training on small or low-diversity datasets. Addressing these challenges is essential for developing stable and reliable multi-resolution forecasting models.

In time series analysis, several data augmentation methods have been proposed to enhance model's generalization performance (Wen et al., 2021b; Iglesias et al., 2023; Debnath et al., 2021; Semenoglou et al., 2023). Despite their widespread use, labeling augmented data remains a significant challenge, particularly in forecasting tasks. This challenge arises because the augmented data may alter the underlying dynamics of the time series and often introduce noise. Consequently, assigning accurate labels for future predictions becomes difficult, as the augmented past values represent artificial, potentially noisy data rather than actual observations. Even slight shifts in trends or periodic patterns can lead to increased prediction errors, thereby limiting the direct applicability of augmented data in time series forecasting. However, the use of diverse and previously unseen data is essential to address the complex dynamics and time-varying heterogeneity of long-term time series data. Therefore, reducing the risk of model overfitting and enhancing generalization ability without manually assigning future values to augmented data is crucial for effectively utilizing data augmentation in time series forecasting.

This study investigates the use of augmented data in long-term time series forecasting through a multi-view learning approach. Multi-view data has demonstrated considerable success in various domains, such as computer vision and natural language processing (Chen et al., 2021; Lim et al., 2020), by leveraging different manifestations of the same data to improve performance. In time series analysis, multi-view data can be represented by features such as varied temporal resolutions or distances, allowing for the extraction of temporal characteristics and dependencies across multiple time intervals, ultimately enhancing the accuracy and robustness of forecasts (Challu et al., 2023; Liu et al., 2022; Shabani et al., 2023; Chen et al., 2024). Many multi-scale modeling techniques integrate features from different time scales but are limited to labeled data. By employing semi-supervised learning, which leverages unlabeled data, we can fully utilize multi-scale features across diverse patterns.

Thus, we propose a semi-supervised approach that improves individual single-view models by enhancing consensus among them using unseen data. Inspired by traditional co-training methods (Blum & Mitchell, 1998; Lim et al., 2020), this approach leverages augmented unseen data to ensure that models not only capture multi-resolution patterns but also provide consistent predictions when faced with new, unseen data. The core of this **CoMRes** (**Co**nsensus promotion of **M**ulti-**Res**olution) lies in maximizing agreement among multi-view models on unseen data, thereby improving the overall performance and reliability of long-term time series forecasting.

In addition, we address the issue of error accumulation in long-horizon forecasting. In transformer-based research for long-term time series forecasting, a common approach is the use of a non-autoregressive decoding scheme, which projects future horizons all at once. Zhang et al. (2023) were the first to conduct a comparative evaluation of several models in terms of autoregressive and non-autoregressive decoding schemes. Motivated by their work, we evaluate our model with respect to both decoding styles. Specifically, we investigate how non-autoregressive models can be employed to minimize error accumulation when predicting values further into the future than the forecast range on which they were trained.

In summary, our contributions are as follows:

- We propose **CoMRes** (**Co**nsensus promotion of **M**ulti-**Res**olution), a novel semi-supervised time series forecasting that utilizes consensus promotion of multi-resolution information and augmented data without requiring explicit future values as labels to enhance long-term time series forecasting. To the best of our knowledge, this is the first work to investigate the influence of a semi-supervised framework in the context of long-term time series forecasting.
- We propose evaluating the forecasting model using both autoregressive and non-autoregressive decoding schemes. Our study focuses on how a forecasting model can extend its prediction horizon beyond the trained range while minimizing error accumulation, particularly when non-autoregressive models are used to generate autoregressive forecasts.
- To further understand CoMRes, we conduct ablation study the impact of incorporating data augmentation and consensus promotion strategies in semi-supervised and supervised learning.
- We evaluate the impact of our CoMRes under various training data size and analyze the effect of different data augmentation techniques to enhance model performance.

## 2 Related Work

**Long-term Time Series Forecasting** Recent research has made significant progress in improving model architectures for time-series forecasting. Transformer based model such as FEDformer (Zhou et al., 2022) and Autoformer (Wu et al., 2021), apply attention mechanisms to multivariate time series data. These models address the quadratic complexity of traditional attention mechanisms by introducing novel mechanisms to reduce computational complexity. PatchTST (Nie et al., 2023) utilized patch-based representations to enhance local pattern recognition, while iTransformer (Liu et al., 2024) and Crossformer (Zhang & Yan, 2023) pushed forward by capturing multivariate correlations through sophisticated attention mechanisms. Meanwhile, several recent works (Das et al. (2023); Wang et al. (2023); Zeng et al. (2023); Wang et al. (2024)) demonstrate the effectiveness of non-transformer-based methods, offering competitive alternatives.

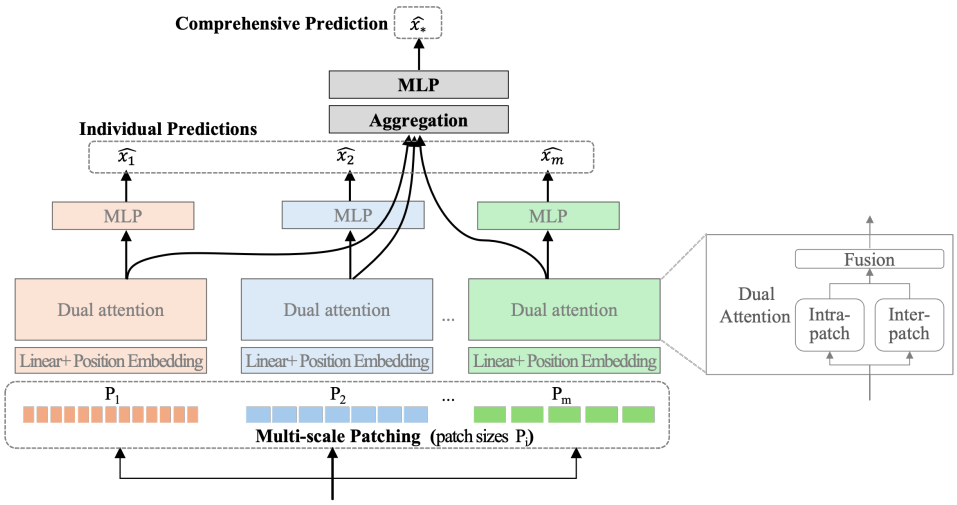



Figure 1: Overall architecture of multi-resolution transformer, CoMRes and MRes. There are $m$ individual blocks and a aggregation block. Each individual block predict *individual* prediction, $\widehat{x}_i$. The aggregation block produce *comprehensive* prediction, $\widehat{x}_*$.

**Multi-scale Modeling for Time Series** Real-world time series often exhibit diverse variations and fluctuations across different temporal scales. To address this, recent research has increasingly focused on leveraging multi-scale information to better capture these complex temporal patterns. NHITS (Challu et al., 2023) employs multi-rate data sampling and hierarchical interpolation to model features of different resolutions. Pyraformer (Liu et al., 2022) and Scaleformer (Shabani et al., 2023) advanced the field with pyramidal attention mechanisms and iterative scale-refinement mechanisms, respectively, to capture multi-scale dependencies. TimeMixer (Wang et al., 2024) introduce a new multi-scale mixing architecture with decomposable mixing strategy. Pathformer (Chen et al., 2024) takes a more comprehensive approach by considering time series features from both different resolutions and temporal distances. Most previous methods integrate complementary forecasting capabilities through an ensemble of multiple predictions that rely on labeled data. By employing semi-supervised learning, which leverages unlabeled data, our approach effectively tackles the challenge of reconciling diverse representations, particularly in the context of unseen data.

**Time Series Data Augmentation** Comprehensive reviews of the most popular data augmentation approaches in time series analysis mostly focus on classification and anomaly detection tasks (Wen et al., 2021b; Iglesias et al., 2023). Data augmentation is easily applicable for time series classification and anomaly detection as small perturbations typically do not change the data label. Nonetheless, it is not straightforward to apply data augmentation techniques in timeseries forecasting as even small perturbations can lead to significant changes in the observations. Debnath et al. (2021) and Semenoglou et al. (2023) explored the impact of data augmentation on time series forecasting, and their findings show that the improvement relies on the quality of the augmented data. Similarly, Wen et al. (2021a) surveys time series data augmentation methods, showing performance boosts in classification and anomaly detection but noting negative effects for certain data/model pairs in forecasting. Our findings (Table 1) show that high-quality augmentation is challenging to achieve. Simply expanding labeled data with standard techniques often worsens performance. In this study, we utilize the same data augmentation technique differently, using it exclusively for consensus promotion, which has shown improvements in most of our experiments.

## 3 METHODOLOGY

In this paper, we study the problem of long-term time-series forecasting: Given historical data of a multivariate time series with a look-back window $L : (x_{t-L}, \ldots, x_t)$, where $x_t$ denotes the observation at timestamp $t$, our objective is to predict the future values for $H$ timestamps $(x_{t+1}, \ldots, x_{t+H})$. We use a similar model architecture to Pathformer, introduced by Chen et al. (2024), as our base model

and extend it to leverage unseen data. Our model architecture for long-term time series forecasting is depicted in Figure 1.

To effectively capture multi-scale characteristics, we employ multi-scale patch division with various patch sizes ($P_1, P_2, \ldots, P_m$) and dual attention transformer. The multi-scale patch division provides different views of the time series at different resolutions. These patches are fed into the transformer encoder, which comprises intra-patch attention within each divided patch and inter-patch attention across different patches. Following the transformer encoder, Multi-Layer Perceptrons (MLPs) generate individual predictions ($\widehat{x}_i$) for each patch size. These predictions are then aggregated using an equal weighted aggregation strategy, resulting in a single comprehensive prediction ($\widehat{x}_*$). Unlike Chen et al. (2024), we use equal weighted aggregator to combine these multi-scale characteristics.

During supervised learning, we train the model to minimize the mean squared error (MSE) between the predictions and the ground truth. Given that there are $m+1$ predictions, we average the errors of each prediction. The supervised loss is optimized as follows:

$$\mathcal{L}_s = \frac{1}{m+1}\left(||\widehat{x_*}_{t+1:t+H} - x_{t+1:t+H}||^2 + \sum_{i=1}^{m} ||\widehat{x}_{i\,t+1:t+H} - x_{t+1:t+H}||^2\right)$$

In the following sections, we provide a detailed description of consensus promotion learning on unseen data and the process of generating unseen data. Subsequently, we explain how to extend prediction horizons beyond the training ranges.

### 3.1 Consensus Promotion Learning

Due to the complex dynamics of long-term time series data, learning models from diverse and unseen data is crucial for improving forecasting accuracy. However, determining the future values of unseen augmented data is challenging, as this data is synthetic and potentially noisy. To address this issue, we propose a method inspired by standard multi-view learning approaches, which aim to jointly optimize multiple models derived from distinct perspectives, each providing complementary information. Drawing from the concept of one model teaching others, as introduced by Blum & Mitchell (1998) and Lim et al. (2020), we suggest using a *consistency loss* that encourages mutual agreement among the models. This approach leverages the unique information provided by different views to enhance comprehensive predictions ($\widehat{x}_*$), effectively capturing the complex, multi-resolution patterns inherent in long-term time series forecasting.

To achieve this, we propose that each individual-view model learns from the comprehensive prediction ($\widehat{x}_*$) by optimizing the following unsupervised loss:

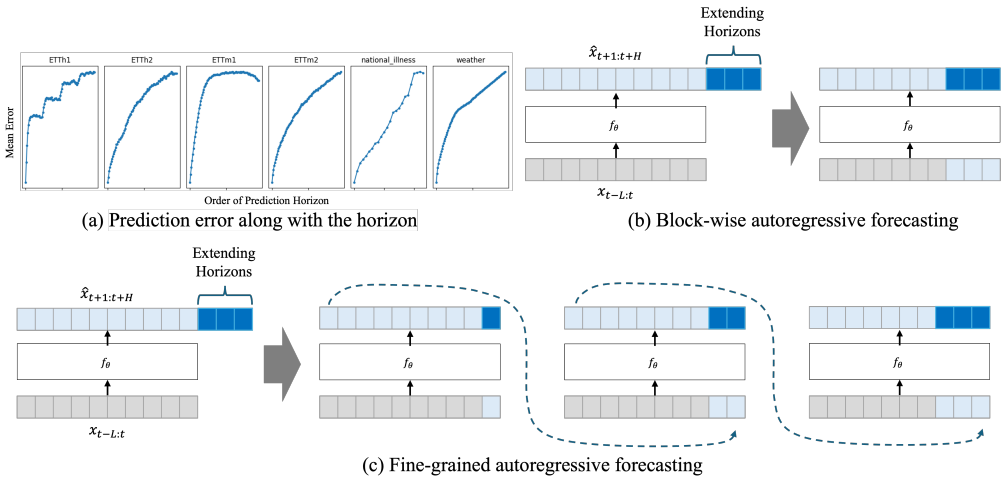
$$\mathcal{L}_u = \frac{1}{m}\left(\sum_{i=1}^{m} ||\widehat{x}_{i\,t+1:t+H} - \widehat{x_*}_{t+1:t+H}||^2\right)$$

This encourages the individual-view models to align their predictions with the comprehensive prediction, thereby enhancing mutual agreement among the models. The overall objective for training the model combines the supervised loss and the unsupervised consistency loss as follows:

$$\mathcal{L} = \mathcal{L}_s + w_u \mathcal{L}_u$$

where $w_u$ represents the weight ot the unsupervised consistency loss $\mathcal{L}_u$. This joint objective ensures that the model not only minimizes the prediction error with respect to the ground truth($\mathcal{L}_s$) but also aligns the individual model predictions with the comprehensive prediction($\mathcal{L}_u$).

The core principle of consensus promotion learning is to achieve best possible forecasting results on unseen data. By leveraging the complementary information from multiple views and aligning individual model predictions with a comprehensive prediction, this approach reduces the variance of individual predictions(see Appendix A.2), leading to more stable and reliable results on unseen data. This is because the consensus or ensemble of predictions can help mitigate the biases and errors that may be present in any single model, allowing for better generalization to new, previously unobserved patterns.




(a) Prediction error along with the horizon

(b) Block-wise autoregressive forecasting

(c) Fine-grained autoregressive forecasting

Figure 2: Extending Prediction Horizons Beyond Training Ranges. Figure (a) demonstrates that prediction error escalates as the prediction horizon lengthens. Figure (b) depicts block-wise autoregressive forecasting, which predicts the entire extended horizon at once, while Figure (c) illustrates fine-grained autoregressive forecasting, which predicts step-by-step. Both approaches use autoregressive forecasting with a non-autoregressive trained model, but the fine-grained approach utilizes the initial prediction due to its relatively small size.

### 3.2 Unseen Data Generation

Consensus promotion aims to improve model generalization by leveraging information from unseen data. To create this unseen data, we use data augmentation techniques such as time warping, interpolation, and noise injection. The objective of generating unseen data is to construct a synthetic dataset that explores new areas of the input space. In supervised learning, data augmentation typically assumes that the relationship between the lookback period and the prediction horizon is well-defined, with known ground-truth labels. In contrast, our consensus promotion approach does not require defining ground truth for augmented data, allowing for greater flexibility in capturing the complex, dynamic nature of time series data.

Various methods for augmenting time series data in deep learning are reviewed in Wen et al. (2021b), Debnath et al. (2021), Iglesias et al. (2023), and Semenoglou et al. (2023). While numerous advanced methods exist, we have chosen to focus on basic augmentation techniques due to their simplicity. We include three variants of CoMRes: interpolation, time warping, and noise injection. Combining multiple augmentation techniques to enhance model performance will be discussed in Section 4.3.2.

**Interpolation** involves estimating values between known data points to produce a smoother and more continuous time series. For example, linear interpolation estimates values between two known points using a straight line. In this study, we adopt linear interpolation.

**Time Warping** includes techniques such as stretching or compressing the time window of data. Similar to dynamic time warping (DTW), this method selects a random time range and then compresses (down-samples) it. In this study, we compress the time window by a factor of 3.

**Noise Injection** adds random noise to the data to introduce variability. This includes injecting Gaussian noise, spike, step-like trend, and slope-like trend, etc. In this study, Gaussian noise, which consists of random values drawn from a normal distribution, is used. After normalization, we add normal noise with a mean of 0 and a standard deviation of 0.01

### 3.3 Extending Prediction Horizons Beyond Training Ranges

In addition to the typical evaluation of forecasting performance, which involves measuring the prediction error over $H$ timestamps (training horizon), we also assess our model's performance when predicting beyond this training horizon. For applications requiring long-term future predictions, such

as stock prices or energy consumption, autoregressive forecasting is essential. However, autoregressive forecasting inherently suffers from error accumulation, especially when the model has not been trained autoregressively over an extended horizon, such as with teacher forcing. As depicted in Figure 2(a), our non-autoregressive model's prediction errors escalate as the prediction horizon lengthens. Our objective is to explore methods that can extend the prediction horizon while minimizing error accumulation in autoregressive forecasting scenarios.

In this study, we evaluate the model trained with consensus promotion through two approaches: Block-wise autoregressive prediction and fine-grained autoregressive prediction. The block-wise autoregressive prediction method, illustrated in Figure 2(b), extends the prediction horizon by utilizing the entire prediction range at once. In contrast, fine-grained autoregressive prediction involves a step-by-step process where the output of one step serves as the input for the next step, emphasizing a recurrent or sequential processing approach, as shown in Figure 2(c). This progression demonstrates the iterative application of the model to refine predictions over multiple steps.

# 4 EXPERIMENTS

## 4.1 TIME SERIES FORECASTING

**Datasets.** We evaluate our model on 8 popular datasets, including 4 ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, Traffic, Electricity, ILI. These datasets have been extensively utilized for benchmarking and publicly available on Wu et al. (2021) and the data statistics are shown in Appendix A.1.1.

**Baselines.** Our experiments aim to evaluate the impact of two core strategies—consensus promotion and data augmentation—on both labeled and unlabeled data. We employed the multi-resolution division architecture presented by Chen et al. (2024). However, we did not implement the adaptive pathways, which are designed to dynamically adjust the multi-resolution modeling process. Our proposed model (**CoMRes**) and the baseline models (**MRes** for **M**ulti-**Res**olution aggregation transformer) share the same architecture, as illustrated in Figure 1. To assess the impact of different unseen data generation methods, we include three variants of CoMRes: interpolation, time warping, and noise injection. We established four baselines:

- MRes (SL): MRes trained exclusively on labeled datasets using supervised learning (SL).
- MRes w. augmentation: MRes Trained on labeled dataset that is augmented with Time Warping.
- MRes w. consensus: MRes Trained on labeled datasets with consensus promotion without data augmentation. Prediction on labeled data is learned not only from the ground truth but also from the comprehensive prediction.
- Pathformer: As originally suggested by Chen et al. (2024), trained with adaptive pathways.

MRes with data augmentation and consensus promotion on all augmented data can be viewed as CoMRes.

**Experimental Setup.** Each dataset was split into training, validation, and test sets. Mean squared error (MSE) and mean absolute error (MAE) were used as evaluation metrics, with each experiment repeated five times, and the average values reported(see Appendix A.4 for error bars). Following the setup from the Pathformer paper, all models used the same input and prediction lengths: $L = 36$ for the ILI dataset and $L = 96$ for the others. Prediction lengths were $H \in \{24, 36, 48, 60\}$ for ILI and $H \in \{96, 192, 336, 720\}$ for the others. More experimental details are in Appendix A.1.

**Results.** The experimental results, summarized in Table 1, indicate that our proposed model outperforms the baselines on most datasets. Notably, CoMRes, which incorporates consensus promotion on unseen data significantly enhanced the model's generalization ability, as evidenced by lower MSE scores on the test sets. Although MRes with consensus promotion showed good results in some datasets, these improvements were inconsistent across others. Both CoMRes and MRes with consensus incorporate consensus promotion; however, CoMRes applies it to unseen data, whereas MRes with consensus applies it to labeled data. This distinction is critical, as adding a consistency loss term alongside supervised loss can improve performance but also carries the risk of overfitting when

Table 1: Multivariate time series forecasting results (MSE). The input length $L = 96$ ($L = 36$ for the ILI dataset). The best results are highlighted in **bold**, and the second-best results are underlined. The $\Delta$ column shows the difference between CoMRes's best results and the baselines, MRes with ablated components and Pathformer, with blue indicating that CoMRes outperforms the baseline and red indicating otherwise.

| Method | | CoMRes (ours) | | | MRes | | | | | | Pathformer | $\Delta$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Time Warping | Interpolation | Noise Injection | w. augmentation | $\Delta$ | w. consensus | $\Delta$ | MRes (SL) | $\Delta$ | | |
| Augmentation | | ○ | ○ | ○ | ○ | | × | | × | | | |
| Consensus Promotion | | ○ | ○ | ○ | × | | ○ | | × | | | |
| Leveraging Unlabeled Data | | ○ | ○ | ○ | × | | × | | × | | | |
| ETTh1 | 96 | **0.378** | 0.379 | 0.379 | 0.406 | 0.028 | 0.379 | 0.001 | 0.382 | 0.004 | 0.382 | 0.004 |
| | 192 | **0.436** | **0.436** | **0.436** | 0.455 | 0.019 | **0.436** | 0.000 | 0.439 | 0.003 | 0.440 | 0.004 |
| | 336 | 0.452 | 0.452 | 0.452 | 0.482 | 0.030 | 0.452 | 0.000 | **0.451** | -0.001 | 0.454 | 0.002 |
| | 720 | **0.474** | 0.478 | 0.478 | 0.475 | 0.001 | 0.478 | 0.004 | 0.481 | 0.007 | 0.479 | 0.005 |
| | Avg. | **0.435** | 0.436 | 0.436 | 0.455 | 0.020 | 0.436 | 0.001 | 0.438 | 0.003 | 0.439 | 0.004 |
| ETTh2 | 96 | **0.275** | **0.275** | **0.275** | 0.278 | 0.003 | **0.275** | 0.000 | 0.277 | 0.002 | 0.279 | 0.004 |
| | 192 | 0.347 | **0.346** | **0.346** | 0.355 | 0.009 | **0.346** | 0.000 | 0.348 | 0.002 | 0.349 | 0.003 |
| | 336 | **0.326** | 0.328 | **0.326** | 0.346 | 0.020 | 0.330 | 0.004 | 0.332 | 0.006 | 0.348 | 0.022 |
| | 720 | **0.393** | 0.398 | 0.397 | 0.404 | 0.011 | 0.405 | 0.012 | 0.406 | 0.013 | 0.398 | 0.005 |
| | Avg. | **0.335** | 0.337 | 0.336 | 0.346 | 0.011 | 0.339 | 0.004 | 0.341 | 0.006 | 0.344 | 0.009 |
| ETTm1 | 96 | **0.311** | **0.311** | **0.311** | 0.326 | 0.016 | **0.311** | 0.001 | 0.312 | 0.001 | 0.316 | 0.005 |
| | 192 | 0.362 | 0.364 | 0.364 | 0.367 | 0.005 | 0.364 | 0.002 | **0.361** | -0.001 | 0.366 | 0.004 |
| | 336 | 0.386 | 0.386 | 0.387 | 0.404 | 0.018 | 0.386 | 0.000 | **0.381** | -0.005 | 0.386 | 0.000 |
| | 720 | 0.456 | 0.456 | 0.456 | 0.467 | 0.011 | 0.456 | 0.000 | **0.454** | -0.002 | 0.460 | 0.004 |
| | Avg. | 0.379 | 0.379 | 0.380 | 0.391 | 0.013 | 0.379 | 0.001 | **0.377** | -0.002 | 0.382 | 0.003 |
| ETTm2 | 96 | **0.163** | **0.163** | **0.163** | 0.168 | 0.005 | 0.166 | 0.003 | 0.167 | 0.004 | 0.170 | 0.007 |
| | 192 | **0.229** | **0.229** | **0.229** | 0.230 | 0.001 | **0.229** | 0.000 | 0.231 | 0.002 | 0.238 | 0.009 |
| | 336 | **0.291** | 0.292 | 0.292 | 0.295 | 0.004 | 0.292 | 0.001 | 0.292 | 0.001 | 0.293 | 0.002 |
| | 720 | **0.369** | **0.369** | **0.369** | 0.385 | 0.016 | 0.383 | 0.014 | **0.369** | 0.000 | 0.390 | 0.021 |
| | Avg. | **0.263** | **0.263** | **0.263** | 0.270 | 0.007 | 0.268 | 0.005 | 0.265 | 0.002 | 0.273 | 0.010 |
| Electricity | 96 | 0.152 | 0.153 | 0.153 | 0.162 | 0.010 | 0.156 | 0.004 | 0.150 | -0.002 | **0.145** | 0.007 |
| | 192 | 0.167 | 0.167 | 0.166 | 0.173 | 0.007 | 0.169 | 0.002 | **0.164** | -0.002 | 0.167 | 0.001 |
| | 336 | **0.181** | **0.181** | **0.181** | 0.186 | 0.005 | 0.184 | 0.003 | **0.181** | 0.000 | 0.186 | 0.005 |
| | 720 | **0.210** | 0.211 | 0.214 | 0.218 | 0.008 | 0.214 | 0.004 | 0.223 | 0.013 | 0.231 | 0.021 |
| | Avg. | **0.178** | **0.178** | 0.179 | 0.185 | 0.007 | 0.181 | 0.003 | 0.180 | 0.002 | 0.182 | 0.004 |
| ILI | 24 | 1.957 | 1.907 | 1.956 | 1.760 | -0.147 | 1.986 | 0.029 | 1.976 | 0.069 | **1.587** | -0.320 |
| | 36 | 1.640 | 1.599 | 1.538 | 1.569 | 0.031 | 1.540 | -0.100 | 1.551 | 0.013 | **1.429** | -0.109 |
| | 48 | 1.529 | 1.516 | 1.490 | 1.514 | 0.024 | **1.456** | -0.073 | 1.497 | 0.007 | 1.505 | 0.015 |
| | 60 | 1.693 | 1.739 | 1.715 | 1.727 | 0.034 | **1.679** | -0.014 | 1.714 | 0.021 | 1.731 | 0.038 |
| | Avg. | 1.705 | 1.690 | 1.675 | 1.643 | -0.032 | 1.665 | -0.039 | 1.685 | 0.010 | **1.563** | 0.112 |
| Traffic | 96 | **0.468** | 0.470 | 0.470 | 0.475 | 0.007 | 0.479 | 0.011 | 0.478 | 0.010 | 0.479 | 0.011 |
| | 192 | **0.465** | 0.466 | 0.466 | 0.482 | 0.017 | 0.479 | 0.014 | 0.476 | 0.011 | 0.484 | 0.019 |
| | 336 | **0.493** | 0.494 | 0.494 | 0.511 | 0.018 | 0.501 | 0.008 | 0.501 | 0.008 | 0.503 | 0.010 |
| | 720 | **0.535** | **0.535** | **0.535** | 0.540 | 0.537 | 0.005 | 0.556 | 0.002 | 0.021 | 0.537 | 0.002 |
| | Avg. | **0.490** | 0.491 | 0.491 | 0.502 | 0.012 | 0.499 | 0.009 | 0.503 | 0.013 | 0.501 | 0.011 |
| Weather | 96 | **0.151** | **0.151** | **0.151** | 0.155 | 0.004 | 0.154 | 0.003 | 0.152 | 0.001 | 0.156 | 0.005 |
| | 192 | **0.199** | 0.200 | 0.200 | 0.203 | 0.004 | 0.201 | 0.002 | **0.199** | 0.000 | 0.206 | 0.007 |
| | 336 | **0.244** | 0.246 | 0.246 | 0.248 | 0.010 | 0.246 | 0.004 | 0.245 | 0.002 | 0.254 | 0.010 |
| | 720 | 0.335 | 0.335 | 0.334 | 0.334 | 0.000 | **0.333** | -0.002 | 0.334 | 0.000 | 0.340 | 0.006 |
| | Avg. | **0.232** | 0.233 | 0.233 | 0.235 | 0.003 | 0.234 | 0.001 | 0.233 | 0.001 | 0.239 | 0.007 |

applied to labeled data. Additionally, MRes with augmentation where augmented data is artificial and potentially noisy data performs poorly. Our results indicate that CoMRes consistently outperforms the baselines across multiple datasets. The case of ILI, where CoMRes underperforms compared to Pathformer, is discussed in detail in Appendix A.7. Interestingly, MRes (SL) outperformed Pathformer in most cases, suggesting that Pathformer's strengths lie in its ability to process multi-resolution information and its inter- and intra-attention design, rather than in the use of adaptive pathways. Among the unseen data generation techniques of CoMRes, time warping achieved slightly better performance compared to interpolation and noise injection. Overall, these findings demonstrate the efficacy of our approach in improving time series forecasting performance. Complementary results for mean absolute error(MAE) are provided in the Appendix A.3.

Figure 3 illustrates the mean error against the prediction horizon, demonstrating how prediction errors increase as the forecast extends further into the future from the lookback period. As shown in Figure 3, our proposed model consistently outperforms the baseline model in terms of prediction accuracy, especially over longer horizons. This comparison underscores the effectiveness of our approach in reducing errors in long-term time series forecasting.

### 4.2 Error Accumulation in Auto-regressive Forecasting

**Experimental Setting.** In this section, we conduct experiments to evaluate our proposed model (CoMRes, Time Warping) and the baseline model (MRes, SL) on test data, focusing on predictions beyond the training horizon. The datasets used are described in Section 4.1.

We employ two approaches for autoregressive prediction:

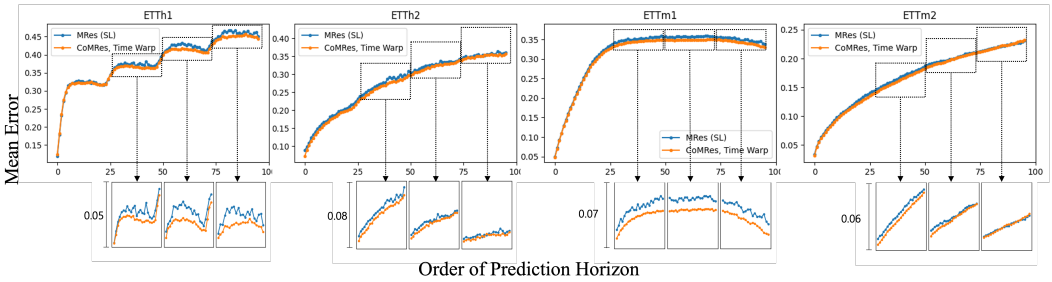



Figure 3: Prediction error along with the prediction horizon. The orange line representing the mean error of the proposed method is lower than that of MRes (SL), the blue line.

Table 2: Autoregressive forecasting result. Mean Squared Error on the extended horizon beyond the trained horizon. The best results are highlighted in **bold**. In both (a) and (b) autoregressive forecasting, CoMRes predicts more accurately than MRes. Additionally, (a) block-wise autoregressive forecasting produces more accurate predictions compared to (b) fine-grained autoregressive forecasting.

| Method | | (a) Block-wise Autoregressive | | (b) Fine-grained Autoregressive | |
|---|---|---|---|---|---|
| Data | Horizon | CoMRes, Time Warp | MRes (SL) | CoMRes, Time Warp | MRes (SL) |
| ETTh1 | 96 + 96 | **0.465** | 0.477 | 0.469 | 0.490 |
| | 192 + 96 | **0.476** | 0.490 | 0.481 | 0.499 |
| | 336 + 96 | **0.473** | 0.491 | 0.481 | 0.506 |
| | 720 + 96 | 0.505 | 0.513 | **0.500** | 0.507 |
| ETTh2 | 96 + 96 | **0.420** | 0.453 | 0.429 | 0.465 |
| | 192 + 96 | **0.428** | 0.451 | 0.430 | 0.455 |
| | 336 + 96 | 0.430 | 0.463 | **0.429** | 0.468 |
| | 720 + 96 | 0.445 | **0.437** | 0.448 | 0.469 |
| ETTm1 | 96 + 96 | **0.412** | 0.414 | 0.424 | 0.442 |
| | 192 + 96 | **0.416** | 0.418 | 0.421 | 0.429 |
| | 336 + 96 | **0.426** | 0.429 | 0.430 | 0.432 |
| | 720 + 96 | **0.449** | 0.451 | 0.454 | 0.455 |
| ETTm2 | 96 + 96 | **0.332** | 0.336 | 0.351 | 0.416 |
| | 192 + 96 | 0.358 | **0.355** | 0.361 | 0.404 |
| | 336 + 96 | **0.374** | 0.377 | 0.384 | 0.378 |
| | 720 + 96 | **0.417** | 0.420 | 0.420 | 0.421 |
| Weather | 96 + 96 | **0.280** | 0.285 | 0.303 | 0.425 |
| | 192 + 96 | **0.303** | 0.305 | **0.303** | 0.425 |
| | 336 + 96 | **0.323** | 0.347 | 0.331 | 0.356 |
| | 720 + 96 | **0.358** | 0.361 | 0.361 | 0.372 |

(a) Block-wise Autoregressive Prediction: The model generates predictions for the entire extended horizon in one step. Specifically, the model uses part of its own prediction as input to produce predictions beyond this range at once.

(b) Fine-grained Autoregressive Prediction: The model generates predictions step-by-step, where each output serves as the input for the next prediction step.As depicted in Figure 2(a), the initial prediction error over the prediction horizon is relatively small. Therefore, if we use the initial prediction as an input to extend the horizon step by step, we can expect to refine the predictions over multiple steps. In this approach, we update the input with the last $k$=1 value using the previous prediction.

In both approaches, the extended prediction horizon is 96 sequences beyond the model's initial prediction horizon.

**Autoregressive Forecasting Results.** Table 2 presents the results of our autoregressive forecasting evaluation. Compared to the errors shown in Table 1, the mean errors increased due to error accumulation. This issue is inherent in autoregressive forecasting, particularly when the model has not been trained autoregressively over an extended horizon, such as with teacher forcing. Despite these challenges, CoMRes consistently outperformed the MRes(SL) across all prediction horizons and both autoregressive methods, demonstrating superior predictive accuracy. The block-wise autoregressive method generally shows smaller error accumulation compared to the fine-grained autoregressive method. This suggests that the block-wise approach handles autoregressive error accumulation better. To investigate the accumulation of errors in block-wise autoregressive forecasting, we compared the prediction error over the entire horizon between a model trained on a longer forecast range and

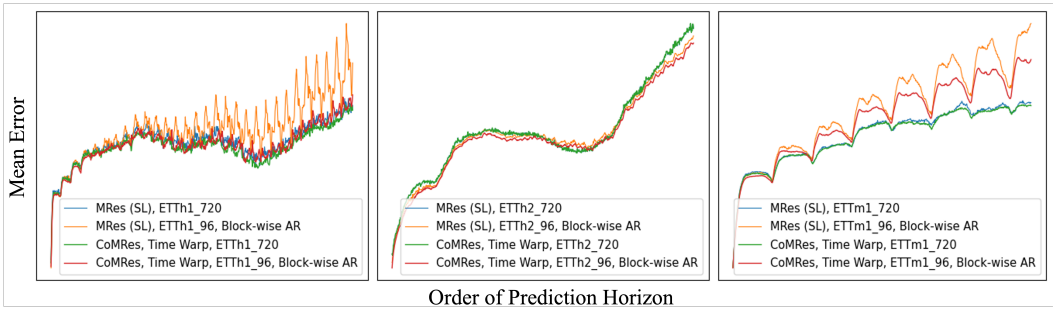



Figure 4: Comparion of Error Propagation. The red line(CoMRes, Time Warp, 96 block-wise AR) shows less error accumulation in long-term forecasting compared to the orange line(MRes(SL), 96 block-wise AR), indicating that the our proposed method is more effective at mitigating error accumulation.

one using block-wise autoregressive predictions made with a shorter model. Figure 4 illustrates the performance of block-wise autoregressive forecasting with a shorter predictor model, specifically one trained to predict 96 steps, which is iteratively applied 8 times to forecast longer horizons (up to 768 time points). This approach is compared against a baseline model trained to predict 720 steps directly.

In the block-wise autoregressive setup, MRes (represented by the orange lines in the 96-step block-wise autoregressive approach) suffers from a substantial increase in mean error as the prediction horizon extends, especially when iterating the shorter model. This greater accumulation of errors in the traditional supervised learning makes it more susceptible to compounding small errors with each iteration. On the other hand, the CoMRes (shown by the green and red lines) demonstrates a stronger capacity to reduce error accumulation, maintaining significantly lower error rates even when using block-wise autoregressive forecasting. While error accumulation occurs in CoMRes, for some data, the block-wise autoregressive model's error is comparable to that of the longer horizon non-autoregressive model. This indicates that it is feasible to effectively extend the prediction horizon of CoMRes using a shorter model.

### 4.3 Ablation Study

#### 4.3.1 Limited-Resource Scenarios

Real-world time series often exhibit complex temporal variations, making it crucial to effectively leverage training data under resource constraints. To assess our method's performance in limited-resource scenarios, we conducted an ablation study by systematically reducing the amount of labeled training data. We compared our CoMRes, Time Warp method against the MRes baseline across several datasets, using 20%, 50%, and 80% of the original training set to evaluate performance.

The results in Table 3 demonstrate that CoMRes consistently achieves the best performance across all label sizes and prediction horizons compared to the baseline. Due to temporal variations, the temporal dynamics in distant periods may differ from those in the test period, leading to irrelevant information being included and causing worse predictions even with a large amount of data. Furthermore, on the ETTh2 and ETTm2 datasets, CoMRes delivers comparable performance to MRes, even at smaller label sizes. Due to resource constraints, we were unable to assess the performance of our approach on Traffic or Electricity datasets. Comparison figures between CoMRes and MRes, evaluated using the same label sizes and dataset, are provided in Appendix A.6 as supplementary material.

#### 4.3.2 Combining Multiple Unseen Data Generation Techniques

In this section, we evaluate the impact of combining multiple unseen data generation techniques. We examine two approaches: (1) Combining all three augmentation techniques—Time Warp, Interpolation, and Noise Injection—and (2) applying Multi Time Warp, using compression factors of 3, 5, and 7. These methods are compared against the best results achieved by CoMRes in 4.1, primarily using Time Warp, which we refer to as Single Time Warp.

The results in Table 4 indicate that the effectiveness of augmentation techniques depends on the dataset's characteristics. For datasets with high variability over time, such as ETTh1, combining

Table 3: Result in Limited-Resource scenarios. The best results among the compared methods are highlighted in **bold**, while the best result of the remaining methods is underlined. The label size is represented as a ratio of the original training data. For example, a value of 0.5 indicates that only the last half of the original data was used for training, meaning that the most recent portion of the training data was used.

| Method | | CoMRes, Time Warp | | | | MRes (SL) | | | |
|---|---|---|---|---|---|---|---|---|---|
| Label Size | | 0.2 | 0.5 | 0.8 | 1.0 | 0.2 | 0.5 | 0.8 | 1.0 |
| ETTh1 | 96 | 0.390 | **0.377** | 0.378 | 0.378 | 0.393 | 0.379 | 0.381 | 0.382 |
| | 192 | 0.444 | **0.435** | 0.437 | 0.436 | 0.448 | 0.436 | 0.438 | 0.439 |
| | 336 | 0.452 | 0.448 | **0.444** | 0.452 | 0.449 | 0.447 | 0.445 | 0.451 |
| | 720 | 0.504 | 0.489 | **0.469** | 0.474 | 0.474 | 0.481 | 0.470 | 0.481 |
| ETTh2 | 96 | 0.296 | 0.277 | 0.276 | **0.275** | 0.295 | 0.280 | 0.277 | 0.277 |
| | 192 | 0.377 | 0.356 | 0.348 | **0.345** | 0.377 | 0.363 | 0.349 | 0.347 |
| | 336 | 0.375 | 0.359 | 0.334 | **0.329** | 0.376 | 0.358 | 0.334 | 0.330 |
| | 720 | 0.452 | 0.438 | 0.404 | **0.399** | 0.460 | 0.434 | 0.416 | 0.409 |
| ETTm1 | 96 | 0.336 | **0.310** | 0.313 | **0.310** | 0.337 | 0.311 | 0.311 | 0.312 |
| | 192 | 0.376 | **0.357** | 0.361 | 0.362 | 0.377 | **0.357** | 0.361 | 0.361 |
| | 336 | 0.401 | **0.378** | 0.390 | 0.386 | 0.399 | **0.378** | 0.384 | 0.381 |
| | 720 | 0.471 | **0.446** | 0.467 | 0.456 | 0.471 | 0.447 | 0.454 | 0.454 |
| ETTm2 | 96 | 0.178 | 0.171 | 0.168 | **0.163** | 0.179 | 0.170 | 0.168 | 0.166 |
| | 192 | 0.248 | 0.234 | 0.231 | **0.229** | 0.248 | 0.234 | 0.232 | **0.229** |
| | 336 | 0.313 | 0.299 | 0.295 | **0.291** | 0.312 | 0.299 | 0.294 | 0.292 |
| | 720 | 0.409 | 0.400 | 0.386 | **0.383** | 0.408 | 0.401 | 0.386 | **0.383** |
| Weather | 96 | 0.164 | 0.160 | 0.155 | **0.151** | 0.175 | 0.163 | 0.156 | 0.152 |
| | 192 | 0.213 | 0.223 | 0.204 | **0.199** | 0.223 | 0.210 | 0.204 | **0.199** |
| | 336 | 0.265 | 0.257 | 0.250 | **0.244** | 0.258 | 0.257 | 0.250 | 0.245 |
| | 720 | 0.351 | 0.342 | 0.338 | 0.335 | 0.343 | 0.343 | 0.338 | **0.334** |

multiple augmentation techniques did not lead to noticeable performance gains. However, in datasets with clear and consistent periodic patterns over time, such as ETTm2, improvements were observed at certain prediction horizons, although rapid long-term trend shift caused performance declines at the 720-horizon. On the other hand, in the Weather dataset, which features strong inter-variable correlations, combining augmentation techniques resulted in performance gains. Overall, while multi-augmentation approaches do not universally improve performance, they can be beneficial for datasets with complex inter-variable relationships. The degree of improvement, however, varies according to the specific features of the dataset and the forecasting task at hand.

Table 4: Results of combining unseen data generation techniques. Performance improvements compared to the Single Time Warp method are in blue, while declines are in red.

| Method | | Combine all three | Multi Time Warp | Single Time Warp |
|---|---|---|---|---|
| ETTh1 | 96 | 0.378 | 0.379 | 0.378 |
| | 192 | 0.436 | 0.436 | 0.436 |
| | 336 | 0.452 | 0.452 | 0.452 |
| | 720 | 0.474 | 0.476 | 0.474 |
| ETTm2 | 96 | 0.164 | 0.163 | 0.163 |
| | 192 | 0.228 | 0.227 | 0.229 |
| | 336 | 0.289 | 0.288 | 0.291 |
| | 720 | 0.384 | 0.382 | 0.369 |
| Weather | 96 | 0.149 | 0.150 | 0.151 |
| | 192 | 0.198 | 0.199 | 0.199 |
| | 336 | 0.244 | 0.244 | 0.244 |
| | 720 | 0.330 | 0.334 | 0.334 |

## 5 Conclusion

This paper introduces a multi-view learning strategy for long-term time series forecasting, demonstrating superior prediction accuracy and robustness compared to traditional supervised learning approaches. This approach leverages augmented unseen data to ensure that models not only capture multi-resolution patterns but also provide consistent predictions when faced with new, unseen data. Our comprehensive experiments highlight the model's ability to generalize well across various datasets and handle challenging forecasting scenarios effectively, especially in autoregressive setups, where it mitigates the issue of error accumulation.

## REFERENCES


Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Ankur Debnath, Govind Waghmare, Hardik Wadhwa, Siddhartha Asthana, and Ankur Arora. Exploring generative data augmentation in multivariate time series forecasting: opportunities and challenges. *Solar-Energy*, 137:52–560, 2021.

Guillermo Iglesias, Edgar Talavera, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval. Data augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, 35(14):10123–10145, 2023.

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

KyungTae Lim, Jay Yoon Lee, Jaime Carbonell, and Thierry Poibeau. Semi-supervised learning on meta structure: Multi-task tagging and parsing in low-resource scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8344–8351, 2020.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. Data augmentation for univariate time series forecasting with neural networks. *Pattern Recognition*, 134: 109132, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.109132. URL `https://www.sciencedirect.com/science/article/pii/S0031320322006124`.

Mohammad Amin Shabani, Amir H. Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 4653–4660. International Joint Conferences on Artificial Intelligence Organization, 8 2021a. doi: 10.24963/ijcai.2021/631. URL https://doi.org/10.24963/ijcai.2021/631. Survey Track.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 4653–4660. International Joint Conferences on Artificial Intelligence Organization, 8 2021b. doi: 10.24963/ijcai.2021/631.

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

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

Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. *arXiv preprint arXiv:2310.07446*, 2023.

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

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

# A Appendix

## A.1 Experimental Details

### A.1.1 Datasets

We use eight popular datasets from Wu et al. (2021) for forecasting. The ETT dataset[1] includes data from two electricity transformers at two stations, capturing metrics such as load and oil temperature. This dataset is recorded at both 15-minute and 1-hour intervals, and is labeled as ETTh1, ETTh2, ETTm1, and ETTm2. The Electricity dataset[2] contains hourly electricity consumption data from 321 users. The ILI dataset[3] collects weekly reports of patients with influenza-like illness, spanning 2002 to 2021, from the Centers for Disease Control and Prevention of the United States. The Traffic dataset[4] records road occupancy rates from various sensors on San Francisco freeways. Finally, the Weather dataset[5] comprises 21 meteorological indicators collected every 10 minutes in Germany.

[1] https://github.com/zhouhaoyi/ETDataset
[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[3] https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
[4] https://pems.dot.ca.gov/
[5] https://www.bgc-jena.mpg.de/wetter/

Table 5: Statistics of popular datasets for benchmark.

| Datasets | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Electricity | ILI | Traffic | Weather |
|---|---|---|---|---|---|---|---|---|
| Features | 7 | 7 | 7 | 7 | 321 | 7 | 862 | 21 |
| Timesteps | 17,420 | 17,420 | 69,680 | 69,680 | 26,304 | 966 | 17,544 | 52,696 |
| Split Ratio | 6:2:2 | 6:2:2 | 6:2:2 | 6:2:2 | 7:1:2 | 7:1:2 | 7:1:2 | 7:1:2 |

### A.1.2 Details of Implementation

To ensure fair comparisons, we use the same hyperparameters as Pathformer in Chen et al. (2024). The model is trained using the Adam optimizer (Kingma, 2014) with a learning rate of $10^{-3}$. The default loss function is L1 Loss, and early stopping is applied after 10 epochs if no improvement is observed. All experiments are implemented in PyTorch and executed on a NVIDIA A6000 48GB GPU.

The baseline model, MRes, includes four different patch sizes, same to those used in Pathformer repository[6]. These patch sizes are selected from the following options: $\{2, 3, 6, 12, 16, 24, 32\}$. Unlike Pathformer, which is composed of three multi-scale blocks, MRes does not utilize a hierarchical architecture. This allows us to argue that MRes achieves comparable predictive performance with fewer model parameters, offering a more efficient design.

CoMRes and MRes atgenerates one comprehensive prediction and $m$ individual predictions, so for the final model performance evaluation, $m + 1$ predictions were used in an ensemble. First, the average value of the $m$ individual predictions was calculated, and then it was averaged with the comprehensive prediction to form the final prediction.

## A.2 Variance of Individual-view Model

The Figure 5 provides clear evidence that applying unsupervised consistency loss in CoMRes improves consistency between individual-view models compared to MRes (SL). CoMRes exhibits a consistently smoother and lower trend than MRes(SL), indicating reduced variability in predictions. This smoother trend suggests that individual models within CoMRes are more aligned and consistent with one another after applying the unsupervised consistency loss, effectively achieving the intended effect of consensus promotion.

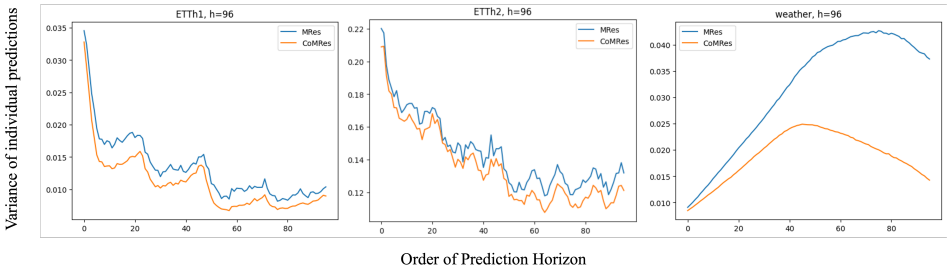



Figure 5: Comparison of individual-view model variability between MRes and CoMRes across three datasets (ETTh1, ETTh2, and Weather) with a forecast horizon of h=96. CoMRes demonstrates consistently reduced variability compared to MRes.

## A.3 Time Series Forecasting

In section 4.1, we reported the model performance using Mean Squared Error (MSE). Additionally, Table 6 presents the results based on Mean Absolute Error (MAE), providing a complementary evaluation of the model's predictive accuracy.

[6]https://github.com/decisionintelligence/pathformer

Table 6: Multivariate time series forecasting results (MAE). The input length L = 96 (L = 36 for ILI dataset). The best results are highlighted in bold, and the second-best results are underlined. The Δ column represents the difference between CoMRes best result and the baselines such as MRes with ablated components and Pathformer.

| Method | | CoMRes (ours) | | | MRes | | | | | | Pathformer | Δ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Time Warping | Interpolation | Noise Injection | w. augmentation | Δ | w. consensus | Δ | MRes (SL) | Δ | | |
| Augmentation | | ○ | ○ | ○ | ○ | | × | | × | | | |
| Consensus Promotion | | ○ | ○ | ○ | × | | ○ | | × | | | |
| Leveraging Unlabeled Data | | ○ | ○ | ○ | × | | × | | × | | | |
| ETTh1 | 96 | **0.386** | **0.386** | **0.386** | 0.415 | 0.029 | **0.386** | 0.000 | 0.387 | 0.001 | 0.400 | 0.014 |
| | 192 | **0.415** | **0.415** | **0.415** | 0.439 | 0.024 | **0.415** | 0.000 | 0.416 | 0.001 | 0.427 | 0.012 |
| | 336 | **0.421** | **0.421** | **0.421** | 0.458 | 0.037 | **0.421** | 0.000 | **0.421** | 0.000 | 0.432 | 0.011 |
| | 720 | 0.453 | 0.453 | **0.451** | 0.476 | 0.025 | 0.453 | 0.002 | 0.454 | 0.003 | 0.461 | 0.010 |
| | Avg. | 0.419 | 0.419 | **0.418** | 0.447 | 0.029 | 0.419 | 0.001 | 0.420 | 0.002 | 0.430 | 0.012 |
| ETTh2 | 96 | **0.327** | **0.327** | **0.327** | 0.332 | 0.005 | **0.327** | 0.000 | 0.329 | 0.002 | 0.331 | 0.004 |
| | 192 | **0.373** | 0.374 | **0.373** | 0.382 | 0.009 | 0.374 | 0.001 | 0.375 | 0.002 | 0.380 | 0.007 |
| | 336 | **0.369** | 0.370 | **0.369** | 0.385 | 0.016 | 0.371 | 0.002 | 0.373 | 0.004 | 0.382 | 0.013 |
| | 720 | 0.422 | 0.422 | **0.419** | 0.430 | 0.011 | 0.426 | 0.007 | 0.427 | 0.008 | 0.424 | 0.005 |
| | Avg. | 0.373 | 0.373 | **0.372** | 0.382 | 0.010 | 0.375 | 0.003 | 0.376 | 0.004 | 0.379 | 0.007 |
| ETTm1 | 96 | **0.339** | **0.339** | **0.339** | 0.360 | 0.021 | **0.339** | 0.000 | 0.341 | 0.002 | 0.346 | 0.007 |
| | 192 | **0.365** | **0.365** | **0.365** | 0.383 | 0.018 | **0.365** | 0.000 | 0.366 | 0.001 | 0.370 | 0.005 |
| | 336 | **0.387** | **0.387** | **0.387** | 0.404 | 0.017 | **0.387** | 0.000 | **0.387** | 0.000 | 0.394 | 0.007 |
| | 720 | **0.424** | 0.425 | 0.425 | 0.439 | 0.015 | 0.425 | 0.001 | 0.426 | 0.002 | 0.432 | 0.008 |
| | Avg. | **0.379** | **0.379** | **0.379** | 0.397 | 0.018 | **0.379** | 0.000 | 0.380 | 0.001 | 0.386 | 0.007 |
| ETTm2 | 96 | **0.245** | 0.246 | 0.246 | 0.250 | 0.005 | 0.246 | 0.001 | 0.247 | 0.002 | 0.248 | 0.002 |
| | 192 | 0.290 | 0.291 | 0.290 | 0.292 | 0.002 | **0.289** | -0.001 | 0.290 | 0.000 | 0.295 | 0.005 |
| | 336 | **0.330** | **0.330** | **0.330** | 0.334 | 0.004 | **0.330** | 0.000 | **0.330** | 0.000 | 0.331 | 0.001 |
| | 720 | **0.378** | **0.378** | **0.378** | 0.389 | 0.011 | 0.385 | 0.007 | **0.378** | 0.000 | 0.389 | 0.011 |
| | Avg. | **0.311** | **0.311** | **0.311** | 0.316 | 0.006 | 0.313 | 0.002 | **0.311** | 0.000 | 0.316 | 0.005 |
| Electricity | 96 | 0.240 | 0.240 | 0.239 | 0.247 | 0.008 | 0.243 | 0.004 | 0.237 | -0.002 | **0.236** | -0.003 |
| | 192 | 0.252 | 0.252 | 0.252 | 0.258 | 0.006 | 0.254 | 0.002 | **0.250** | -0.002 | 0.256 | 0.004 |
| | 336 | **0.267** | **0.267** | **0.267** | 0.272 | 0.005 | 0.269 | 0.002 | 0.269 | 0.002 | 0.275 | 0.008 |
| | 720 | 0.293 | 0.293 | **0.292** | 0.299 | 0.007 | 0.294 | 0.002 | 0.302 | 0.010 | 0.309 | 0.017 |
| | Avg. | **0.263** | **0.263** | **0.263** | 0.269 | 0.006 | 0.265 | 0.002 | 0.265 | 0.002 | 0.269 | 0.007 |
| ILI | 24 | 0.830 | 0.823 | 0.834 | 0.799 | -0.024 | 0.837 | 0.014 | 0.840 | 0.010 | **0.758** | -0.065 |
| | 36 | 0.755 | 0.764 | 0.778 | 0.741 | -0.014 | 0.749 | -0.006 | 0.758 | 0.003 | **0.711** | -0.044 |
| | 48 | 0.766 | 0.769 | 0.767 | 0.749 | -0.017 | 0.744 | -0.022 | 0.761 | -0.005 | **0.742** | -0.024 |
| | 60 | 0.811 | 0.821 | 0.808 | 0.811 | 0.003 | 0.808 | 0.000 | 0.810 | 0.002 | **0.799** | -0.009 |
| | Avg. | 0.791 | 0.794 | 0.797 | 0.775 | -0.016 | 0.785 | -0.006 | 0.792 | 0.001 | **0.753** | -0.038 |
| Traffic | 96 | **0.270** | 0.271 | 0.271 | 0.275 | 0.005 | 0.275 | 0.005 | 0.273 | 0.003 | 0.283 | 0.013 |
| | 192 | **0.285** | 0.286 | 0.286 | 0.292 | 0.007 | 0.282 | -0.003 | 0.284 | -0.001 | 0.292 | 0.007 |
| | 336 | **0.295** | **0.295** | **0.295** | 0.300 | 0.005 | **0.295** | 0.000 | 0.296 | 0.001 | 0.299 | 0.004 |
| | 720 | 0.324 | 0.324 | 0.324 | 0.333 | 0.009 | 0.323 | -0.001 | **0.321** | -0.003 | 0.322 | -0.002 |
| | Avg. | **0.294** | **0.294** | **0.294** | 0.300 | 0.007 | 0.294 | 0.000 | **0.294** | 0.000 | 0.299 | 0.006 |
| Weather | 96 | **0.190** | **0.190** | **0.190** | 0.195 | 0.005 | 0.193 | 0.003 | **0.190** | 0.000 | 0.192 | 0.002 |
| | 192 | 0.237 | 0.237 | **0.236** | 0.239 | 0.002 | 0.237 | 0.000 | **0.236** | 0.000 | 0.240 | 0.004 |
| | 336 | 0.276 | 0.276 | **0.275** | 0.279 | 0.003 | 0.276 | 0.000 | 0.276 | 0.001 | 0.282 | 0.007 |
| | 720 | 0.331 | 0.332 | 0.332 | 0.333 | 0.002 | **0.330** | -0.001 | 0.331 | 0.000 | 0.336 | 0.005 |
| | Avg. | 0.259 | 0.259 | **0.258** | 0.262 | 0.003 | 0.259 | 0.001 | **0.258** | 0.000 | 0.263 | 0.005 |

### A.4 Error Bars

In this paper, we repeat all the experiments five times. In Table 7, we report the standard deviation of our model and the baseline model.

### A.5 Additional Baselines

To provide additional evaluation results and contextualize current progress in long-term time series forecasting, we include several advanced baselines: Pathformer (Chen et al., 2024), PatchTST (Nie et al., 2023), TimeMixer (Wang et al., 2024), NLinear (Zeng et al., 2023), Scaleformer (Shabani et al., 2023), and TiDE (Das et al., 2023). All models use the same input length ($H = 36$ for the ILI dataset and $H = 96$ for others) and prediction lengths ($F \in \{24, 36, 48, 60\}$ for the ILI dataset and $F \in \{96, 192, 336, 720\}$ for others). We evaluate model performance using two common metrics in time series forecasting: Mean Absolute Error (MAE) and Mean Squared Error (MSE). The results for TimeMixer are taken from Wang et al. (2024), those for PatchTST are sourced from Nie et al. (2023), and the remaining results are obtained from Chen et al. (2024).

### A.6 Limited-Resource Scenarios

As noted in Section 4.3.1, CoMRes consistently outperforms MRes across all label sizes and prediction horizons in limited resource scenarios. Not only does CoMRes achieve the best scores, as presented in Table 3, but it also demonstrates higher prediction accuracy when trained on the same amount of labeled data. This suggests that CoMRes possesses generalization ability.

The table 9 shows the results of training with distant data from the test period. Compared to Table 3, the overall best scores have changed due to intricate temporal variations. This suggests that the

Table 7: We provide standard deviation for our method and the MRes baselines over 5 independent runs.

| Method | | CoMRes (ours) | | | MRes | | |
|---|---|---|---|---|---|---|---|
| | | Time Warping | Interpolation | Noise Injection | w. augmentation | w. consensus | MRes (SL) |
| Augmentation | | ○ | ○ | ○ | ○ | × | × |
| Consensus Promotion | | ○ | ○ | ○ | × | ○ | × |
| ETTh1 | 96 | 0.378±0.001 | 0.379±0.001 | 0.379±0.001 | 0.406±0.006 | 0.379±0.001 | 0.382±0.001 |
| | 192 | 0.436±0.002 | 0.436±0.001 | 0.436±0.001 | 0.455±0.006 | 0.436±0.001 | 0.439±0.001 |
| | 336 | 0.452±0.001 | 0.452±0.001 | 0.452±0.001 | 0.482±0.005 | 0.452±0.001 | 0.451±0.001 |
| | 720 | 0.474±0.003 | 0.478±0.003 | 0.478±0.003 | 0.475±0.004 | 0.478±0.003 | 0.481±0.002 |
| ETTh2 | 96 | 0.275±0.002 | 0.275±0.002 | 0.275±0.002 | 0.278±0.005 | 0.275±0.004 | 0.277±0.003 |
| | 192 | 0.347±0.002 | 0.346±0.002 | 0.346±0.001 | 0.355±0.009 | 0.346±0.003 | 0.348±0.001 |
| | 336 | 0.326±0.001 | 0.328±0.005 | 0.326±0.001 | 0.346±0.006 | 0.330±0.004 | 0.332±0.008 |
| | 720 | 0.393±0.002 | 0.398±0.009 | 0.397±0.009 | 0.404±0.015 | 0.405±0.006 | 0.406±0.009 |
| ETTm1 | 96 | 0.310±0.001 | 0.311±0.001 | 0.311±0.001 | 0.326±0.002 | 0.311±0.001 | 0.312±0.001 |
| | 192 | 0.362±0.001 | 0.364±0.001 | 0.364±0.001 | 0.367±0.002 | 0.363±0.002 | 0.361±0.002 |
| | 336 | 0.386±0.001 | 0.386±0.001 | 0.387±0.001 | 0.404±0.004 | 0.386±0.001 | 0.381±0.001 |
| | 720 | 0.456±0.001 | 0.456±0.002 | 0.456±0.001 | 0.467±0.002 | 0.456±0.002 | 0.454±0.001 |
| ETTm2 | 96 | 0.163±0.000 | 0.163±0.000 | 0.163±0.000 | 0.168±0.001 | 0.166±0.001 | 0.167±0.001 |
| | 192 | 0.229±0.001 | 0.229±0.001 | 0.229±0.001 | 0.230±0.000 | 0.229±0.001 | 0.231±0.001 |
| | 336 | 0.291±0.001 | 0.292±0.001 | 0.292±0.001 | 0.295±0.001 | 0.292±0.002 | 0.292±0.001 |
| | 720 | 0.369±0.003 | 0.369±0.003 | 0.369±0.003 | 0.385±0.002 | 0.383±0.002 | 0.369±0.003 |
| Electricity | 96 | 0.152±0.004 | 0.153±0.003 | 0.153±0.003 | 0.162±0.003 | 0.156±0.003 | 0.150±0.003 |
| | 192 | 0.167±0.001 | 0.167±0.001 | 0.166±0.002 | 0.173±0.003 | 0.169±0.003 | 0.164±0.003 |
| | 336 | 0.181±0.009 | 0.181±0.007 | 0.181±0.007 | 0.186±0.006 | 0.184±0.006 | 0.181±0.006 |
| | 720 | 0.210±0.002 | 0.211±0.002 | 0.214±0.003 | 0.218±0.005 | 0.214±0.005 | 0.223±0.005 |
| ILI | 96 | 1.957±0.072 | 1.907±0.062 | 1.956±0.080 | 1.760±0.067 | 1.986±0.112 | 1.976±0.051 |
| | 192 | 1.640±0.036 | 1.599±0.076 | 1.538±0.124 | 1.569±0.079 | 1.540±0.055 | 1.551±0.137 |
| | 336 | 1.529±0.097 | 1.516±0.075 | 1.490±0.101 | 1.514±0.126 | 1.456±0.048 | 1.497±0.040 |
| | 720 | 1.693±0.091 | 1.739±0.064 | 1.715±0.049 | 1.727±0.047 | 1.679±0.092 | 1.714±0.048 |
| Traffic | 96 | 0.468±0.004 | 0.470±0.003 | 0.470±0.003 | 0.475±0.003 | 0.479±0.003 | 0.478±0.003 |
| | 192 | 0.465±0.001 | 0.466±0.001 | 0.466±0.002 | 0.482±0.003 | 0.479±0.003 | 0.476±0.003 |
| | 336 | 0.493±0.009 | 0.494±0.007 | 0.494±0.007 | 0.511±0.006 | 0.501±0.006 | 0.501±0.006 |
| | 720 | 0.535±0.002 | 0.535±0.002 | 0.535±0.003 | 0.540±0.005 | 0.537±0.005 | 0.556±0.005 |
| Weather | 96 | 0.151±0.000 | 0.151±0.000 | 0.151±0.000 | 0.155±0.001 | 0.154±0.000 | 0.152±0.001 |
| | 192 | 0.199±0.001 | 0.200±0.003 | 0.200±0.003 | 0.203±0.001 | 0.201±0.001 | 0.199±0.001 |
| | 192 | 0.244±0.001 | 0.246±0.002 | 0.246±0.002 | 0.248±0.002 | 0.246±0.000 | 0.245±0.001 |
| | 192 | 0.335±0.004 | 0.335±0.004 | 0.334±0.004 | 0.334±0.003 | 0.333±0.002 | 0.334±0.004 |

relevance of the data is more important than the amount of data used. As the same as result of the recent data, CoMRes consistently achieves the best performance across all label sizes and prediction horizons compared to the baseline.

### A.7 Discussion - Result on ILI dataset

The ILI dataset presents unique challenges due to its highly non-stationary patterns, which make accurate prediction particularly difficult. While CoMRes demonstrates strong performance across most datasets, its performance on the ILI dataset is less competitive compared to Pathformer, especially when $h = 24$ and $h = 36$, as shown in Table 1. Notably, we were unable to reproduce Pathformer's reported performance on the ILI dataset; however, crediting the authors, Table 1 includes the performance metrics as reported in the original Pathformer paper.

The ILI dataset is characterized by a rapid long-term uptrend, which poses significant challenges for forecasting models. Recent advanced models, such as those proposed by Shabani et al. (2023) and Nie et al. (2023), exhibit high prediction errors (MSE above 2), while others, such as Wang et al. (2024) and Das et al. (2023), exclude the ILI dataset from their experiments, further reflecting its inherent difficulty.

We hypothesize that the rapid long-term uptrend in the ILI dataset limits the advantages of multi-scale modeling, as long-term dependency patterns may not effectively leverage multi-resolution features. Nonetheless, our experimental results show that CoMRes improves prediction accuracy compared to MRes (SL). This suggests that the unsupervised consistency promotion mechanism in CoMRes has a positive impact on enhancing model generalization, even in challenging scenarios like the ILI dataset.

Table 8: Multivariate time series forecasting results for additional baselines. The input length $L = 96$ ($L = 36$ for the ILI dataset). The best results are highlighted in **bold**. CoMRes's performance represents the average values over 5 runs. The results for TimeMixer are taken from Wang et al. (2024), the results for PatchTST are sourced from Nie et al. (2023), and the remaining results are obtained from Chen et al. (2024).

| Method | | CoMRes | | Pathformer | | PatchTST | | TimeMixer | | Nlinear | | Scaleformer | | TiDE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 96 | 0.378 | **0.386** | 0.382 | 0.400 | 0.394 | 0.408 | **0.375** | 0.400 | 0.386 | 0.392 | 0.396 | 0.440 | 0.427 | 0.450 |
| | 192 | 0.436 | **0.415** | 0.440 | 0.427 | 0.446 | 0.438 | **0.429** | 0.421 | 0.440 | 0.430 | 0.434 | 0.460 | 0.472 | 0.486 |
| | 336 | **0.452** | **0.421** | 0.454 | 0.432 | 0.485 | 0.550 | 0.484 | 0.458 | 0.480 | 0.443 | 0.462 | 0.476 | 0.527 | 0.527 |
| | 720 | **0.474** | **0.453** | 0.479 | 0.461 | 0.495 | 0.474 | 0.498 | 0.482 | 0.486 | 0.472 | 0.494 | 0.500 | 0.644 | 0.605 |
| | Avg. | **0.435** | **0.419** | 0.439 | 0.430 | 0.455 | 0.468 | 0.447 | 0.440 | 0.448 | 0.434 | 0.447 | 0.469 | 0.518 | 0.517 |
| ETTh2 | 96 | **0.275** | **0.327** | 0.279 | 0.331 | 0.294 | 0.343 | 0.289 | 0.341 | 0.290 | 0.339 | 0.364 | 0.407 | 0.304 | 0.359 |
| | 192 | **0.346** | **0.373** | 0.349 | 0.380 | 0.378 | 0.394 | 0.372 | 0.392 | 0.379 | 0.395 | 0.466 | 0.458 | 0.394 | 0.422 |
| | 336 | **0.326** | **0.369** | 0.348 | 0.382 | 0.382 | 0.410 | 0.386 | 0.414 | 0.421 | 0.431 | 0.479 | 0.476 | 0.385 | 0.421 |
| | 720 | **0.393** | **0.419** | 0.398 | 0.424 | 0.412 | 0.433 | 0.412 | 0.434 | 0.436 | 0.453 | 0.487 | 0.492 | 0.463 | 0.475 |
| | Avg. | **0.335** | **0.372** | 0.344 | 0.379 | 0.367 | 0.395 | 0.365 | 0.395 | 0.382 | 0.405 | 0.449 | 0.458 | 0.387 | 0.419 |
| ETTm1 | 96 | **0.310** | **0.339** | 0.316 | 0.346 | 0.324 | 0.361 | 0.320 | 0.357 | 0.339 | 0.369 | 0.355 | 0.398 | 0.356 | 0.381 |
| | 192 | 0.362 | **0.365** | 0.366 | 0.370 | 0.362 | 0.383 | **0.361** | 0.381 | 0.379 | 0.386 | 0.428 | 0.455 | 0.391 | 0.399 |
| | 336 | **0.386** | **0.387** | **0.386** | 0.394 | 0.39 | 0.402 | 0.390 | 0.404 | 0.411 | 0.407 | 0.524 | 0.487 | 0.424 | 0.423 |
| | 720 | 0.456 | **0.424** | 0.460 | 0.432 | 0.461 | 0.438 | 0.454 | 0.441 | 0.478 | 0.442 | 0.558 | 0.517 | 0.48 | 0.456 |
| | Avg. | **0.379** | **0.379** | 0.382 | 0.386 | 0.384 | 0.396 | 0.381 | 0.396 | 0.402 | 0.401 | 0.466 | 0.464 | 0.413 | 0.415 |
| ETTm2 | 96 | **0.163** | **0.245** | 0.170 | 0.248 | 0.177 | 0.260 | 0.175 | 0.258 | 0.177 | 0.257 | 0.182 | 0.275 | 0.182 | 0.264 |
| | 192 | **0.229** | **0.290** | 0.238 | 0.295 | 0.248 | 0.306 | 0.237 | 0.299 | 0.241 | 0.297 | 0.251 | 0.318 | 0.256 | 0.323 |
| | 336 | **0.291** | **0.330** | 0.293 | 0.331 | 0.304 | 0.342 | 0.298 | 0.340 | 0.302 | 0.337 | 0.34 | 0.375 | 0.313 | 0.354 |
| | 720 | **0.369** | **0.378** | 0.390 | 0.389 | 0.403 | 0.397 | 0.391 | 0.396 | 0.405 | 0.396 | 0.435 | 0.433 | 0.419 | 0.410 |
| | Avg. | **0.263** | **0.311** | 0.273 | 0.316 | 0.283 | 0.326 | 0.275 | 0.323 | 0.281 | 0.322 | 0.302 | 0.350 | 0.293 | 0.338 |
| Electricity | 96 | 0.152 | 0.239 | **0.145** | **0.236** | 0.18 | 0.264 | 0.153 | 0.247 | 0.185 | 0.266 | 0.182 | 0.297 | 0.194 | 0.277 |
| | 192 | **0.166** | **0.252** | 0.167 | 0.256 | 0.188 | 0.275 | **0.166** | 0.256 | 0.189 | 0.276 | 0.188 | 0.300 | 0.193 | 0.280 |
| | 336 | **0.181** | **0.267** | 0.186 | 0.275 | 0.206 | 0.291 | 0.185 | 0.277 | 0.204 | 0.289 | 0.210 | 0.324 | 0.206 | 0.296 |
| | 720 | **0.210** | **0.292** | 0.231 | 0.309 | 0.247 | 0.328 | 0.225 | 0.310 | 0.245 | 0.319 | 0.232 | 0.339 | 0.242 | 0.328 |
| | Avg. | **0.177** | **0.263** | 0.182 | 0.269 | 0.205 | 0.290 | 0.182 | 0.273 | 0.206 | 0.288 | 0.203 | 0.315 | 0.209 | 0.295 |
| ILI | 96 | 1.907 | 0.830 | **1.587** | **0.758** | 1.724 | 0.843 | - | - | 2.725 | 1.069 | 0.232 | 0.339 | 2.154 | 0.992 |
| | 192 | 1.538 | 0.755 | **1.429** | **0.711** | 1.536 | 0.752 | - | - | 2.530 | 1.032 | 2.745 | 1.075 | 2.436 | 1.042 |
| | 336 | **1.490** | 0.766 | 1.505 | **0.742** | 1.821 | 0.832 | - | - | 2.510 | 1.031 | 2.748 | 1.072 | 2.532 | 1.051 |
| | 720 | **1.693** | 0.811 | 1.731 | **0.799** | 1.923 | 0.842 | - | - | 2.492 | 1.026 | 2.793 | 1.059 | 2.748 | 1.142 |
| | Avg. | 1.657 | 0.791 | **1.563** | **0.753** | 1.751 | 0.817 | - | - | 2.564 | 1.040 | 2.130 | 0.886 | 2.468 | 1.057 |
| Traffic | 96 | 0.468 | **0.270** | 0.479 | 0.283 | 0.492 | 0.324 | **0.462** | 0.285 | 0.645 | 0.388 | 2.678 | 1.071 | 0.568 | 0.352 |
| | 192 | **0.465** | **0.285** | 0.484 | 0.292 | 0.487 | 0.303 | 0.473 | 0.296 | 0.599 | 0.365 | 0.564 | 0.351 | 0.612 | 0.371 |
| | 336 | **0.493** | **0.295** | 0.503 | 0.299 | 0.505 | 0.317 | 0.498 | 0.296 | 0.606 | 0.367 | 0.57 | 0.349 | 0.605 | 0.374 |
| | 720 | 0.535 | 0.324 | 0.537 | 0.322 | 0.542 | 0.337 | **0.506** | **0.313** | 0.645 | 0.388 | 0.576 | 0.349 | 0.647 | 0.410 |
| | Avg. | 0.490 | **0.294** | 0.501 | 0.299 | 0.507 | 0.320 | **0.485** | 0.298 | 0.624 | 0.377 | 1.097 | 0.530 | 0.608 | 0.377 |
| Weather | 96 | **0.151** | **0.19** | 0.156 | 0.192 | 0.177 | 0.218 | 0.163 | 0.209 | 0.168 | 0.208 | 0.288 | 0.365 | 0.202 | 0.261 |
| | 192 | **0.199** | **0.236** | 0.206 | 0.24 | 0.224 | 0.258 | 0.208 | 0.250 | 0.217 | 0.255 | 0.368 | 0.425 | 0.242 | 0.298 |
| | 336 | **0.244** | **0.275** | 0.254 | 0.282 | 0.277 | 0.297 | 0.251 | 0.287 | 0.267 | 0.292 | 0.447 | 0.469 | 0.287 | 0.335 |
| | 720 | **0.334** | **0.331** | 0.340 | 0.336 | 0.35 | 0.345 | 0.339 | 0.341 | 0.351 | 0.346 | 0.640 | 0.574 | 0.351 | 0.386 |
| | Avg. | **0.232** | **0.258** | 0.239 | 0.263 | 0.257 | 0.280 | 0.240 | 0.272 | 0.251 | 0.275 | 0.436 | 0.458 | 0.271 | 0.320 |

### A.8 VISUALIZATION

We visualize the prediction results of CoMRes and Pathformer on ETTh1, Weather, Traffic dataset. As shown in Figures 7, 8 and 9, the prediction curve closely aligns with the ground truth, demonstrating the excellent predictive performance of CoMRes. Additionally, we present the prediction results of CoMRes, MRes, and Pathformer in an autoregressive forecasting scenario. Figure 10 illustrates that CoMRes effectively mitigates error accumulation.

### A.9 LIMITATIONS AND FUTURE WORK

CoMRes has demonstrated favorable performance in long-term time series forecasting; however, there is room for further exploration and refinement of its design options. First, our framework assumes a Euclidean distance to measure consensus between the aggregated prediction and each view. While this approach is straightforward and widely used, it may not be optimal for all scenarios. Alternative metrics, such as soft-DTW (Dynamic Time Warping, Cuturi & Blondel (2017)), which better capture temporal alignment and dependencies, or differentiable loss functions, could be investigated to improve performance. Additionally, this study focuses on three basic augmentation strategies—time warping, noise injection, and interpolation. Although these techniques are effective and simple to implement, they may not fully leverage the potential of more complex augmentation methods. As demonstrated in Section 4.3.2, the effectiveness of augmentation techniques varies depending on

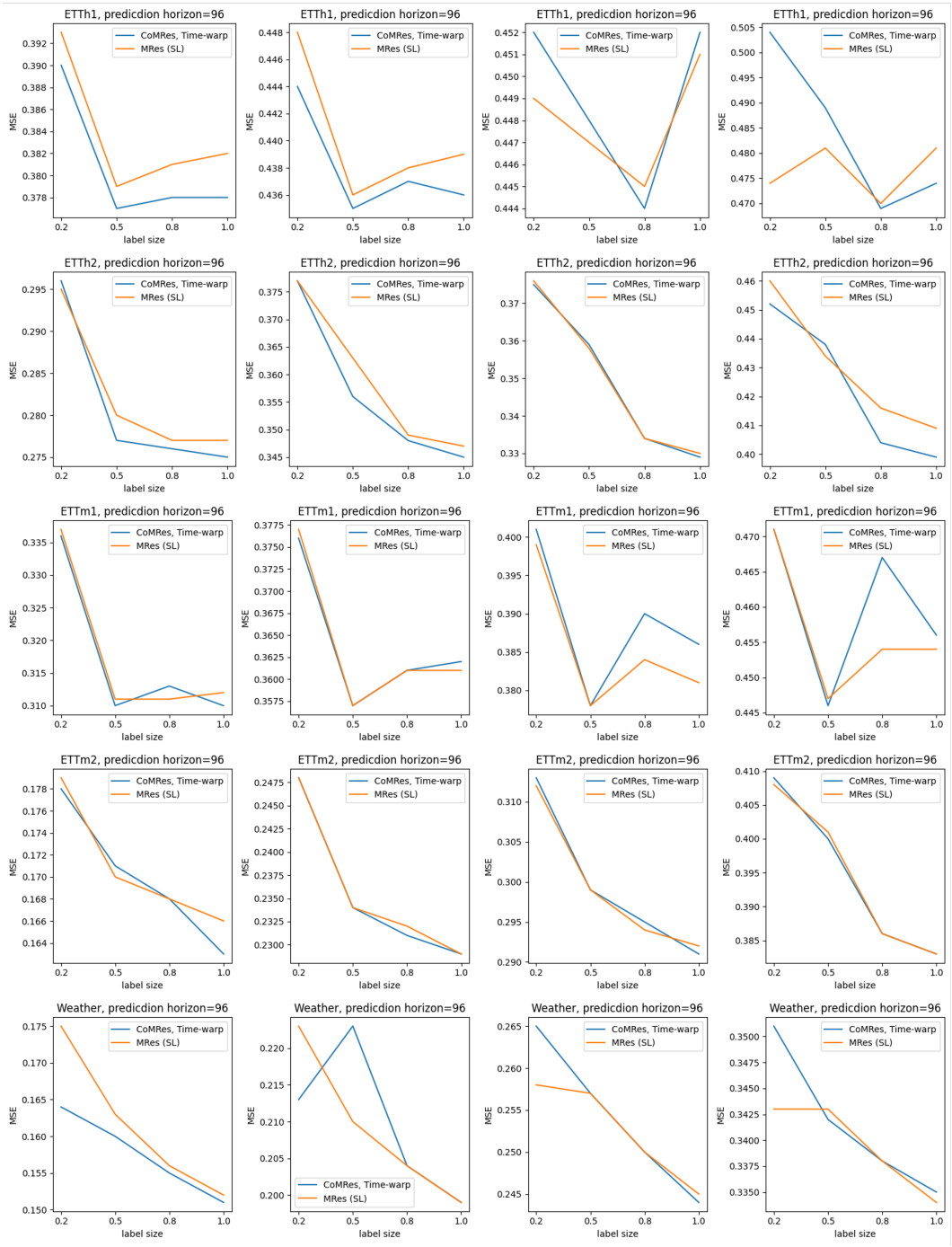



Figure 6: Comparison figures between CoMRes and MRes, evaluated using the same label sizes and dataset

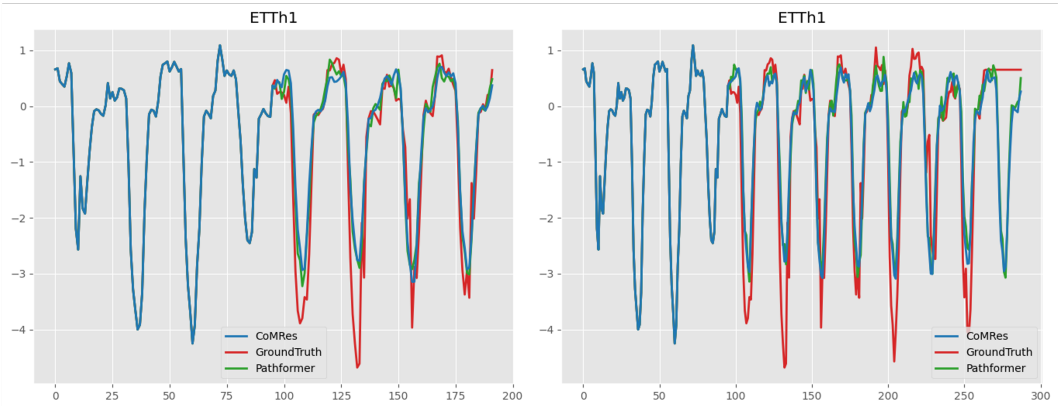
the dataset's characteristics. Future work could explore the performance of our framework with advanced augmentation strategies, such as learned augmentations or domain-specific transformations, to provide deeper insights and enhance the applicability and robustness of the proposed approach. Lastly, we employ multi-scale patch division with various patch sizes, which is similar to the model architecture used in Pathformer (Chen et al., 2024). In contrast, other multi-scale models, such as TimeMixer(Wang et al., 2024) and Scaleformer(Shabani et al., 2023), generate multi-scale time

Table 9: Results in Limited-Resource Scenarios: When using the oldest data. The best results among the compared methods are highlighted in bold, while the best result of the remaining methods is underlined. The label size is represented as a ratio of the original training data. For example, a value of 0.5 indicates that only the first half of the original data was used for training, meaning that the most oldest portion of the training data was used.

| Method | | SSL, Time Warp (ours) | | | | SL, Baseline | | | |
|---|---|---|---|---|---|---|---|---|---|
| Label Size | | 0.2 | 0.5 | 0.8 | 1.0 | 0.2 | 0.5 | 0.8 | 1.0 |
| ETTh1 | 96 | 0.412 | 0.398 | **0.378** | **0.378** | 0.409 | 0.396 | 0.381 | 0.382 |
| | 192 | 0.486 | 0.455 | 0.437 | **0.436** | 0.482 | 0.456 | 0.438 | 0.439 |
| | 336 | 0.533 | 0.478 | 0.454 | 0.452 | 0.523 | 0.479 | 0.452 | **0.451** |
| | 720 | 0.579 | 0.502 | 0.481 | **0.474** | 0.544 | 0.506 | 0.484 | 0.481 |
| ETTh2 | 96 | 0.286 | 0.278 | **0.270** | 0.275 | 0.286 | 0.279 | 0.271 | 0.277 |
| | 192 | 0.357 | 0.353 | **0.343** | 0.345 | 0.361 | 0.357 | 0.347 | 0.347 |
| | 336 | 0.343 | 0.334 | 0.333 | **0.329** | 0.346 | 0.338 | 0.333 | 0.330 |
| | 720 | 0.453 | **0.385** | 0.389 | 0.399 | 0.434 | 0.388 | 0.395 | 0.409 |
| ETTm1 | 96 | 0.383 | 0.387 | 0.312 | **0.310** | 0.381 | 0.382 | 0.318 | 0.312 |
| | 192 | 0.412 | 0.416 | 0.368 | 0.362 | 0.412 | 0.416 | 0.371 | **0.361** |
| | 336 | 0.436 | 0.438 | 0.392 | 0.386 | 0.436 | 0.436 | 0.397 | **0.381** |
| | 720 | 0.513 | 0.511 | 0.458 | 0.456 | 0.514 | 0.508 | 0.466 | **0.454** |
| ETTm2 | 96 | 0.180 | 0.172 | 0.166 | **0.163** | 0.180 | 0.170 | 0.168 | 0.166 |
| | 192 | 0.242 | 0.233 | 0.231 | **0.229** | 0.243 | 0.234 | 0.232 | **0.229** |
| | 336 | 0.306 | 0.297 | 0.292 | **0.291** | 0.305 | 0.294 | 0.292 | 0.292 |
| | 720 | 0.409 | 0.390 | 0.386 | **0.383** | 0.406 | 0.389 | 0.384 | **0.383** |

series through pooling. Incorporating consensus mechanisms between these pooled series could be a promising direction for extending our framework.

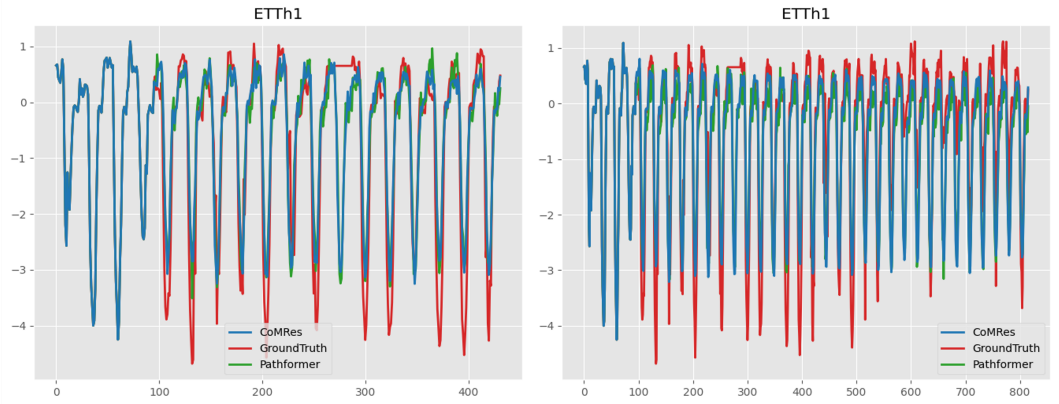
For future research, we aim to explore further improvements in non-Markovian frameworks, where the model would benefit from considering the entire history rather than relying on limited past observations. Another promising direction is investigating how to ensure consistency in multi-scaling predictions across different time horizons, enhancing the overall reliability of forecasts. These studies could further strengthen the capability of our multi-view learning strategy as a tool for long-term time series forecasting across diverse applications.




(a) Prediciton Horizon H = 96




(b) Prediciton Horizon H = 192

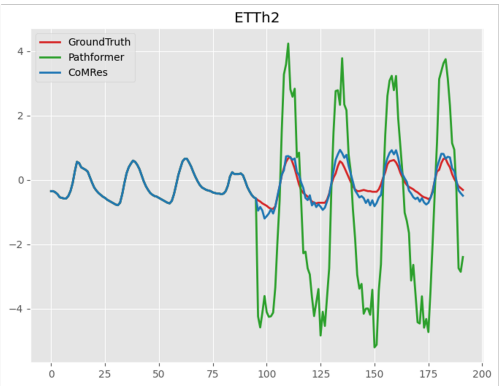



(c) Prediciton Horizon H = 336

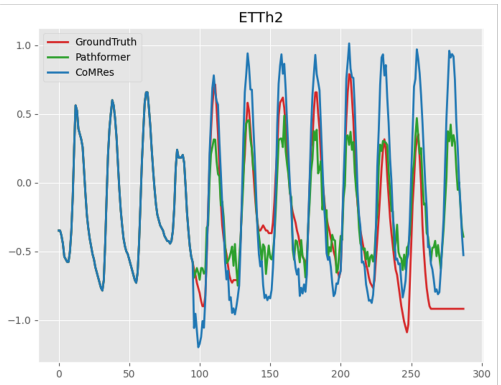



(d) Prediciton Horizon H = 720

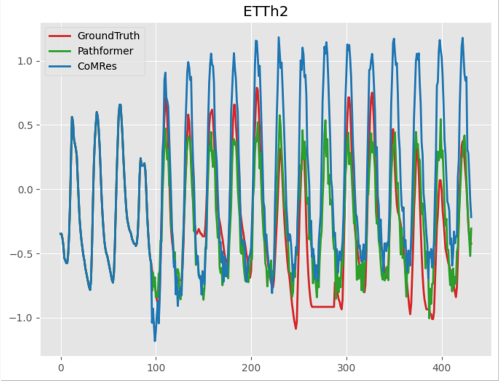
Figure 7: Visualization of CoMRes's and Pathformer's prediction results on ETTh1. The input length H = 96



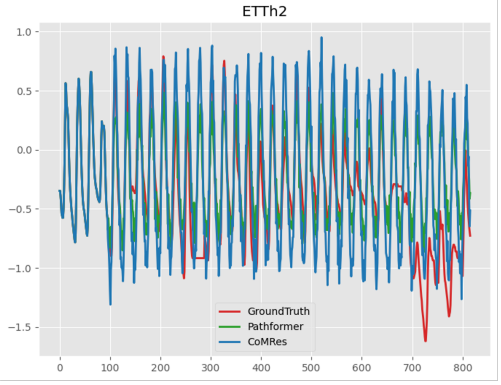

(a) Prediciton Horizon H = 96




(b) Prediciton Horizon H = 192

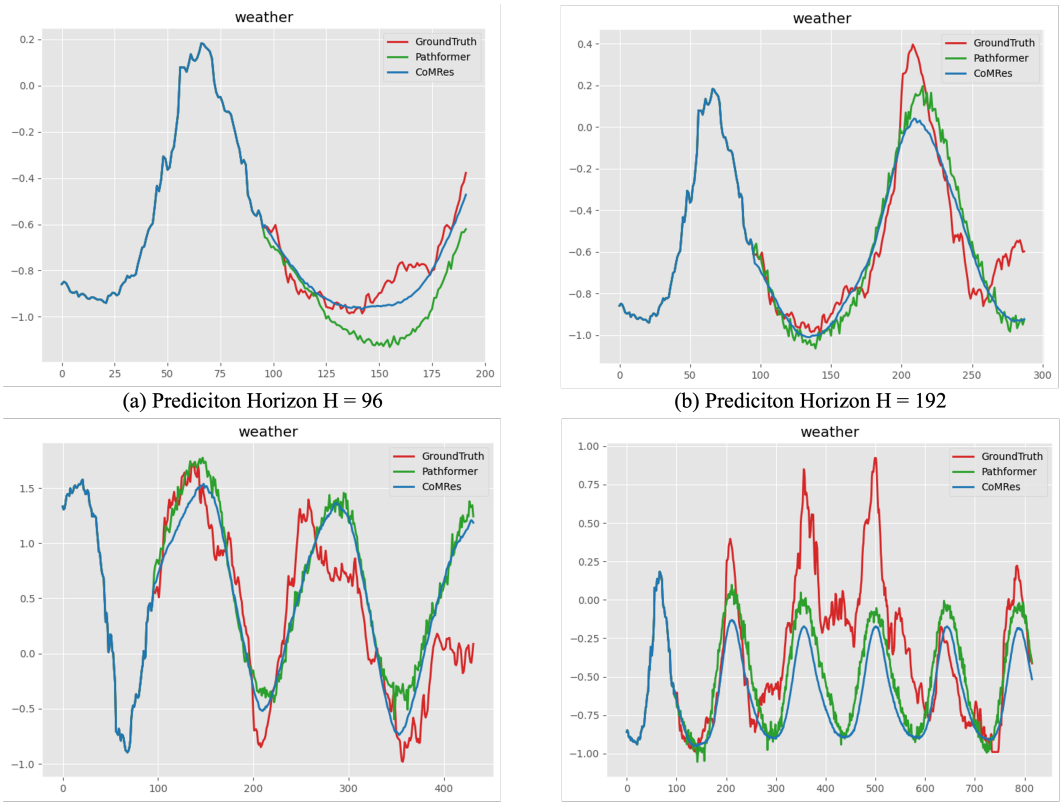



(c) Prediciton Horizon H = 336




(d) Prediciton Horizon H = 720

Figure 8: Visualization of CoMRes's and Pathformer's prediction results on ETTh2. The input length H = 96



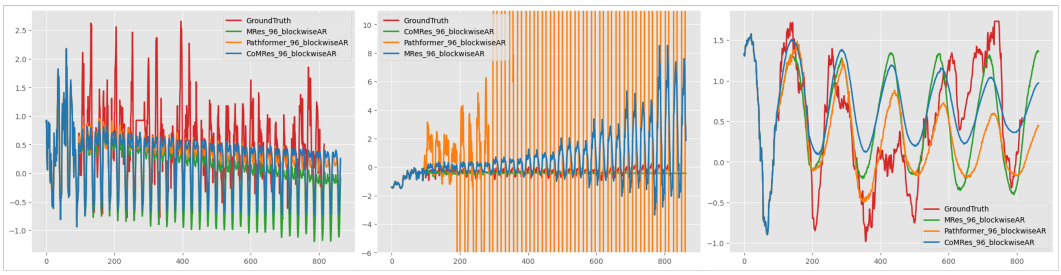

(a) Prediciton Horizon H = 96

(b) Prediciton Horizon H = 192

(c) Prediciton Horizon H = 336

(d) Prediciton Horizon H = 720

Figure 9: Visualization of CoMRes's and Pathformer's prediction results on Weather. The input length H = 96




(a) ETTh1

(b) ETTh2

(c) Weather

Figure 10: Visualization of CoMRes's, MRes's and Pathformer's autoregressive prediction results on ETTh1, ETTh2 and Weather.