Beyond Linear Decoders: Dynamic Expert-Coupled Optimal Decoding for Time Series Forecasting

Binwu Wang[1,2], **Zhipeng Liu**[3,*], **Zhengyang Zhou**[1,2], **Pengkun Wang**[1,2], **Yang Wang**[1,2,*]
[1]University of Science and Technology of China (USTC), Hefei, Anhui, China
[2]Suzhou Institute for Advanced Research, USTC, Suzhou, Jiangsu, China
[3]School of Software, Northeastern University, Shenyang, China
{wbw2024, zzy0929, pengkun, angyan}@ustc.edu.cn
2310543@stu.neu.edu.cn

Abstract

Current multivariate time series forecasting methods mainly rely on static linear decoders, but these often suffer from severe representational bottlenecks. In this paper, we propose a novel architecture called DecoderTS (**D**ynamic **E**xpert-**C**oupled **O**ptimal **DE**coder for Time Series Forecasting), which replaces the conventional static prediction head with a heterogeneous expert bank. DecoderTS adopts a divide-and-conquer strategy to disentangle complex temporal dynamics, such as long-term trends and abrupt changes. Crucially, we introduce an optimal transport (OT)-based dynamic routing mechanism that can adaptively assign customized combinations of experts to different variables. By imposing OT marginal constraints, DecoderTS is theoretically proven to achieve collaborative load balancing among experts and effectively eliminate representation collapse. In extensive experiments on **14** datasets, DecoderTS achieves state-of-the-art forecasting accuracy while maintaining high computational efficiency and low memory usage. Our code is available on GitHub.

1 Introduction

Deep representation learning has achieved remarkable progress in time series forecasting (TSF) (Shao et al., 2024; Liu et al., 2022; Wen et al., 2022; Kim et al., 2025). Research on both Transformer-based variants (Nie et al., 2023; Wu et al., 2021; Ma et al., 2026a) and MLP-based architectures (Wang et al., 2024b; Zeng et al., 2023; Zhang et al., 2026e; Yue et al., 2025) has focused extensively on designing powerful encoders to capture complex temporal dependencies.

Compared with the rapid progress in encoder architectures, decoder design for time series forecasting remains underexplored. Although advanced methods can leverage powerful encoders to learn high-quality latent representations from historical observations, most of them use a simple decoder (e.g., a linear layer) to map latent features directly to future sequences (Zeng et al., 2023; Fei et al., 2025). As shown in Figure 1(a), this design implicitly assumes that complex future dynamics can be captured by a shared decoding rule. However, in multivariate time series, future evolution is typically driven by multiple patterns, including trend, periodicity, and local fluctuations (Zeng et al., 2023; Dai et al., 2024). Such structural heterogeneity is difficult to model with a shared linear mapping. Furthermore, linear decoders are sensitive to missing values and distribution shifts. Missing observations can bias learned representations, while fixed linear mappings struggle to compensate for information loss and evolving dynamics, limiting generalization.

An effective forecasting decoder should explicitly model the structure of future sequences by jointly characterizing trends, periodicity, and local fluctuations. As illustrated in Figure 1(b), this approach

*Zhipeng Liu and Yang Wang are corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

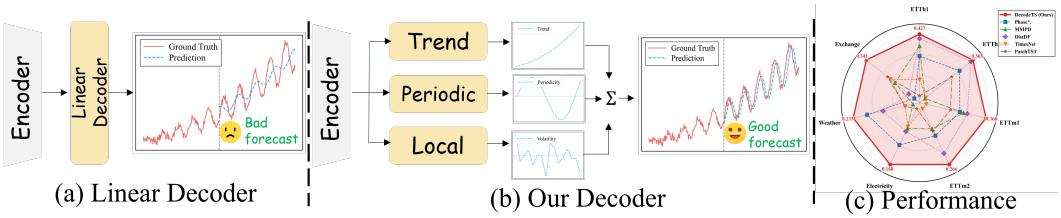


Figure 1: Comparison of the forward computation in (a) a linear decoder and (b) the proposed decoder, with (c) a forecasting performance comparison.

reformulates the forecasting task as a collaborative modeling process over diverse dynamic patterns, enabling a more accurate characterization of temporal dynamics. This is particularly important when overlapping patterns and dynamic coupling make direct forecasting challenging. Explicit structural modeling provides an inductive bias for capturing heterogeneous temporal dependencies and improving robustness to missing observations and distribution shifts, reducing reliance on a single fixed decoding rule. Nevertheless, integrating multiple temporal structures within the decoding process remains challenging because their interactions can be complex, tightly coupled, and variable across channels and forecasting horizons. Designing expressive decoders that disentangle and adaptively combine diverse temporal patterns therefore remains important and underexplored.

To address these challenges, we propose **D**ynamic **E**xpert-**C**oupled **O**ptimal **DE**coder (DecoderTS). Moving beyond traditional linear decoders, DecoderTS utilizes a heterogeneous expert bank with diverse architectures to capture long-term trends, high-frequency abrupt changes, and complex periodicities via a divide-and-conquer strategy. Furthermore, we introduce an optimal transport (OT)-based dynamic routing mechanism for customized forecasting. By evaluating historical latent distributions, this mechanism adaptively assigns optimal expert combinations while leveraging OT marginal constraints to enforce collaborative specialization and prevent pattern redundancy. Finally, a lightweight cross-variable optimization module reconstructs the joint distribution of the multivariate time series through inter-channel constraints. Experiments on diverse real-world datasets demonstrate the state-of-the-art performance of DecoderTS. Our main **contributions** are:

- We design a heterogeneous expert decoder to replace traditional linear decoders. The decoder uses a divide-and-conquer strategy to decouple complex temporal dynamics, substantially improving its capacity to represent diverse time series.
- We design an OT-based dynamic routing mechanism that adaptively assigns optimal expert combinations to different prediction targets. By utilizing the marginal constraints of the optimal transport algorithm, it enforces a collaborative division of labor among experts, effectively avoiding pattern homogenization and redundancy.
- We provide a theoretical foundation for achieving balanced expert allocation and avoiding routing collapse through the proposed OT-based routing mechanism.
- Extensive experiments across diverse real-world datasets demonstrate the strong forecasting performance and computational efficiency of DecoderTS. Comparisons with competitive baselines and ablation studies validate the benefits of heterogeneous expert decoding and optimal transport-based routing, while additional analyses demonstrate its robustness to missing observations and distribution shifts.

2 Related Work

Time series forecasting is widely applied across diverse domains (Miller et al., 2024; Guo et al., 2026; Wang et al., 2025b; Yu and Mi, 2026). Recent research has primarily focused on model architectures, particularly temporal encoders. Popular approaches fall into three broad categories. ❶ Transformer-based models capture long-range dependencies through self-attention (Wu et al., 2021; Ma et al., 2025a; Zhang et al., 2025; Niu et al., 2026), with improvements based on temporal decoupling (Ma et al., 2026a), frequency-domain techniques (Zhou et al., 2022), patching (Nie et al., 2023), and embedding strategies (Zhang et al., 2025). Related studies emphasize frequency-aware multiscale representations and periodic structure (Wang et al., 2023a; Zhang et al., 2024; Wang

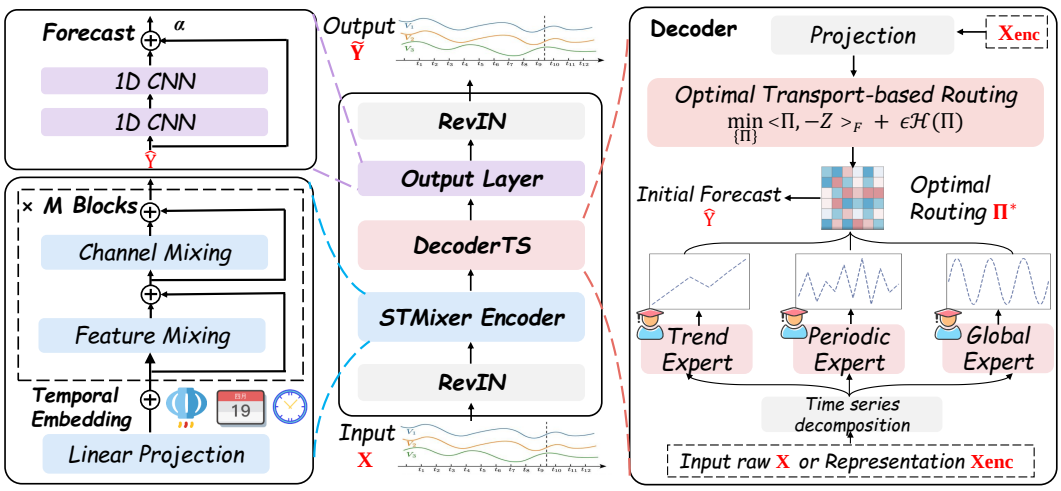


Figure 2: Overview of DecoderTS. An MLP-based STMixer encoder models temporal dynamics and dependencies among variables. A heterogeneous expert decoder then uses optimal transport-based routing to generate future predictions.

et al., 2026d). ❷ MLP-based models offer lightweight architectures with competitive forecasting performance (Lin et al., 2024; Stitsyuk and Choi, 2025; Wang et al., 2024b; Murad et al., 2025). For example, PatchMLP (Tang and Zhang, 2025a) uses a block-based architecture with channel mixing. ❸ TCN-based models efficiently capture local context using convolutions (Wu et al., 2023; Wang et al., 2026c).

Mamba-based models (Wang et al., 2025a), KAN-based models (Huang et al., 2025), and architectures that draw on large language models (Jin et al., 2023; Liu et al., 2025b,a) have also been explored. Multimodal learning (Liu et al., 2024d) and diffusion models (Tashiro et al., 2021) further extend the range of forecasting approaches. Other studies address forecasting with missing variables or distribution shifts (Yu et al., 2024; Li et al., 2026).

For time series forecasting, mixture-of-experts (MoE) methods address temporal heterogeneity by adapting expert selection to the patterns present in different series or time segments. Existing studies use pattern-specific experts to handle patch-level distribution shifts (Sun et al., 2025), introduce token-level specialization in time series foundation models (Liu et al., 2024c), and combine convolutional and Fourier-based experts to model temporal continuity and local periodicity (Ortigossa et al., 2026). Building on these motivations, DecoderTS combines trend, periodic, and global predictions through variable-level OT routing, explicitly balancing expert utilization during forecast generation. Further discussion is provided in **Appendix A**.

3 Methodology

Given a historical multivariate time series observation $\mathbf{X} \in \mathbb{R}^{N \times L}$ containing N variables with a look-back window of length L, our goal is to forecast the future sequence $\mathbf{Y} \in \mathbb{R}^{N \times P}$ over P time steps. To achieve accurate predictions, DecoderTS uses a linear encoder with temporal prior embeddings to capture temporal dynamics, followed by a multi-expert decoder with optimal transport (OT) constraints to generate the final predictions.

3.1 Spatiotemporal Mixer Encoder with Temporal Prior Embedding

We propose a lightweight encoder to capture temporal dynamics and inter-variable correlations. Specifically, the input is first normalized using reversible instance normalization (RevIN) (Kim et al., 2021). Subsequently, $\mathbf{X} \in \mathbb{R}^{N \times L}$ is mapped into a d-dimensional latent space via a linear projection. To preserve global periodic patterns and the characteristics of each variable, the latent representation is augmented, through a broadcasting mechanism, with temporal embeddings (e.g., hour-of-day and day-of-week embeddings) as well as learnable variable embeddings $\mathbf{E}_{var} \in \mathbb{R}^{N \times d}$. This yields a

unified initial latent representation $\mathbf{H}^{(0)} \in \mathbb{R}^{N \times d}$:

$$\mathbf{H}^{(0)} = \text{MLP}(\mathbf{X}^\top) + \mathbf{E}_{hour} + \mathbf{E}_{day} + \mathbf{E}_{var} \tag{1}$$

To capture both intra-variable temporal evolution and inter-variable spatial correlations, $\mathbf{H}^{(0)}$ is subsequently processed by a latent time series mixer encoder composed of M stacked blocks. To ensure highly efficient representation learning, each block relies exclusively on MLPs applied along alternating dimensions. For the m-th block, the representation first undergoes feature mixing along the latent dimension d to extract deep temporal dynamics, followed by channel mixing along the variable dimension N to capture cross-variable dependencies:

$$\begin{aligned} \tilde{\mathbf{H}}^{(m)} &= \mathbf{H}^{(m-1)} + \text{MLP}_f\Big(\text{BatchNorm}\Big(\mathbf{H}^{(m-1)}\Big)\Big) \in \mathbb{R}^{N \times d}, \\ \mathbf{H}^{(m)} &= \tilde{\mathbf{H}}^{(m)} + \Big(\text{MLP}_c\Big(\text{BatchNorm}\Big(\tilde{\mathbf{H}}^{(m)\top}\Big)\Big)\Big)^\top \in \mathbb{R}^{N \times d} \end{aligned} \tag{2}$$

where MLP_f operates along the latent dimension d, and MLP_c operates across the N variables to exchange cross-channel information. The final encoder output is $\mathbf{X}_{enc} = \mathbf{H}^{(M)} \in \mathbb{R}^{N \times d}$.

3.2 Multiscale Decoupling via Heterogeneous Experts

To explicitly handle the intertwined patterns in time series, we first employ a series decomposition block based on a moving average to split the raw temporal sequence $\mathbf{X}$ into a macroscopic trend component $\overline{\mathbf{X}}$ and a seasonal component $\widetilde{\mathbf{X}}$. Unlike standard MoE architectures that use identical structural blocks, we design heterogeneous experts tailored to different time series characteristics. Each expert takes either the decoupled temporal components ($\overline{\mathbf{X}}$, $\widetilde{\mathbf{X}}$) or the encoded latent state $\mathbf{X}_{enc}$ as input, generating a preliminary forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times P}$. The details are as follows:

❶ **Trend Expert**: To capture low-frequency macroscopic trends, we use a single linear layer applied directly to the decomposed trend sequence $\overline{\mathbf{X}}$:

$$\mathcal{E}_1(\overline{\mathbf{X}}) = \overline{\mathbf{X}}\mathbf{W}_t + \mathbf{b}_t \in \mathbb{R}^{N \times P} \tag{3}$$

❷ **Periodic Expert**: To capture long-term temporal semantics and anomalies, we apply a 1D CNN with a bottleneck structure to the seasonal component $\widetilde{\mathbf{X}}$, followed by a linear projection:

$$\mathcal{E}_2(\widetilde{\mathbf{X}}) = \mathbf{W}_p\big(\text{GELU}\big(\text{Conv}_{\times 2}(\text{GELU}(\text{Conv}_{\times 1}(\widetilde{\mathbf{X}})))\big)\big) + \mathbf{b}_p \in \mathbb{R}^{N \times P} \tag{4}$$

where $\times 1$ and $\times 2$ denote convolutions applied separately along the two dimensions of $\widetilde{\mathbf{X}}$.

❸ **Global Expert**: To extract complex, nonlinear global dependencies, we process the mixed latent representation $\mathbf{X}_{enc}$ using an MLP layer:

$$\mathcal{E}_3(\mathbf{X}_{enc}) = \text{MLP}_{global}(\mathbf{X}_{enc}) \in \mathbb{R}^{N \times P} \tag{5}$$

3.3 Optimal Transport-Based Latent Routing

In MoE architectures, heuristic routing mechanisms frequently suffer from representation collapse—a phenomenon where the gating network overwhelmingly favors a small subset of experts, leaving the remaining experts unused and severely restricting the model's effective capacity. To address this issue, we propose Optimal Transport-based Latent Routing. By formulating the routing process as a constrained distribution-alignment problem, this method maximizes the utilization of heterogeneous experts through strict load balancing and globally optimal matching.

Specifically, given the encoded latent states $\mathbf{X}_{enc} \in \mathbb{R}^{N \times d}$, we project them into an expert preference matrix (logits) $\mathbf{Z} \in \mathbb{R}^{N \times K}$, where K is the number of experts (i.e., K=3). Conventional MoE architectures typically adopt a noisy Top-k gating network, defined as:

$$\mathbf{w}_i = \text{Softmax}(\text{Top}_k(\mathbf{z}_i/\tau) + \boldsymbol{\epsilon}) \tag{6}$$

where τ is a temperature scalar and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise. $\mathbf{z}_i \in \mathbb{R}^K$ denotes the logit vector for the i-th variable across all experts. However, this heuristic approach inherently suffers from representation collapse. Due to the non-differentiable nature of the Top-k operation and the fact that

the optimization process inherently favors already well-performing experts, the routing distribution becomes skewed, resulting in only a few experts being activated (Liu et al., 2024a). Consequently, the model tends to rely solely on a small number of experts operating at a single scale. This makes it difficult to adequately capture the rich and diverse temporal patterns in time series data, thereby diminishing the model's generalization capability.

Kantorovich Optimal Transport Formulation. To address this issue, instead of relying on a soft auxiliary load-balancing penalty, we impose exact load balancing as a hard marginal constraint within a discrete optimal transport (OT) framework. Let $\mathbf{r} = \frac{1}{N}\mathbf{1}_N$ denote the uniform empirical distribution of the N variate tokens, and $\mathbf{c} = \frac{1}{K}\mathbf{1}_K$ denote the uniform capacity distribution of the K experts. We define the transportation cost matrix as $\mathbf{C} = -\mathbf{Z}$. Our objective is to find an **expert routing matrix** $\mathbf{\Pi} \in \mathbb{R}^{N\times K}$ within the transportation polytope $\mathcal{U}(\mathbf{r}, \mathbf{c})$:

$$\mathcal{U}(\mathbf{r}, \mathbf{c}) = \left\{\mathbf{\Pi} \in \mathbb{R}_+^{N\times K} \mid \mathbf{\Pi}\mathbf{1}_K = \mathbf{r}, \mathbf{\Pi}^\top \mathbf{1}_N = \mathbf{c}\right\} \tag{7}$$

We minimize the Frobenius inner product of the transport plan and cost matrix:

$$\min_{\mathbf{\Pi}\in\mathcal{U}(\mathbf{r},\mathbf{c})} \langle \mathbf{\Pi}, \mathbf{C}\rangle_F = \sum_{i=1}^{N}\sum_{j=1}^{K} \mathbf{\Pi}_{i,j}\mathbf{C}_{i,j} \tag{8}$$

where $\cdot_{i,j}$ denotes the element in the i-th row and j-th column of the matrix. By strictly enforcing $\mathbf{\Pi} \in \mathcal{U}(\mathbf{r}, \mathbf{c})$, we mathematically guarantee that every expert receives exactly $1/K$ of the total probability mass, eradicating representation collapse by definition.

Entropy Regularization and Sinkhorn Projections. Since our router transports N variate tokens to a small number of experts K, it forms a rectangular $N \times K$ OT problem rather than a dense balanced $N \times N$ assignment problem. We adopt entropy-regularized OT with Sinkhorn iterations to obtain smooth, differentiable routing. Therefore, we add a negative Shannon entropy regularization term, which is convex over the positive orthant:

$$\min_{\mathbf{\Pi}\in\mathcal{U}(\mathbf{r},\mathbf{c})} \langle \mathbf{\Pi}, \mathbf{C}\rangle_F + \epsilon \sum_{i=1}^{N}\sum_{j=1}^{K} \mathbf{\Pi}_{i,j} \log \mathbf{\Pi}_{i,j} \tag{9}$$

where $\epsilon > 0$ controls the exploration-exploitation trade-off. This negative-entropy regularization transforms the objective into a strictly convex problem, guaranteeing a unique global minimum in the form of a scaled exponential matrix: $\mathbf{\Pi}_{i,j} = u_i \exp(-\mathbf{C}_{i,j}/\epsilon)v_j$. We approximate the dual scaling vectors $\mathbf{u} \in \mathbb{R}^N$ and $\mathbf{v} \in \mathbb{R}^K$ efficiently using the iterative Sinkhorn-Knopp algorithm, which performs alternating Bregman projections:

$$\mathbf{u}^{(t+1)} = \mathbf{r} \oslash \left(\mathbf{K}\mathbf{v}^{(t)}\right), \quad \mathbf{v}^{(t+1)} = \mathbf{c} \oslash \left(\mathbf{K}^\top \mathbf{u}^{(t+1)}\right) \tag{10}$$

where $\mathbf{K} = \exp(-\mathbf{C}/\epsilon)$ and $\oslash$ denotes element-wise division. After T iterations, the **optimal expert routing matrix** is denoted as $\mathbf{\Pi}^* = \mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v}) \in \mathbb{R}^{N\times K}$. We derive this formula in **Appendix B**. The optimized routing weights combine the expert predictions:

$$\hat{\mathbf{Y}} = \sum_{e=1}^{K} \mathbf{A}_{:,e} \odot \mathcal{E}_e(\mathbf{X}, \mathbf{X}_{enc}), \quad \mathbf{A} = \mathrm{diag}(\mathbf{r})^{-1}\mathbf{\Pi}^*. \tag{11}$$

3.4 Final Refinement and Output

To refine the aggregated forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times P}$, we apply a final cross-channel mixing module parameterized by 1×1 convolutions along the variate dimension:

$$\widetilde{\mathbf{Y}} = \hat{\mathbf{Y}} + \alpha \cdot \mathrm{Conv1D}_{1\times1}(\mathrm{GELU}(\mathrm{Conv1D}_{1\times1}(\hat{\mathbf{Y}}))) \tag{12}$$

where α is a learnable scaling parameter that stabilizes training. Finally, $\widetilde{\mathbf{Y}}$ is denormalized through the RevIN module to produce the final forecast.

3.5 Theoretical Analysis

We analyze the entropy-regularized routing mechanism from an information-geometric perspective. The following theorem shows that it performs an optimal probability projection that enforces expert load balancing to eliminate representation collapse while approximating the model's original unconstrained preferences as closely as possible.

Theorem 1 (Information-Geometric KL Projection). The entropy-regularized optimal transport objective is mathematically equivalent to projecting the unnormalized expert preference matrix onto the transportation polytope under the Kullback-Leibler (KL) divergence.

Proof. Let the Gibbs kernel matrix be defined as $\mathbf{K} = \exp(-\mathbf{C}/\epsilon) = \exp(\mathbf{Z}/\epsilon)$, which represents the original unconstrained exponential preference mapping from tokens to experts. The entropy-regularized objective can be algebraically rewritten as:

$$\begin{aligned} \mathcal{L}_\epsilon(\mathbf{\Pi}) &= \langle \mathbf{\Pi}, \mathbf{C} \rangle_F + \epsilon \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbf{\Pi}_{i,j} \log \mathbf{\Pi}_{i,j} \\ &= \epsilon \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbf{\Pi}_{i,j} \log \frac{\mathbf{\Pi}_{i,j}}{\exp(-\mathbf{C}_{i,j}/\epsilon)} \\ &= \epsilon \cdot \mathrm{KL}(\mathbf{\Pi} \,\|\, \mathbf{K}) \end{aligned} \tag{13}$$

Since $\epsilon > 0$, minimizing $\mathcal{L}_\epsilon(\mathbf{\Pi})$ over the polytope $\mathcal{U}(\mathbf{r}, \mathbf{c})$ that satisfies the marginal capacity constraints is equivalent to computing the information projection (I-projection):

$$\mathbf{\Pi}^* = \arg \min_{\mathbf{\Pi} \in \mathcal{U}(\mathbf{r}, \mathbf{c})} \mathrm{KL}(\mathbf{\Pi} \,\|\, \mathbf{K}) \tag{14}$$

This proves that the optimal routing plan $\mathbf{\Pi}^*$ is the closest valid joint distribution to the original model outputs $\mathbf{K}$ in the information-geometric space. Consequently, by preserving the model's original routing preferences as closely as possible under strict capacity constraints, this mechanism ensures that the specialization of heterogeneous experts is not degraded by forced uniform allocation. □

Complexity Analysis. The proposed model consists of an STMixer encoder and an OT-based fusion module. The STMixer encoder relies on linear projection, MLP-based mixing, and lightweight expert predictors, leading to a complexity of $\mathcal{O}(L)$ with respect to the input sequence length L. The OT-based fusion module introduces additional routing and Sinkhorn iterations, whose cost depends on the number of experts and routing steps, both of which are set to small constants in our implementation. Therefore, the overall complexity of the proposed model remains linear in L, i.e., $\mathcal{O}(L)$.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate the proposed method on widely used real-world datasets, including the ETT series (Zhou et al., 2021a), Electricity, Weather (Wu et al., 2021), Exchange, AQWan, CzeLan, and ZafNoo (Qiu et al., 2025) for long-term time series forecasting, as well as Air Quality, ILI, Mobile Traffic (Barlacchi et al., 2015), and ST-EVCDP (Li et al., 2025) for short-term time series forecasting. For the short-term forecasting benchmarks, the look-back window L is set to 96, and the prediction horizon P is chosen from $\{12, 24, 48\}$. For the long-term forecasting benchmarks, the look-back window L is fixed at 96, while the prediction horizon P is selected from $\{96, 192, 336, 720\}$. These datasets exhibit substantial diversity in terms of scale, number of variables, and application domains. Detailed statistics are summarized in Table 3.

Implementation Details We implement all experiments in PyTorch and run them on an NVIDIA GeForce RTX 4090D GPU with 24 GB of memory. We use the AdamW optimizer (Kingma, 2014) and retain the final incomplete batch during training. For most datasets, the learning rate is selected from {0.003, 0.001, 0.0005}, and the model dimension is selected from {64, 128, 256}. The kernel size of the periodic expert is fixed at 3. For routing, the temperature parameter τ, which controls the sharpness of the expert assignment distribution, is set to 1, while the Gaussian noise standard deviation σ is set to 0. In the OT-based router, the entropy regularization coefficient ϵ is set to 0.05,

Table 1: Multivariate time series forecasting results with a fixed look-back length of $L = 96$. The abbreviation "*." denotes "former" in model names. Results are averaged across all forecasting horizons. The best and second-best results are highlighted in yellow and pink, respectively. Full results are provided in Tables 4 and 5.

Model	DecoderTS (Ours)		Phase*. (2026)		MMPD (2026)		MixLinear (2026)		DistDF (2026)		TimeFilter (2025)		WPMixer (2025)		iTrans*. (2024)		TimesNet (2023)		PatchTST (2023)		TimeMixer (2023)		DLinear (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Long-term forecasting results, with prediction window lengths of {96, 192, 336, 720}.																								
ETTh1	0.427	0.419	0.442	0.422	0.435	0.431	0.452	0.454	0.432	0.429	0.450	0.439	0.455	0.441	0.454	0.448	0.458	0.450	0.469	0.455	0.445	0.439	0.456	0.452
ETTh2	0.365	0.390	0.379	0.397	0.404	0.421	0.427	0.447	0.371	0.396	0.380	0.406	0.382	0.407	0.383	0.407	0.414	0.427	0.387	0.407	0.369	0.391	0.559	0.515
ETTm1	0.368	0.372	0.396	0.386	0.380	0.392	0.444	0.442	0.378	0.394	0.376	0.395	0.400	0.405	0.407	0.410	0.400	0.406	0.387	0.400	0.378	0.392	0.403	0.407
ETTm2	0.266	0.308	0.279	0.323	0.282	0.329	0.380	0.443	0.277	0.321	0.272	0.321	0.278	0.322	0.288	0.332	0.291	0.333	0.281	0.326	0.279	0.325	0.350	0.401
Electricity	0.168	0.260	0.181	0.275	0.192	0.338	0.234	0.324	0.190	0.282	0.169	0.265	0.194	0.285	0.178	0.270	0.193	0.295	0.205	0.290	0.185	0.274	0.212	0.300
Weather	0.233	0.255	0.257	0.271	0.263	0.284	0.293	0.344	0.248	0.275	0.242	0.270	0.252	0.280	0.258	0.278	0.259	0.287	0.259	0.273	0.246	0.276	0.265	0.317
Exchange	0.341	0.394	0.393	0.421	0.372	0.415	0.395	0.444	0.387	0.441	0.436	0.466	0.360	0.403	0.416	0.443	0.371	0.408	0.364	0.415	0.359	0.414	0.356	0.414
AQWan	0.844	0.489	0.885	0.513	0.888	0.516	0.947	0.583	0.956	0.604	0.860	0.508	0.866	0.511	0.896	0.522	0.863	0.517	0.881	0.519	0.883	0.519	0.856	0.539
CzeLan	0.215	0.252	0.260	0.276	0.271	0.303	0.288	0.326	0.273	0.315	0.244	0.279	0.246	0.282	0.271	0.304	0.265	0.308	0.257	0.292	0.286	0.318	0.556	0.507
ZafNoo	0.539	0.431	0.587	0.453	0.595	0.481	0.602	0.514	0.739	0.590	0.543	0.451	0.554	0.455	0.578	0.471	0.602	0.490	0.554	0.462	0.601	0.488	0.544	0.472
Short-term forecasting results, with prediction window lengths of {12, 24, 48}.																								
Air	0.797	0.681	0.893	0.723	0.870	0.749	1.318	0.868	1.006	0.810	0.937	0.725	0.864	0.727	0.863	0.721	0.938	0.785	0.836	0.708	0.874	0.745	0.878	0.739
MTraffic	0.237	0.302	0.377	0.401	0.382	0.502	1.388	0.840	1.072	0.758	0.270	0.319	0.422	0.475	0.293	0.337	0.331	0.381	0.310	0.353	0.466	0.509	0.802	0.692
ILI	1.494	0.732	2.450	1.034	3.756	1.412	5.821	1.794	4.428	1.560	2.244	0.996	2.106	1.102	1.834	0.865	2.470	0.998	3.870	1.382	3.541	1.188	3.938	1.429
ST-EVCDP	0.636	0.511	0.731	0.560	0.694	0.566	1.139	1.009	1.072	0.758	0.741	0.578	0.821	0.648	0.711	0.553	0.766	0.581	0.696	0.570	0.740	0.604	1.736	0.836
count of 1st	14	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

and the number of Sinkhorn iterations T is set to 10 for most datasets. We use mean absolute error (MAE) as the training loss and report MAE and mean squared error (MSE) as evaluation metrics.

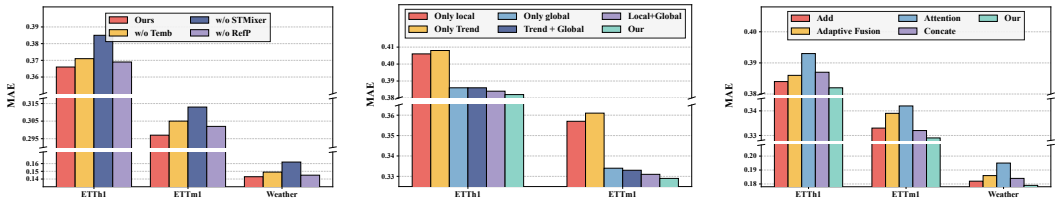
Baselines We select multiple time series forecasting baselines, including DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), PatchTST (Nie et al., 2023), CrossGNN (Huang et al., 2023), FITS (Xu et al., 2023), FreTS (Yi et al., 2023), iTransformer (Liu et al., 2024b), FilterNet (Yi et al., 2024), SOFTS (Han et al., 2024), Crossformer (Zhang and Yan, 2023), CycleNet (Lin et al., 2024), TimeMixer (Wang et al., 2024b), Duet (Qiu et al., 2025), xPatch (Stitsyuk and Choi, 2025), WPMixer (Murad et al., 2025), SimpleTM (Chen et al., 2025), Amplifier (Fei et al., 2025), MultiPatchformer (Naghashi et al., 2025), PatchMLP (Tang and Zhang, 2025b), TimeKAN (Huang et al., 2025), TimeFilter (Hu et al., 2025), TFPS (Sun et al., 2025), PhaseFormer (Niu et al., 2026), MMPD (Zhang et al., 2026d), MixLinear (Ma et al., 2026b), interPDN (Kong and Hong, 2026), FreqEvo (Wang et al., 2025c), MDMLP-EIA (Zhang et al., 2026a), SSformer (Liu et al., 2026b), TimeSAF (Zhang et al., 2026c), PMDFormer (Hu et al., 2026), and DistDF (Wang et al., 2026b).

4.2 Analysis of Experimental Results

As shown in Table 1, our DecoderTS achieves competitive performance. For long-term forecasting tasks, DLinear performs poorly because it relies solely on a simple linear layer to model temporal dynamics. In contrast, MixLinear depends on linear transformations and Fourier transforms, which limit its ability to capture complex temporal patterns. TimeMixer introduces a decomposition mechanism and therefore achieves better forecasting performance than DLinear. Transformer-based architectures, such as PatchTST, effectively model long-range dependencies through the attention mechanism. PhaseFormer builds an efficient variable interaction mechanism through compact phase embeddings and thus achieves strong performance. TimeFilter models correlations among variables using a sparse graph interaction mechanism, which also leads to strong forecasting results. DistDF better fits the data distribution through joint distribution alignment, thereby achieving strong performance. Our DecoderTS achieves state-of-the-art results across all metrics on all datasets, which demonstrates that a powerful decoder has great potential to improve forecasting performance.

4.3 Ablation Study

(a) General Modules. We conduct ablation studies on three modules: STMixer, the temporal prior, and prediction refinement. We design three variants: (1) w/o Tem, which removes the temporal embedding prior; (2) w/o STMixer, which uses only an MLP as the temporal encoder; and (3) w/o



(a) General Modules (b) Heterogeneous Experts (c) Expert Fusion Strategy

Figure 3: Ablation Study

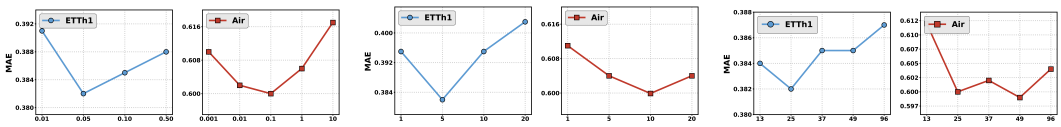
ReFP, which directly outputs $\hat{\mathbf{Y}}$ in Eq. 11 as the prediction. As shown in Figure 3(a), the w/o Tem variant yields larger errors, indicating that temporal prior information helps the model better capture temporal dynamics. The w/o STMixer variant also performs poorly, suggesting that a simple MLP is insufficient to model complex temporal dynamics.

(b) Heterogeneous Experts. We further evaluate the effectiveness of each expert. First, we generate predictions using each expert individually. Then, we combine the Global expert with the Trend expert and the Periodic expert. The experimental performance is shown in Figure 3(b). "Only Global" can be regarded as a linear decoder, while progressively incorporating Periodic and Trend experts introduces information at different scales, thereby improving the model's predictive performance.

(c) Expert Fusion Strategy. We further compare different expert fusion strategies. "Add" means directly summing the predictions of the three experts, while "Adaptive Fusion" uses learnable parameters for adaptive combination. "Attention" employs a Transformer as the fusion strategy, and "Concat" denotes directly concatenating the outputs of the three experts. The performance is shown in Figure 3(c). Among these methods, "Attention" performs the worst, possibly because the complexity of the Transformer leads to overfitting. "Add" and "Concat" achieve similar performance, but both are inferior to our optimal-transport-based routing strategy, which avoids overreliance on any single expert and thus improves generalization.

4.4 Hyperparameter Analysis

In this section, we evaluate sensitivity to three hyperparameters: the entropy regularization coefficient ϵ and the number of Sinkhorn iterations T in OT-based routing, as well as the decomposition kernel size. The experimental results are shown in Figure 4. On ETTh1, the lowest MAE is obtained at $\epsilon = 0.05$ and $T = 5$ among the evaluated settings, whereas on Air the lowest MAE is obtained at $T = 10$; we therefore use $T = 10$ as the default for most datasets. A larger ϵ strengthens entropy regularization and produces smoother routing weights; if it is too large, the routing weights can become nearly uniform and less sensitive to expert preferences. A smaller ϵ yields a more concentrated transport plan, but an excessively small value can make the assignment overly sharp and the Sinkhorn computation less numerically stable. Too few Sinkhorn iterations may leave the transport plan insufficiently converged, while additional iterations increase computation and do not necessarily improve forecasting performance. For the decomposition kernel size, a small kernel captures local variations but may inadequately smooth the trend, whereas a large kernel may smooth out useful temporal details.

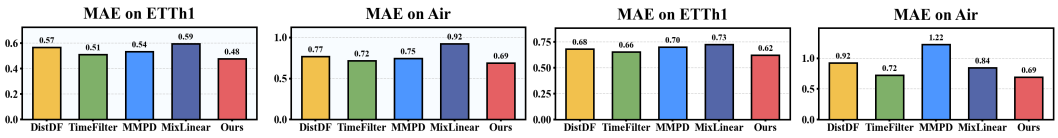


(a) Entropy Regularization ϵ (b) Sinkhorn Iterations T (c) Decomposition Kernel Size

Figure 4: Hyperparameter Analysis

4.5 Experimental Analysis of Robustness to Complex Scenarios

We further compare the performance of different models under complex scenarios: (1) a missing-data scenario, in which 50% of the data in the training set were randomly masked; and (2) a distribution-shift scenario, in which the train/validation/test ratio was reset to 3:2:5. The performance of our model and several state-of-the-art models is shown in Figure 5. The results demonstrate the superiority of our model. The predictors used by these models rely on simple mapping rules, making it difficult for them to capture sufficient information to support prediction in complex scenarios. In contrast, our multi-expert decoding mechanism enables the model to capture rich information at multiple granularities, thereby improving its robustness in these complex scenarios.



(a) Scenario with 50% Missing Data (b) Distribution-Shift Scenario

Figure 5: Comparison of predictive performance under complex scenarios.

4.6 Model Efficiency Analysis

Table 2 reports efficiency on an NVIDIA RTX 4090 D. Baselines use default architecture parameters and start at batch size 128, halved after runtime failures; DecoderTS keeps its reproducible configuration. After warm-up, we average three complete training epochs and three full-test-set inference passes. Timing includes CPU batch preparation, host-to-device transfer, and model computation, excluding validation, checkpointing, and evaluation metrics. Baseline epochs use 2917 training windows; DecoderTS retains its original incomplete-batch handling and uses 2816. All timed models predict the same 853 test windows. Counted FLOPs cover supported matrix, convolution, and attention operations; MAC equivalents equal half these counts. DecoderTS requires **1.747 s** per epoch, **0.537 s** for test-set inference, and **2394.87 MiB** of peak allocated GPU memory. Accuracy was not re-evaluated under these efficiency configurations.

Table 2: Efficiency comparison on MTraffic (900 variables, $P = 12$). MACs: multiply–accumulate operations; FLOPs: floating-point operations; M: million (10^6); G: billion (10^9); MiB: mebibytes; s: seconds. Red arrows (↑): percentage increases over DecoderTS; "Epoch Time": seconds per epoch.

Model	MSE	# MACs	# FLOPs	Epoch Time	GPU Memory	Inference Time (s)
TimeFilter	0.247↑17.06%	88.09G↑21932.73%	176.19G↑21932.73%	276.77s↑15738.54%	17462.09 MiB↑629.15%	28.58s↑5226.32%
Crossformer	0.263↑24.64%	43.13G↑10687.02%	86.26G↑10687.02%	59.02s↑3277.26%	17473.89 MiB↑629.64%	4.91s↑814.96%
TimeMixer	0.414↑96.20%	1.16G↑188.92%	2.31G↑188.92%	11.21s↑541.60%	16896.99 MiB↑605.55%	1.29s↑139.77%
MMPD	0.315↑49.28%	9.27G↑2218.14%	18.54G↑2218.14%	32.10s↑1736.74%	14249.01 MiB↑494.98%	1.65s↑207.76%
iTransformer	0.251↑18.96%	7.41G↑1753.58%	14.82G↑1753.58%	8.39s↑380.22%	12696.54 MiB↑430.16%	0.75s↑39.91%
PatchTST	0.248↑17.54%	68.37G↑16999.40%	136.74G↑16999.40%	52.30s↑2892.71%	15394.37 MiB↑542.81%	5.65s↑952.63%
Ours	**0.211**	**399.83M**	**799.65M**	**1.75s**	**2394.87 MiB**	**0.54s**

4.7 OT Routing and Multi-Expert Analysis

Using ETTh1 with 96 input steps and 96 prediction steps as an example, we extract the expert routing matrix and compare the activation distributions with and without OT routing, counting one activation for the expert with the highest routing probability. As shown in Figure 6, without OT, expert routing is biased and the model mainly relies on the Trend expert, which, as shown on the right, fails to provide accurate predictions. With OT, expert activation becomes more balanced, improving robustness. Moreover, the global expert, which can be viewed as a linear decoder, shows larger prediction errors, while our decoder integrates multiscale information and achieves better forecasting performance.

5 Conclusion

In this paper, we proposed DecoderTS, a dynamic expert-coupled decoder for multivariate time series forecasting. The framework replaces a shared static prediction head with heterogeneous experts

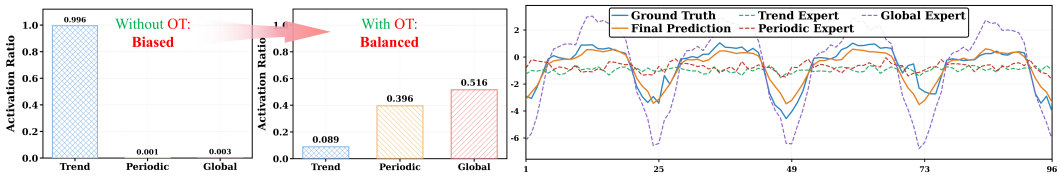


Figure 6: Expert activation distributions and forecast visualizations.

that capture trend, periodic, and global patterns. Optimal transport-based routing combines their predictions for individual variables while imposing explicit constraints on expert utilization. Together with a lightweight encoder and cross-variable refinement, this design supports expressive and efficient forecasting. Experiments on 14 datasets against approximately 30 baselines demonstrate strong performance under standard conditions, missing observations, and distribution shifts. Ablation and routing analyses further illustrate the contributions of expert specialization and balanced allocation.

These findings highlight decoder design as an important direction for improving time series forecasting. Future work will extend the framework to irregularly sampled and streaming data and explore richer expert organizations and routing mechanisms for complex temporal dynamics.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work is partially supported by the National Natural Science Foundation of China (Grant No. 12227901) and the Natural Science Foundation of Jiangsu Province (Grant No. BK20250482). AI-driven experiments, simulations, and model training used the Chinese Academy of Sciences' robotic AI-Scientist platform.

References

Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. 2015. A multi-source dataset of urban life in the city of Milan and the Province of Trentino. *Scientific data* 2, 1 (2015), 150055.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. 2025. SimpleTM: A Simple Baseline for Multivariate Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. 2024. Periodicity decoupling framework for long-term series forecasting. In *The twelfth international conference on learning representations*.

Shiming Fan, Hua Wang, and Fan Zhang. 2026. TRED: Exponential feature decomposition three-channel networks for multivariate time series forecasting. *Applied Soft Computing* 189 (2026), 114453. doi:10.1016/j.asoc.2025.114453

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. 2025. Amplifier: Bringing Attention to Neglected Low-Energy Components in Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 11645–11653.

Qi Guo, Baolin Zhao, Mingchen Song, and Guoqiang Zhong. 2026. A Survey of Deep Learning for Time Series Forecasting: Taxonomy, Analysis and Future Directions. *IEEE Transactions on Knowledge and Data Engineering* (2026).

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. 2024. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems* 37 (2024), 64145–64175.

Ao Hu, Liangjian Wen, Jiang Duan, Yong Dai, YAN HE, Dongkai Wang, Jun Wang, Yukun Zhang, Ruoxi Jiang, and Zenglin Xu. 2026. PMDformer: Patch-Mean Decoupling Information Transformer for Long-term Forecasting. In *The Fourteenth International Conference on Learning Representations*.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. 2025. TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting. *arXiv preprint arXiv:2501.13041* (2025).

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. CrossGNN: Confronting Noisy Multivariate Time Series Via Cross Interaction Refinement. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., 46885–46902. doi:10.52202/075280-2031

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. 2025. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. *arXiv preprint arXiv:2502.06910* (2025).

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).

Changhun Kim, Yechan Mun, Hyeongwon Jang, Eunseo Lee, Sangchul Hahn, and Eunho Yang. 2025. Delta-XAI: A Unified Framework for Explaining Prediction Changes in Online Time Series Monitoring. *arXiv preprint arXiv:2511.23036* (2025).

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

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

Linghao Kong and Xiaopeng Hong. 2026. Time Series Forecasting via Direct Per-Step Probability Distribution Modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 22653–22661.

Han Li, Haohao Qu, Xiaojun Tan, Linlin You, Rui Zhu, and Wenqi Fan. 2025. UrbanEV: An Open Benchmark Dataset for Urban Electric Vehicle Charging Demand Prediction. *Scientific Data* 12 (2025), 523.

Yujie Li, Zezhi Shao, Chengqing Yu, Yisong Fu, Tao Sun, Yongjun Xu, and Fei Wang. 2026. APT: Affine Prototype-Timestamp for Time Series Forecasting Under Distribution Shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 15180–15188. doi:10.1609/aaai.v40i18.38542

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. 2024. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems* 37 (2024), 106315–106345.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

Chenxi Liu, Hao Miao, Qianxiong Xu, Shaowen Zhou, Cheng Long, Yan Zhao, Ziyue Li, and Rui Zhao. 2025a. Efficient Multivariate Time Series Forecasting via Calibrated Language Models with Privileged Knowledge Distillation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. 3165–3178. doi:10.1109/icde65448.2025.00237

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. 2025b. TimeCMA: Towards LLM-Empowered Multivariate Time Series Forecasting via Cross-Modality Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 18780–18788. doi:10.1609/aaai.v39i18.34067

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. 2024d. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems* 37 (2024), 77888–77933.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. 2024c. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469* (2024).

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

Ying Liu, Bo Liu, Sheng Huang, Gang Luo, Wenbo Hu, Meng Wang, and Richang Hong. 2026b. Sparse-Scale Transformer with Bidirectional Awareness for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 23899–23907.

Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. 2022. Multivariate time-series forecasting with temporal polynomial graph neural networks. *Advances in neural information processing systems* 35 (2022), 19414–19426.

Zhipeng Liu, Peibo Duan, Xuan Tang, Baixin Li, Yongsheng Huang, Mingyang Geng, Changsheng Zhang, Bin Zhang, and Binwu Wang. 2026a. TimeFormer: Transformer with Attention Modulation Empowered by Temporal Characteristics for Time Series Forecasting. *Expert Systems with Applications* 307 (2026), 131040. doi:10.1016/j.eswa.2025.131040

Aitian Ma, Dongsheng Luo, and Mo Sha. 2026b. MixLinear: Extreme Low Resource Multivariate Time Series Forecasting with $0.1K$ Parameters. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=QUj0KuCumD

Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Binwu Wang, and Yang Wang. 2026a. Phat: Modeling period heterogeneity for multivariate time series forecasting. *arXiv preprint arXiv:2602.00654* (2026).

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. 2025a. MoFo: Empowering Long-term Time Series Forecasting with Periodic Pattern Modeling. In *Advances in Neural Information Processing Systems*, Vol. 38. Curran Associates, Inc., 94558–94589. doi:10.52202/085713-2844

Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. 2025b. BiST: A Lightweight and Efficient Bi-directional Model for Spatiotemporal Prediction. *Proceedings of the VLDB Endowment* 18, 6 (2025), 1663–1676. doi:10.14778/3725688.3725697

Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. 2025c. Robust Spatio-Temporal Centralized Interaction for OOD Learning. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 267)*. PMLR, 42165–42192. https://proceedings.mlr.press/v267/ma25s.html

John A Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I Budak Arpinar, and Ninghao Liu. 2024. A survey of deep learning and foundation models for time series forecasting. *arXiv preprint arXiv:2401.13912* (2024).

Hesam Mosalli, Saba Sanami, Yu Yang, Hen-Geul Yeh, and Amir G Aghdam. 2025. Dynamic load balancing for ev charging stations using reinforcement learning and demand prediction. In *2025 IEEE International systems Conference (SysCon)*. IEEE, 1–7.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. 2025. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 19581–19588.

Vahid Naghashi, Mounir Boukadoum, and Abdoulaye Banire Diallo. 2025. A multiscale model for multivariate time series forecasting. *Scientific Reports* 15, 1 (2025), 1565.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Yiming Niu, Jinliang Deng, and Yongxin Tong. 2026. PhaseFormer: From Patches to Phases for Efficient and Effective Time Series Forecasting. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=Lk9SqMQzhX

Evandro S Ortigossa, Guy Lutsker, and Eran Segal. 2026. MoHETS: Long-term Time Series Forecasting with Mixture-of-Heterogeneous-Experts. *arXiv preprint arXiv:2601.21866* (2026).

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proc. VLDB Endow.* 17, 9 (2024), 2363–2377.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. 2025. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. In *SIGKDD*. 1185–1196.

Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. 2025. BLAST: Balanced Sampling Time Series Corpus for Universal Forecasting Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. 2502–2513. doi:10.1145/3711896.3736860

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. 2024. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering* 37, 1 (2024), 291–305.

Artyom Stitsyuk and Jaesik Choi. 2025. xPatch: Dual-Stream Time Series Forecasting with Exponential Seasonal-Trend Decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 20601–20609.

Yanru Sun, Zongxia Xie, Emadeldeen Eldele, Dongyue Chen, Qinghua Hu, and Min Wu. 2025. Learning Pattern-Specific Experts for Time Series Forecasting Under Patch-level Distribution Shift. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=CtoIG9Iwas

Peiwang Tang and Weitai Zhang. 2025a. Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 12640–12648.

Peiwang Tang and Weitai Zhang. 2025b. Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 12640–12648.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems* 34 (2021), 24804–24816.

Binwu Wang, Gaoyun Lin, Jiaming Ma, Qihe Huang, Zhengyang Zhou, Xu Wang, Pengkun Wang, and Yang Wang. 2026a. Crisp: A Spectral-Based Interaction Strategy for Multivariate Time Series Forecasting. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 128034–128050. https://proceedings.mlr.press/v306/wang26ec.html

Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. 2024a. STONE: A Spatio-temporal OOD Learning Framework Kills Both Spatial and Temporal Shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA, 2948–2959. doi:10.1145/3637528.3671680

Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. 2023b. Knowledge Expansion and Consolidation for Continual Traffic Prediction With Expanding Graphs. *IEEE Transactions on Intelligent Transportation Systems* 24, 7 (2023), 7190–7201. doi:10.1109/TITS.2023.3263904

Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. 2023c. Pattern Expansion and Consolidation on Evolving Graphs for Continual Traffic Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA, 2223–2232. doi:10.1145/3580305.3599463

Eric Wang, Licheng Pan, Yuan Lu, Zhixuan Chu, Xiaoxi Li, Shuting He, Zi Ciu Chan, Haoxuan Li, Qingsong Wen, and Zhouchen Lin. 2026b. DistDF: Time-series Forecasting Needs Joint-distribution Wasserstein Alignment. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=VrdLwUmzBy

Guohong Wang, Xianhan Tan, Zengming Lin, Binli Luo, Shangjian Zhong, and Kele Xu. 2025c. FreqEvo: Enhancing Time Series Forecasting With Multi-Level Frequency Domain Feature Extraction. *IEEE Transactions on Knowledge and Data Engineering* 38, 2 (2025), 1099–1113.

Hao Wang, Licheng Pan, Zhichao Chen, Xu Chen, Qingyang Dai, Lei Wang, Haoxuan Li, and Zhouchen Lin. 2025b. Time-o1: Time-series forecasting needs transformed label alignment. *arXiv preprint arXiv:2505.17847* (2025).

Huiqiang Wang, Jieming Shi, and Li Qing. 2026c. FACT: Fine-grained Across-variable Convolution for Multivariate Time Series Forecasting. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=j3gNYqrHtl

Min Wang, Hua Wang, and Fan Zhang. 2023a. FAMC-Net: Frequency Domain Parity Correction Attention and Multi-Scale Dilated Convolution for Time Series Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2554–2563. doi:10.1145/3583780.3614876

Min Wang, Hua Wang, and Fan Zhang. 2026d. Correctformer: A transformer architecture for correcting periodic drift in time-series forecasting. *Neural Networks* 196 (2026), 108375. doi:10.1016/j.neunet.2025.108375

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. 2024b. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. 2025a. Is mamba effective for time series forecasting? *Neurocomputing* 619 (2025), 129178.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125* (2022).

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*.

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

Mingyuan Xia, Chunxu Zhang, Zijian Zhang, Hao Miao, Qidong Liu, Yuanshao Zhu, and Bo Yang. 2025. TimeEmb: A Lightweight Static-Dynamic Disentanglement Framework for Time Series Forecasting. *arXiv preprint arXiv:2510.00461* (2025). doi:10.48550/arXiv.2510.00461

Zhijian Xu, Ailing Zeng, and Qiang Xu. 2023. FITS: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756* (2023).

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. 2024. Filternet: Harnessing frequency filters for time series forecasting. *Advances in Neural Information Processing Systems* 37 (2024), 55115–55140.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2023. Frequency-domain MLPs are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems* 36 (2023), 76656–76679.

Chengqing Yu and Xiwei Mi. 2026. Wind Speed Foundation Model. *Expert Systems with Applications* (2026), 135004. doi:10.1016/j.eswa.2026.135004

Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, and Yongjun Xu. 2024. GinAR: An End-To-End Multivariate Time Series Forecasting Model Suitable for Variable Missing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3989–4000. doi:10.1145/3637528.3672055

Wenzhen Yue, Yong Liu, Haoxuan Li, Hao Wang, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. 2025. Olinear: A linear model for time series forecasting in orthogonally transformed domain. *arXiv preprint arXiv:2505.08550* (2025).

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

Boya Zhang, Shuaijie Yin, Huiwen Zhu, and Xing He. 2026e. FreqCycle: A Multi-Scale Time-Frequency Analysis Method for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 28159–28167.

Fan Zhang, Shiming Fan, and Hua Wang. 2026b. Time-TK: A Multi-Offset Temporal Interaction Framework Combining Transformer and Kolmogorov-Arnold Networks for Time Series Forecasting. *arXiv preprint arXiv:2602.11190* (2026). doi:10.48550/arXiv.2602.11190

Fan Zhang, Shiming Fan, and Hua Wang. 2026c. TimeSAF: Towards LLM-Guided Semantic Asynchronous Fusion for Time Series Forecasting. *arXiv preprint arXiv:2604.12648* (2026).

Fan Zhang, Lele Yuan, Wenchang Zhang, Mingli Zhang, and Hua Wang. 2026f. Multi-scale temporal correlation multi-dimensional decomposition network for time series analysis. *Pattern Recognition* 175 (2026), 113140. doi:10.1016/j.patcog.2026.113140

Hu Zhang, Zhien Dai, Zhaohui Tang, and Yongfang Xie. 2026a. MDMLP-EIA: Multi-domain Dynamic MLPs with Energy Invariant Attention for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 28221–28229.

Wenchang Zhang, Hua Wang, and Fan Zhang. 2024. Skip-Timeformer: Skip-Time Interaction Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 5499–5507. doi:10.24963/ijcai.2024/608

Yunhao Zhang, Wenyao Hu, Jiale Zheng, Lujia Pan, and Junchi Yan. 2026d. MMPD: Diverse Time Series Forecasting via Multi-Mode Patch Diffusion Loss. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=NEUgHT8dvH

Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*.

Zhiwei Zhang, Xinyi Du, Xuanchi Guo, Weihao Wang, and Wenjuan Han. 2025. EMAformer: Enhancing Transformer through Embedding Armor for Time Series Forecasting. *arXiv preprint arXiv:2511.08396* (2025).

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021a. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021b. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.

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

A Related Work

Time series forecasting has broad applications in energy management, weather prediction, and transportation (Liu et al., 2026a; Ma et al., 2025a). These applications require models to capture heterogeneous temporal patterns and dependencies among variables or locations (Wang et al., 2026a; Huang et al., 2023; Ma et al., 2025b). Practical forecasting settings also involve expanding traffic networks (Wang et al., 2023c,b) and distribution shifts across time or space (Wang et al., 2024a; Ma et al., 2025c). These diverse demands motivate forecasting methods that combine accurate temporal modeling with adaptability and computational efficiency.

Complementary studies investigate temporal decomposition and dependencies across offsets or scales (Fan et al., 2026; Zhang et al., 2026b,f), and disentangle persistent from time-varying components (Xia et al., 2025). These directions motivate explicit temporal modeling. Pattern-balanced pretraining corpora also emphasize coverage of heterogeneous temporal patterns (Shao et al., 2025).

A.1 Mixture-of-Experts Models and Routing Strategies for Time Series Forecasting

In recent years, an increasing number of studies on time series forecasting have adopted mixture-of-experts (MoE) mechanisms. TFPS (Sun et al., 2025) introduces pattern-specific experts by identifying evolving temporal patterns in different data segments under segment-level distribution shifts. Moirai-MoE (Liu et al., 2024c) incorporates sparse MoE into time-series foundation models, enabling token-level specialization for different temporal patterns. MoHETS (Ortigossa et al., 2026) further demonstrates the effectiveness of heterogeneous experts by combining convolutional and Fourier-based expert modules for long-term multivariate time series forecasting. These studies confirm that expert specialization is a promising direction for modeling heterogeneous temporal dynamics.

The main contributions of our work are threefold:

First, we focus on improving the forecasting decoder. Specifically, DecoderTS treats forecast generation itself as a structured decoding process. It decomposes the input into trend and seasonal components, assigns them to architecturally distinct decoding experts, and performs variable-level expert aggregation at the output stage. Our emphasis is on coordinating forecast generators with different temporal inductive biases through a shared, constrained routing mechanism.

Second, we develop a new routing mechanism that integrates multi-granularity information. This strategy is based on optimal-transport routing and is specifically designed for time-series decoding. Instead of using optimal transport as a generic token allocator, we route variable-level latent states to different temporal experts through explicit marginal constraints, thereby encouraging balanced expert utilization while preserving variable-specific decoding preferences.

Third, we design a lightweight spatiotemporal encoder to efficiently capture complex temporal dynamics. The decoder has linear complexity, and extensive comparisons with more than 30 baselines on 14 real-world datasets from multiple domains demonstrate that DecoderTS achieves competitive performance with high computational efficiency.

B Mathematical Proof

In this section, we derive the formula $\mathbf{\Pi}^* = \mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v}) \in \mathbb{R}^{N\times K}$.

Let $\mathbf{\Pi}^*$ denote the unique optimizer of the entropy-regularized optimal transport problem

$$\min_{\mathbf{\Pi}\in\mathcal{U}(\mathbf{r},\mathbf{c})} \left(\langle \mathbf{\Pi}, \mathbf{C}\rangle_F + \epsilon \sum_{i=1}^{N}\sum_{j=1}^{K} \Pi_{ij}\log\Pi_{ij} \right), \tag{15}$$

where $\epsilon > 0$ and

$$\mathcal{U}(\mathbf{r},\mathbf{c}) = \left\{ \mathbf{\Pi} \in \mathbb{R}_+^{N\times K} \;\middle|\; \mathbf{\Pi}\mathbf{1}_K = \mathbf{r},\ \mathbf{\Pi}^\top \mathbf{1}_N = \mathbf{c} \right\}. \tag{16}$$

Then there exist vectors $\mathbf{u} \in \mathbb{R}_{++}^N$ and $\mathbf{v} \in \mathbb{R}_{++}^K$ such that

$$\Pi_{ij}^* = u_i K_{ij} v_j, \tag{17}$$

where

$$K_{ij} = \exp\left(-\frac{C_{ij}}{\epsilon}\right). \tag{18}$$

Algorithm 1 Forward Pass of the DecoderTS Model

Input: Historical multivariate time series $\mathbf{X} \in \mathbb{R}^{L \times N}$; number of experts K; entropy coefficient ϵ; Sinkhorn iterations T_{sinkhorn}
Output: Predicted future sequence $\mathbf{Y}_{pred} \in \mathbb{R}^{P \times N}$
1. Preprocessing & Temporal Prior Embedding

$\mathbf{X}_{norm} \leftarrow \text{RevIN}(\mathbf{X})$
$\mathbf{X}_{var} \leftarrow \mathbf{X}_{norm}^{\top} \in \mathbb{R}^{N \times L}$
$\mathbf{Z}^{(0)} \leftarrow \text{MLP}(\mathbf{X}_{var}) + \mathbf{E}_{time} + \mathbf{E}_{var}$ // $\mathbf{E}_{time}$: temporal prior embedding; $\mathbf{E}_{var}$: learnable variable embedding
2. Spatiotemporal Mixer Encoder

for $m = 1$ to M **do**
  $\tilde{\mathbf{Z}}^{(m)} \leftarrow \mathbf{Z}^{(m-1)} + \text{MLP}_f\left(\text{BatchNorm}(\mathbf{Z}^{(m-1)})\right)$
  $\mathbf{Z}^{(m)} \leftarrow \tilde{\mathbf{Z}}^{(m)} + \left(\text{MLP}_c\left(\text{BatchNorm}\left((\tilde{\mathbf{Z}}^{(m)})^{\top}\right)\right)\right)^{\top}$
end for
$\mathbf{X}_{enc} \leftarrow \mathbf{Z}^{(M)} \in \mathbb{R}^{N \times d}$
3. Multiscale Decoupling & Heterogeneous Experts

$(\overline{\mathbf{X}}, \widetilde{\mathbf{X}}) \leftarrow \text{SeriesDecomp}(\mathbf{X}_{var})$ // $\overline{\mathbf{X}}$: trend component; $\widetilde{\mathbf{X}}$: seasonal/local component
$\mathbf{Y}_{trend} \leftarrow \mathcal{E}_1(\overline{\mathbf{X}}) = \overline{\mathbf{X}}\mathbf{W}_t + \mathbf{b}_t$ // Trend expert
$\mathbf{Y}_{periodic} \leftarrow \mathcal{E}_2(\widetilde{\mathbf{X}}) = \mathbf{W}_p\left(\text{GELU}\left(\text{Conv}_{\times 2}\left(\text{GELU}\left(\text{Conv}_{\times 1}(\widetilde{\mathbf{X}})\right)\right)\right)\right) + \mathbf{b}_p$ // Periodic/local expert
$\mathbf{Y}_{global} \leftarrow \mathcal{E}_3(\mathbf{X}_{enc}) = \text{MLP}_{global}(\mathbf{X}_{enc})$ // Latent global expert
$\mathcal{Y}_{experts} \leftarrow \{\mathbf{Y}_{trend}, \mathbf{Y}_{periodic}, \mathbf{Y}_{global}\}$
4. Optimal Transport-Based Latent Routing

$\mathbf{Z}_{route} \leftarrow \text{Linear}_{route}(\mathbf{X}_{enc}) \in \mathbb{R}^{N \times K}$
$\mathbf{C} \leftarrow -\mathbf{Z}_{route}$
Initialize OT marginals: $\mathbf{r} \leftarrow \frac{1}{N}\mathbf{1}_N \in \mathbb{R}^N$, $\mathbf{c} \leftarrow \frac{1}{K}\mathbf{1}_K \in \mathbb{R}^K$ // $\mathbf{r}$: variate marginal; $\mathbf{c}$: expert capacity marginal
$\mathbf{K}_{ot} \leftarrow \exp(-\mathbf{C}/\epsilon)$
Initialize scaling vectors $\mathbf{u}^{(0)} \leftarrow \mathbf{1}_N$, $\mathbf{v}^{(0)} \leftarrow \mathbf{1}_K$
for $t = 0$ to $T_{\text{sinkhorn}} - 1$ **do**
  $\mathbf{u}^{(t+1)} \leftarrow \mathbf{r} \oslash \left(\mathbf{K}_{ot}\mathbf{v}^{(t)}\right)$
  $\mathbf{v}^{(t+1)} \leftarrow \mathbf{c} \oslash \left(\mathbf{K}_{ot}^{\top}\mathbf{u}^{(t+1)}\right)$
end for
$\mathbf{\Pi}^* \leftarrow \text{diag}(\mathbf{u})\mathbf{K}_{ot}\text{diag}(\mathbf{v}) \in \mathbb{R}^{N \times K}$ // Joint OT coupling
$\mathbf{A} \leftarrow \text{diag}(\mathbf{r})^{-1}\mathbf{\Pi}^*$ // Convert the joint coupling into row-stochastic routing weights
$\hat{\mathbf{Y}} \leftarrow \sum_{e=1}^{K} \mathbf{A}_{:,e} \odot \mathbf{Y}_e$ // $\mathbf{A}$ is used for expert prediction aggregation
5. Final Refinement & Output

$\widetilde{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}} + \alpha \cdot \text{Conv1D}_{1\times 1}\left(\text{GELU}\left(\text{Conv1D}_{1\times 1}(\hat{\mathbf{Y}})\right)\right)$
$\mathbf{Y}_{pred} \leftarrow \text{RevIN}^{-1}(\widetilde{\mathbf{Y}})^{\top}$
return $\mathbf{Y}_{pred}$

Equivalently, the optimal transport plan admits the matrix form

$$\mathbf{\Pi}^* = \mathrm{Diag}(\mathbf{u})\,\mathbf{K}\,\mathrm{Diag}(\mathbf{v}). \tag{19}$$

Proof. Since $\epsilon > 0$ and the marginal distributions $\mathbf{r}$ and $\mathbf{c}$ are strictly positive, the optimizer of the negative-entropy-regularized OT problem lies in the interior of the transportation polytope, i.e., $\Pi^*_{ij} > 0$ for all (i, j). Intuitively, the derivative of the negative entropy term satisfies

$$\frac{d}{d\Pi_{ij}} \left(\Pi_{ij} \log \Pi_{ij}\right) = \log \Pi_{ij} + 1 \to -\infty \quad \text{as } \Pi_{ij} \to 0^+. \tag{20}$$

Thus, a boundary solution with $\Pi_{ij} = 0$ cannot satisfy the first-order optimality condition. Equivalently, the non-negativity constraints are inactive at the optimum, and the corresponding KKT multipliers are zero. The Lagrangian thus contains only marginal equality constraints:

$$\begin{aligned} \mathcal{J}(\mathbf{\Pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = &\sum_{i=1}^{N}\sum_{j=1}^{K} \Pi_{ij} C_{ij} + \epsilon \sum_{i=1}^{N}\sum_{j=1}^{K} \Pi_{ij} \log \Pi_{ij} \\ &+ \sum_{i=1}^{N} \alpha_i \left(r_i - \sum_{j=1}^{K} \Pi_{ij}\right) + \sum_{j=1}^{K} \beta_j \left(c_j - \sum_{i=1}^{N} \Pi_{ij}\right). \end{aligned} \tag{21}$$

Stationarity with respect to Π_{ij} gives

$$\frac{\partial \mathcal{J}}{\partial \Pi_{ij}} = C_{ij} + \epsilon(\log \Pi_{ij} + 1) - \alpha_i - \beta_j = 0. \tag{22}$$

Rearranging yields

$$\log \Pi^*_{ij} = \frac{\alpha_i + \beta_j - C_{ij}}{\epsilon} - 1. \tag{23}$$

Exponentiating both sides gives

$$\Pi^*_{ij} = \exp\left(\frac{\alpha_i}{\epsilon} - 1\right) \exp\left(-\frac{C_{ij}}{\epsilon}\right) \exp\left(\frac{\beta_j}{\epsilon}\right). \tag{24}$$

Define

$$u_i := \exp\left(\frac{\alpha_i}{\epsilon} - 1\right), \qquad v_j := \exp\left(\frac{\beta_j}{\epsilon}\right), \qquad K_{ij} := \exp\left(-\frac{C_{ij}}{\epsilon}\right). \tag{25}$$

Then

$$\Pi^*_{ij} = u_i K_{ij} v_j, \tag{26}$$

which proves that

$$\mathbf{\Pi}^* = \mathrm{Diag}(\mathbf{u})\,\mathbf{K}\,\mathrm{Diag}(\mathbf{v}). \tag{27}$$

This completes the proof. □

C Additional Experiments

C.1 Datasets

ETT Dataset: The ETTh1, ETTh2, ETTm1, and ETTm2 datasets were introduced with Informer (Zhou et al., 2021b). They contain oil temperature and electricity load measurements collected at an electricity transformer station. These datasets cover the period from July 2016 to July 2018 and vary mainly in sampling frequency. Specifically, ETTh1 and ETTh2 were collected at an hourly resolution, with each dataset comprising 17,420 time steps and 7 variables. By contrast, ETTm1 and ETTm2 were recorded every 15 minutes, resulting in a substantially larger scale of 69,680 time steps while preserving the same 7-channel structure. These standard long-term forecasting benchmarks capture short-term fluctuations and long-term patterns in electricity usage and ambient conditions.

Exchange Dataset: The Exchange-Rate dataset spans the period from 1990 to 2016 and contains 7,588 daily observations of exchange rates for eight major foreign currencies, including those of Australia, the United Kingdom, Canada, Switzerland, China, Japan, New Zealand, and Singapore. By

covering multiple representative currencies over a long time horizon, the dataset reflects both broad global economic trends and short-term market fluctuations.

Electricity Dataset: The Electricity dataset covers the period from 2012 to 2014 and contains 26,304 hourly observations of electricity consumption collected from 321 residential and commercial users in Portugal. Its fine temporal resolution and diverse customer profiles support studies of consumption behavior, seasonal demand, and peak load forecasting.

Weather Dataset: The Weather dataset is a multivariate meteorological benchmark for long-term forecasting. It is typically based on the Jena Climate dataset, which was collected at the weather station of the Max Planck Institute for Biogeochemistry in Jena, Germany. The dataset contains multiple atmospheric variables, such as temperature, pressure, humidity, vapor-related measurements, and wind information, recorded at 10-minute intervals over several years.

ILI Dataset: The ILI dataset spans the period from 2002 to 2021 and contains 966 weekly time steps with 7 variables, recording the ratio of influenza-like illness patients to total patient visits in the United States as reported by the CDC. Its seasonality, epidemic fluctuations, and complex temporal dependencies make it a common benchmark for public-health forecasting.

ZafNoo, AQWan, and CzeLan Datasets: The ZafNoo, AQWan, and CzeLan datasets are included in the TFB benchmark (Qiu et al., 2024) as representative multivariate time series from natural and environmental domains. Among them, ZafNoo and CzeLan are derived from the Sapflux data project and contain sap flow measurements together with related environmental variables. Both datasets are sampled at 30-minute intervals and comprise 11 variables, with sequence lengths of 19,225 and 19,934 time steps, respectively. In contrast, AQWan is an air quality dataset collected from a monitoring station over a period of four years. It contains 35,064 hourly observations across 11 variables. Owing to their diverse temporal patterns and domain characteristics, these datasets provide challenging benchmarks for evaluating multivariate long-term forecasting models.

ST-EVCDP Dataset: The ST-EVCDP dataset (Mosalli et al., 2025) is a real-world spatiotemporal benchmark designed for electric vehicle charging demand forecasting. The data were collected from public EV charging piles in Shenzhen, China, and cover 18,061 charging piles over a 30-day period from June 19 to July 18, 2022. The data are recorded at hourly intervals. For spatial modeling, the dataset is organized as a graph consisting of 13 nodes.

Table 3: Detailed information about the datasets.

Attribute	ETTh1	ETTh2	ETTm1	ETTm2	Exchange	Electricity	Weather	ILI	Mobile traffic	ZafNoo	AQWan	CzeLan	Air Quality	ST-EVCDP
Timesteps	14,400	14,400	57,600	57,600	7,588	52,560	34,272	966	4,320	19,225	35,064	19,934	11,688	4,727,520
Split	6:2:2	6:2:2	6:2:2	6:2:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2
Frequency	1 hour	1 hour	15 mins	15 mins	1 day	10 mins	10 mins	1 week	30 mins	30 mins	1 hour	30 mins	1 hour	1 hour
Domain	Electricity	Electricity	Electricity	Electricity	Economic	Energy	Transportation	Health	Nature	Nature	Environment	Nature	Environment	Transportation

C.2 Experimental Performance Analysis

From Tables 6 and 7, we observe that DecoderTS achieves the best or second-best performance on most datasets and forecasting horizons, demonstrating strong accuracy and robustness. In particular, on ETTm1, Weather, and Electricity, DecoderTS achieves the best MSE and MAE in most prediction lengths, suggesting that the proposed dynamic expert decoder is more effective at capturing diverse temporal patterns, including trends, periodicity, and local fluctuations. Among the baselines, TimeSAF shows strong performance due to its ability to model multiscale temporal patterns and dynamic dependencies, especially under longer forecasting horizons. PatchMLP benefits from patch-wise modeling, which improves the extraction of local temporal structures and yields competitive results for medium- and long-term forecasting. Crossformer and FilterNet also perform well on several datasets, likely because they enhance representation learning through cross-variable interactions and frequency-domain or filtering mechanisms, respectively. In addition, FITS and FreTS remain competitive on some benchmarks, indicating the effectiveness of frequency-domain modeling for time series with pronounced periodicity. In contrast, DecoderTS further improves forecasting performance by introducing heterogeneous expert decoding and optimal transport-based dynamic routing, which enable finer-grained pattern decomposition and adaptive expert assignment. These results highlight the importance of decoder design for time series forecasting.

Table 4: Multivariate time series forecasting results with a fixed look-back length of $L = 96$. The abbreviation "*." denotes "former" in model names, and "Avg." denotes the average across forecasting horizons. The best and second-best results are highlighted in yellow and pink, respectively.

Model		DecoderTS (Ours)		Phase*. (2026)		MMPD (2026)		MixLinear (2026)		DistDF (2026)		interPDN (2026)		TimeFilter (2025)		WPMixer (2025)		iTrans*. (2024)		TimesNet (2023)		PatchTST (2023)		TimeMixer (2023)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.366	0.382	0.383	0.386	0.376	0.393	0.389	0.417	0.373	0.393	0.375	0.396	0.390	0.401	0.386	0.397	0.386	0.405	0.384	0.402	0.414	0.419	0.378	0.400	0.386	0.400
	192	0.421	0.413	0.444	0.418	0.425	0.424	0.434	0.446	0.428	0.425	0.429	0.416	0.445	0.435	0.436	0.431	0.441	0.436	0.436	0.429	0.460	0.445	0.432	0.429	0.437	0.432
	336	0.459	0.432	0.474	0.436	0.467	0.448	0.489	0.464	0.472	0.449	0.472	0.449	0.487	0.451	0.487	0.451	0.487	0.458	0.491	0.469	0.501	0.466	0.481	0.451	0.481	0.459
	720	0.460	0.448	0.469	0.450	0.472	0.461	0.495	0.487	0.453	0.453	0.462	0.451	0.478	0.467	0.511	0.485	0.503	0.491	0.521	0.500	0.500	0.488	0.493	0.477	0.519	0.516
	Avg	0.427	0.419	0.442	0.422	0.435	0.431	0.452	0.454	0.432	0.429	0.434	0.428	0.450	0.439	0.455	0.441	0.454	0.448	0.458	0.450	0.469	0.455	0.445	0.439	0.456	0.452
ETTh2	96	0.281	0.327	0.293	0.337	0.322	0.364	0.337	0.393	0.287	0.336	0.310	0.345	0.291	0.340	0.299	0.350	0.297	0.349	0.340	0.374	0.302	0.348	0.284	0.331	0.333	0.387
	192	0.362	0.381	0.380	0.388	0.416	0.422	0.431	0.446	0.373	0.388	0.367	0.383	0.363	0.397	0.371	0.395	0.380	0.400	0.402	0.414	0.388	0.400	0.364	0.382	0.477	0.476
	336	0.406	0.418	0.426	0.424	0.436	0.442	0.475	0.460	0.408	0.421	0.413	0.420	0.408	0.427	0.426	0.433	0.428	0.432	0.452	0.452	0.426	0.433	0.406	0.416	0.594	0.541
	720	0.412	0.432	0.419	0.437	0.440	0.454	0.467	0.487	0.416	0.435	0.417	0.434	0.458	0.461	0.433	0.449	0.427	0.445	0.462	0.468	0.431	0.446	0.423	0.436	0.831	0.657
	Avg	0.365	0.390	0.379	0.397	0.404	0.421	0.427	0.447	0.371	0.396	0.377	0.396	0.380	0.406	0.382	0.407	0.383	0.407	0.414	0.427	0.387	0.407	0.369	0.391	0.559	0.515
ETTm1	96	0.297	0.329	0.326	0.344	0.313	0.350	0.374	0.419	0.316	0.357	0.324	0.347	0.319	0.357	0.329	0.364	0.334	0.368	0.338	0.375	0.329	0.367	0.315	0.343	0.345	0.372
	192	0.350	0.359	0.398	0.376	0.358	0.376	0.403	0.429	0.358	0.380	0.367	0.383	0.355	0.379	0.383	0.395	0.377	0.391	0.374	0.387	0.367	0.385	0.358	0.380	0.380	0.389
	336	0.382	0.380	0.425	0.401	0.400	0.404	0.470	0.443	0.392	0.404	0.409	0.417	0.388	0.405	0.400	0.408	0.426	0.420	0.410	0.411	0.399	0.410	0.385	0.400	0.413	0.413
	720	0.445	0.420	0.436	0.424	0.450	0.438	0.530	0.477	0.448	0.437	0.461	0.434	0.448	0.437	0.487	0.453	0.491	0.459	0.478	0.450	0.454	0.439	0.455	0.443	0.474	0.453
	Avg	0.368	0.372	0.396	0.386	0.380	0.392	0.444	0.442	0.378	0.394	0.390	0.395	0.376	0.395	0.400	0.405	0.407	0.410	0.400	0.406	0.387	0.400	0.378	0.392	0.403	0.407
ETTm2	96	0.164	0.242	0.179	0.258	0.175	0.260	0.207	0.326	0.174	0.256	0.173	0.250	0.169	0.255	0.175	0.257	0.180	0.264	0.187	0.267	0.175	0.259	0.178	0.260	0.193	0.292
	192	0.227	0.285	0.233	0.296	0.242	0.305	0.311	0.386	0.239	0.298	0.237	0.293	0.235	0.299	0.238	0.297	0.250	0.309	0.249	0.309	0.241	0.302	0.246	0.306	0.284	0.362
	336	0.287	0.323	0.310	0.340	0.308	0.349	0.417	0.462	0.300	0.338	0.292	0.332	0.293	0.336	0.305	0.342	0.311	0.348	0.321	0.351	0.305	0.343	0.296	0.339	0.369	0.427
	720	0.385	0.383	0.392	0.398	0.403	0.403	0.583	0.596	0.397	0.394	0.394	0.391	0.390	0.393	0.392	0.392	0.412	0.407	0.408	0.403	0.402	0.400	0.396	0.396	0.554	0.522
	Avg	0.266	0.308	0.279	0.323	0.282	0.329	0.380	0.443	0.277	0.321	0.274	0.317	0.272	0.321	0.278	0.322	0.288	0.332	0.291	0.333	0.281	0.326	0.279	0.325	0.350	0.401
Electricity	96	0.141	0.235	0.154	0.246	0.161	0.457	0.214	0.314	0.167	0.265	0.845	0.761	0.139	0.237	0.165	0.259	0.148	0.240	0.168	0.272	0.181	0.270	0.156	0.248	0.197	0.282
	192	0.157	0.249	0.167	0.263	0.180	0.277	0.210	0.313	0.179	0.275	0.166	0.266	0.158	0.253	0.177	0.266	0.162	0.253	0.184	0.289	0.188	0.274	0.170	0.259	0.196	0.285
	336	0.172	0.267	0.183	0.283	0.194	0.295	0.237	0.320	0.203	0.287	0.183	0.279	0.175	0.272	0.196	0.286	0.178	0.269	0.198	0.300	0.204	0.293	0.186	0.276	0.209	0.301
	720	0.200	0.290	0.221	0.309	0.233	0.324	0.274	0.350	0.212	0.302	0.236	0.314	0.202	0.299	0.238	0.327	0.225	0.317	0.220	0.320	0.246	0.324	0.228	0.312	0.245	0.333
	Avg	0.168	0.260	0.181	0.275	0.192	0.338	0.234	0.324	0.190	0.282	0.184	0.278	0.169	0.265	0.194	0.285	0.178	0.270	0.193	0.295	0.205	0.290	0.185	0.274	0.212	0.300
Weather	96	0.143	0.179	0.172	0.204	0.183	0.222	0.213	0.286	0.164	0.209	0.163	0.197	0.156	0.202	0.169	0.215	0.174	0.214	0.172	0.220	0.177	0.210	0.163	0.210	0.196	0.255
	192	0.195	0.232	0.219	0.246	0.232	0.267	0.252	0.326	0.212	0.252	0.219	0.247	0.204	0.247	0.216	0.257	0.221	0.254	0.219	0.261	0.225	0.250	0.208	0.251	0.237	0.296
	336	0.256	0.278	0.276	0.288	0.294	0.313	0.320	0.361	0.270	0.295	0.275	0.288	0.261	0.290	0.272	0.298	0.278	0.296	0.280	0.306	0.278	0.290	0.268	0.295	0.283	0.335
	720	0.339	0.332	0.362	0.346	0.344	0.337	0.386	0.401	0.348	0.345	0.345	0.336	0.345	0.342	0.352	0.348	0.358	0.349	0.365	0.359	0.354	0.340	0.346	0.348	0.345	0.381
	Avg	0.233	0.255	0.257	0.271	0.263	0.284	0.293	0.344	0.248	0.275	0.250	0.267	0.242	0.270	0.252	0.280	0.258	0.278	0.259	0.287	0.259	0.273	0.246	0.276	0.265	0.317
Exchange	96	0.081	0.198	0.086	0.205	0.087	0.205	0.095	0.243	0.089	0.221	0.085	0.203	0.112	0.246	0.086	0.206	0.107	0.234	0.088	0.205	0.092	0.218	0.085	0.218	0.088	0.218
	192	0.172	0.296	0.196	0.312	0.199	0.301	0.189	0.347	0.203	0.303	0.177	0.298	0.237	0.362	0.177	0.299	0.226	0.344	0.176	0.299	0.181	0.317	0.175	0.315	0.176	0.315
	336	0.319	0.411	0.356	0.432	0.337	0.440	0.355	0.456	0.376	0.515	0.341	0.421	0.387	0.470	0.331	0.417	0.367	0.448	0.320	0.413	0.324	0.431	0.323	0.427	0.321	0.427
	720	0.794	0.670	0.935	0.735	0.865	0.714	0.939	0.730	0.883	0.726	0.870	0.703	1.008	0.786	0.847	0.691	0.964	0.746	0.901	0.714	0.856	0.693	0.851	0.695	0.839	0.695
	Avg	0.341	0.394	0.393	0.421	0.372	0.415	0.395	0.444	0.387	0.441	0.368	0.406	0.436	0.466	0.360	0.403	0.416	0.443	0.371	0.408	0.364	0.415	0.359	0.414	0.356	0.414
AQWan	96	0.787	0.463	0.803	0.479	0.831	0.486	0.855	0.553	0.912	0.550	0.797	0.470	0.779	0.474	0.787	0.481	0.828	0.496	0.846	0.510	0.803	0.489	0.809	0.490	0.784	0.510
	192	0.832	0.484	0.878	0.506	0.875	0.516	0.899	0.602	0.963	0.563	0.863	0.495	0.851	0.506	0.858	0.505	0.885	0.514	0.841	0.508	0.865	0.513	0.877	0.515	0.843	0.535
	336	0.853	0.495	0.908	0.524	0.889	0.524	0.967	0.586	0.967	0.570	0.885	0.507	0.872	0.514	0.880	0.517	0.912	0.532	0.853	0.515	0.905	0.526	0.894	0.524	0.865	0.547
	720	0.904	0.513	0.951	0.546	0.957	0.539	1.065	0.592	0.982	0.736	0.947	0.531	0.936	0.539	0.937	0.539	0.957	0.546	0.910	0.533	0.950	0.546	0.951	0.547	0.932	0.563
	Avg	0.844	0.489	0.885	0.513	0.888	0.516	0.947	0.583	0.956	0.604	0.873	0.500	0.860	0.508	0.866	0.511	0.896	0.522	0.863	0.517	0.881	0.519	0.883	0.519	0.856	0.539
CzLan	96	0.166	0.214	0.211	0.243	0.211	0.258	0.219	0.270	0.215	0.294	0.203	0.254	0.185	0.236	0.193	0.242	0.217	0.262	0.206	0.268	0.203	0.252	0.229	0.276	0.352	0.422
	192	0.199	0.235	0.237	0.258	0.252	0.289	0.257	0.304	0.253	0.296	0.235	0.283	0.216	0.261	0.224	0.264	0.246	0.287	0.261	0.304	0.235	0.276	0.262	0.297	0.471	0.475
	336	0.225	0.260	0.266	0.282	0.277	0.308	0.298	0.337	0.280	0.317	0.275	0.314	0.250	0.288	0.260	0.297	0.282	0.313	0.261	0.304	0.265	0.299	0.297	0.328	0.597	0.527
	720	0.270	0.302	0.328	0.323	0.346	0.358	0.376	0.391	0.343	0.351	0.352	0.359	0.327	0.331	0.306	0.327	0.341	0.354	0.330	0.356	0.324	0.340	0.358	0.371	0.803	0.604
	Avg	0.215	0.252	0.260	0.276	0.271	0.303	0.288	0.326	0.273	0.315	0.266	0.302	0.244	0.279	0.246	0.282	0.271	0.304	0.265	0.308	0.257	0.292	0.286	0.318	0.556	0.507
ZaNoo	96	0.453	0.381	0.465	0.385	0.476	0.419	0.492	0.482	0.786	0.650	0.458	0.387	0.450	0.398	0.453	0.402	0.473	0.416	0.487	0.425	0.454	0.409	0.484	0.425	0.454	0.433
	192	0.519	0.421	0.557	0.437	0.558	0.462	0.556	0.524	0.845	0.680	0.536	0.425	0.521	0.439	0.532	0.442	0.558	0.461	0.557	0.469	0.525	0.448	0.567	0.470	0.521	0.465
	336	0.553	0.441	0.617	0.471	0.626	0.501	0.628	0.517	0.608	0.488	0.599	0.464	0.557	0.462	0.578	0.472	0.598	0.483	0.639	0.514	0.577	0.474	0.626	0.504	0.561	0.483
	720	0.630	0.483	0.709	0.517	0.721	0.544	0.731	0.532	0.719	0.545	0.703	0.539	0.647	0.506	0.651	0.505	0.684	0.524	0.724	0.550	0.661	0.515	0.728	0.552	0.640	0.506
	Avg	0.539	0.431	0.587	0.453	0.595	0.481	0.602	0.514	0.739	0.590	0.574	0.454	0.543	0.451	0.554	0.455	0.578	0.471	0.602	0.490	0.554	0.462	0.601	0.488	0.544	0.472

C.3 Performance of Various Models at Different Missing Rates

Figure 7 compares the performance of different models under varying missing rates. TimeMixer achieves relatively good performance, potentially because it employs decomposition techniques to capture multiscale information, and such enriched representations can improve model robustness to missing values. PhaseFormer introduces frequency-domain techniques to enhance periodicity modeling, thereby alleviating the learning difficulties caused by missing data. Our multi-expert decoder delivers strong performance across all evaluated missing rates.

C.4 Integration with Other Forecasting Models

We combine our multi-expert decoder with iTransformer and DLinear to evaluate the compatibility of our decoder with other architectures. The performance on the Air and MTraffic datasets is shown in Table 8. The experimental results demonstrate that our model can effectively replace the linear decoder, thereby improving forecasting performance.

Table 5: Multivariate time series forecasting results with a fixed look-back length of $L = 96$. The abbreviation "*." denotes "former" in model names, and "Avg." denotes the average across forecasting horizons. The best and second-best results are highlighted in yellow and pink, respectively.

Model		DecoderTS (Ours)		Phase*. (2026)		MMPD (2026)		MixLinear (2026)		DistDF (2026)		interPDN (2026)		TimeFilter (2025)		WPMixer (2025)		iTrans*. (2024)		TimesNet (2023)		PatchTST (2023)		TimeMixer (2023)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Air	12	0.635	0.600	0.773	0.659	0.742	0.654	0.969	0.794	1.022	0.799	0.737	0.647	0.769	0.648	0.735	0.663	0.740	0.657	0.914	0.766	0.712	0.640	0.830	0.712	0.832	0.715
	24	0.828	0.697	0.906	0.726	0.888	0.729	1.077	0.870	1.042	0.848	0.876	0.718	0.898	0.735	0.901	0.744	0.881	0.721	0.910	0.785	0.838	0.713	0.841	0.733	0.850	0.730
	48	0.929	0.748	1.000	0.783	0.979	0.863	1.908	0.941	0.955	0.783	0.956	0.790	1.145	0.791	0.956	0.773	0.967	0.785	0.991	0.804	0.957	0.771	0.951	0.791	0.951	0.772
	Avg	0.797	0.681	0.893	0.723	0.870	0.749	1.318	0.868	1.006	0.810	0.856	0.718	0.937	0.725	0.864	0.727	0.863	0.721	0.938	0.785	0.836	0.708	0.874	0.745	0.878	0.739
MTraffic	12	0.211	0.280	0.280	0.336	0.315	0.402	0.845	0.714	0.904	0.693	0.277	0.335	0.247	0.297	0.389	0.451	0.251	0.298	0.292	0.351	0.248	0.312	0.414	0.437	0.761	0.674
	24	0.235	0.302	0.380	0.401	0.392	0.514	1.418	0.856	1.232	0.824	0.369	0.397	0.268	0.320	0.361	0.392	0.290	0.341	0.308	0.362	0.307	0.349	0.434	0.523	0.820	0.700
	48	0.266	0.323	0.470	0.467	0.438	0.590	1.900	0.950	1.079	0.757	0.424	0.429	0.297	0.341	0.517	0.583	0.338	0.371	0.394	0.430	0.374	0.398	0.550	0.568	0.826	0.701
	Avg	0.237	0.302	0.377	0.401	0.382	0.502	1.388	0.840	1.072	0.758	0.357	0.387	0.270	0.319	0.422	0.475	0.293	0.337	0.331	0.381	0.310	0.353	0.466	0.509	0.802	0.692
ILI	12	1.485	0.714	2.168	0.966	3.367	1.332	5.442	1.612	4.051	1.494	1.719	0.783	2.332	0.982	1.967	0.967	1.515	0.728	2.223	0.951	2.523	1.131	3.460	1.035	4.061	1.463
	24	1.495	0.739	2.536	1.001	3.819	1.423	5.661	1.827	4.324	1.563	1.729	0.798	2.375	1.051	2.037	1.048	2.170	0.982	2.260	0.948	3.281	1.277	3.562	1.225	3.851	1.416
	48	1.503	0.743	2.645	1.135	4.083	1.482	6.359	1.944	4.908	1.623	1.565	0.762	2.024	0.955	2.315	1.291	1.816	0.885	2.926	1.096	5.807	1.737	3.601	1.304	3.901	1.409
	Avg	1.494	0.732	2.450	1.034	3.756	1.412	5.821	1.794	4.428	1.560	1.671	0.781	2.244	0.996	2.106	1.102	1.834	0.865	2.470	0.998	3.870	1.382	3.541	1.188	3.938	1.429
ST-EVCDP	12	0.365	0.378	0.475	0.439	0.436	0.444	0.816	0.873	0.904	0.692	0.425	0.410	0.502	0.461	0.524	0.497	0.420	0.415	0.439	0.434	0.467	0.447	0.544	0.491	0.401	0.406
	24	0.631	0.512	0.697	0.559	0.672	0.586	1.256	1.034	1.233	0.825	0.683	0.532	0.709	0.570	0.803	0.633	0.718	0.565	0.792	0.600	0.722	0.582	0.735	0.625	3.850	1.416
	48	0.912	0.643	1.021	0.682	0.974	0.668	1.345	1.120	1.079	0.757	0.995	0.669	1.012	0.703	1.136	0.814	0.995	0.679	1.067	0.709	0.899	0.681	0.941	0.696	0.957	0.686
	Avg	0.636	0.511	0.731	0.560	0.694	0.566	1.139	1.009	1.072	0.758	0.701	0.537	0.741	0.578	0.821	0.648	0.711	0.553	0.766	0.581	0.696	0.570	0.740	0.604	1.736	0.836

Table 6: Multivariate time series forecasting results with a fixed look-back length of $L = 96$. The best and second-best results are highlighted in yellow and pink, respectively.

Method		DecoderTS Ours		PMDFormer (2026)		TimeSAF (2026)		SSformer (2026)		MDMLP-EIA (2026)		FreqEvo (2026)		DUET (2025)		SimpleTM (2025)		xPatch (2025)		MultiPatch* (2025)		Amplifier (2025)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.297	0.329	0.323	0.347	0.311	0.351	0.318	0.355	0.305	0.335	0.310	0.352	0.324	0.354	0.323	0.364	0.332	0.369	0.315	0.355	0.309	0.337
	192	0.350	0.359	0.367	0.373	0.357	0.372	0.360	0.379	0.359	0.362	0.354	0.378	0.369	0.379	0.371	0.389	0.371	0.387	0.360	0.383	0.364	0.365
	336	0.382	0.380	0.398	0.396	0.390	0.399	0.393	0.402	0.391	0.386	0.386	0.390	0.404	0.402	0.411	0.413	0.399	0.407	0.398	0.410	0.396	0.388
	720	0.445	0.420	0.469	0.437	0.450	0.438	0.449	0.440	0.459	0.424	0.450	0.439	0.463	0.437	0.460	0.445	0.463	0.443	0.467	0.447	0.471	0.431
ETTm2	96	0.164	0.242	0.177	0.255	0.171	0.255	0.171	0.254	0.168	0.247	0.173	0.255	0.174	0.255	0.178	0.256	0.177	0.262	0.176	0.258	0.173	0.252
	192	0.227	0.285	0.240	0.300	0.236	0.299	0.235	0.302	0.233	0.292	0.238	0.297	0.243	0.302	0.242	0.300	0.244	0.306	0.241	0.304	0.239	0.295
	336	0.287	0.323	0.306	0.347	0.294	0.336	0.307	0.338	0.292	0.329	0.292	0.334	0.304	0.341	0.309	0.344	0.304	0.342	0.303	0.342	0.299	0.334
	720	0.385	0.383	0.401	0.397	0.388	0.393	0.403	0.395	0.391	0.388	0.392	0.393	0.399	0.397	0.401	0.399	0.404	0.399	0.412	0.408	0.391	0.389
ETTh1	96	0.366	0.382	0.380	0.395	0.369	0.391	0.373	0.393	0.374	0.383	0.377	0.398	0.377	0.393	0.379	0.396	0.383	0.401	0.382	0.401	0.384	0.389
	192	0.421	0.413	0.437	0.433	0.423	0.420	0.432	0.431	0.446	0.437	0.423	0.427	0.429	0.425	0.438	0.431	0.431	0.427	0.435	0.437	0.435	0.419
	336	0.459	0.432	0.474	0.456	0.451	0.438	0.483	0.432	0.473	0.439	0.461	0.447	0.471	0.446	0.474	0.445	0.483	0.455	0.466	0.451	0.482	0.441
	720	0.460	0.448	0.490	0.494	0.462	0.466	0.519	0.495	0.483	0.472	0.473	0.471	0.496	0.480	0.489	0.477	0.486	0.474	0.498	0.486	0.486	0.480
ETTh2	96	0.281	0.327	0.292	0.341	0.286	0.338	0.281	0.332	0.306	0.341	0.285	0.338	0.296	0.345	0.308	0.356	0.291	0.343	0.294	0.347	0.285	0.332
	192	0.362	0.381	0.382	0.401	0.370	0.390	0.367	0.385	0.381	0.395	0.361	0.389	0.368	0.389	0.383	0.404	0.372	0.392	0.374	0.395	0.364	0.386
	336	0.406	0.418	0.448	0.451	0.391	0.410	0.394	0.418	0.421	0.439	0.403	0.420	0.411	0.422	0.422	0.436	0.423	0.433	0.418	0.431	0.413	0.419
	720	0.412	0.432	0.822	0.617	0.425	0.438	0.461	0.487	0.451	0.479	0.418	0.438	0.412	0.434	0.442	0.454	0.430	0.451	0.436	0.454	0.429	0.436
Weather	96	0.143	0.179	0.164	0.202	0.157	0.200	0.181	0.224	0.156	0.195	0.163	0.208	0.163	0.202	0.175	0.219	0.178	0.221	0.163	0.212	0.166	0.204
	192	0.195	0.232	0.221	0.246	0.204	0.247	0.227	0.265	0.203	0.238	0.209	0.250	0.218	0.252	0.217	0.257	0.231	0.263	0.211	0.254	0.211	0.245
	336	0.256	0.278	0.279	0.295	0.262	0.287	0.281	0.289	0.261	0.283	0.269	0.294	0.274	0.294	0.282	0.302	0.283	0.300	0.273	0.299	0.266	0.284
	720	0.339	0.332	0.355	0.339	0.362	0.351	0.361	0.357	0.339	0.333	0.348	0.345	0.349	0.343	0.357	0.351	0.360	0.350	0.351	0.348	0.344	0.335
Electricity	96	0.141	0.235	0.179	0.256	0.143	0.237	0.145	0.247	0.181	0.257	0.141	0.237	0.145	0.233	0.176	0.258	0.191	0.277	0.174	0.260	0.151	0.263
	192	0.157	0.249	0.184	0.264	0.160	0.254	0.199	0.286	0.193	0.264	0.269	0.254	0.163	0.248	0.184	0.267	0.193	0.278	0.182	0.267	0.181	0.267
	336	0.172	0.267	0.198	0.279	0.176	0.266	0.174	0.268	0.207	0.299	0.177	0.272	0.175	0.262	0.204	0.289	0.206	0.291	0.199	0.285	0.184	0.273
	720	0.200	0.290	0.228	0.306	0.219	0.300	0.209	0.303	0.211	0.305	0.207	0.300	0.204	0.291	0.244	0.321	0.244	0.321	0.237	0.317	0.211	0.298

C.5 Forecast Visualizations

Figures 8 and 9 compare expert forecasts, their aggregation, and ground truth. MTraffic uses a 96-step history and a 12-step forecast horizon. The global expert, essentially a linear decoder, yields relatively large errors alone; multiscale integration improves accuracy.

D Discussion

D.1 Clarification of Novelty

The novelty of our work does not lie in introducing an isolated new component, but in reformulating the forecasting decoder as a structured decoding process with dynamic routing. Specifically, DecoderTS differs from conventional time series forecasting decoders in four key aspects:

Table 7: Multivariate time series forecasting results with a fixed look-back length of $L = 96$. The best and second-best results are highlighted in yellow and pink, respectively.

Method		DecoderTS Ours		TFPS (2025)		PatchMLP (2025)		TimeKAN (2025)		CycleNet (2024)		SOFTS (2024)		Crossformer (2024)		FilterNet (2024)		FITS (2024)		FreTS (2023)		CrossGNN (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.297	0.329	0.327	0.367	0.329	0.367	0.326	0.363	0.325	0.363	0.310	0.339	0.404	0.426	0.313	0.341	0.338	0.354	0.341	0.363	0.335	0.373
	192	0.350	0.359	0.374	0.395	0.377	0.394	0.359	0.384	0.366	0.382	0.367	0.369	0.450	0.451	0.369	0.368	0.385	0.377	0.386	0.387	0.372	0.390
	336	0.382	0.380	0.392	0.401	0.408	0.413	0.390	0.407	0.396	0.401	0.400	0.392	0.532	0.515	0.399	0.391	0.418	0.398	0.415	0.407	0.403	0.411
	720	0.445	0.420	0.479	0.446	0.485	0.454	0.442	0.435	0.457	0.433	0.469	0.434	0.666	0.589	0.466	0.429	0.485	0.436	0.481	0.448	0.461	0.442
ETTm2	96	0.164	0.242	0.170	0.255	0.178	0.262	0.177	0.259	0.166	0.248	0.172	0.249	0.310	0.331	0.171	0.250	0.183	0.259	0.184	0.272	0.176	0.266
	192	0.227	0.285	0.235	0.296	0.244	0.307	0.242	0.304	0.233	0.291	0.238	0.294	0.734	0.725	0.235	0.293	0.248	0.300	0.251	0.318	0.240	0.307
	336	0.287	0.323	0.297	0.335	0.307	0.343	0.304	0.344	0.293	0.330	0.301	0.334	0.750	0.735	0.294	0.330	0.308	0.350	0.309	0.354	0.304	0.345
	720	0.385	0.383	0.401	0.397	0.420	0.413	0.400	0.401	0.395	0.389	0.403	0.394	0.769	0.765	0.393	0.389	0.409	0.417	0.417	0.420	0.406	0.400
ETTh1	96	0.366	0.382	0.398	0.413	0.392	0.405	0.384	0.396	0.378	0.391	0.376	0.391	0.423	0.448	0.378	0.389	0.412	0.416	0.397	0.404	0.382	0.398
	192	0.421	0.413	0.447	0.441	0.442	0.434	0.437	0.425	0.426	0.419	0.432	0.422	0.471	0.474	0.442	0.423	0.493	0.463	0.444	0.429	0.427	0.425
	336	0.459	0.432	0.484	0.461	0.483	0.455	0.476	0.439	0.464	0.439	0.470	0.440	0.570	0.546	0.490	0.446	0.493	0.463	0.487	0.453	0.465	0.445
	720	0.460	0.448	0.488	0.476	0.502	0.484	0.468	0.470	0.461	0.460	0.464	0.461	0.653	0.621	0.492	0.463	0.534	0.513	0.557	0.537	0.472	0.468
ETTh2	96	0.281	0.327	0.327	0.367	0.309	0.359	0.306	0.353	0.285	0.335	0.289	0.336	0.287	0.366	0.280	0.328	0.298	0.345	0.352	0.395	0.309	0.359
	192	0.362	0.381	0.374	0.395	0.405	0.419	0.375	0.392	0.373	0.391	0.366	0.386	0.414	0.492	0.362	0.381	0.388	0.400	0.411	0.428	0.390	0.406
	336	0.406	0.418	0.413	0.424	0.432	0.441	0.425	0.435	0.421	0.433	0.417	0.423	0.597	0.542	0.412	0.421	0.423	0.430	0.483	0.476	0.426	0.444
	720	0.412	0.432	0.479	0.456	0.460	0.446	0.471	0.464	0.453	0.458	0.415	0.436	1.730	1.042	0.422	0.437	0.432	0.447	0.605	0.549	0.445	0.464
Weather	96	0.143	0.179	0.154	0.202	0.165	0.211	0.163	0.209	0.170	0.216	0.165	0.198	0.162	0.208	0.155	0.193	0.173	0.213	0.174	0.212	0.159	0.218
	192	0.195	0.232	0.205	0.249	0.212	0.252	0.209	0.252	0.222	0.259	0.215	0.246	0.207	0.249	0.204	0.241	0.221	0.255	0.213	0.249	0.211	0.266
	336	0.256	0.278	0.262	0.289	0.283	0.305	0.264	0.292	0.275	0.296	0.271	0.287	0.263	0.290	0.265	0.285	0.279	0.295	0.263	0.294	0.267	0.310
	720	0.339	0.332	0.344	0.342	0.357	0.352	0.340	0.343	0.349	0.345	0.351	0.339	0.338	0.340	0.354	0.342	0.359	0.342	0.338	0.352	0.352	0.362
Electricity	96	0.141	0.235	0.149	0.236	0.156	0.257	0.177	0.267	0.141	0.234	0.858	0.759	0.158	0.230	0.183	0.259	0.206	0.281	0.171	0.251	0.173	0.275
	192	0.157	0.249	0.162	0.253	0.174	0.270	0.182	0.272	0.155	0.247	0.859	0.759	0.206	0.277	0.189	0.267	0.203	0.281	0.181	0.262	0.195	0.288
	336	0.172	0.267	0.200	0.310	0.199	0.297	0.198	0.287	0.172	0.264	0.868	0.763	0.272	0.335	0.205	0.284	0.220	0.302	0.198	0.280	0.206	0.300
	720	0.200	0.290	0.220	0.320	0.242	0.331	0.239	0.321	0.210	0.296	0.896	0.773	0.398	0.418	0.246	0.317	0.275	0.354	0.241	0.318	0.231	0.335

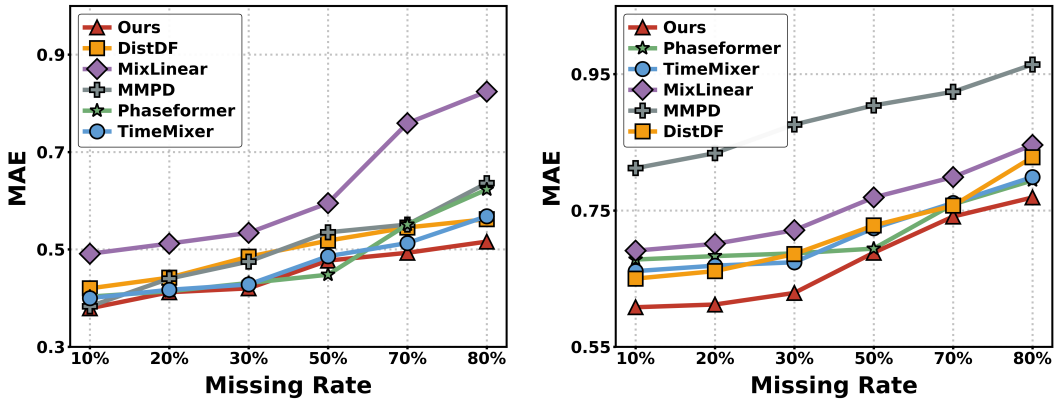


Figure 7: The predictive performance of models at different missing rates.

❶ It replaces a shared linear head with heterogeneous experts for trends, seasonal and local variations, and global nonlinear dependencies.

❷ It performs variable-wise adaptive expert allocation, allowing different variables to be decoded by different combinations of experts rather than relying on a universal decoding rule.

❸ It introduces an optimal transport (OT)-based routing mechanism with explicit marginal constraints, which ensures balanced expert utilization and effectively prevents routing collapse.

❹ It is supported by a theoretical analysis showing that the proposed OT-based routing mechanism provides a principled foundation for balanced expert allocation and the prevention of routing collapse.

Table 8: Model compatibility with different backbones.

Model	DLinear		DLinear+ours		iTransformer		iTransformer+ours	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Air	0.832	0.715	0.814	0.694	0.740	0.657	0.723	0.639
MTraffic	0.761	0.674	0.674	0.661	0.251	0.298	0.298	0.284

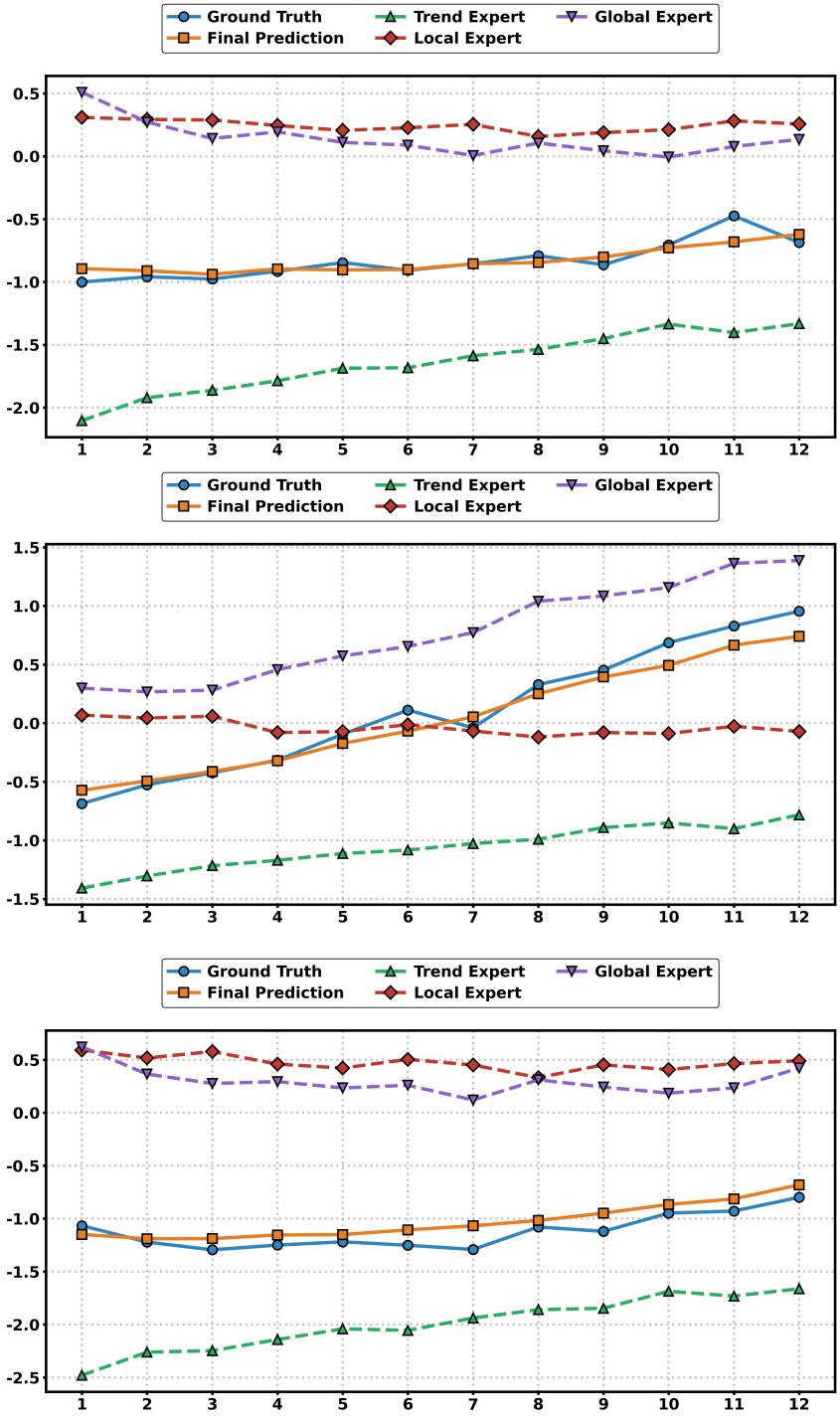


Figure 8: Forecast visualizations on the MTraffic dataset with L=96 and P=12

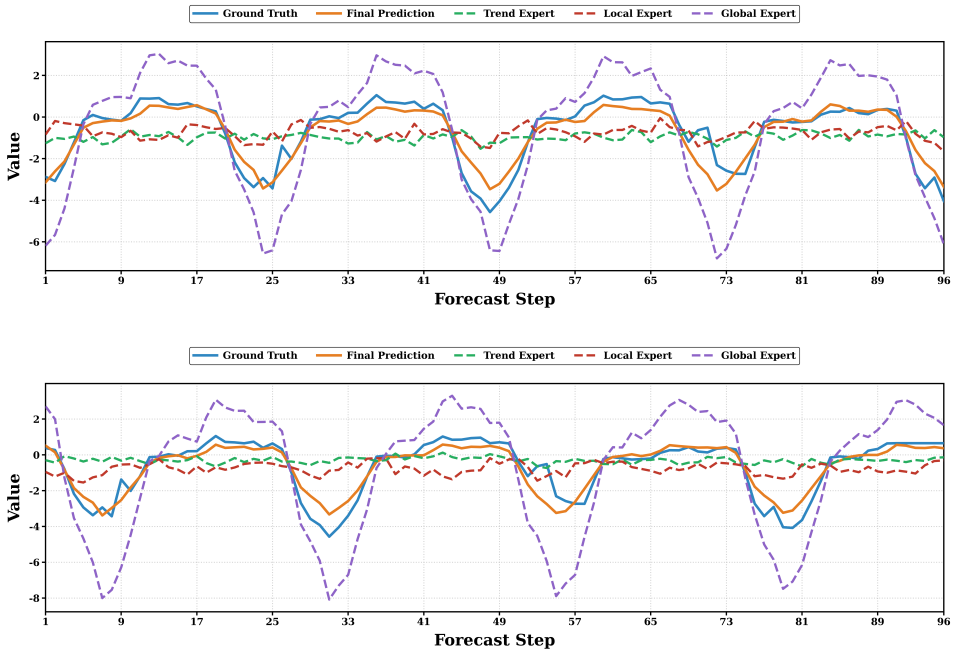


Figure 9: Forecast visualizations on the ETTh1 dataset with L=96

These components play complementary roles: series decomposition exposes pattern-specific signals, heterogeneous experts provide specialized decoding functions, and OT-based routing supports balanced expert utilization. Their interaction enables structured and adaptive forecast generation. Thus, **DecoderTS represents a decoder design paradigm for time series forecasting that integrates decomposition, expert specialization, and constrained routing.**

D.2 Future Work

Future research may extend DecoderTS to a wider range of forecasting scenarios, including irregularly sampled time series, series with missing observations, and streaming data. These settings would help assess the applicability of structured decoder design under more realistic conditions. Another direction is to explore richer expert organizations and routing mechanisms, such as hierarchical expert collaboration and transport-based assignment at multiple granularities. Such extensions may improve the flexibility and scalability of the framework when modeling complex temporal dynamics.

E Broader Impacts

This work advances time series forecasting by improving the accuracy and efficiency of multivariate prediction models. It may have positive societal impacts in applications such as energy demand forecasting, traffic management, environmental monitoring, and public-health surveillance, where more accurate forecasts can support better resource allocation and earlier decision-making. The improved efficiency of DecoderTS may also reduce computational cost and make forecasting models more accessible in resource-constrained settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the proposed DecoderTS framework, including heterogeneous expert decoding, OT-based routing, theoretical analysis, and empirical evaluation. The claims are supported by the methodology, ablation studies, robustness experiments, and efficiency analysis.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Future research directions are discussed in Appendix Section D.2.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides a theoretical analysis of the entropy-regularized OT routing mechanism and includes the derivation of the optimal transport plan in the appendix. The assumptions on positive marginals, entropy regularization, and the transportation polytope are stated in the corresponding formulation and proof.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper reports the datasets, forecasting horizons, train/validation/test splits, optimizer, hyperparameter search ranges, evaluation metrics, and architecture details. Algorithm 1 further summarizes the forward pass of the proposed model.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments are conducted on publicly available benchmark datasets, and the implementation details are described in the paper. Our code is available on GitHub.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup specifies dataset splits, look-back and prediction lengths, optimizer, learning rates, model dimensions, OT routing hyperparameters, and evaluation metrics. Additional dataset details are provided in the appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The paper reports standard forecasting metrics across multiple datasets and horizons but does not include error bars, confidence intervals, or formal statistical significance tests. This is due to the high computational cost of repeatedly running all baselines across the full benchmark suite.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the GPU used for experiments and reports memory usage, epoch time, FLOPs, MACs, and inference time in the efficiency analysis. These details help estimate the computational resources required to reproduce the experiments.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work focuses on time series forecasting methodology and uses publicly available or standard benchmark datasets. It does not involve human-subject experiments, sensitive personal data collection, or high-risk deployment.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper includes a broader impacts discussion covering potential benefits in energy, transportation, environmental monitoring, and public-health forecasting, and risks associated with inaccurate forecasts, distribution shifts, and high-stakes deployment.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models such as generative models, large language models, or scraped sensitive datasets. Therefore, additional safeguards for releases with a high risk of misuse are not applicable.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the original sources of the benchmark datasets and baseline methods. However, the current version does not explicitly list the licenses or terms of use for every dataset and code asset.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces a new model architecture but does not introduce a new dataset or benchmark asset. If code is released, documentation and instructions for reproducing the results will be provided with the released implementation.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject experiments.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human-subject research, so IRB approval or equivalent review is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not use LLMs as a methodological component. Any use of language models, if applicable, is limited to writing, editing, or formatting assistance and does not affect the scientific contribution.