OTformer: Non-Stationarity-Aware Adaptive Optimal Transport Attention for Time Series Forecasting

Zhaowei Liu[1], **Kaixiang Wang**[1], **Sheng Liu**[1], **Zengyang Zhang**[1], **Haitao Yang**[1], **Rufei Gao**[2], **Zongxing Zhao**[3], **Yao Shan**[1], **Yongchao Song**[1], **Yanwei Yu**[4], **Anzuo Jiang**[1*]
[1]Computer and Control Engineering College, Yantai University, Yantai, Shandong, China
[2]Yantai Institute of Technology, Yantai, Shandong, China
[3]School of Computer Science and Technology, Soochow University, Suzhou, China
[4]Faculty of Information Science and Engineering, Ocean University of China, Qingdao, China
{lzw, shanyao, ycsong, azjiang403}@ytu.edu.cn
{xiuxiu521, 15763803605, zhangzengyang, yanghaitao}@s.ytu.edu.cn
gaorufei@yitsd.edu.cn, zxzhaozzx@stu.suda.edu.cn, yuyanwei@ouc.edu.cn

Abstract

Time series forecasting is important in applications such as energy management, climate modeling, traffic control, and industrial monitoring. Recent variate-tokenization Transformers model cross-variate dependencies effectively, but Softmax attention still suffers from uneven cross-variate routing and fixed attention sharpness under varying non-stationarity. To address these limitations, OTformer is introduced as a non-stationarity-aware Optimal Transport framework for time series forecasting. A lightweight scorer estimates sample-wise non-stationarity from raw pre-normalization statistics and uses this signal to adapt the Sinkhorn entropy regularization coefficient ε, yielding smoother transport plans for more non-stationary inputs and sharper routing for more stationary ones. A shared phase embedding further injects cycle-aware context into variate tokens. A matched-control analysis attributes the gain to near-balanced column marginals rather than to exact double stochasticity, and shows that the adaptive coefficient matches a validation-tuned constant without a per-dataset search. Experiments on eight standard benchmarks and 32 forecasting settings show that OTformer improves upon its strong iTransformer backbone on the majority of settings, with average MSE reductions of 4.4% on ECL and 12.5% on Solar-Energy across four forecasting horizons.

1 Introduction

Time series forecasting is a core problem in modern data-driven systems, with applications in energy scheduling, climate modeling, traffic management, and industrial monitoring [10, 28]. Recent progress in long-term forecasting has explored diverse architectural biases, including efficient temporal Transformers, decomposition-based models, lightweight linear baselines, patch-based representations, and multiscale mixing architectures [39, 31, 40, 37, 18, 26, 32]. Among these developments, a particularly influential direction is the shift from time-step tokenization to variate tokenization, where each variable is treated as a token and cross-variate dependencies are modeled directly in the variate dimension [38, 13]. Despite their success, existing variate-tokenization models still rely heavily on Softmax dot-product attention for cross-variate interaction. This leads to two limitations. First, Softmax enforces unit row sums but leaves column sums unconstrained. In the variate-token setting, this means that some variables may receive disproportionately low total

*Corresponding author

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

routing mass across all queries, resulting in highly uneven cross-variate information flow. We refer to this structural effect as column-sum imbalance in cross-variate attention. Such imbalance is particularly undesirable in multivariate forecasting, where predictive signals are often distributed across multiple complementary channels. Second, Softmax applies the same routing sharpness to all samples, regardless of whether the input is relatively stationary or strongly non-stationary. This is problematic because real-world time series frequently exhibit evolving distributional properties, regime changes, and varying levels of volatility over time [7, 15, 5, 11]. Volatile inputs may benefit from smoother and more robust routing, whereas more stable inputs may benefit from sharper and more selective cross-variate matching. Although OT-inspired attention mechanisms have explored Sinkhorn-based routing or flow-conservation perspectives [23, 22, 30], they typically use a fixed routing regularization or normalization mechanism and therefore do not explicitly adapt attention smoothness to sample-wise distributional variation.

To address these issues, OTformer is introduced as a non-stationarity-aware Optimal Transport framework for time series forecasting. OTformer replaces Softmax routing with Sinkhorn-based transport attention in the variate dimension, which promotes more balanced cross-variate interaction through near-doubly-stochastic transport plans. More importantly, a lightweight NonStationarityScorer estimates the non-stationarity level of each sample from raw pre-normalization statistics and uses this signal to adapt the Sinkhorn entropy regularization strength on a per-sample basis. As a result, the attention geometry itself becomes input-adaptive: more non-stationary inputs are assigned smoother transport plans, while more stable inputs are allowed to form sharper and more selective routing patterns. In addition, a shared Phase Embedding injects cycle-aware context into variate tokens, reflecting the importance of periodic structure in long-term forecasting [11].

Experiments on eight real-world benchmarks and 32 forecasting settings show that OTformer improves upon the strong iTransformer backbone on the majority of settings, with especially clear gains on datasets exhibiting stronger distribution shift, such as ECL and Solar-Energy. Matched routing controls isolate where these gains come from: near-balanced column marginals account for the improvement, whereas a learned attention temperature or the ground cost alone does not, and the adaptive coefficient matches a validation-tuned constant without a per-dataset search. Ablation and robustness analyses further support the effectiveness of the proposed routing and phase-aware encoding.

The main contributions of this work are summarized as follows.

- A column-sum imbalance in Softmax-based cross-variate attention is identified and quantified for variate-tokenization forecasters, showing that the total routing mass received by different variates can vary substantially across queries.
- A non-stationarity-aware Optimal Transport attention mechanism is proposed, where the Sinkhorn entropy regularization is modulated on a per-sample basis by a learned non-stationarity score computed from raw pre-normalization statistics. The mechanism is analyzed against strictly matched controls, which attribute the gain to near-balanced column marginals and show that adaptation removes the need for per-dataset ε tuning.
- OTformer is instantiated as a variate-tokenization forecaster that combines adaptive transport attention with a shared phase embedding. Experiments on eight benchmarks show that OTformer improves over the iTransformer backbone on the majority of forecasting settings, and the same routing replacement transfers to a PatchTST backbone when cross-variate competition is substantial.

2 Related Work

2.1 Transformer-based Time Series Forecasting

Transformer architectures [24] have become a dominant paradigm for time series forecasting. Early Transformer-based forecasting models mainly focused on improving temporal dependency modeling while reducing the quadratic complexity of self-attention. Informer [39] introduced ProbSparse attention to reduce the computational cost of long-sequence modeling. Autoformer [31] replaced standard attention with an Auto-Correlation mechanism to aggregate sub-series patterns across temporal lags, and FEDformer [40] further exploited frequency-domain sparsity for efficient forecasting.

At the same time, simpler baselines such as DLinear [37] and TiDE [3] showed that strong forecasting performance can also arise from carefully designed lightweight architectures, prompting renewed attention to which inductive biases are truly essential in long-horizon forecasting. Recent methods further explore alternative forecasting biases, including multiscale MLP-based mixing in TimeMixer [26], lightweight frequency-domain modeling in FITS [32], MLP-mixing architectures [4], modern convolutional backbones [17], and state-space sequence models [1].

A major recent shift is the move from time-step tokenization to variate-aware modeling. PatchTST [18] adopts channel-independent patch tokenization, which is effective for capturing temporal patterns within each variate while reducing sequence length. Crossformer [38] explicitly models cross-dimension dependencies through two-stage attention, and iTransformer [13] pushes this idea further by treating each variate's entire look-back window as a single token and performing self-attention directly in the variate dimension. Following this trend, CARD [33] further emphasizes the importance of channel alignment and robust cross-variate modeling, while TimeXer [27] incorporates variate-wise attention for leveraging exogenous variables in forecasting. These models establish variate-level interaction as a powerful design principle for multivariate forecasting.

However, existing variate-tokenization models still rely on Softmax dot-product attention for cross-variate interaction. Although Softmax attention is effective at assigning query-wise importance, it does not explicitly regulate the total amount of information received by each variate across all queries. As a result, cross-variate routing may become highly imbalanced, especially when some variables dominate the attention distribution. OTformer differs from these methods by replacing unconstrained Softmax routing with transport-based routing in the variate dimension, aiming to promote more balanced cross-variate information flow.

2.2 Non-stationarity and Distribution Shift in Time Series

Non-stationarity is a pervasive challenge in real-world time series forecasting, where distributional properties evolve over time due to seasonal transitions, concept drift, and abrupt external events [15]. Existing methods typically address this issue through decomposition, normalization, or architectural redesign.

Decomposition-based methods, such as Autoformer [31] and FEDformer [40], separate input series into trend and seasonal components before further modeling, thereby reducing the burden of non-stationarity on the backbone network. Normalization-based methods provide another influential line of work. RevIN [7] applies reversible instance normalization to reduce train-test distribution mismatch, while Non-stationary Transformer [15] argues that excessive normalization may suppress useful non-stationary cues and therefore introduces de-stationary attention. Dish-TS [5] further improves distribution alignment with a two-component normalization scheme. More recently, Koopa [14] models non-stationary dynamics through Koopman predictors by separating time-variant and time-invariant components.

Beyond instance-level normalization, recent methods further revisit non-stationarity from adaptive and frequency-aware perspectives. SAN [16] applies adaptive normalization from a temporal-slice perspective, while FAN [34] extends reversible normalization to frequency components for handling dynamic trend and seasonal variation. TimeBridge [12] further argues that non-stationarity has different effects on short-term local dependency modeling and long-term cross-variate relationships. Different from these methods, OTformer does not primarily modify the input normalization or feature decomposition. Instead, it estimates non-stationarity from raw pre-normalization statistics and uses this signal to adapt the entropy regularization of transport attention, making the cross-variate routing geometry explicitly sample-adaptive.

Periodic structures also play an important role in long-term forecasting, since many real-world time series contain recurring cycles at daily, weekly, or seasonal scales. Recent work such as CycleNet [11] explicitly models periodic patterns to improve forecasting performance. Motivated by this observation, OTformer introduces a shared Phase Embedding to inject cycle-aware context into variate tokens before transport-based routing.

2.3 Optimal Transport in Deep Learning and Attention

Optimal transport (OT) provides a principled framework for comparing probability distributions through minimum-cost mass transport [25]. The introduction of entropy-regularized OT and the Sinkhorn algorithm [2] made OT practical for large-scale machine learning, and comprehensive treatments can be found in [20]. OT has since been successfully applied in generative modeling, domain adaptation, and structured representation learning, where global geometric alignment is often more suitable than pointwise matching.

OT and flow-based perspectives have also been explored in attention mechanisms. Sparse Sinkhorn Attention [23] uses differentiable sorting to induce sparse attention patterns, while Flowformer [30] reinterprets attention as information flow and emphasizes source competition and sink allocation under conservation constraints. More directly related to OTformer, Sinkformers [22] replace row-wise Softmax normalization with Sinkhorn normalization, yielding attention matrices driven toward a near-doubly-stochastic form. These methods provide geometrically grounded alternatives to standard dot-product attention, where the resulting attention matrix can better reflect global routing constraints rather than independent query-wise normalization.

However, prior OT-based attention is mainly studied in NLP or vision settings, and typically uses a fixed entropy regularization coefficient or routing normalization mechanism for all inputs. This assumption is suboptimal for time series forecasting, where different samples may exhibit very different levels of non-stationarity and therefore may require different routing smoothness. OTformer extends this line of work to time series forecasting in two important ways. First, transport-based attention is applied in the variate dimension within a variate-tokenization forecasting backbone. Second, the entropy regularization coefficient is made per-sample adaptive through a learned non-stationarity scorer computed from raw input statistics. This allows the transport plan to become smoother for more non-stationary samples and sharper for more stationary ones, aligning OT attention with the distributional characteristics of each forecasting instance.

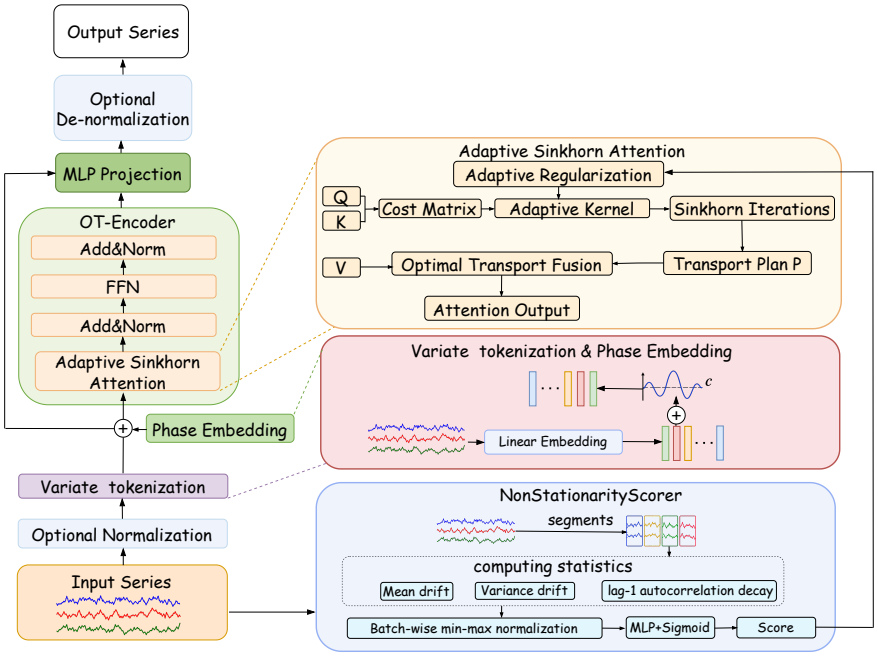


Figure 1: Overall architecture of OTformer. The model processes multivariate inputs through three stages: (i) variate tokenization with a shared phase embedding ($\oplus$); (ii) Sinkhorn-based optimal transport attention with sample-adaptive entropy regularization ε_b driven by a non-stationarity score; and (iii) a residual MLP projection head. Since attention is applied along the variate dimension and the encoder/projection weights are shared across variates, the number of learnable parameters in the encoder does not scale with the number of variates N.

3 Method

3.1 Problem Setup

Given a multivariate historical window $\mathbf{X} \in \mathbb{R}^{L \times N}$, where L is the look-back length and N is the number of variates, the goal is to predict the future horizon $\hat{\mathbf{Y}} \in \mathbb{R}^{T \times N}$. For a mini-batch of size B, the encoder input is $\mathbf{X}_{enc} \in \mathbb{R}^{B \times L \times N}$ and the forecast output is $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times T \times N}$.

OTformer is built on a variate-tokenization backbone. Instead of attending over time steps, it treats each variate as a token and models cross-variate interaction in the variate dimension. Its key idea is to replace Softmax routing with non-stationarity-aware Optimal Transport routing.

3.2 OTformer Overview

Figure 1 shows the overall architecture. Given the raw input $\mathbf{X}_{\text{enc}} \in \mathbb{R}^{B \times L \times N}$, OTformer proceeds in two parallel branches. The *scoring branch* computes a per-sample non-stationarity score from a small set of statistics on the raw, pre-normalization input,

$$\mathbf{s} \;=\; \mathrm{NSScorer}(\mathbf{X}_{\text{enc}}) \;\in\; (0,1)^B. \tag{1}$$

The *forecasting branch* applies optional reversible instance normalization [7] to obtain $\tilde{\mathbf{X}}_{\text{enc}}$, performs variate tokenization,

$$\mathbf{Z}_0 \;=\; \mathrm{Embed}(\tilde{\mathbf{X}}_{\text{enc}}) \;\in\; \mathbb{R}^{B \times N \times d}, \tag{2}$$

and adds a shared phase embedding indexed by the cycle position c of each sample,

$$\mathbf{Z} \;=\; \mathbf{Z}_0 \;+\; \mathbf{E}_\phi(c). \tag{3}$$

The phase-aware tokens are encoded by stacked OT-Encoder layers, in which the regularization ε_b of Sinkhorn attention is conditioned on $\mathbf{s}$:

$$\mathbf{H} \;=\; \mathrm{OTEncoder}(\mathbf{Z}\,;\,\mathbf{s}). \tag{4}$$

A global residual connection and an MLP projector then map the encoded tokens to the prediction horizon,

$$\hat{\mathbf{Y}} \;=\; \mathrm{Proj}(\mathbf{H} + \mathbf{Z}), \tag{5}$$

followed by de-normalization when input normalization is enabled.

3.3 Non-stationarity-aware Transport Attention

NonStationarityScorer. To adapt routing to the input distribution, we estimate the non-stationarity level of each sample *before* normalization, since instance normalization removes much of the first- and second-order distributional variation that is informative for stationarity estimation.

For each sample, the input window is partitioned into four equal segments, from which we compute three lightweight statistics: mean drift, variance drift, and lag-1 autocorrelation decay across the segments. These three scalars form a feature vector $\mathbf{r}_b$, which is min-max normalized across the mini-batch to $\tilde{\mathbf{r}}_b$ and passed through a small MLP with sigmoid activation:

$$s_b \;=\; \sigma\big(\mathrm{MLP}(\tilde{\mathbf{r}}_b)\big), \qquad b = 1, \ldots, B. \tag{6}$$

The raw statistics are deterministic for each sample; the batch normalization is the only batch-dependent step and only calibrates their relative scale before the MLP. The resulting scalar $s_b \in (0,1)$ indicates the degree of non-stationarity of sample b, with larger values corresponding to a higher estimated degree of distribution shift.

Adaptive Sinkhorn Attention. Within each encoder layer, variate tokens are projected to per-head queries, keys, and values $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B \times H \times N \times d_h}$, where H is the number of heads and d_h is the per-head dimension. We replace the Softmax routing matrix with an entropy-regularized optimal transport plan whose cost is the scaled squared Euclidean distance between queries and keys,

$$C_{ij} \;=\; \frac{\|\mathbf{q}_i - \mathbf{k}_j\|_2^2}{\sqrt{d_h}}, \tag{7}$$

where $i, j \in \{1, \ldots, N\}$ index variate tokens; $\mathbf{C}$ is computed separately for each sample and head, and we drop these indices when unambiguous.

Squared Euclidean distance is used rather than a dot-product similarity because, in a transport problem, C is a ground cost rather than a score: it must be non-negative so that the kernel $\exp(-C/\varepsilon)$ stays bounded. The squared Euclidean cost satisfies this, underlies the 2-Wasserstein distance, and gives an interpretable geometry in which closer tokens exchange more mass, whereas $-\mathbf{q}_i^\top \mathbf{k}_j$ is unbounded below and scales with feature norms, making the choice of ε depend on activation magnitude.

Computational cost. With batch size B, H heads, N variates, per-head dimension d_h, and K Sinkhorn iterations, forming the cost and applying the plan costs $O(BHN^2 d_h)$, as in Softmax attention, and the iterations add $O(KBHN^2)$. The total is $O\big(BHN^2(d_h+K)\big)$ time and $O(BHN^2)$ memory, so OT routing adds a constant factor but no new asymptotic order in N.

A central design choice of OTformer is to make the entropy regularization ε *per-sample adaptive*. Specifically, the non-stationarity score $s_b \in (0, 1)$ is linearly mapped to

$$\varepsilon_b \;=\; \varepsilon_{\min} \;+\; (\varepsilon_{\max} - \varepsilon_{\min})\, s_b, \tag{8}$$

so that more non-stationary samples (larger s_b) receive a smoother transport plan, while more stationary samples (smaller s_b) keep a sharper routing. Note that ε_b is a per-sample scalar that is broadcast over the head, query, and key dimensions of the $N \times N$ cost matrix.

Given ε_b, the transport plan is computed by K log-space Sinkhorn iterations with unit marginals $\boldsymbol{\mu} = \boldsymbol{\nu} = \mathbf{1}_N$, so that $\mathbf{P}_b$ is directly comparable to a Softmax matrix whose rows sum to one. Initializing $\log \mathbf{u}^{(0)} = \log \mathbf{v}^{(0)} = \mathbf{0} \in \mathbb{R}^N$, we iterate

$$\log u_i^{(k+1)} = \log \mu_i \;-\; \mathrm{logsumexp}_j\big(-C_{ij}/\varepsilon_b + \log v_j^{(k)}\big), \tag{9}$$

$$\log v_j^{(k+1)} = \log \nu_j \;-\; \mathrm{logsumexp}_i\big(-C_{ij}/\varepsilon_b + \log u_i^{(k+1)}\big). \tag{10}$$

The transport plan for sample b is then

$$\mathbf{P}_b \;=\; \exp\big(-\mathbf{C}/\varepsilon_b + \log \mathbf{u}^{(K)} \oplus \log \mathbf{v}^{(K)}\big) \;\in\; \mathbb{R}^{N\times N}, \tag{11}$$

where $\oplus$ denotes outer addition (broadcasting). Writing $\mathbf{P}_{b,h}$ for the plan of sample b and head h, the attention output is

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{b,h} \;=\; \mathbf{P}_{b,h}\, \mathbf{V}_{b,h}, \tag{12}$$

and the head outputs are concatenated as in standard multi-head attention. Because each iteration ends with the column update, every column of $\mathbf{P}_b$ sums exactly to one, while the row sums approach one as K grows; this is the reverse of Softmax attention, which fixes the row sums and leaves the column sums unconstrained.

3.4 Phase-aware Prediction Head

Before entering the OT-Encoder, OTformer injects a shared phase embedding indexed by the sample's cycle position. This provides periodic context, such as hour-of-day or other cycle information, while remaining shared across variates within the same sample.

After OT encoding, the model uses a global residual connection and a 3-layer MLP projector to map token representations to the prediction horizon. Compared with a shallow linear readout, this head provides stronger forecasting capacity while keeping the overall architecture simple. The final output is de-normalized when input normalization is enabled.

4 Experiments

4.1 Experimental Setup

Datasets. Experiments are conducted on eight widely used real-world benchmarks. The four ETT datasets (**ETTh1**, **ETTh2**, **ETTm1**, **ETTm2**) contain electricity transformer temperature measurements at hourly and 15-minute resolutions. **Weather** contains 21 meteorological variables recorded every 10 minutes. **ECL** records electricity consumption of 321 clients. **Traffic** contains road occupancy rates collected from 862 sensors in San Francisco [9], and **Solar-Energy** records solar power

Table 1: Average multivariate long-term forecasting results across four prediction horizons $T \in \{96, 192, 336, 720\}$. For each dataset and metric, the best and second-best results are shown in **bold** and underlined, respectively. The full comparison against an extended set of baselines is provided in Appendix Table A2.

Model	**OTformer (Ours)**		DUET (2025)		Leddam (2024)		SOFTS (2024)		iTransformer (2024)		PatchTST (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.441	**0.428**	0.443	0.436	**0.431**	0.429	0.449	0.443	0.454	0.448	0.469	0.455
ETTh2	**0.372**	**0.392**	**0.372**	0.398	0.373	0.399	0.373	0.400	0.383	0.407	0.387	0.407
ETTm1	0.388	**0.383**	0.390	0.393	**0.386**	0.397	0.399	0.405	0.401	0.408	0.387	0.400
ETTm2	**0.279**	**0.317**	0.280	0.324	0.281	0.325	0.288	0.330	0.288	0.332	0.281	0.326
ECL	0.171	**0.255**	0.172	0.259	**0.169**	0.263	0.174	0.264	0.178	0.270	0.216	0.304
Solar-Energy	**0.207**	**0.232**	0.237	0.233	0.228	0.262	0.239	0.267	0.236	0.266	0.270	0.307
Traffic	0.429	**0.261**	0.451	0.269	0.467	0.294	**0.409**	0.267	0.430	0.282	0.555	0.362
Weather	0.252	**0.268**	0.251	0.273	**0.242**	0.272	0.255	0.278	0.258	0.278	0.259	0.281
1^{st} Count	**11**		1		4		1		0		0	

production from 137 plants in Alabama [9]. Following standard practice, all datasets use chronological train/validation/test splits of 7:1:2. We evaluate prediction horizons $T \in \{96, 192, 336, 720\}$ with a fixed look-back window $L = 96$. Detailed dataset statistics are provided in Appendix Table A1.

Baselines. We compare OTformer against a set of representative multivariate forecasting models, including **iTransformer** [13] as the direct backbone baseline, recent state-of-the-art models such as **DUET** [21], **Leddam** [36], **FreTS** [35], and **SOFTS** [6], as well as established baselines including **PatchTST** [18], **TimesNet** [29], **DLinear** [37], **TiDE** [3], and **Stationary** [15]. Baseline results are taken from the original papers or reproduced under matched settings when official implementations are available.

Implementation Details. OTformer is implemented in PyTorch [19] and trained with Adam [8] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) for up to 10 epochs with early stopping (patience = 3) based on validation MSE. The initial learning rate is 10^{-4} and is decayed using a standard scheduler. Unless otherwise specified, the hidden dimension is $d = 256$, the number of encoder layers is $L_e = 2$, the number of attention heads is $H = 4$, the FFN dimension is 512, the number of Sinkhorn iterations is $K = 3$, and the adaptive regularization range is $\varepsilon_{\min} = 0.03$, $\varepsilon_{\max} = 0.30$. The default cycle length is set according to dataset granularity (e.g., 24 for hourly datasets and 96 for 15-minute ETT datasets), with additional sensitivity analysis reported in Section A.4.

4.2 Main Results

Table 1 reports horizon-averaged forecasting performance on eight benchmarks. OTformer improves over its direct backbone iTransformer on the majority of (dataset, metric) pairs, and is competitive with or better than recent strong baselines such as DUET [21], Leddam [36], and SOFTS [6]. The largest gains are observed on ECL and Solar-Energy, two datasets that exhibit clear non-stationarity and where adaptive transport routing is expected to be most useful. On Traffic, the gap to iTransformer narrows; this is consistent with the observation that Traffic dependencies are more strongly determined by sparse sensor topology than by transport-style geometric similarity in the variate space. Per-horizon results and the comparison against an extended set of baselines are provided in Appendix Table A2.

4.3 Ablation Study

To isolate the contribution of each proposed component, we evaluate four variants on ETTh1, Weather, and ECL with prediction horizon $T = 96$:

- **A1 — Full OTformer**: the complete proposed model.
- **A2 — Fixed ε**: adaptive entropy regularization is disabled by setting $\varepsilon_{\min} = \varepsilon_{\max} = 0.05$. The NonStationarityScorer is retained but no longer affects ε.

- **A3 — w/o Phase Emb**: the phase embedding is replaced by a zero tensor, removing cycle-phase information.
- **A4 — Softmax**: the iTransformer backbone with standard Softmax dot-product attention and its linear head, trained under the same setup.

Table 2: Ablation study (T=96, mean MSE / MAE over three runs). **Bold**: best result in each column.

	ETTh1		Weather		ECL	
Variant	MSE	MAE	MSE	MAE	MSE	MAE
A1 **Full OTformer**	**0.373**	**0.390**	**0.166**	**0.198**	**0.144**	**0.229**
A2 Fixed ε	0.376	0.392	0.171	0.204	0.154	0.239
A3 w/o Phase Emb	0.375	0.394	0.170	0.202	0.146	0.231
A4 Softmax	0.382	0.395	0.172	0.203	0.151	0.235

Table 2 shows that all three design choices contribute across ETTh1, Weather, and ECL. Disabling input conditioning of ε (A2) causes the largest degradation on Weather and an even more pronounced drop on ECL. Because A2 fixes $\varepsilon = 0.05$ rather than tuning it, this comparison isolates the benefit of input conditioning relative to an untuned constant, and we separate it from the tuned-constant comparison in Section 4.4. Removing the shared phase embedding (A3) yields a smaller but consistent drop across all three datasets, suggesting that cycle-aware context is complementary to transport routing rather than redundant with it. Replacing OT attention with standard Softmax (A4) is the largest structural change and consistently hurts performance, with ETTh1 MSE increasing from 0.373 to 0.382 and ECL MSE from 0.144 to 0.151. Since A4 also uses the linear head and no phase embedding, this gap reflects the full OTformer design rather than the routing rule alone; the matched controls below isolate the routing mechanism.

4.4 Mechanism Analysis and Reproducibility

What drives the gain. To test whether the improvement is due to balanced column marginals rather than to transport machinery in general, we ran strictly matched controls on ECL, where the gap to Softmax is largest. All variants share the backbone, the head, and the training protocol, and differ only in the routing rule. Table 3 reports three-seed means.

Table 3: Matched routing controls on ECL (three seeds, mean $\pm$ std). All variants share the same backbone and training protocol. Column residual is the mean ℓ_1 deviation of the per-head routing matrix column sums from 1. Bold marks the lowest MSE.

Routing variant	ECL MSE	Rel. gain	Column residual
Dot-product Softmax	0.145595 ± 0.000188	reference	1.000989
NSS-temperature Softmax	0.145491 ± 0.000254	-0.07%	–
Same-cost row-Softmax	0.151588 ± 0.000123	$+4.1\%$	–
Forced column normalization	0.144290 ± 0.000035	-0.90%	0.685255
Validation-selected fixed ε	$\mathbf{0.143715 \pm 0.000113}$	$\mathbf{-1.29\%}$	2.5×10^{-7}
Adaptive Sinkhorn (Ours)	0.143830 ± 0.000373	-1.21%	2.5×10^{-7}

Three conclusions follow. First, neither a learned attention temperature nor the ground cost alone reproduces the gain: NSS-temperature Softmax changes MSE by only 0.07%, and the same squared-Euclidean cost with row-wise Softmax is 4.1% worse than standard Softmax, so the benefit requires transport-based marginal balancing. Second, a naive column normalization already recovers most of the improvement, and the error decreases monotonically along Softmax $\rightarrow$ forced column normalization $\rightarrow$ full Sinkhorn as the column residual shrinks, which supports the column-sum-imbalance account illustrated in Figure 2. Third, the residual need not reach zero: forced column normalization already captures most of the gain, and the two Sinkhorn variants reach a comparable MSE with the same residual, so the benefit saturates once routing is approximately column-balanced. We therefore attribute the improvement to near-balanced marginals rather than to exact double stochasticity.

Adaptive ε versus a tuned constant. Table 3 also shows that adaptive ε and a fixed ε selected on validation data differ by about 0.08% in MSE on ECL, and by similarly small margins on ETTh1

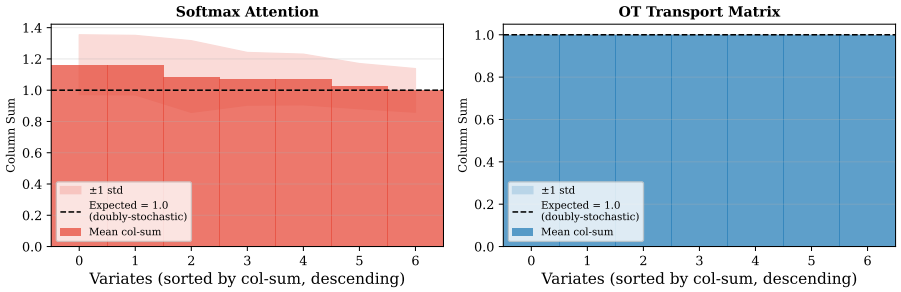


Figure 2: Column-sum statistics of cross-variate routing matrices on ETTh1 (7 variates). **Left (Softmax):** column sums vary substantially across variates, indicating uneven total received attention. **Right (OT):** column sums concentrate tightly around 1.0, consistent with the near-doubly-stochastic transport constraint.

and Weather (Appendix A.5). We therefore do not claim that sample-wise adaptation raises the attainable accuracy. Its role is to obtain input-conditioned transport plans without a per-dataset hyperparameter search: a single $(\varepsilon_{\min}, \varepsilon_{\max})$ pair performs comparably to a value that has to be tuned separately for each benchmark. At the same time, the conditioning signal is not interchangeable with noise: reversing the score-to-ε mapping at inference time raises MSE from 0.166546 to 0.175133 on Weather ($+5.16\%$) and from 0.143830 to 0.158787 on ECL ($+10.40\%$), whereas shuffling the scores only within a batch leaves MSE nearly unchanged. The aggregate, batch-level conditioning of ε is thus load-bearing, whereas fine-grained ordering of samples inside a batch is not a major source of the gain.

Reproducibility across seeds. To quantify seed sensitivity, we reran all 32 OTformer settings in an independent rerun with three different random seeds. The coefficient of variation of MSE is below 1% in 29 settings, with a median of 0.265%; at $T{=}96$, ECL gives 0.143830 ± 0.000373, Weather 0.166546 ± 0.001164, and ETTh1 0.375772 ± 0.000599. Differences smaller than these spreads should not be read as significant, which is why we report the tuned-constant comparison alongside the claims above.

Transfer to another backbone. To check that the mechanism is not specific to our implementation, we applied the same routing replacement to PatchTST under an identical training protocol (Appendix A.6). On high-dimensional ECL, the adaptive OT variant reduces MSE by 8.79% relative to PatchTST and by 2.40% relative to a structurally matched Softmax adapter. On low-dimensional ETTh1, the variant yields no further gain, indicating that the benefit appears where cross-variate competition is substantial rather than universally.

4.5 Model Analysis and Interpretability

Figure 2 illustrates the column-sum imbalance directly: across ETTh1 test batches Softmax attention yields noticeably uneven column sums, so some variates receive far less total routing mass than others, whereas OTformer's plan keeps them tightly concentrated around 1.0, consistent with the marginal-balancing account of the gain.

4.6 Robustness Analysis on ETTh1

Figure 3 reports forecasting error on ETTh1 with $T{=}96$ under additive Gaussian noise. With clean inputs the two models start at comparable error levels; as the noise grows both degrade, but OTformer degrades more slowly in absolute MSE and relative to the clean baseline. We read this as evidence that balanced transport routing avoids over-committing to a few dominant cross-variate links, so the encoder is less brittle under perturbation. The experiment uses a single dataset and a single noise model and is not a global robustness claim.

Figure 4 compares the computational profile of OTformer and iTransformer on ETTh1 with $T{=}96$. OTformer has more parameters than iTransformer, yet training time per batch is nearly unchanged, so

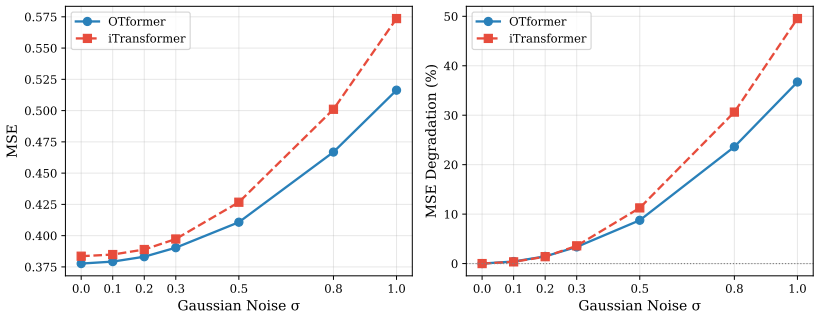


Figure 3: Forecasting error on ETTh1 with T=96 under additive Gaussian input noise ($\sigma \in \{0, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$). **Left:** absolute MSE. **Right:** relative MSE degradation normalized to the clean baseline. OTformer degrades more slowly than iTransformer as σ increases.

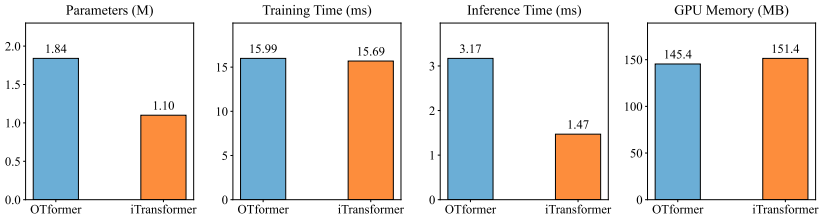


Figure 4: Efficiency comparison between OTformer and iTransformer on ETTh1 with T=96. From left to right: parameters, training time, inference time, and GPU memory. OTformer adds little training overhead and comparable memory usage, but increases inference latency due to Sinkhorn iterations.

the scorer and Sinkhorn updates add limited training overhead. Inference is slower, as expected from the added Sinkhorn iterations, while peak GPU memory remains comparable to, and slightly below, that of iTransformer. The model therefore incurs a moderate inference-time cost while remaining practical on standard benchmarks.

5 Conclusion

OTformer replaces Softmax routing with sample-adaptive Sinkhorn transport plans, promoting balanced cross-variate information flow while adapting routing geometry to input non-stationarity. On eight benchmarks and 32 settings it improves over iTransformer on the majority of settings. Matched controls attribute the gain to near-balanced column marginals; adaptive ε matches a validation-tuned constant without per-dataset search; the mechanism transfers to PatchTST when cross-variate competition is substantial.

Limitations

Several limitations remain in the current study. The non-stationarity score uses hand-crafted statistics, which may not fully capture richer distributional change; stronger end-to-end descriptors may improve adaptivity. The Sinkhorn iterations introduce moderate inference-time overhead, which may be relevant in latency-sensitive deployments. The benefit of transport-based routing may further depend on the dependency structure of the data. On datasets such as Traffic, where sparse sensor topology appears to play a stronger role, the advantage over the backbone becomes smaller, suggesting that the proposed routing mechanism is not uniformly beneficial across all multivariate forecasting scenarios. The current evaluation uses a fixed look-back length L=96 and standard benchmarks; robustness is examined only on ETTh1 under additive Gaussian noise, so behavior under substantially different context lengths or broader distribution shifts is not yet characterized.

Acknowledgments

We thank all the reviewers and the area chair for their helpful feedback, which aided us in greatly improving the paper. This work was supported in part by the National Natural Science Foundation of China under Grant 62572420, the Shandong Province Key R&D Program under Grant 2024CXGC010801, and the Yantai Science and Technology Plan Project under Grants 2025ZDCX012 and 2025ZDCX020.

References

[1] Xiuding Cai, Yaoyao Zhu, Xueyao Wang, and Yu Yao. Mambats: Improved selective state space models for long-term time series forecasting. *arXiv preprint arXiv:2405.16440*, 2024.

[2] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[3] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

[4] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 459–469, 2023.

[5] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529, 2023.

[6] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *arXiv preprint arXiv:2404.14197*, 2024.

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

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[10] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, 379(2194), 2021.

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

[12] Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. *arXiv preprint arXiv:2410.04442*, 2024.

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

[14] Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.

[15] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. 2022.

[16] Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems*, 36:14273–14292, 2023.

[17] Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.

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

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[20] Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

[21] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1185–1196, 2025.

[22] Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022.

[23] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International conference on machine learning*, pages 9438–9447. PMLR, 2020.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[25] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[26] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

[27] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

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

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

[30] Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. *arXiv preprint arXiv:2202.06258*, 2022.

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

[32] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

[33] Wang Xue, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. *arXiv preprint arXiv:2305.12095*, 2023.

[34] Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37:31350–31379, 2024.

[35] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[36] Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In *Forty-first International Conference on Machine Learning*, 2024.

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

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

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

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

A Additional Experimental Details

A.1 Dataset statistics.

Table A1 summarizes the statistics of the eight benchmark datasets used in the experiments. These datasets cover a wide range of application domains, including electricity, meteorology, transportation, and solar energy forecasting. They also vary substantially in the number of variates, from small-scale ETT datasets with 7 variables to large-scale Traffic and ECL datasets with hundreds of variables. This diversity allows the evaluation to test whether OTformer remains effective under different sampling frequencies, variable dimensions, and domain-specific temporal patterns.

Table A1: Dataset statistics. N: number of variates; $|\mathcal{D}|$: total time steps; Freq.: sampling frequency.

Dataset	N	$\lvert\mathcal{D}\rvert$	**Freq.**	**Domain**
ETTh1 / ETTh2	7	17,420	1 h	Electricity
ETTm1 / ETTm2	7	69,680	15 min	Electricity
Weather	21	52,696	10 min	Meteorology
ECL	321	26,304	1 h	Electricity
Traffic	862	17,544	1 h	Transport
Solar-Energy	137	52,560	10 min	Energy

A.2 Additional Qualitative Visualization of Adaptive OT Transport

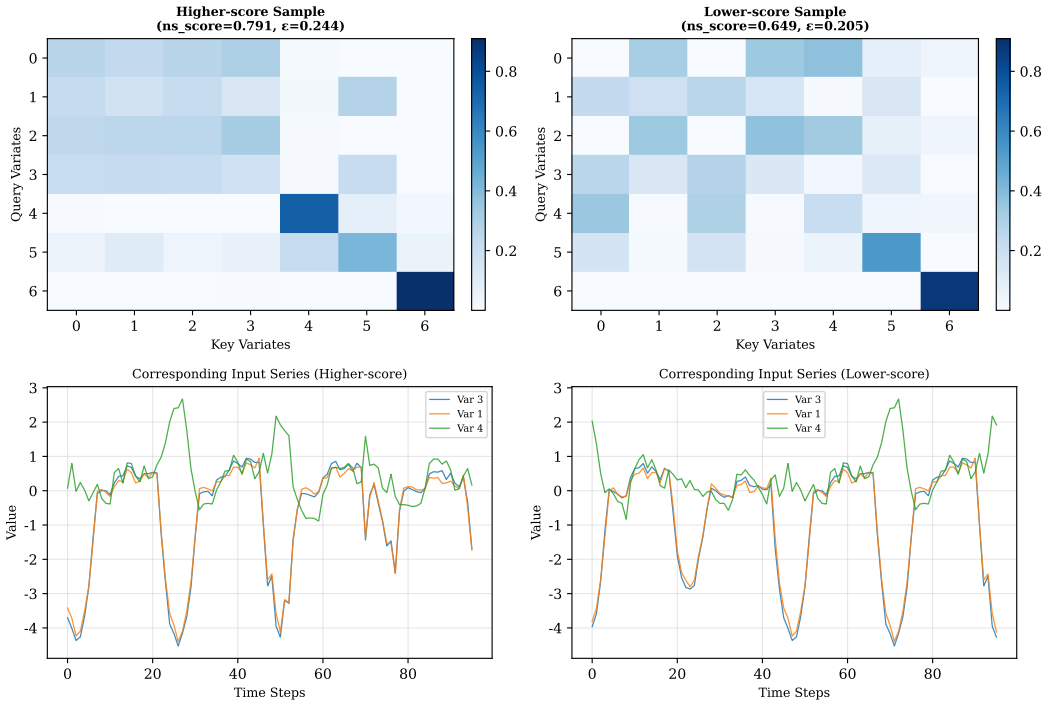


Figure 5: Additional qualitative visualization of OT transport matrices for two ETTh1 test samples (7×7 variates). **Left:** a sample with a relatively higher non-stationarity score, for which a larger ε yields a smoother and more diffuse transport plan. **Right:** a sample with a comparatively lower score, for which the transport plan becomes sharper and more concentrated. **Bottom:** corresponding raw input series.

Figure 5 provides additional qualitative evidence for the adaptive behavior of the transport matrix. For a sample with a relatively higher estimated non-stationarity score, the model assigns a larger

entropy coefficient and produces a smoother, more diffuse transport plan, distributing routing mass more broadly across variates. For a sample with a comparatively lower score, the transport plan becomes sharper and more concentrated on a smaller set of variate correspondences. This behavior is qualitatively consistent with the design of adaptive ε: more volatile inputs encourage smoother routing, whereas relatively stable inputs allow more selective cross-variate matching.

A.3 Full MSE/MAE Comparison Against Expanded Baselines

Table A2 provides the complete MSE/MAE comparison against an expanded set of baselines. Counting (dataset, horizon, metric) triplets, OTformer attains the largest number of best entries (the per-method count is shown in the bottom row of the table), substantially more than any competing method. The improvements are especially clear on Solar-Energy and ECL, moderate on the four ETT datasets, smaller on Weather, and least pronounced on Traffic in MSE. This pattern aligns with the main-paper analysis: the proposed adaptive OT routing is most beneficial when balanced cross-variate interaction and sample-wise routing adaptation play a central role, while its advantage is reduced in settings where other inductive biases are more dominant.

Table A2: Per-horizon multivariate long-term forecasting results (MSE / MAE, lower is better). **Bold**: best result per row. Underline: second best. OTformer results are averaged over three independent runs.

Models		OTformer (ours)		DUET (2025)		Leddam (2024)		FreTS (2024)		SOFTS (2024)		iTransformer (2024)		PatchTST (2023)		DLinear (2023)		TiDE (2023)		TimesNet (2023)		Stationary (2022)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.373**	**0.390**	0.377	0.393	0.377	0.394	0.399	0.412	0.381	0.399	0.386	0.405	0.414	0.419	0.386	0.400	0.479	0.464	0.384	0.402	0.591	0.524
	192	0.430	**0.420**	0.429	0.425	**0.424**	0.422	0.453	0.443	0.435	0.431	0.441	0.436	0.460	0.445	0.437	0.432	0.525	0.492	0.436	0.429	0.615	0.540
	336	0.472	**0.440**	0.471	0.446	**0.459**	0.442	0.503	0.475	0.480	0.452	0.487	0.458	0.501	0.466	0.481	0.459	0.565	0.515	0.491	0.469	0.632	0.551
	720	0.489	0.463	0.496	0.480	**0.463**	**0.459**	0.596	0.565	0.499	0.488	0.503	0.491	0.500	0.488	0.519	0.516	0.594	0.558	0.521	0.500	0.828	0.658
ETTh2	96	**0.291**	**0.335**	0.296	0.345	0.292	0.343	0.350	0.403	0.297	0.347	0.297	0.349	0.302	0.348	0.333	0.387	0.400	0.440	0.340	0.374	0.347	0.387
	192	**0.362**	**0.381**	0.368	0.389	0.367	0.389	0.472	0.475	0.373	0.394	0.380	0.400	0.388	0.400	0.477	0.476	0.528	0.509	0.402	0.414	0.379	0.418
	336	0.416	0.419	0.411	0.422	0.412	0.424	0.564	0.528	0.410	0.426	0.428	0.432	0.426	0.433	0.594	0.541	0.643	0.571	0.452	0.452	**0.358**	**0.413**
	720	0.418	**0.433**	0.412	0.434	0.419	0.438	0.815	0.654	**0.411**	**0.433**	0.427	0.445	0.431	0.446	0.831	0.657	0.874	0.679	0.462	0.468	0.422	0.457
ETTm1	96	**0.312**	**0.339**	0.324	0.354	0.319	0.359	0.339	0.374	0.325	0.361	0.334	0.368	0.329	0.367	0.345	0.372	0.364	0.387	0.338	0.375	0.415	0.410
	192	**0.365**	**0.368**	0.369	0.379	0.369	0.383	0.382	0.397	0.377	0.391	0.380	0.394	0.367	0.385	0.380	0.389	0.374	0.387	0.374	0.387	0.494	0.451
	336	0.402	**0.392**	0.404	0.402	**0.394**	0.402	0.421	0.426	0.426	0.420	0.400	0.410	0.399	0.410	0.413	0.413	0.410	0.411	0.410	0.411	0.577	0.490
	720	0.474	**0.433**	0.463	0.437	0.460	0.442	0.485	0.462	0.466	0.447	0.491	0.459	**0.454**	0.439	0.474	0.453	0.487	0.461	0.478	0.450	0.636	0.535
ETTm2	96	**0.172**	**0.249**	0.174	0.255	0.176	0.257	0.190	0.282	0.180	0.261	0.180	0.264	0.175	0.259	0.193	0.292	0.207	0.305	0.187	0.267	0.210	0.294
	192	**0.240**	**0.293**	0.243	0.302	0.243	0.303	0.260	0.329	0.246	0.306	0.250	0.309	0.241	0.302	0.284	0.362	0.290	0.364	0.249	0.309	0.338	0.373
	336	**0.303**	**0.334**	0.304	0.341	**0.303**	0.341	0.373	0.405	0.319	0.352	0.311	0.348	0.305	0.343	0.369	0.427	0.377	0.422	0.321	0.351	0.432	0.416
	720	**0.399**	**0.391**	**0.399**	0.397	0.400	0.398	0.517	0.499	0.405	0.401	0.412	0.407	0.402	0.400	0.554	0.522	0.558	0.524	0.408	0.403	0.554	0.476
Weather	96	0.166	**0.198**	0.163	0.202	**0.156**	0.202	0.184	0.239	0.166	0.208	0.174	0.214	0.177	0.218	0.196	0.255	0.202	0.261	0.172	0.220	0.188	0.242
	192	0.220	**0.247**	0.218	0.252	**0.207**	0.250	0.223	0.275	0.217	0.253	0.221	0.254	0.225	0.259	0.237	0.296	0.242	0.298	0.219	0.261	0.241	0.290
	336	0.273	**0.287**	0.274	0.294	**0.262**	0.291	0.272	0.316	0.282	0.300	0.278	0.296	0.278	0.297	0.283	0.335	0.287	0.335	0.280	0.306	0.341	0.341
	720	0.350	**0.339**	0.349	0.343	0.343	0.343	**0.340**	0.363	0.356	0.351	0.358	0.347	0.354	0.348	0.345	0.381	0.351	0.386	0.365	0.359	0.403	0.388
ECL	96	0.144	**0.229**	0.145	0.233	**0.141**	0.235	0.183	0.269	0.143	0.233	0.148	0.240	0.195	0.285	0.197	0.282	0.237	0.329	0.168	0.272	0.171	0.274
	192	0.159	**0.244**	0.163	0.248	0.159	0.252	0.187	0.276	**0.158**	0.248	0.162	0.253	0.199	0.289	0.196	0.285	0.236	0.330	0.184	0.289	0.180	0.283
	336	0.175	**0.260**	0.175	0.262	**0.173**	0.268	0.202	0.292	0.178	0.269	0.178	0.269	0.215	0.305	0.209	0.301	0.249	0.344	0.198	0.300	0.204	0.305
	720	0.204	**0.286**	0.204	0.291	**0.201**	0.295	0.237	0.325	0.218	0.305	0.225	0.317	0.256	0.337	0.245	0.333	0.284	0.373	0.220	0.320	0.221	0.319
Traffic	96	0.396	**0.246**	0.407	0.252	0.426	0.276	0.519	0.315	**0.376**	0.251	0.397	0.268	0.544	0.359	0.650	0.396	0.805	0.493	0.593	0.321	0.604	0.330
	192	0.414	**0.253**	0.431	0.262	0.458	0.289	0.521	0.325	**0.398**	0.261	0.417	0.276	0.540	0.354	0.598	0.370	0.756	0.474	0.617	0.336	0.610	0.338
	336	0.436	**0.264**	0.456	0.269	0.486	0.297	0.533	0.327	**0.415**	0.269	0.437	0.283	0.551	0.358	0.605	0.373	0.762	0.477	0.629	0.336	0.626	0.341
	720	0.468	**0.282**	0.509	0.292	0.498	0.313	0.580	0.344	**0.447**	0.287	0.469	0.302	0.586	0.375	0.645	0.394	0.719	0.449	0.640	0.350	0.643	0.347
Solar	96	**0.185**	0.215	0.200	**0.207**	0.197	0.241	0.192	0.225	0.203	0.237	0.205	0.246	0.234	0.286	0.290	0.378	0.312	0.399	0.250	0.292	0.381	0.398
	192	**0.205**	**0.226**	0.228	0.233	0.231	0.264	0.229	0.252	0.233	0.261	0.237	0.267	0.267	0.310	0.320	0.398	0.339	0.416	0.296	0.318	0.395	0.386
	336	**0.217**	**0.234**	0.262	0.244	0.242	0.269	0.243	0.269	0.250	0.275	0.250	0.276	0.290	0.315	0.353	0.415	0.368	0.430	0.319	0.330	0.410	0.394
	720	**0.219**	0.251	0.258	**0.249**	0.240	0.272	0.245	0.272	0.270	0.295	0.252	0.275	0.289	0.317	0.356	0.413	0.370	0.425	0.338	0.337	0.377	0.376
1st Count		41		3		12		1		7		0		1		0		0		0		2	

A.4 Hyperparameter Sensitivity Analysis

Table A3: MSE sensitivity to $\varepsilon_{\min}$ / $\varepsilon_{\max}$ on ETTh1 ($T = 96$) and Weather ($T = 96$). The default setting is shown in **bold**.

$\varepsilon_{\min}$	$\varepsilon_{\max}$	ETTh1 MSE	Weather MSE
0.01	0.10	0.384	0.167
0.02	0.20	0.379	0.164
0.03	**0.30**	**0.373**	**0.166**
0.05	0.50	0.381	0.165
0.10	1.00	0.390	0.170

Regularization bounds $\varepsilon_{\min}$ / $\varepsilon_{\max}$. Table A3 reports MSE under several $(\varepsilon_{\min}, \varepsilon_{\max})$ settings on ETTh1 ($T = 96$) and Weather ($T = 96$). Performance degrades when the two values become too close, effectively shrinking the adaptive range, or when both are set too large, which over-smooths

transport plans even for relatively stable inputs. The default configuration provides the most favorable trade-off on the evaluated datasets.

Sinkhorn iterations K. Varying $K \in \{1, 2, 3, 5, 10\}$ shows that very small values lead to under-converged transport plans and reduced accuracy, while performance improves rapidly up to around $K = 3$ and then saturates. Larger K yields only marginal gains while increasing inference cost approximately linearly. We therefore use $K = 3$ as the default trade-off between quality and efficiency.

Cycle length. The effectiveness of Phase Embedding depends on how well the cycle length matches a meaningful periodic scale in the data. In our experiments, values aligned with the dominant temporal granularity of the dataset consistently perform better than mismatched settings. This suggests that phase-aware encoding is most useful when the chosen cycle length corresponds to a physically or statistically meaningful periodic pattern.

A.5 Matched Mechanism Controls

Table 3 in the main text reports the matched routing controls on ECL, where the separation between Softmax and transport routing is largest. The same protocol was applied to ETTh1 and Weather, and we summarize the conclusions here rather than duplicating the table. Across all three datasets the ordering of the variants is unchanged: the same-cost row-Softmax variant is the weakest of the transport-inspired controls, forced column normalization recovers most of the gap to Softmax, and the full Sinkhorn variants improve only slightly further. The difference between the validation-selected fixed ε baseline and adaptive ε is at most about 0.08% in relative MSE on each of the three datasets, which is within the seed-to-seed spread reported in Section 4.4; the two are therefore statistically indistinguishable at this scale. We report this explicitly so that the role of adaptive ε is not overstated: it provides input-conditioned plans without per-dataset tuning, not a higher accuracy ceiling.

A.6 Cross-Backbone Transfer to PatchTST

To test whether the routing mechanism is specific to our iTransformer-style implementation, we integrated it into PatchTST. **PT-B** is the original PatchTST with its default channel-independent head. **PT-S** is a structurally matched Softmax adapter that replaces the original prediction head with the same extra adapter block used by PT-A, so that PT-S and PT-A differ only in the normalization used for cross-variate routing. **PT-A** additionally replaces Softmax routing with adaptive OT routing. All variants use the same training protocol and three random seeds.

Table A4: Cross-backbone transfer to PatchTST (three seeds, mean $\pm$ std MSE). PT-S and PT-A differ only in the cross-variate normalization rule.

Dataset	PT-B: Original	PT-S: Softmax adapter	PT-A: Adaptive OT
ECL	0.184691 ± 0.000410	0.172608 ± 0.000372	$\mathbf{0.168463 \pm 0.001143}$
ETTh1	0.377044 ± 0.001422	0.376293 ± 0.001593	0.379635 ± 0.002108

On high-dimensional ECL, PT-A reduces MSE by 8.79% relative to the original PatchTST and by 2.40% relative to the structurally matched Softmax adapter, so the gain cannot be attributed solely to adding an extra adapter block. On low-dimensional ETTh1, with only seven variates, PT-A does not produce a further gain and is slightly worse than PT-S ($+0.9\%$ MSE). This is consistent with the main-text analysis: transport routing acts on cross-variate competition, which is limited when the variate dimension is small. These results establish modularity and a reproducible benefit under substantial cross-variate competition, without implying universal improvement across all datasets.

A.7 Broader Impacts

The proposed method is a forecasting architecture for multivariate time series and is primarily intended for benign applications such as energy management, traffic planning, climate-related monitoring, and industrial system analysis. More accurate and robust forecasting in these settings may support

better resource allocation, improved scheduling, and earlier detection of system-level anomalies or operational risks.

At the same time, forecasting errors in high-stakes settings may also lead to negative impacts if the model is deployed without appropriate safeguards. For example, inaccurate forecasts may affect energy dispatch decisions, traffic control policies, or industrial monitoring pipelines. In addition, although the method is not designed for surveillance, profiling, or generative misuse, forecasts derived from operational data could still be over-interpreted in automated decision systems. For this reason, the method is best used as a decision-support tool rather than as a fully autonomous decision-maker, and practical deployment should include domain validation, monitoring, and human oversight.

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 Section 1 state the main contributions of the paper, including identifying column-sum imbalance in Softmax-based cross-variate attention, proposing sample-adaptive Sinkhorn regularization, and instantiating OTformer with phase-aware encoding. These claims are supported by the method in Section 3 and the experiments in Section 4 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated limitations paragraph is included at the end of Section 4.6. It discusses the hand-crafted nature of the non-stationarity statistics, the moderate inference-time overhead introduced by Sinkhorn iterations, the dependency-structure sensitivity of transport routing (e.g., the reduced advantage on Traffic), and the restricted evaluation scope in terms of fixed look-back length, perturbation type, and benchmark coverage.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: [N/A]

   Justification: The paper does not present formal theoretical results such as theorems, lemmas, or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

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: Section 3 fully describes the architecture (NonStationarityScorer, adaptive Sinkhorn attention, phase embedding, and MLP projection head) and Section 3.3 gives the per-sample cost-matrix construction and the Sinkhorn update rules. Section 4.1 specifies the datasets, chronological 7:1:2 splits, forecasting horizons, baselines, optimizer, learning rate, early-stopping policy, number of random seeds, and all architectural hyperparameters. Dataset statistics are provided in Appendix A.1, sensitivity to regularization bounds and Sinkhorn iterations in Appendix A.4, and the seed-to-seed variability required to reproduce the reported spreads in Section 4.4. Appendix A.6 specifies the PatchTST transfer protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [No]

   Justification: All data used in this work is open-source. Due to institutional release policy, the source code cannot be made public at submission time. We plan to open-source the implementation once internal review is completed, with the goal of public release upon paper acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: Section 4.1 reports the datasets, chronological 7:1:2 splits, forecasting horizons, baselines, optimizer (Adam) with learning rate and early-stopping policy, and the full set of model hyperparameters (hidden dimension, number of encoder layers, attention heads, FFN dimension, number of Sinkhorn iterations, and adaptive regularization range). Sensitivity to the regularization bounds is further analyzed in Appendix A.4, and dataset statistics are provided in Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

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: All OTformer results are averaged over three independent runs, and Section 4.4 reports the seed-to-seed spread from an independent rerun with three different random seeds: across the 32 forecasting settings the coefficient of variation of MSE is below 1% in 29 settings, with a median of 0.265% (e.g., ECL 0.143830 ± 0.000373, Weather 0.166546 ± 0.001164, ETTh1 0.375772 ± 0.000599). The reported $\pm$ values are standard deviations over random seeds (initialization and data-order shuffling) under a fixed train/validation/test split; no confidence intervals or formal significance tests are reported, and we state in the text that differences smaller than the seed spread should not be read as significant. Matched routing controls (Table 3) are likewise reported as three-seed means with standard deviations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: Section 4.6 (Figure 4) reports the parameter count, training time per batch, inference time per batch, and peak GPU memory usage of OTformer and the iTransformer baseline, summarizing the quantitative compute profile of the main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: To the best of current knowledge, the research conforms to the NeurIPS Code of Ethics. The work uses standard forecasting benchmarks, does not involve human subjects, and does not introduce intentionally harmful methodology.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: A broader impacts discussion is provided in Appendix A.7. It describes positive applications of more accurate and robust forecasting in domains such as energy management, traffic planning, climate-related monitoring, and industrial system analysis, and also discusses possible negative impacts arising from forecasting errors in high-stakes settings or over-interpretation of forecasts in automated decision systems, together with the recommendation that the method be used as a decision-support tool with appropriate human oversight.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 generative models, scraped datasets, or other assets that require special safeguards for responsible access.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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 benchmark datasets, baseline methods, and software libraries used in the experiments. However, the exact license names and terms of use for all external assets are not explicitly documented in the current manuscript or appendix.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

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 does not introduce or release a new dataset, benchmark, or other standalone asset in the current submission.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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 research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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 research with human subjects, so no IRB or equivalent approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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: LLMs are not part of the core model, training procedure, or experimental methodology of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.