DISTDF: TIME-SERIES FORECASTING NEEDS JOINT-DISTRIBUTION WASSERSTEIN ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Training time-series forecast models requires aligning the conditional distribution of model forecasts with that of the label sequence. The standard direct forecast (DF) approach resorts to minimizing the conditional negative log-likelihood of the label sequence, typically estimated using the mean squared error. However, this estimation proves to be biased in the presence of label autocorrelation. In this paper, we propose DistDF, which achieves alignment by alternatively minimizing a discrepancy between the conditional forecast and label distributions. Because conditional discrepancies are difficult to estimate from finite time-series observations, we introduce a newly proposed joint-distribution Wasserstein discrepancy for time-series forecasting, which provably upper bounds the conditional discrepancy of interest. This discrepancy admits tractable, differentiable estimation from empirical samples and integrates seamlessly with gradient-based training. Extensive experiments show that DistDF improves the performance of diverse forecast models and achieves the state-of-the-art forecasting performance. Code is available at https://anonymous.4open.science/r/DistDF-F66B.

1 INTRODUCTION

Time-series forecasting, which entails predicting future values based on historical observations, plays a critical role in numerous applications, such as stock trend analysis in finance (Li et al., 2025a), website traffic prediction in e-commerce (Chen et al., 2023), and trajectory forecasting in robotics (Fan et al., 2023). In the era of deep learning, the development of effective forecast models hinges on two aspects (Wang et al., 2025f): *(1) How to design neural architecture serving as the forecast models?* and *(2) How to design learning objective driving model training?* Both aspects are essential for achieving high forecast performance.

The design of neural architectures has been extensively investigated in recent studies. A central challenge involves effectively capturing the autocorrelation structures inherent in the input sequences. To this end, a variety of neural architectures have been proposed (Wang et al., 2023b; Lin et al., 2024). Recent discourse emphasizes the comparison between Transformer-based models—which leverage self-attention mechanisms to capture autocorrelation and scale effectively (Nie et al., 2023; Liu et al., 2024; Piao et al., 2024)—and linear models, which use linear projections to model autocorrelation and often achieve competitive performance with reduced complexity (Yi et al., 2023b; Zeng et al., 2023; Yue et al., 2025). These developments illustrate a rapidly evolving aspect in time-series forecasting.

In contrast, the design of learning objectives remains comparatively under-explored (Li et al., 2025c; Qiu et al., 2025a; Kudrat et al., 2025b). Current approaches typically define the learning objective by estimating the conditional likelihood of the label sequence. In practice, this is often implemented as the mean squared error (MSE), which has become a standard objective for training forecast models (Lin et al., 2025). However, MSE neglects the autocorrelation structure of the label sequence, leading to biased likelihood estimation (Wang et al., 2025g). Some efforts transform the label sequence into conditionally decorrelated components to eliminate the bias (Wang et al., 2025f;g). Nevertheless, as demonstrated in this work, such conditional decorrelation cannot be guaranteed in practice; thus, the bias persists. *Therefore, likelihood-based methods are fundamentally limited by biased likelihood estimation that impedes model training.*

To bypass the limitation of previous widely used likelihood-based methods, we propose Distribution-aware Direct Forecast (DistDF), which trains forecast models by minimizing the discrepancy between

the conditional distributions of forecast and label sequences. Since directly estimating conditional discrepancies is intractable given finite time-series observations, we introduce the joint-distribution Wasserstein discrepancy for unbiased time-series forecasting. It upper-bounds the conditional discrepancy of interest, enables differentiation, and can be estimated from finite time-series observations, making it well-suited for integration with gradient-based optimization of time-series forecast models.

Our main contributions are summarized as follows:

- We demonstrate a fundamental limitation in prevailing likelihood-based learning objectives for time-series forecasting: biased likelihood estimation that hampers effective model training.
- We propose DistDF, a training framework that aligns the conditional distributions of forecasts and labels, with a newly proposed joint-distribution Wasserstein discrepancy, ensuring the alignment of conditional distributions and admitting tractable estimation from finite time-series observations.
- We perform comprehensive empirical evaluations to demonstrate the effectiveness of DistDF, which enhances the performance of state-of-the-art forecast models across diverse datasets.

2 PRELIMINARIES

2.1 PROBLEM DEFINITION

In this paper, we focus on the multi-step time-series forecasting problem. We use uppercase letters (*e.g.*, X) to denote matrices and lowercase letters (*e.g.*, x) to denote scalars. Given a time-series dataset S with D covariates, the historical sequence at time step n is defined as $X = [S_{n-\mathrm{H}+1}, \ldots, S_n] \in \mathbb{R}^{\mathrm{H}\times\mathrm{D}}$, and the label sequence is defined as $Y = [S_{n+1}, \ldots, S_{n+\mathrm{T}}] \in \mathbb{R}^{\mathrm{T}\times\mathrm{D}}$, where H is the lookback window size and T is the forecast horizon. Modern models adopt a direct forecasting (DF) approach, generating all T forecast steps simultaneously (Liu et al., 2024). Thus, the target is to learn a model $g : \mathbb{R}^{\mathrm{H}\times\mathrm{D}} \to \mathbb{R}^{\mathrm{T}\times\mathrm{D}}$ that maps X to a forecast sequence $\hat{Y}$ approximating Y[1].

The development of forecast models encompasses two principal aspects: (1) neural network architectures that effectively encode historical sequences (Zeng et al., 2023; Liu et al., 2024), and (2) learning objectives for training neural networks (Wang et al., 2025f;g). It is important to emphasize that this work focuses on the design of learning objectives rather than proposing novel architectures. Nevertheless, we provide a concise review of both aspects for contextual completeness.

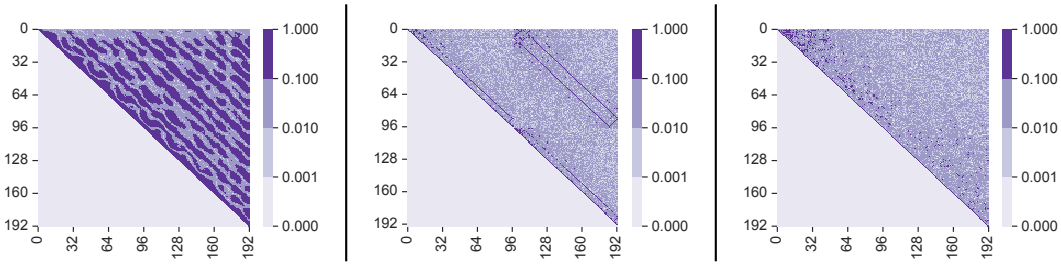
2.2 NEURAL NETWORK ARCHITECTURES IN TIME-SERIES FORECASTING

Architectural developments aim to encode historical sequences to obtain informative representation (Wu et al., 2025; Qiu et al., 2025b). Representative classic architectures include recurrent neural networks (Gu et al., 2021), convolutional neural networks (Luo and Wang, 2024), and graph neural networks (Yi et al., 2023a). A central theme in recent literature is the comparison of Transformer and non-Transformer architectures. Transformers (*e.g.*, PatchTST (Nie et al., 2023), TQNet (Lin et al., 2025), TimeBridge (Liu et al., 2025)) demonstrate strong scalability on large datasets but often entail substantial computational cost. In contrast, non-Transformer models (*e.g.*, TimeMixer (Wang et al., 2024), FreTS (Yi et al., 2023b)) offer greater computational efficiency but may be less scalable. Recent advances include hybrid architectures that combine Transformer and non-Transformer components for their complementary strengths (Lin et al., 2024), as well as the integration of Fourier analysis for efficient learning (Piao et al., 2024; Yi et al., 2025).

2.3 LEARNING OBJECTIVES IN TIME-SERIES FORECASTING

Learning objective developments have largely focused on aligning the conditional distributions of model forecasts $\mathbb{P}(\hat{Y}|X)$ with those of the label sequence $\mathbb{P}(Y|X)$. To this end, the most common objective is the MSE, which measures the point-wise error between the forecast and label sequences

[1]Hereafter, we consider the univariate case ($\mathrm{D} = 1$) for clarity. In the multivariate case, each variable can be treated as a separate univariate case when computing the learning objectives.



(a) Raw labels. (b) FreDF components. (c) Time-o1 components.

Figure 1: The conditional correlation of label components given X, where the forecast horizon is set to $\mathrm{T} = 192$. The correlation matrices are computed for the raw labels (a), the frequency components in FreDF (b) (Wang et al., 2025g) and the principal components in Time-o1 (c) (Wang et al., 2025f).

(Dai et al., 2024; Chen et al., 2025; Lin et al., 2025):

$$\mathcal{L}_{\text{mse}} = \left\| Y_{|X} - \hat{Y}_{|X} \right\|_2^2 = \sum_{t=1}^{\mathrm{T}} \left(Y_{|X,t} - \hat{Y}_{|X,t} \right)^2, \tag{1}$$

where $Y_{|X}$ is the label sequence given historical sequence X, $\hat{Y}_{|X}$ is the forecast sequence. However, the MSE objective is known to be biased since it overlooks the presence of label autocorrelation (Wang et al., 2025g). To mitigate this issue, several alternative learning objectives have been proposed. One line of work advocates aligning the overall shape of the forecast and label sequence (*e.g.*, Dilate (Le Guen and Thome, 2019) and PS (Kudrat et al., 2025a)). These approaches accommodate autocorrelation by emphasizing sequence-level differences, but lack theoretical guarantees for achieving an unbiased objective. Another line of work transforms labels into decorrelated components before alignment. This strategy reduces bias and improves forecasting performance (Wang et al., 2025f;g), showcasing the benefits of refining learning objectives for time-series forecasting.

3 Methodology

3.1 Motivation

The primary objective in training time-series forecast models is to align the conditional distribution of model-generated forecasts with that of the label sequence. Likelihood-based approaches seek this by maximizing the conditional likelihood of the label sequence. A common practice is to estimate the negative log-likelihood through the mean squared error (MSE), which has become the predominant objective for training time-series forecast models (Lin et al., 2025). However, MSE treats each future step as an independent prediction task and thus ignores the autocorrelation structure of the label sequence, where each observation typically depends on its predecessors (Zeng et al., 2023). Such an oversight renders MSE biased from the true negative log-likelihood of the label sequence. This issue is termed as autocorrelation bias and formalized in Theorem 3.1.

Theorem 3.1 (Autocorrelation bias)**.** *Suppose $Y_{|X} \in \mathbb{R}^{\mathrm{T}}$ is the label sequence given historical sequence X, $\hat{Y}_{|X} \in \mathbb{R}^{T}$ is the forecast sequence, $\Sigma_{|X} \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$ is the conditional covariance of $Y_{|X}$. The bias of MSE from the negative log-likelihood of the label sequence given X is expressed as:*

$$\text{Bias} = \left\| Y_{|X} - \hat{Y}_{|X} \right\|_{\Sigma_{|X}^{-1}}^2 - \left\| Y_{|X} - \hat{Y}_{|X} \right\|_2^2. \tag{2}$$

where $\|v\|_{\Sigma_{|X}^{-1}}^2 = v^\top \Sigma_{|X}^{-1} v$. It vanishes if the conditional covariance $\Sigma_{|X}$ is the identity matrix[2].

Some might argue that the bias can be eliminated by first transforming the label sequence into conditionally decorrelated components and then applying MSE component-wise. For example,

[2]The pioneering work (Wang et al., 2025f) derives the bias from the marginal likelihood of Y assuming it follows a Gaussian distribution. In contrast, this work clarifies that it is the conditional distribution of Y given X that is Gaussian. Consequently, we derive the bias from the conditional log-likelihood of Y.

FreDF (Wang et al., 2025g) uses Fourier transform to obtain frequency components; **Time-o1** (Wang et al., 2025f) employs principal component analysis to obtain principal components. This strategy does eliminate the bias if the resulting components were truly conditionally decorrelated (see Theorem 3.1). However, one key distinction warrants emphasis: both Fourier and principal component transformations guarantee only *marginally decorrelated* of the obtained components (*i.e.*, diagonal Σ), not the required *conditional decorrelation* (*i.e.*, diagonal $\Sigma_{|X}$)[3]; thus the bias persists. *Hence, likelihood-based methods are limited by biased likelihood estimation which hampers model training.*

Case study. We conduct a case study on the Traffic dataset to illustrate the limitations of likelihood-based methods. As shown in Fig. 1(a), the conditional correlation matrix reveals substantial off-diagonal values—over 50.3% exceed 0.1—illustrating the presence of autocorrelation effects. In contrast, Fig. 1(b) presents the conditional correlations of the latent components extracted by FreDF and Time-o1 (Wang et al., 2025g;f). While the non-diagonal elements are notably reduced, residual correlations remain, indicating that these methods do not fully eliminate autocorrelation in the transformed components. Consequently, applying a point-wise loss to these transformed components continues to ignore autocorrelation and yields bias.

Given the substantial challenges faced by likelihood-based methods, it is worthwhile to explore alternative strategies to align conditional distributions for model training. One plain strategy is directly minimizing a *distributional discrepancy between the conditional distributions* (Courty et al., 2017), which can effectively achieve alignment while bypassing the complexity of likelihood estimation. Importantly, there are two questions that warrant investigation. *How to devise a discrepancy to align the two conditional distributions? Does it effectively improve forecast performance?*

3.2 Aligning conditional distributions via joint-distribution balancing

In this section, we aim to align the conditional distributions, *i.e.*, $\mathbb{P}_{\hat{Y}|X}$ and $\mathbb{P}_{Y|X}$, by minimizing a discrepancy metric between them. As with general distribution alignment tasks, the choice of discrepancy metric is crucial (Xu et al., 2021). We select the Wasserstein discrepancy from optimal transport theory, which measures the discrepancy between two distributions as the minimum cost required to transform one into the other. Its ability to remain informative for distributions with disjoint supports, combined with its robust theoretical properties and proven empirical success, makes it a principled choice for this work (Courty et al., 2017). An informal definition is provided in Definition 3.2.

Definition 3.2 (Wasserstein discrepancy)**.** *Let α and β be random variables with probability distributions $\mathbb{P}_\alpha$ and $\mathbb{P}_\beta$; $\mathcal{S}_\alpha = [\alpha_1, ..., \alpha_{\mathrm{n}}]$ and $\mathcal{S}_\beta = [\beta_1, ..., \beta_{\mathrm{m}}]$ be empirical samples from $\mathbb{P}_\alpha$ and $\mathbb{P}_\beta$. The optimization problem seeks a feasible plan $P \in \mathbb{R}_+^{\mathrm{n}\times\mathrm{m}}$ to transport α to β at the minimum cost:*

$$\begin{aligned} \mathcal{W}_p(\mathbb{P}_\alpha, \mathbb{P}_\beta) &:= \min_{P\in\Pi(\alpha,\beta)} \langle D, P\rangle\,, \\ \Pi(\mathbb{P}_\alpha, \mathbb{P}_\beta) &:= \begin{cases} P_{i,1} + ... + P_{i,\mathrm{m}} = a_i, i = 1, ..., \mathrm{n}, \\ P_{1,j} + ... + P_{\mathrm{n},j} = b_j, j = 1, ..., \mathrm{m}, \\ P_{i,j} \geq 0, i = 1, ..., \mathrm{n}, j = 1, ..., \mathrm{m}, \end{cases} \end{aligned} \tag{3}$$

where $\mathcal{W}_p$ denotes the p-Wasserstein discrepancy; $D \in \mathbb{R}_+^{\mathrm{n}\times\mathrm{m}}$ represents the pairwise distances calculated as $D_{i,j} = \|\alpha_i - \beta_j\|_p^p$; $a = [a_1, \ldots, a_{\mathrm{n}}]$ and $b = [b_1, \ldots, b_{\mathrm{m}}]$ are the weights of samples in α and β, respectively; n *and* m *are the numbers of samples;* Π *defines the set of constraints.*

A natural approach to aligning the conditional distributions is to minimize the Wasserstein discrepancy $\mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X})$. However, this approach suffers from an **estimation difficulty**. For any given X, a typical dataset often provides only a single associated label sequence Y, and the forecast model produces only a single output $\hat{Y}$. Thus, the empirical sets ($\mathcal{S}_{Y|X}$ and $\mathcal{S}_{\hat{Y}|X}$) each contain only a single sample, which is insufficient to represent the underlying conditional distributions and renders the discrepancy uninformative. Crucially, this limitation is not unique to the Wasserstein discrepancy; any distributional discrepancy metric becomes degenerate in the absence of multiple samples.

[3]According to Theorem 3.3 (Wang et al., 2025g) and Lemma 3.2 (Wang et al., 2025f), the components obtained by Fourier and principal component transformations are marginal decorrelated.

Lemma 3.3 (Kim et al. (2022)). *For any $p \geq 1$, the joint-distribution Wasserstein discrepancy upper bounds the expected conditional-distribution Wasserstein discrepancy:*

$$\int \mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X}) d\mathbb{P}(X) \leq \mathcal{W}_p(\mathbb{P}_{X,Y}, \mathbb{P}_{X,\hat{Y}}). \tag{4}$$

where the equality holds if $p = 1$ or the conditional Wasserstein term is constant with respect to X.

To bypass this estimation difficulty, we advocate the joint-distribution Wasserstein discrepancy, $\mathcal{W}_p(\mathbb{P}_{X,Y}, \mathbb{P}_{X,\hat{Y}})$, for training time-series forecast models. This proxy is advantageous for two reasons. First, it provides a provable **upper bound** on the expected conditional discrepancy (see Lemma 3.3), ensuring that minimizing the joint discrepancy effectively aligns the conditional distributions of interest. Second, it is readily **estimable** from finite time-series observations, since the empirical samples $\mathcal{S}_{X,Y}$ and $\mathcal{S}_{X,\hat{Y}}$ can be constructed from the entire dataset, yielding sufficient samples to compute a meaningful and informative discrepancy.

Theorem 3.4 (Alignment property). *The conditional distributions are aligned, i.e., $\mathbb{P}_{Y|X} = \mathbb{P}_{\hat{Y}|X}$ if the joint-distribution Wasserstein discrepancy is minimized to zero, i.e., $\mathcal{W}_p(\mathbb{P}_{X,Y}, \mathbb{P}_{X,\hat{Y}}) = 0$.*

Lemma 3.5 (Peyré and Cuturi (2019)). *Suppose $\mathbb{P}_{X,Y}$ and $\mathbb{P}_{X,\hat{Y}}$ obey Gaussian distributions $\mathcal{N}(\mu_{X,Y}, \Sigma_{X,Y})$ and $\mathcal{N}(\mu_{X,\hat{Y}}, \Sigma_{X,\hat{Y}})$, respectively. The squared $\mathcal{W}_2$ discrepancy can be calculated as the Bures-Wasserstein discrepancy:*

$$\mathcal{BW}(\mu_{X,Y}, \mu_{X,\hat{Y}}, \Sigma_{X,Y}, \Sigma_{X,\hat{Y}}) = \left\| \mu_{X,Y} - \mu_{X,\hat{Y}} \right\|_2^2 + \mathcal{B}(\Sigma_{X,Y}, \Sigma_{X,\hat{Y}}), \tag{5}$$

where $\mathcal{B}(\Sigma_{X,Y}, \Sigma_{X,\hat{Y}}) = \mathrm{Tr}\left(\Sigma_{X,Y} + \Sigma_{X,\hat{Y}} - 2\sqrt{\Sigma_{X,Y}^{1/2} \Sigma_{X,\hat{Y}} \Sigma_{X,Y}^{1/2}}\right)$, $\mathrm{Tr}(\cdot)$ denotes matrix trace.

Theoretical Justification. Theorem 3.4 shows that minimizing the joint-distribution Wasserstein discrepancy to zero guarantees the alignment of conditional distributions. This result enables using the joint discrepancy as a learning objective for training forecast models. Under a Gaussian assumption (likewise MSE), this discrepancy has an analytical form (Lemma 3.5), obviating the need to solve the complex transport problem of Definition 3.2. The proof is available in Appendix A.

The use of Wasserstein discrepancy for distribution alignment is highly inspired by domain adaptation field (Courty et al., 2017). However, one key distinction warrants emphasis. Domain adaptation dominantly aligns the *marginal distributions of inputs* to improve generalization; in contrast, we align the *conditional distributions* of model outputs and labels to perform supervised training. To our knowledge, this represents a technically innovative strategy.

3.3 Model Implementation

In this section, we present the implementation specifics of DistDF, a framework that leverages the joint-distribution Wasserstein discrepancy to enhance the training of time-series forecast models. The principal steps of the algorithm are formalized in Algorithm 1.

Algorithm 1 The workflow of DistDF.

Input: X: historical sequences, Y: label sequences.
Parameter: α: the relative weight of the discrepancy, g: the forecast model to generate forecast sequence.
Output: $\mathcal{L}_\alpha$: the obtained learning objective.

1: $\hat{Y} \leftarrow g(X)$
2: $Z \leftarrow \mathrm{concate}(X, Y)$, $\hat{Z} \leftarrow \mathrm{concate}(X, Y)$
3: $\mu_Z \leftarrow \mathrm{mean}(Z)$, $\Sigma_Z \leftarrow \mathrm{cov}(Z)$
4: $\mu_{\hat{Z}} \leftarrow \mathrm{mean}(\hat{Z})$, $\Sigma_{\hat{Z}} \leftarrow \mathrm{cov}(\hat{Z})$
5: $\mathcal{L}_{\mathrm{dist}} \leftarrow \mathcal{BW}(\mu_Z, \mu_{\hat{Z}}, \Sigma_Z, \Sigma_{\hat{Z}})$
6: $\mathcal{L}_{\mathrm{mse}} \leftarrow \|Y - \hat{Y}\|_2^2$
7: $\mathcal{L}_\alpha := \alpha \cdot \mathcal{L}_{\mathrm{dist}} + (1 - \alpha) \cdot \mathcal{L}_{\mathrm{mse}}$

Given historical sequences X and corresponding label sequences $Y \in \mathbb{R}^{\mathrm{B} \times \mathrm{T}}$, where B denotes batch size and T denotes forecast horizon; the forecast model g is employed to generate the forecast sequences, denoted as $\hat{Y}$ (step 1). Subsequently, we define two joint sequences, which are constructed by concatenating X with Y and $\hat{Y}$ along the time axis, respectively (step 2), expressed as $Z = [X, Y]$ and $\hat{Z} = [X, \hat{Y}]$.

To quantify the discrepancy term $\mathcal{L}_{\mathrm{dist}}$, we compute the first- and second-order statistics of Z and $\hat{Z}$, *i.e.*, the mean vectors (μ_Z and $\mu_{\hat{Z}}$) and covariance matrices (Σ_Z and $\Sigma_{\hat{Z}}$) (steps 3-4). The discrepancy term $\mathcal{L}_{\mathrm{dist}}$ is then evaluated using the Bures-Wasserstein metric (step 5), as defined in Lemma 3.5.

Table 1: Long-term forecasting performance.

Models	**DistDF (Ours)**		TimeBridge (2025)		Fredformer (2024)		iTransformer (2024)		FreTS (2023)		TimesNet (2023)		MICN (2023)		TiDE (2023)		PatchTST (2023)		DLinear (2023)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.378**	**0.394**	0.387	0.400	0.387	0.398	0.411	0.414	0.414	0.421	0.438	0.430	0.396	0.421	0.413	0.407	0.389	0.400	0.403	0.407
ETTm2	**0.277**	**0.321**	0.281	0.326	0.280	0.324	0.295	0.336	0.316	0.365	0.302	0.334	0.308	0.364	0.286	0.328	0.303	0.344	0.342	0.392
ETTh1	**0.430**	**0.429**	0.442	0.440	0.447	0.434	0.452	0.448	0.489	0.474	0.472	0.463	0.533	0.519	0.448	0.435	0.459	0.451	0.456	0.453
ETTh2	**0.367**	**0.393**	0.377	0.403	0.377	0.402	0.386	0.407	0.524	0.496	0.409	0.420	0.620	0.546	0.378	0.401	0.390	0.413	0.529	0.499
ECL	**0.172**	**0.267**	0.176	0.271	0.191	0.284	0.179	0.270	0.199	0.288	0.212	0.306	0.192	0.302	0.215	0.292	0.195	0.286	0.212	0.301
Traffic	**0.417**	**0.279**	0.426	0.282	0.486	0.336	0.426	0.285	0.538	0.330	0.631	0.338	0.529	0.312	0.624	0.373	0.468	0.298	0.625	0.384
Weather	**0.248**	**0.275**	0.252	0.277	0.261	0.282	0.269	0.289	0.249	0.293	0.271	0.295	0.264	0.321	0.272	0.291	0.267	0.288	0.265	0.317
PEMS03	**0.104**	**0.215**	0.112	0.223	0.146	0.260	0.122	0.233	0.149	0.261	0.126	0.230	0.106	0.223	0.316	0.370	0.170	0.282	0.216	0.322
PEMS08	**0.123**	**0.223**	0.139	0.239	0.171	0.271	0.149	0.247	0.174	0.275	0.152	0.243	0.153	0.258	0.318	0.378	0.201	0.303	0.249	0.332

Note: We fix the input length as 96 following Liu et al. (2024). **Bold** and underlined denote best and second-best results, respectively. *Avg* indicates average results over horizons: T=96, 192, 336 and 720. DistDF employs the top-performing baseline on each dataset as its underlying forecast model.

Given the complexity of directly optimizing the Bures–Wasserstein discrepancy and its lack of inherent pairing awareness, we integrate it with the mean squared error to promote training stability and facilitate convergence (steps 6–7), following the established practices (Wang et al., 2025f;g):

$$\mathcal{L}_\alpha := \alpha \cdot \mathcal{L}_{\text{dist}} + (1-\alpha) \cdot \mathcal{L}_{\text{mse}}. \tag{6}$$

where $0 \leq \alpha \leq 1$ balances the contribution of the distributional discrepancy term.

By minimizing the distributional discrepancy, DistDF effectively aligns the conditional distributions of the forecast and label sequences, thereby refining the model's forecast performance. DistDF preserves the principal benefits of the canonical DF framework (Zeng et al., 2023; Liu et al., 2024), such as efficient inference and multi-task learning capability. Moreover, DistDF is model-agnostic, which renders it a plugin-and-play component to improve the training of different forecast models.

4 Experiments

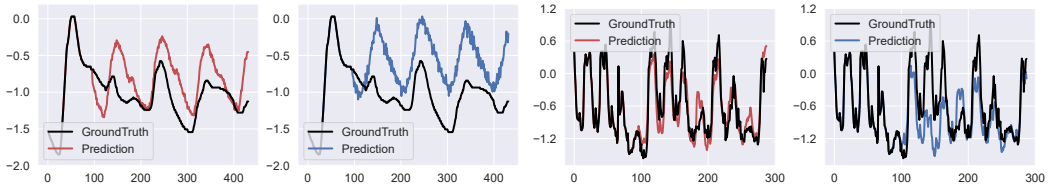
To demonstrate the efficacy of DistDF, the following aspects deserve empirical investigation:

1. **Performance:** *Does DistDF perform well?* In Section 4.2, we benchmark DistDF against state-of-the-art baselines, and in Section 4.3, we compare it with alternative learning objectives.
2. **Gain:** *Why does it work?* In section 4.4, we perform an ablative study, dissecting the individual components of DistDF and clarifying their contributions to forecast accuracy.
3. **Generality:** *Does it support other models and discrepancy measures?* In Section 4.5, we examine its compatibility with various models and discrepancies, with further results in Appendix D.4.
4. **Sensitivity:** *Is it sensitive to hyperparameters?* In Section 4.6, we analyze the sensitivity of DistDF to the hyperparameter α, showing stable performance across a broad parameter range.
5. **Efficiency:** *What is the computational cost of it?* In Appendix D.7, we evaluate the running cost of DistDF across different scenarios.

4.1 Setup

Datasets. We evaluate our methods using several standard public benchmarks for long-term time-series forecasting, following Wu et al. (2023). Specifically, we use the ETT dataset (four subsets), ECL, Traffic, Weather, and PEMS (Liu et al., 2024). All datasets are split chronologically into training, validation, and test sets. Comprehensive dataset statistics are presented in Appendix C.1.

Baselines. We compare DistDF to a range of competitive baselines, categorized as: (1) Transformer-based models—PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), Fredformer (Piao et al., 2024) and TimeBridge (Liu et al., 2025); (2) Non-Transformer based models— DLinear (Zeng et al., 2023), TiDE (Das et al., 2023), MICN (Wang et al., 2023b) and FreTS (Yi et al., 2023b).



(a) ETTm2 snapshot. (b) ECL snapshot.

Figure 2: The forecast sequence of DF (in blue) and DistDF (in red), with historical length H = 96.

Table 2: Comparative results with other objectives for time-series forecasting.

Loss		**DistDF**		Time-o1		FreDF		Koopman		Dilate		Soft-DTW		DF	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeBridge	ETTm1	**0.383**	**0.397**	0.383	0.397	0.386	0.398	0.460	0.438	0.387	0.400	0.395	0.402	0.387	0.400
	ETTh1	**0.434**	**0.436**	0.439	0.438	0.439	0.436	0.459	0.449	0.464	0.452	0.452	0.445	0.442	0.440
	ECL	**0.172**	**0.267**	0.175	0.268	0.175	0.267	0.182	0.277	0.176	0.271	0.173	0.268	0.176	0.271
	Weather	**0.248**	**0.275**	0.250	0.275	0.254	0.276	0.269	0.293	0.252	0.277	0.260	0.280	0.252	0.277
Fredformer	ETTm1	**0.378**	0.394	0.379	**0.393**	0.384	0.394	0.389	0.400	0.389	0.400	0.397	0.402	0.387	0.398
	ETTh1	**0.430**	**0.429**	0.431	0.429	0.438	0.434	0.452	0.443	0.453	0.442	0.460	0.449	0.447	0.434
	ECL	**0.173**	**0.266**	0.178	0.270	0.179	0.272	0.190	0.282	0.187	0.280	0.206	0.298	0.191	0.284
	Weather	**0.255**	0.277	0.255	**0.276**	0.256	0.277	0.257	0.279	0.258	0.280	0.261	0.280	0.261	0.282

Note: **Bold** and underlined denote best and second-best results, respectively. The reported results are averaged over forecast horizons: T=96, 192, 336 and 720. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

Implementation. Baseline implementations closely follow the official codebase from Piao et al. (2024). To ensure fair comparison, the drop-last trick is disabled for all models, as recommended in Qiu et al. (2024). All models are trained with the Adam optimizer (Kingma and Ba, 2015). When integrating DistDF into a baseline forecast model, we retain all hyperparameters from the public benchmarks (Liu et al., 2024; Piao et al., 2024), only tuning α and the learning rate. Experiments are run on Intel(R) Xeon(R) Platinum 8383C CPUs with 32 NVIDIA RTX H100 GPUs. Further implementation details are provided in Appendix C.

4.2 Overall Performance

Table 1 reports the long-term forecasting results. DistDF consistently enhances the performance of base models across all evaluated datasets. For instance, on ETTh1, DistDF reduces the MSE of TimeBridge by 0.016. Similar improvements observed on other benchmarks confirm its robustness and generalizability. We attribute these empirical improvements to DistDF's ability to align conditional distributions, a property supported by its theoretical guarantees (Theorem 3.4).

Showcases. To further illustrate the practical benefits, we compare the forecast sequences of DF and DistDF in Fig. 2. While a model trained with the standard DF objective captures the overall trend, it fails to accurately track fine-grained variations, such as rapid changes between steps 100 and 200. In contrast, DistDF produces forecasts that more precisely reflect these subtle and rapid changes, highlighting its effectiveness in improving real-world forecasting accuracy.

4.3 Learning Objective Comparison

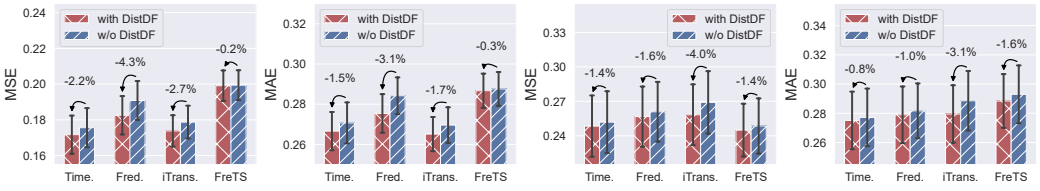
Table 2 presents a comparison between DistDF and several established time-series learning objectives: Time-o1 (Wang et al., 2025f), FreDF (Wang et al., 2025g), Koopman (Lange et al., 2021), Dilate (Le Guen and Thome, 2019), Soft-DTW (Cuturi and Blondel, 2017), and DPTA (Sakoe and Chiba, 2003). In this comparison, all methods are integrated into both TimeBridge and Fredformer using their official implementations for ensuring fairness.

In general, shape alignment objectives (Dilate, Soft-DTW, DPTA) improve marginally over standard DF, consistent with findings by Le Guen and Thome (2019). This suggests that heuristic shape-level alignment does not guarantee alignment of conditional distributions. FreDF and Time-o1 reduce

Table 3: Ablation study results.

Model	Align μ	Align Σ	Data	T=96		T=192		T=336		T=720		Avg	
				MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DF	✗	✗	ETTm1	0.326	0.361	0.365	0.382	0.396	0.404	0.459	0.444	0.387	0.398
			ETTh1	0.377	0.396	0.437	0.425	0.486	0.449	0.488	0.467	0.447	0.434
			ECL	0.142	0.239	0.161	0.257	0.182	0.278	0.217	0.309	0.176	0.271
			Weather	0.168	0.211	0.214	0.254	0.273	0.297	0.353	0.347	0.252	0.277
DistDF†	✓	✗	ETTm1	0.318	0.359	0.361	0.382	0.393	0.404	0.453	0.440	0.381	0.396
			ETTh1	0.375	0.394	0.435	0.426	0.471	0.446	0.457	0.455	0.435	0.430
			ECL	0.142	0.239	0.160	0.257	0.180	0.273	0.217	0.307	0.175	0.269
			Weather	0.168	0.211	0.213	0.253	0.273	0.296	0.349	0.348	0.251	0.277
DistDF‡	✗	✓	ETTm1	0.328	0.365	0.364	0.385	0.395	0.406	0.457	0.441	0.386	0.399
			ETTh1	0.374	0.396	0.430	0.430	0.476	0.451	0.476	0.472	0.439	0.437
			ECL	0.141	0.239	0.161	0.257	0.179	0.273	0.216	0.307	0.174	0.269
			Weather	0.168	0.211	0.214	0.253	0.270	0.296	0.353	0.347	0.251	0.277
DistDF	✓	✓	ETTm1	**0.316**	**0.357**	**0.359**	**0.381**	**0.392**	**0.404**	**0.448**	**0.437**	**0.379**	**0.395**
			ETTh1	**0.373**	**0.393**	**0.428**	**0.425**	**0.466**	**0.445**	**0.453**	**0.453**	**0.430**	**0.429**
			ECL	**0.137**	**0.235**	**0.159**	**0.257**	**0.178**	**0.272**	**0.212**	**0.302**	**0.172**	**0.267**
			Weather	**0.164**	**0.209**	**0.212**	**0.252**	**0.270**	**0.295**	**0.348**	**0.345**	**0.248**	**0.275**

Note: **Bold** and underlined denote best and second-best results, respectively. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.



(a) ECL with MSE (b) ECL with MAE (c) Weather with MSE (d) Weather with MAE

Figure 3: Improvement of DistDF applied to different forecast models, shown with colored bars for means over forecast lengths (96, 192, 336, 720) and error bars for 50% confidence intervals.

the bias in likelihood estimation and improve performance. However, as established in Section 3.1, residual bias remains, preventing unbiased alignment of conditional distributions. DistDF minimizes the discrepancy between conditional distributions, achieving unbiased alignment with theoretical guarantees (see Theorem3.4), and consequently delivers superior performance.

4.4 Ablation Studies

Table 3 examines the two components in the joint-distribution Wasserstein discrepancy (5): mean alignment and covariance alignment. The main findings are as follows:

- DistDF† augments DF by aligning only the means of the joint distributions, omitting the $\mathcal{B}(\cdot)$ in (5). This approach outperforms DF, illustrating that mean alignment of joint distributions can improve the alignment of the conditional distributions between label and forecast sequence.
- DistDF‡ improves DF by aligning only the variance of joint distributions, exclusively involving $\mathcal{B}(\cdot)$ in (5). This approach also leads to improvements over DF in most cases, illustrating that variance alignment of joint distributions improves the alignment of the conditional distributions.
- DistDF combines both mean and variance alignment for comprehensive joint distribution matching. It yields the best results, demonstrating a synergistic effect when both components are integrated.

4.5 Generalization Studies

In this section, we assess the generalizability of DistDF by applying it to different distribution discrepancy measures and across various forecast models.

Varying discrepancy. We evaluate alternative discrepancy measures to align the joint distribution and report the results in Table 4. Specifically, we consider Kullback-Leibler (KL) divergence, maxi-

Table 4: Comparative results with other discrepancies for aligning the joint distributions.

Discrepancy		**Ours**		EMD		MMD@Linear		MMD@RBF		KL		DF	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeBridge	ETTm1	**0.383**	**0.398**	0.388	0.400	0.385	0.400	0.387	0.399	0.387	0.400	0.387	0.400
	ETTh1	**0.433**	0.437	0.441	0.439	0.438	**0.437**	0.441	0.440	0.437	0.438	0.442	0.440
	ECL	**0.172**	0.267	0.177	0.272	0.174	0.269	0.172	**0.266**	0.176	0.271	0.176	0.271
	Weather	**0.248**	**0.275**	0.251	0.276	0.253	0.278	0.250	0.276	0.253	0.277	0.252	0.277
Fredformer	ETTm1	**0.379**	**0.395**	0.386	0.397	0.380	0.395	0.385	0.397	0.385	0.397	0.387	0.398
	ETTh1	**0.429**	**0.431**	0.445	0.435	0.437	0.432	0.444	0.435	0.444	0.435	0.447	0.434
	ECL	**0.183**	**0.275**	0.187	0.280	0.188	0.280	0.187	0.280	0.187	0.279	0.191	0.284
	Weather	**0.257**	**0.279**	0.261	0.282	0.262	0.282	0.262	0.282	0.261	0.282	0.261	0.282

Note: **Bold** and underlined denote best and second-best results, respectively. The reported results are averaged over forecast horizons: T=96, 192, 336 and 720. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

Table 5: Varying α results of TimeBridge

α	ETTh2		ECL		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE
0	0.377	0.403	0.176	0.271	0.252	0.277
0.001	0.378	0.402	0.172	0.267	0.250	0.276
0.002	0.377	0.402	0.173	0.267	0.250	0.276
0.005	0.376	0.401	**0.172**	**0.267**	0.250	**0.276**
0.01	0.376	0.400	0.172	0.267	**0.249**	0.276
0.02	0.376	0.400	0.174	0.269	0.249	0.276
0.05	**0.375**	0.399	0.174	0.268	0.252	0.278
0.1	0.375	**0.399**	0.174	0.269	0.254	0.280
0.2	0.376	0.399	0.177	0.270	0.258	0.282
0.5	0.378	0.400	0.186	0.277	0.261	0.285
1	0.381	0.402	0.197	0.282	0.265	0.286

Note: **Bold** and underlined denote the best and second-best results. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

Table 6: Varying α results of Fredformer.

α	ETTh2		ECL		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE
0	0.377	0.402	0.191	0.284	0.261	0.282
0.001	0.371	0.397	0.182	0.275	0.257	0.279
0.002	0.372	0.398	**0.181**	**0.274**	0.257	0.279
0.005	0.372	0.398	0.182	0.275	0.257	0.280
0.01	0.370	0.397	0.183	0.275	**0.257**	**0.279**
0.02	**0.369**	**0.395**	0.182	0.275	0.258	0.280
0.05	0.370	0.396	0.187	0.279	0.259	0.281
0.1	0.371	0.397	0.196	0.287	0.261	0.283
0.2	0.372	0.398	0.209	0.298	0.263	0.285
0.5	0.376	0.399	0.230	0.317	0.266	0.287
1	0.386	0.406	0.239	0.326	0.268	0.290

Note: **Bold** and underlined denote the best and second-best results. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

mum mean discrepancy (MMD) with RBF and linear kernels, and earth mover discrepancy (EMD). All discrepancies show improvements over the standard DF, indicating the benefit of incorporating distribution alignment for training forecast models. The joint-distribution Wasserstein discrepancy achieves the best overall results in 14 out of 16 cases, underscoring its effectiveness in reliably aligning distributions (Peyré and Cuturi, 2019).

Varying forecast models. We further demonstrate the flexibility of DistDF by integrating it into several representative models: TimeBridge, FredFormer, iTransformer, and FreTS. As shown in Fig. 3, DistDF consistently enhances forecast performance across all tested models. For example, on the ECL dataset, iTransformer and FredFormer augmented with DistDF achieve substantial MSE reductions—up to 2.7% and 4.3%, respectively. These results highlight DistDF's potential as a general, plug-and-play enhancement for supporting various forecast models.

4.6 Hyperparameter Sensitivity

In this section, we analyze how varying the weight of the distributional discrepancy influences DistDF's performance, as summarized in Table 5 and Table 6. On the one hand, increasing α from 0 generally leads to improved performance; for example, FredFormer achieves a 0.01 reduction in MSE on ECL, showcasing the utility of incorporating the discrepancy term into the learning objective of training forecast models. On the other hand, the optimal performance is typically observed for $\alpha < 1$. This underscores the complementary role of MSE, which is both easy to optimize and straightforward for ensuring point-wise accuracy between the forecast and label sequences.

Conclusion

In this study, we demonstrate that existing likelihood-based approaches suffer from biased likelihood estimation. Instead, we propose DistDF, which trains forecast models by minimizing the discrepancy between the conditional distributions of forecasts and labels. Recognizing the intractability of directly estimating conditional discrepancies, we propose the use of a joint-distribution Wasserstein discrepancy, which serves as a tractable upper bound and can be efficiently estimated from observed data.

By minimizing this quantity, DistDF provably aligns the conditional distributions with theoretical guarantees for training forecast models. Extensive experiments corroborate that DistDF consistently yields improvements in forecast accuracy.

Limitations. According to Lemma 3.5, DistDF quantifies the divergence between the mean and covariance of the joint distributions, thereby capturing global distributional properties. However, it discards elementwise correspondences between forecast and label sequences—information critical for forecasting tasks. Therefore, DistDF is most effective when employed as a regularization term alongside the standard MSE loss, where MSE recovers elementwise correspondences and fully unleashes the potential of the proposed DistDF.

Reproducibility Statement

The anonymous downloadable source code is available at https://anonymous.4open.science/r/DistDF-F66B. For theoretical results, a complete proof of the claims is included in the Appendix A; For datasets used in the experiments, a complete description of the dataset statistics and processing workflow is provided in the Appendix C.

References

Jason M. Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1964–1974, 2017.

Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. *ACM Trans. Graph.*, 30(6):158, 2011.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhichao Chen, Leilei Ding, Zhixuan Chu, Yucheng Qi, Jianmin Huang, and Hao Wang. Monotonic neural ordinary differential equation: Time-series forecasting for cumulative data. In *Proc. ACM Int. Conf. Inf. Knowl. Manag.*, pages 4523–4529, 2023.

Zhichao Chen, Haoxuan Li, Fangyikang Wang, Haotian Zhang, Hu Xu, Xiaoyu Jiang, Zhihuan Song, and Hao Wang. Rethinking the diffusion models for missing data imputation: A gradient flow perspective. In *Proc. Adv. Neural Inf. Process. Syst.*, 2024.

Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.*, 87(314):2563–2609, 2018.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1853–1865, 2017.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proc. Int. Conf. Mach. Learn.*, pages 894–903. PMLR, 2017.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *Proc. Int. Conf. Learn. Represent.*, 2024.

Abhimanyu Das, Weihao Kong, Andrew Leach, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *Trans. Mach. Learn. Res.*, 2023.

Jiajun Fan, Yuzheng Zhuang, Yuecheng Liu, Hao Jianye, Bin Wang, Jiangcheng Zhu, Hao Wang, and Shu-Tao Xia. Learnable behavior control: Breaking atari human world records via sample-efficient behavior selection. In *Proc. Int. Conf. Learn. Represent.*, pages 1–9, 2023.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *Proc. Int. Conf. Learn. Represent.*, 2021.

Leonid V Kantorovich. On the translocation of masses. *J. Math. Sci.*, 133(4):1381–1382, 2006.

Young-geun Kim, Kyungbok Lee, and Myunghee Cho Paik. Conditional wasserstein generator. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7208–7219, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learn. Represent.*, pages 1–9, 2015.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In *Proc. Int. Conf. Mach. Learn.*, 2025a.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In *Proc. Int. Conf. Mach. Learn.*, 2025b.

Henning Lange, Steven L Brunton, and J Nathan Kutz. From fourier to koopman: Spectral methods for long-term time series prediction. *Journal of Machine Learning Research*, 22(41):1–38, 2021.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 32, 2019.

Haoxuan Li, Kunhan Wu, Chunyuan Zheng, Yanghao Xiao, Hao Wang, Zhi Geng, Fuli Feng, Xiangnan He, and Peng Wu. Removing hidden confounding in recommendation: a unified multi-task learning approach. *Proc. Adv. Neural Inf. Process. Syst.*, 36:54614–54626, 2024a.

Haoxuan Li, Chunyuan Zheng, Shuyi Wang, Kunhan Wu, Eric Wang, Peng Wu, Zhi Geng, Xu Chen, and Xiao-Hua Zhou. Relaxing the accurate imputation assumption in doubly robust learning for debiased collaborative filtering. In *Proc. Int. Conf. Mach. Learn.*, volume 235, pages 29448–29460, 2024b.

Haoxuan Li, Chunyuan Zheng, Wenjie Wang, Hao Wang, Fuli Feng, and Xiao-Hua Zhou. Debiased recommendation with noisy feedback. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, page 1576–1586, 2024c.

Jianxin Li, Xiong Hui, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proc. AAAI Conf. Artif. Intell.*, 2021.

Junjie Li, Yang Liu, Weiqing Liu, Shikai Fang, Lewen Wang, Chang Xu, and Jiang Bian. Mars: a financial market simulation engine powered by generative foundation model. In *Proc. Int. Conf. Learn. Represent.*, 2025a.

Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *Proc. Int. Conf. Learn. Represent.*, 2025b.

Xinyu Li, Yuchen Luo, Hao Wang, Haoxuan Li, Liuhua Peng, Feng Liu, Yandong Guo, Kun Zhang, and Mingming Gong. Towards accurate time series forecasting via implicit decoding. *Proc. Adv. Neural Inf. Process. Syst.*, 2025c.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 37, pages 106315–106345, 2024.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proc. Int. Conf. Mach. Learn.*, 2025.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: time series modeling and forecasting with sample convolution and interaction. In *Proc. Adv. Neural Inf. Process. Syst.*, 2022.

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. In *Proc. Int. Conf. Mach. Learn.*, 2025.

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 *Proc. Int. Conf. Learn. Represent.*, 2024.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *Proc. Int. Conf. Learn. Represent.*, pages 1–43, 2024.

Simone Di Marino and Augusto Gerolin. An optimal transport approach for the schrödinger bridge problem and convergence of sinkhorn algorithm. *J. Sci. Comput.*, 85(2):27, 2020.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *Proc. Int. Conf. Learn. Represent.*, 2023.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11 (5-6):355–607, 2019.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2400–2410, 2024.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. In *Proc. VLDB Endow.*, pages 2363–2377, 2024.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. Dbloss: Decomposition-based loss function for time series forecasting. *Proc. Adv. Neural Inf. Process. Syst.*, 2025a.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pages 1185–1196, 2025b.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Signal Process.*, 26(1):43–49, 2003.

Hao Wang, Zhichao Chen, Jiajun Fan, Haoxuan Li, Tianqiao Liu, Weiming Liu, Quanyu Dai, Yichao Wang, Zhenhua Dong, and Ruiming Tang. Optimal transport for treatment effect estimation. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, pages 5404–5418, 2023a.

Hao Wang, Zhichao Chen, Zhaoran Liu, Xu Chen, Haoxuan Li, and Zhouchen Lin. Proximity matters: Local proximity enhanced balancing for treatment effect estimation. *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2025a.

Hao Wang, Zhichao Chen, Zhaoran Liu, Haozhe Li, Degui Yang, Xinggao Liu, and Haoxuan Li. Entire space counterfactual learning for reliable content recommendations. *IEEE Trans. Inf. Forensics Security*, 20:1755–1764, 2025b.

Hao Wang, Zhichao Chen, Yuan Shen, Hui Zheng, Degui Yang, Dangjun Zhao, and Buge Liang. Robust missing value imputation with proximal optimal transport for low-quality iiot data. In *IEEE Trans. Neural Netw. Learn. Syst.*, 2025c.

Hao Wang, Zhichao Chen, Honglei Zhang, Zhengnan Li, Licheng Pan, Haoxuan Li, and Mingming Gong. Debiased recommendation via wasserstein causal balancing. *ACM Trans. Inf. Syst.*, 2025d.

Hao Wang, Xinggao Liu, Zhaoran Liu, Haozhe Li, Yilin Liao, Yuxin Huang, and Zhichao Chen. Lspt-d: Local similarity preserved transport for direct industrial data imputation. *IEEE Trans. Autom. Sci. Eng.*, 22:9438–9448, 2025e.

Hao Wang, Licheng Pan, Zhichao Chen, Xu Chen, Qingyang Dai, Lei Wang, Haoxuan Li, and Zhouchen Lin. Time-o1: Time-series forecasting needs transformed label alignment. *Proc. Adv. Neural Inf. Process. Syst.*, 2025f.

Hao Wang, Licheng Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. Fredf: Learning to forecast in the frequency domain. In *Proc. Int. Conf. Learn. Represent.*, pages 1–9, 2025g.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *Proc. Int. Conf. Learn. Represent.*, 2023b.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *Proc. Int. Conf. Learn. Represent.*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Proc. Adv. Neural Inf. Process. Syst.*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proc. Int. Conf. Learn. Represent.*, 2023.

Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. In *Proc. Adv. Neural Inf. Process. Syst.*, 2025.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32, 2019.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. Vocabulary learning via optimal transport for neural machine translation. In *ACL/IJCNLP (1)*, pages 7361–7373. Association for Computational Linguistics, 2021.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. In *Proc. Adv. Neural Inf. Process. Syst.*, 2023a.

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 *Proc. Adv. Neural Inf. Process. Syst.*, 2023b.

Kun Yi, Qi Zhang, Wei Fan, Longbing Cao, Shoujin Wang, Hui He, Guodong Long, Liang Hu, Qingsong Wen, and Hui Xiong. A survey on deep learning based time series analysis with frequency transformation. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pages 6206–6215, 2025.

Wenzhen Yue, Yong Liu, Haoxuan Li, Hao Wang, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. Olinear: A linear model for time series forecasting in orthogonally transformed domain. *Proc. Adv. Neural Inf. Process. Syst.*, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proc. AAAI Conf. Artif. Intell.*, 2023.

A THEORETICAL JUSTIFICATION

Theorem A.1 (Autocorrelation bias, Theorem 3.1 in the main text)**.** *Suppose $Y_{|X} \in \mathbb{R}^T$ is the label sequence given historical sequence X, $\hat{Y}_{|X} \in \mathbb{R}^T$ is the forecast sequence, $\Sigma_{|X} \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$ is the conditional covariance of $Y_{|X}$. The bias of MSE from the negative log-likelihood of the label sequence given X is expressed as:*

$$\text{Bias} = \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2_{\Sigma^{-1}_{|X}} - \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2. \tag{7}$$

where $\|v\|^2_{\Sigma^{-1}_{|X}} = v^\top \Sigma^{-1}_{|X} v$. It vanishes if the conditional covariance $\Sigma_{|X}$ is identity matrix[4].

Proof. The proof follows the narrative in Wang et al. (2025f) but highlights that it is the conditional distribution of Y given X that obeys Gaussian distribution, instead of the marginal distribution of Y.

Suppose the label sequence given X follows a multivariate normal distribution with mean vector $\hat{Y}_{|X} = [\hat{Y}_{|X,1}, \hat{Y}_{|X,2}, \ldots, \hat{Y}_{|X,\mathrm{T}}]$ and covariance matrix $\Sigma_{|X}$. The conditional likelihood of Y is:

$$\mathbb{P}_{Y|X} = \frac{1}{(2\pi)^{0.5\mathrm{T}}|\Sigma_{|X}|^{0.5}} \exp(-\frac{1}{2}\left\|Y_{|X} - \hat{Y}_{|X}\right\|^2_{\Sigma^{-1}_{|X}}) \tag{8}$$

On the basis, the conditional negative log-likelihood of Y is:

$$-\log \mathbb{P}_{Y|X} = \frac{1}{2}\left(\mathrm{T}\log(2\pi) + \log|\Sigma_{|X}| + \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2_{\Sigma^{-1}_{|X}}\right).$$

Removing the terms unrelated to $\hat{Y}_{|X}$, the terms used for updating $\hat{Y}_{|X}$, namely practical negative log-likelihood (PNLL), is expressed as follows:

$$\text{PNLL} = \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2_{\Sigma^{-1}_{|X}}. \tag{9}$$

On the other hand, the MSE loss can be expressed as:

$$\text{MSE} = \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2_2. \tag{10}$$

The difference between PNLL and MSE is computed as:

$$\text{Bias} = \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2_{\Sigma^{-1}_{|X}} - \left\|Y_{|X} - \hat{Y}_{|X}\right\|^2, \tag{11}$$

which diminishes to zero if the label sequence is conditionally decorrelated, *i.e.*, $\Sigma_{|X}$ is identity matrix. The proof is completed. □

Lemma A.2 (Lemma 3.3 in the main text)**.** *For any $p \geq 1$, the joint-distribution Wasserstein discrepancy upper bounds the expected conditional-distribution Wasserstein discrepancy:*

$$\int \mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X}) d\mathbb{P}(X) \leq \mathcal{W}_p(\mathbb{P}_{X,Y}, \mathbb{P}_{X,\hat{Y}}). \tag{12}$$

where the equality holds if $p = 1$ or the conditional Wasserstein term is constant with respect to X.

Proof. The proof can be found in Theorem 2 of Kim et al. (2022). □

Theorem A.3 (Alignment property, Theorem 3.4 in the main text)**.** *The conditional distributions are aligned, i.e., $\mathbb{P}_{Y|X} = \mathbb{P}_{\hat{Y}|X}$ if the joint-distribution Wasserstein discrepancy is minimized to zero, i.e., $\mathcal{W}_p(\mathbb{P}_{X,Y}, \mathbb{P}_{X,\hat{Y}}) = 0$.*

[4]The pioneering work (Wang et al., 2025g) identifies the bias under the first-order Markov assumption on the label sequence. This study generalizes this bias without the first-order Markov assumption.

Proof. By Lemma 3.3, we have

$$\int \mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X})\, d\mathbb{P}(X) \leq \mathcal{W}_p(\mathbb{P}_{X,Y}, \mathbb{P}_{X,\hat{Y}}).$$

Thus, if RHS $= 0$, we have $\int \mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X})\, d\mathbb{P}(X) = 0$. Since $\mathcal{W}_p$ is non-negative (Peyré and Cuturi, 2019), this implies that $\mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X}) = 0$ for almost every X. Therefore, it suffices to prove that for two distributions $\mathbb{P}_\alpha = \mathbb{P}_{Y|X}$ and $\mathbb{P}_\beta = \mathbb{P}_{\hat{Y}|X}$, $\mathcal{W}_p(\mathbb{P}_\alpha, \mathbb{P}_\beta) = 0$ implies $\mathbb{P}_\alpha = \mathbb{P}_\beta$.

Suppose $\mathcal{S}_\alpha = [\alpha_1, ..., \alpha_{\mathrm{n}}]$ and $\mathcal{S}_\beta = [\beta_1, ..., \beta_{\mathrm{n}}]$ are the empirical samples from $\mathbb{P}_\alpha$ and $\mathbb{P}_\beta$, respectively, with corresponding mass vectors a and b We are given that $\mathcal{W}_p(\mathbb{P}_\alpha, \mathbb{P}_\beta) = 0$. By Definition 3.2, this means the minimum value of the cost function is zero. Let P^* be an optimal transport plan that solves the minimization problem. Then,

$$\mathcal{W}_p(\mathbb{P}_\alpha, \mathbb{P}_\beta) = \langle D, P^* \rangle = \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{m}} P^*_{i,j} \left\| \alpha_i - \beta_j \right\|_p^p = 0. \tag{13}$$

From the constraints, we know the elements of the transport plan are non-negative, $P^*_{i,j} \geq 0$. The distance term is also non-negative, $\left\| \alpha_i - \beta_j \right\|_p^p \geq 0$. Since the total sum of these non-negative terms is zero, each individual term in the summation must be zero:

$$P^*_{i,j} \left\| \alpha_i - \beta_j \right\|_p^p = 0, \quad \forall i = 1, ..., \mathrm{n}, j = 1, ..., \mathrm{m}. \tag{14}$$

This condition implies that if any mass is moved from a point α_i to a point β_j (*i.e.*, $P^*_{i,j} > 0$), then the distance between these points must be zero (*i.e.*, $\left\| \alpha_i - \beta_j \right\|_p^p = 0$), which means $\alpha_i = \beta_j$. In other words, the optimal plan only transports mass between identical points.

Let's consider the total probability mass assigned to an arbitrary value z that exists in the support of either distribution. The total mass at z, *i.e.*, probability density, for distribution $\mathbb{P}_\alpha$ is $P_\alpha(z) = \sum_{i:\alpha_i = z} a_i$. Using the constraints from $\Pi(\mathbb{P}_\alpha, \mathbb{P}_\beta)$ in Equation 3, we can express this as:

$$P_\alpha(z) = \sum_{i:\alpha_i = z} a_i = \sum_{i:\alpha_i = z} \left(\sum_{j=1}^{\mathrm{m}} P^*_{i,j} \right). \tag{15}$$

As established, $P^*_{i,j}$ can only be non-zero if $\beta_j = \alpha_i$. Therefore, for the outer sum where $\alpha_i = z$, the inner sum over j is non-zero only for those indices j where $\beta_j = z$. Thus, we can write:

$$P_\alpha(z) = \sum_{i:\alpha_i = z} \sum_{j:\beta_j = z} P^*_{i,j}. \tag{16}$$

Similarly, the mass at z for distribution $\mathbb{P}_\beta$ is $P_\beta(z) = \sum_{j:\beta_j = z} b_j$. Using the other set of constraints from $\Pi(\mathbb{P}_\alpha, \mathbb{P}_\beta)$:

$$P_\beta(z) = \sum_{j:\beta_j = z} b_j = \sum_{j:\beta_j = z} \left(\sum_{i=1}^{\mathrm{n}} P^*_{i,j} \right). \tag{17}$$

Again, since $P^*_{i,j}$ is non-zero only if $\alpha_i = \beta_j$, for the terms in the outer sum where $\beta_j = z$, the inner sum over i is non-zero only for those indices i where $\alpha_i = z$. This gives:

$$P_\beta(z) = \sum_{j:\beta_j = z} \sum_{i:\alpha_i = z} P^*_{i,j}. \tag{18}$$

By comparing the resulting expressions for $P_\alpha(z)$ and $P_\beta(z)$, we find they are identical:

$$P_\alpha(z) = P_\beta(z). \tag{19}$$

Since this equality holds for any value z, the probability mass functions of $\mathbb{P}_\alpha$ and $\mathbb{P}_\beta$ are identical, which implies $\mathbb{P}_\alpha = \mathbb{P}_\beta$[5]. Applying this result to our conditional distributions, $\mathcal{W}_p(\mathbb{P}_{Y|X}, \mathbb{P}_{\hat{Y}|X}) = 0$ implies $\mathbb{P}_{Y|X} = \mathbb{P}_{\hat{Y}|X}$ for almost every X. This completes the proof. □

[5]A discrete probability is completely characterized by two components: its support and its probability mass function.

Lemma A.4 (Lemma 3.5 in the main text). *Suppose $\mathbb{P}_{X,Y}$ and $\mathbb{P}_{X,\hat{Y}}$ obey Gaussian distributions $\mathcal{N}(\mu_{X,Y}, \Sigma_{X,Y})$ and $\mathcal{N}(\mu_{X,\hat{Y}}, \Sigma_{X,\hat{Y}})$, respectively. The squared $\mathcal{W}_2$ discrepancy can be calculated as the Bures-Wasserstein discrepancy:*

$$\mathcal{BW}(\mu_{X,Y}, \mu_{X,\hat{Y}}, \Sigma_{X,Y}, \Sigma_{X,\hat{Y}}) = \left\|\mu_{X,Y} - \mu_{X,\hat{Y}}\right\|_2^2 + \mathcal{B}(\Sigma_{X,Y}, \Sigma_{X,\hat{Y}}), \tag{20}$$

where $\mathcal{B}(\Sigma_{X,Y}, \Sigma_{X,\hat{Y}}) = \mathrm{Tr}\left(\Sigma_{X,Y} + \Sigma_{X,\hat{Y}} - 2\sqrt{\Sigma_{X,Y}^{1/2}\Sigma_{X,\hat{Y}}\Sigma_{X,Y}^{1/2}}\right)$, $\mathrm{Tr}(\cdot)$ denotes matrix trace.

Proof. The proof can be found in Remark 2.31 of Peyré and Cuturi (2019). □

Additional notes on the Gaussian assumption. Lemma 3.5 presents the $\mathcal{BW}$ discrepancy under the Gaussian assumption, yielding a tractable and efficient form. However, the Bures-Wasserstein discrepancy measures differences only in the first- and second-order moments—i.e., the mean and covariance. While these two moments fully characterize Gaussian distributions, real-world datasets do not necessarily adhere to Gaussianity, additionally requiring higher-order moments for complete characterization. Nonetheless, the mean and covariance remain essential descriptors for any distribution. As a result, in cases where data deviate from strict Gaussianity, $\mathcal{BW}$ remains a valuable tool for distribution alignment by matching these fundamental moments.

B Overview of Discrete Optimal Transport and Wasserstein discrepancy

This section outlines the foundational concepts of optimal transport (OT) and the Wasserstein discrepancy. Our analysis is specifically confined to discrete probability measures, as the broader theory involving general measures is beyond the scope of this work. For a comprehensive treatment of the continuous case, readers are directed to the seminal works by Peyré and Cuturi (2019).

The classical framing of OT, known as the Monge problem, can be illustrated with a simple scenario: transporting goods from n warehouses to m factories (Peyré and Cuturi, 2019). Let the i-th warehouse hold a_i units of material and the j-th factory require b_j units. The objective is to find a transport map that moves all material from the warehouses to satisfy the factories' demands. This problem is subject to several constraints: the entire stock from each warehouse must be shipped, all factory demands must be met, and the mapping must be deterministic (i.e., each warehouse ships its entire stock to a single factory). The optimal map is the one that minimizes the total cost, which is aggregated from the cost of moving a unit of material from a given warehouse to a factory.

Definition B.1 (Monge Problem for Discrete Measures). *Let $\alpha = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i}$ and $\beta = \sum_{j=1}^m b_j \delta_{\mathbf{y}_j}$ be two discrete probability measures. The Monge problem seeks a transport map $\mathbb{T} : \{\mathbf{x}_i\}_{i=1}^n \to \{\mathbf{y}_j\}_{j=1}^m$ that pushes the mass of α forward to match β, denoted by $\mathbb{T}_\sharp \alpha = \beta$. This condition implies that for each j, the total mass received, b_j, must equal the sum of the masses sent from all locations mapped to it: $b_j = \sum_{i:\mathbb{T}(\mathbf{x}_i)=\mathbf{y}_j} a_i$. The objective is to find the map $\mathbb{T}$ that minimizes the total transportation cost:*

$$\min_{\mathbb{T}:\mathbb{T}_\sharp\alpha=\beta} \left\{ \sum_{i=1}^n c(\mathbf{x}_i, \mathbb{T}(\mathbf{x}_i)) a_i \right\}. \tag{21}$$

While intuitive, the Monge formulation is restrictive; a solution is not guaranteed to exist, particularly when mass splitting is required (e.g., one warehouse supplying multiple factories). To address this limitation, Kantorovich (2006) introduced a relaxed formulation. Instead of a deterministic map, Kantorovich's approach seeks a probabilistic coupling or "transport plan" that allows mass from a single source to be distributed among multiple destinations. This reframes the problem within the versatile framework of linear programming. When the measures are probability distributions (i.e., $\sum a_i = \sum b_j = 1$), the resulting optimal cost defines a distance metric.

Definition B.2 (Kantorovich Problem). *Let $\alpha = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i}$ and $\beta = \sum_{j=1}^m b_j \delta_{\mathbf{y}_j}$ be two discrete probability distributions supported on samples $\{\mathbf{x}_i\}_{i=1}^n$ and $\{\mathbf{y}_j\}_{j=1}^m$, respectively. The optimal transport problem is to find a transport plan $\boldsymbol{\pi} \in \mathbb{R}_+^{\mathrm{n}\times\mathrm{m}}$ that minimizes the total cost:*

$$\mathcal{W}_c(\alpha, \beta) := \min_{\boldsymbol{\pi}\in\Pi(a,b)} \langle \mathbf{C}, \boldsymbol{\pi} \rangle_F, \tag{22}$$

Table 7: Dataset description.

Dataset	D	Forecast length	Train / validation / test	Frequency	Domain
ETTh1	7	96, 192, 336, 720	8545/2881/2881	Hourly	Health
ETTh2	7	96, 192, 336, 720	8545/2881/2881	Hourly	Health
ETTm1	7	96, 192, 336, 720	34465/11521/11521	15min	Health
ETTm2	7	96, 192, 336, 720	34465/11521/11521	15min	Health
Weather	21	96, 192, 336, 720	36792/5271/10540	10min	Weather
ECL	321	96, 192, 336, 720	18317/2633/5261	Hourly	Electricity
Traffic	862	96, 192, 336, 720	12185/1757/3509	Hourly	Transportation
PEMS03	358	12, 24, 36, 48	15617/5135/5135	5min	Transportation
PEMS08	170	12, 24, 36, 48	10690/3548/265	5min	Transportation

Note: *D* denotes the number of variates. *Frequency* denotes the sampling interval of time points. *Train, Validation, Test* denotes the number of samples employed in each split. The taxonomy aligns with (Wu et al., 2023).

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot product. The cost matrix $\mathbf{C} \in \mathbb{R}_+^{\mathrm{n} \times \mathrm{m}}$ contains the pairwise costs, e.g., $\mathbf{C}_{ij} = c(\mathbf{x}_i, \mathbf{y}_j)$. The set of feasible transport plans, $\Pi(a, b)$, is defined by the constraints that preserve the total mass of the source and target measures:

$$\Pi(a, b) := \left\{ \boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \boldsymbol{\pi} \mathbf{1}_m = a, \boldsymbol{\pi}^\top \mathbf{1}_n = b \right\}. \tag{23}$$

Here, a and b are the weight vectors for the measures α and β. If the cost is a metric distance raised to a power p, $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$, the p-th root of the optimal cost defines the p-Wasserstein discrepancy, $\mathcal{W}_p(\alpha, \beta)$.

Contemporary research in discrete optimal transport primarily progresses along two paths. The first focuses on computational efficiency. Exact solutions via linear programming are often infeasible for large-scale problems due to their high computational complexity, typically $\mathcal{O}(n^3 \log n)$ where n is the number of support points (Bonneel et al., 2011). This has motivated the development of faster, approximate methods, such as entropic regularization (leading to the Sinkhorn algorithm) with nearly quadratic complexity (Altschuler et al., 2017) and sliced OT, which reduces the problem to one-dimensional computations and achieves near-linear complexity. The second path involves adapting the OT framework to address specific challenges across various domains, such as domain adaptation (Chizat et al., 2018), causal inference (Wang et al., 2025a; 2023a), generative modeling (Marino and Gerolin, 2020; Chen et al., 2024), missing data imputation (Wang et al., 2025e;c), graph comparison (Xu et al., 2019) and recommendation system (Wang et al., 2025d).

C Reproduction Details

C.1 Dataset descriptions

Our empirical evaluation is conducted on a diverse collection of widely-used time series forecasting benchmarks. Each dataset presents distinct characteristics in terms of dimensionality and temporal resolution. A summary is provided in Table 7.

- **ETT** (Li et al., 2021): Contains seven metrics related to electricity transformers, recorded from July 2016 to July 2018. It is divided into four subsets based on sampling frequency: ETTh1 and ETTh2 (hourly), and ETTm1 and ETTm2 (every 15 minutes).
- **Weather** (Wu et al., 2021): Comprises 21 meteorological variables from the Max Planck Biogeochemistry Institute's weather station, captured every 10 minutes throughout 2020.
- **ECL** (Wu et al., 2021): Features the hourly electricity consumption of 321 clients.
- **Traffic** (Wu et al., 2021): Documents the hourly occupancy rates of 862 sensors on San Francisco Bay Area freeways, spanning from 2015 to 2016.
- **PEMS** (Liu et al., 2022): Consists of public traffic data from the California highway system, aggregated in 5-minute intervals. We utilize two common subsets, PEMS03 and PEMS08.

Following established protocols (Qiu et al., 2024; Liu et al., 2024), all datasets are chronologically partitioned into training, validation, and test sets. For the ETT, Weather, ECL, and Traffic datasets, we use a fixed historical sequence length of 96 and evaluate performance across four prediction horizons with lengths of 96, 192, 336, and 720. For the PEMS datasets, we also use an historical length of 96 but evaluate on shorter prediction horizons of 12, 24, 36, and 48 steps. During the final evaluation on the test set, we ensure that no data is discarded from the last batch: a technique referred to as the *dropping-last trick* is disabled throughout our experiments.

C.2 Implementation details of model training

To establish a fair comparison, we reproduced all baseline models using their official, publicly available implementations, primarily sourcing from the iTransformer (Liu et al., 2024) and Fredformer (Piao et al., 2024) repositories. The reproducibility of these baseline results was verified prior to our experiments. All models were trained to minimize the MSE loss function using the Adam optimizer (Kingma and Ba, 2015). The learning rate for each baseline was selected from the set $\{10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}\}$ based on the best performance on the validation set. To prevent overfitting, we employed an early stopping mechanism that terminates training if the validation loss fails to improve for three consecutive epochs.

When integrating our proposed distributional discrepancy component, DistDF, with an existing forecasting model, we maintain the original model's optimized hyperparameters as reported in their respective benchmarks (Liu et al., 2024; Piao et al., 2024). Our tuning is therefore focused and conservative, limited to two key parameters: the learning rate and the weight of the discrepancy term, $\alpha \in (0, 1]$. Adjusting the learning rate is necessary as the distributional discrepancy term has varying overall magnitude and gradient dynamics on different datasets. The tuning is driven by selecting the combination that yields the lowest MSE on the validation set.

C.3 Implementation details of conditional correlation computation

A key challenge in analyzing time series is to accurately quantify the autocorrelation structure within the label sequence without the confounding influence of the historical sequence Wang et al. (2025b); Li et al. (2024b). Standard metrics like the Pearson correlation are insufficient for this task, as they cannot disentangle the dependencies among future time steps from their shared dependence on the past Li et al. (2024a;c).

To address this, we employ the partial correlation coefficient to measure the conditional autocorrelation. This allows us to assess the relationship between any two time steps in the label sequence while controlling for the linear effects of the entire historical sequence. Our implementation is based on the standard procedure for computing partial correlation, which is also implemented in statistical software like MATLAB's 'partialcorr' function.[6]

The procedure can be described as follows. Let X be the historical sequence (the control variables) and Y be the label sequence. To compute the partial correlation between two time steps, Y_t and $Y_{t'}$, conditioned on X, we follow a two-stage regression process. We first isolate the variance in Y_t and $Y_{t'}$ that cannot be explained by X. This is achieved by training two separate linear regression models using ordinary least squares (OLS). The residuals from these models, ϵ_t and $\epsilon_{t'}$, represent the parts of Y_t and $Y_{t'}$ that are linearly independent of X. The partial correlation between Y_t and $Y_{t'}$, conditioned on X, is then calculated as the standard Pearson correlation between their respective residuals. This process effectively measures the linear relationship between Y_t and $Y_{t'}$ after accounting for the influence of the historical context X.

D More Experimental Results

D.1 Overall performance

Additional experimental results of overall performance are available in Table 8, where the performance given different T is reported.

[6]Implementation is available at https://www.mathworks.com/help/stats/partialcorr.html

Table 8: Full results on the multi-step forecasting task. The length of history window is set to 96 for all baselines. Avg indicates the results averaged over forecasting lengths: T=96, 192, 336 and 720.

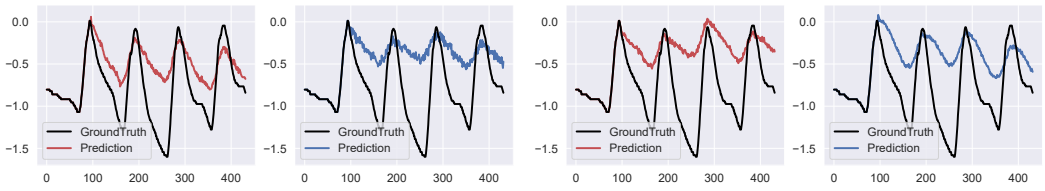
Models		**DistDF (Ours)**		TimeBridge (2025)		Fredformer (2024)		iTransformer (2024)		FreTS (2023)		TimesNet (2023)		MICN (2023)		TiDE (2023)		PatchTST (2023)		DLinear (2023)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.316**	**0.357**	0.323	0.361	0.326	0.361	0.338	0.372	0.342	0.375	0.368	0.394	0.319	0.366	0.353	0.374	0.325	0.364	0.346	0.373
	192	**0.358**	**0.380**	0.366	0.385	0.365	0.382	0.382	0.396	0.385	0.400	0.406	0.409	0.364	0.395	0.391	0.393	0.363	0.383	0.380	0.390
	336	**0.392**	**0.404**	0.398	0.408	0.396	0.404	0.427	0.424	0.416	0.421	0.454	0.444	0.395	0.425	0.423	0.414	0.404	0.413	0.413	0.414
	720	**0.448**	**0.437**	0.461	0.445	0.459	0.444	0.496	0.463	0.513	0.489	0.527	0.474	0.505	0.499	0.486	0.448	0.463	0.442	0.472	0.450
	Avg	**0.378**	**0.394**	0.387	0.400	0.387	0.398	0.411	0.414	0.414	0.421	0.438	0.430	0.396	0.421	0.413	0.407	0.389	0.400	0.403	0.407
ETTm2	96	**0.174**	**0.256**	0.177	0.259	0.177	0.260	0.182	0.265	0.188	0.279	0.184	0.262	0.178	0.277	0.182	0.265	0.180	0.266	0.188	0.283
	192	**0.239**	**0.298**	0.243	0.303	0.242	0.300	0.257	0.315	0.264	0.329	0.257	0.308	0.266	0.343	0.247	0.304	0.285	0.339	0.280	0.356
	336	0.300	**0.338**	0.303	0.343	0.302	0.340	0.320	0.354	0.322	0.369	0.315	0.345	**0.299**	0.354	0.307	0.343	0.309	0.347	0.375	0.420
	720	**0.397**	**0.394**	0.401	0.399	0.399	0.397	0.423	0.411	0.489	0.482	0.452	0.421	0.489	0.482	0.408	0.398	0.437	0.422	0.526	0.508
	Avg	**0.277**	**0.321**	0.281	0.326	0.280	0.324	0.295	0.336	0.316	0.365	0.302	0.334	0.308	0.364	0.286	0.328	0.303	0.344	0.342	0.392
ETTh1	96	**0.373**	**0.393**	0.373	0.395	0.377	0.396	0.385	0.405	0.398	0.409	0.399	0.418	0.381	0.416	0.387	0.395	0.381	0.400	0.389	0.404
	192	0.428	0.425	**0.428**	0.426	0.437	0.425	0.440	0.437	0.451	0.442	0.452	0.451	0.497	0.489	0.439	**0.425**	0.450	0.443	0.442	0.440
	336	**0.466**	**0.445**	0.471	0.451	0.486	0.449	0.480	0.457	0.501	0.472	0.488	0.469	0.589	0.555	0.482	0.447	0.501	0.470	0.488	0.467
	720	**0.453**	**0.453**	0.495	0.487	0.488	0.467	0.504	0.492	0.608	0.571	0.549	0.515	0.665	0.617	0.484	0.471	0.504	0.492	0.505	0.502
	Avg	**0.430**	**0.429**	0.442	0.440	0.447	0.434	0.452	0.448	0.489	0.474	0.472	0.463	0.533	0.519	0.448	0.435	0.459	0.451	0.456	0.453
ETTh2	96	**0.287**	**0.336**	0.294	0.344	0.293	0.344	0.301	0.349	0.315	0.374	0.321	0.358	0.351	0.398	0.291	0.340	0.299	0.349	0.330	0.383
	192	**0.358**	**0.381**	0.371	0.394	0.372	0.391	0.383	0.397	0.466	0.467	0.418	0.417	0.492	0.489	0.376	0.392	0.383	0.404	0.439	0.450
	336	**0.408**	**0.421**	0.421	0.429	0.420	0.433	0.425	0.432	0.522	0.502	0.464	0.454	0.656	0.582	0.417	0.427	0.439	0.444	0.589	0.538
	720	**0.416**	**0.435**	0.423	0.443	0.421	0.439	0.436	0.448	0.792	0.643	0.434	0.450	0.981	0.718	0.429	0.446	0.438	0.455	0.757	0.626
	Avg	**0.367**	**0.393**	0.377	0.403	0.377	0.402	0.386	0.407	0.524	0.496	0.409	0.420	0.620	0.546	0.378	0.401	0.390	0.413	0.529	0.499
ECL	96	**0.137**	**0.235**	0.142	0.239	0.161	0.258	0.150	0.242	0.180	0.266	0.170	0.272	0.170	0.281	0.197	0.274	0.170	0.264	0.197	0.282
	192	**0.159**	**0.257**	0.161	0.257	0.174	0.269	0.168	0.259	0.184	0.272	0.183	0.282	0.185	0.297	0.197	0.277	0.179	0.273	0.197	0.286
	336	**0.178**	**0.272**	0.182	0.278	0.194	0.290	0.182	0.274	0.199	0.290	0.203	0.302	0.190	0.298	0.212	0.292	0.195	0.288	0.209	0.301
	720	**0.212**	**0.302**	0.217	0.309	0.235	0.319	0.214	0.304	0.234	0.322	0.294	0.366	0.221	0.329	0.254	0.325	0.234	0.320	0.245	0.334
	Avg	**0.172**	**0.267**	0.176	0.271	0.191	0.284	0.179	0.270	0.199	0.288	0.212	0.306	0.192	0.302	0.215	0.292	0.195	0.286	0.212	0.301
Traffic	96	**0.380**	**0.262**	0.391	0.268	0.461	0.327	0.397	0.271	0.531	0.323	0.590	0.316	0.498	0.298	0.646	0.386	0.444	0.284	0.649	0.397
	192	**0.407**	**0.275**	0.418	0.276	0.470	0.326	0.416	0.279	0.519	0.321	0.624	0.336	0.521	0.309	0.599	0.362	0.454	0.291	0.598	0.371
	336	**0.429**	0.284	0.432	**0.284**	0.492	0.338	0.429	0.286	0.529	0.327	0.641	0.345	0.529	0.314	0.606	0.363	0.469	0.298	0.605	0.373
	720	**0.452**	**0.297**	0.464	0.301	0.521	0.353	0.462	0.303	0.573	0.346	0.670	0.356	0.567	0.326	0.643	0.383	0.506	0.319	0.646	0.395
	Avg	**0.417**	**0.279**	0.426	0.282	0.486	0.336	0.426	0.285	0.538	0.330	0.631	0.338	0.529	0.312	0.624	0.373	0.468	0.298	0.625	0.384
Weather	96	**0.164**	**0.209**	0.168	0.211	0.180	0.220	0.171	0.210	0.174	0.228	0.183	0.229	0.179	0.244	0.192	0.232	0.189	0.230	0.194	0.253
	192	**0.212**	**0.252**	0.214	0.254	0.222	0.258	0.246	0.278	0.213	0.266	0.242	0.276	0.242	0.310	0.240	0.270	0.228	0.262	0.238	0.296
	336	**0.270**	**0.295**	0.273	0.297	0.283	0.301	0.296	0.313	0.270	0.316	0.293	0.312	0.273	0.330	0.292	0.307	0.288	0.305	0.282	0.332
	720	0.348	**0.345**	0.353	0.347	0.358	0.348	0.362	0.353	**0.337**	0.362	0.366	0.361	0.360	0.399	0.364	0.353	0.362	0.354	0.347	0.385
	Avg	**0.248**	**0.275**	0.252	0.277	0.261	0.282	0.269	0.289	0.249	0.293	0.271	0.295	0.264	0.321	0.272	0.291	0.267	0.288	0.265	0.317
PEMS03	12	**0.068**	**0.174**	0.070	0.176	0.081	0.191	0.072	0.179	0.085	0.198	0.094	0.201	0.096	0.217	0.117	0.226	0.092	0.210	0.105	0.220
	24	**0.094**	**0.205**	0.099	0.211	0.121	0.240	0.104	0.217	0.129	0.244	0.116	0.221	0.095	0.210	0.233	0.322	0.144	0.263	0.183	0.297
	36	0.116	0.229	0.126	0.240	0.180	0.292	0.137	0.251	0.173	0.286	0.134	0.237	**0.107**	**0.223**	0.379	0.418	0.200	0.309	0.258	0.361
	48	0.138	0.252	0.153	0.267	0.201	0.316	0.174	0.285	0.207	0.315	0.161	0.262	**0.125**	**0.242**	0.535	0.516	0.245	0.344	0.319	0.410
	Avg	**0.104**	**0.215**	0.112	0.223	0.146	0.260	0.122	0.233	0.149	0.261	0.126	0.230	0.106	0.223	0.316	0.370	0.170	0.282	0.216	0.322
PEMS08	12	**0.076**	**0.177**	0.080	0.184	0.091	0.199	0.084	0.187	0.096	0.205	0.111	0.208	0.161	0.274	0.121	0.233	0.106	0.223	0.113	0.225
	24	**0.107**	**0.210**	0.119	0.224	0.138	0.245	0.123	0.227	0.151	0.258	0.139	0.232	0.127	0.237	0.232	0.325	0.162	0.275	0.199	0.302
	36	**0.139**	**0.240**	0.159	0.259	0.199	0.303	0.170	0.268	0.203	0.303	0.168	0.260	0.148	0.252	0.376	0.427	0.234	0.331	0.295	0.371
	48	**0.171**	**0.265**	0.198	0.289	0.255	0.338	0.218	0.306	0.247	0.334	0.189	0.272	0.175	0.270	0.543	0.527	0.301	0.382	0.389	0.429
	Avg	**0.123**	**0.223**	0.139	0.239	0.171	0.271	0.149	0.247	0.174	0.275	0.152	0.243	0.153	0.258	0.318	0.378	0.201	0.303	0.249	0.332
1st Count		**40**	**41**	**1**	**1**	0	0	0	0	**1**	0	0	0	**3**	**2**	0	**1**	0	0	0	0

D.2 SHOWCASE

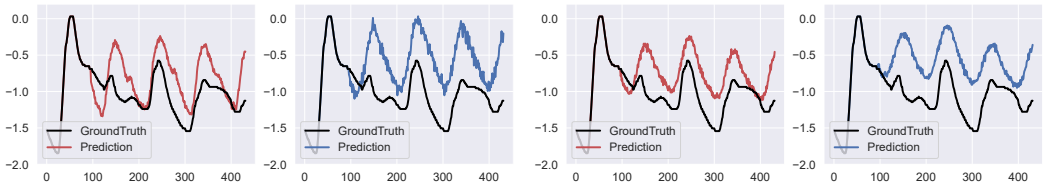
Additional experimental results of showcases are available in Fig. 4 and Fig. 5, where two datasets and two forecast models are involved.

D.3 COMPARISON WITH DIFFERENT LEARNING OBJECTIVES

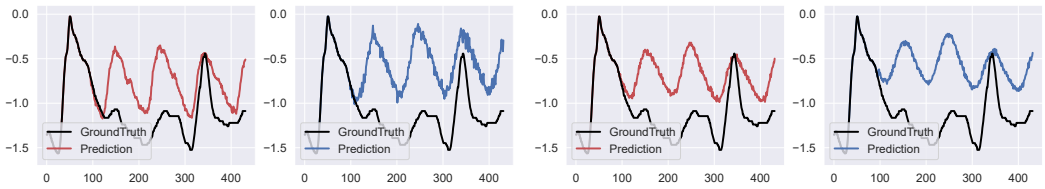
Additional experimental results of learning objective comparison are available in Table 9, where two forecast models are evaluated across different T values.



(a) Snapshot 1 with TimeBridge (b) Snapshot 1 with Fredformer

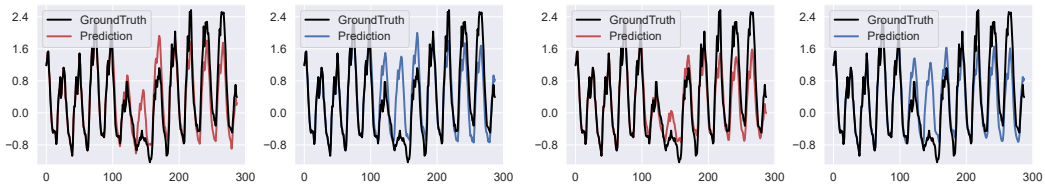


(c) Snapshot 2 with TimeBridge (d) Snapshot 2 with Fredformer

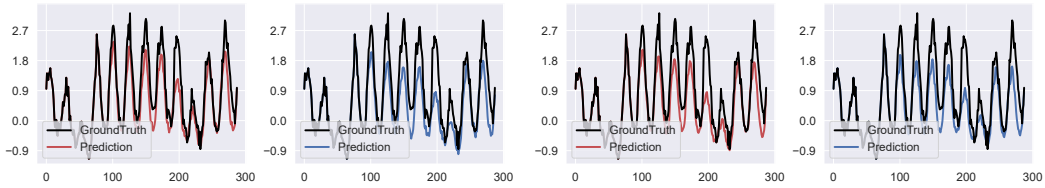


(e) Snapshot 3 with TimeBridge (f) Snapshot 3 with Fredformer

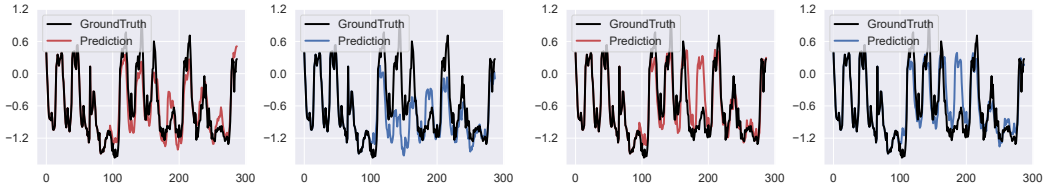
Figure 4: The forecast sequences generated with DF and DistDF. The forecast length is set to 336 and the experiment is conducted on ETTm2.



(a) Snapshot 1 with TimeBridge (b) Snapshot 1 with Fredformer



(c) Snapshot 2 with TimeBridge (d) Snapshot 2 with Fredformer

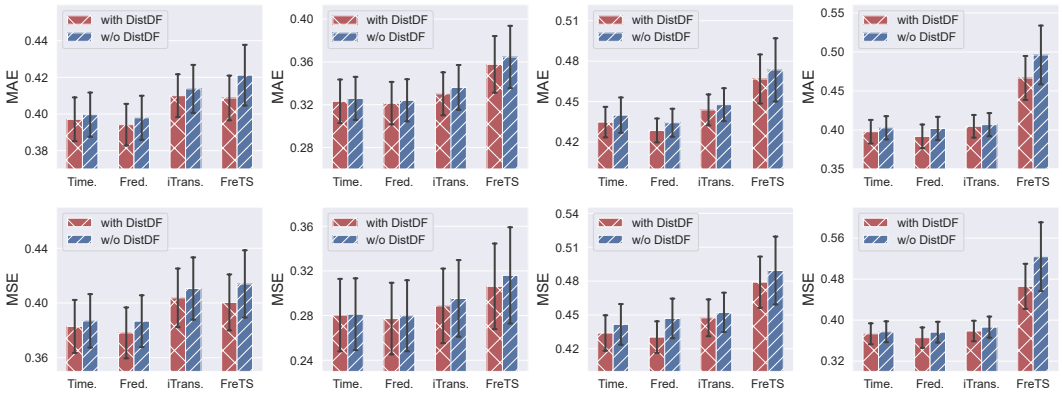


(e) Snapshot 3 with TimeBridge (f) Snapshot 3 with Fredformer

Figure 5: The forecast sequences generated with DF and DistDF. The forecast length is set to 192 and the experiment is conducted on ECL.

Table 9: Comparative results with different learning objectives.

Loss		**DistDF**		Time-o1		FreDF		Koopman		Dilate		Soft-DTW		DF	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Forecast model: TimeBridge															
ETTm1	96	0.319	0.358	0.318	0.356	0.325	0.361	0.572	0.493	0.321	0.360	0.321	0.359	0.323	0.361
	192	0.363	0.383	0.363	0.382	0.373	0.385	0.410	0.407	0.366	0.386	0.368	0.385	0.366	0.385
	336	0.394	0.405	0.396	0.407	0.398	0.406	0.397	0.408	0.397	0.409	0.405	0.410	0.398	0.408
	720	0.455	0.442	0.456	0.443	0.450	0.438	0.460	0.445	0.462	0.447	0.486	0.453	0.461	0.445
	Avg	0.383	0.397	0.383	0.397	0.386	0.398	0.460	0.438	0.387	0.400	0.395	0.402	0.387	0.400
ETTh1	96	0.372	0.392	0.372	0.391	0.373	0.391	0.376	0.397	0.376	0.396	0.376	0.395	0.373	0.395
	192	0.424	0.429	0.422	0.423	0.425	0.421	0.426	0.430	0.430	0.433	0.425	0.427	0.428	0.426
	336	0.467	0.450	0.468	0.450	0.467	0.442	0.483	0.461	0.498	0.469	0.481	0.458	0.471	0.451
	720	0.472	0.471	0.495	0.488	0.493	0.490	0.551	0.509	0.552	0.509	0.529	0.499	0.495	0.487
	Avg	0.434	0.436	0.439	0.438	0.439	0.436	0.459	0.449	0.464	0.452	0.452	0.445	0.442	0.440
ECL	96	0.137	0.235	0.148	0.240	0.137	0.232	0.170	0.266	0.142	0.240	0.139	0.235	0.142	0.239
	192	0.159	0.257	0.156	0.251	0.159	0.254	0.161	0.258	0.160	0.257	0.160	0.257	0.161	0.257
	336	0.178	0.272	0.177	0.273	0.179	0.273	0.182	0.277	0.182	0.277	0.178	0.274	0.182	0.278
	720	0.212	0.302	0.220	0.308	0.224	0.310	0.217	0.308	0.218	0.309	0.215	0.305	0.217	0.309
	Avg	0.172	0.267	0.175	0.268	0.175	0.267	0.182	0.277	0.176	0.271	0.173	0.268	0.176	0.271
Weather	96	0.164	0.209	0.166	0.209	0.174	0.213	0.215	0.261	0.168	0.211	0.169	0.209	0.168	0.211
	192	0.212	0.252	0.212	0.252	0.223	0.255	0.239	0.271	0.214	0.254	0.215	0.251	0.214	0.254
	336	0.270	0.295	0.270	0.294	0.271	0.292	0.271	0.295	0.273	0.297	0.275	0.296	0.273	0.297
	720	0.348	0.345	0.352	0.347	0.350	0.346	0.350	0.345	0.353	0.347	0.379	0.364	0.353	0.347
	Avg	0.248	0.275	0.250	0.275	0.254	0.276	0.269	0.293	0.252	0.277	0.260	0.280	0.252	0.277
Forecast model: FredFormer															
ETTm1	96	0.316	0.357	0.321	0.357	0.326	0.355	0.335	0.368	0.337	0.367	0.332	0.363	0.326	0.361
	192	0.358	0.380	0.360	0.378	0.363	0.380	0.366	0.384	0.364	0.384	0.370	0.386	0.365	0.382
	336	0.392	0.404	0.389	0.400	0.392	0.400	0.399	0.408	0.397	0.406	0.406	0.409	0.396	0.404
	720	0.448	0.437	0.447	0.435	0.455	0.440	0.456	0.441	0.457	0.443	0.478	0.450	0.459	0.444
	Avg	0.378	0.394	0.379	0.393	0.384	0.394	0.389	0.400	0.389	0.400	0.397	0.402	0.387	0.398
ETTh1	96	0.373	0.393	0.368	0.391	0.370	0.392	0.375	0.397	0.378	0.399	0.376	0.398	0.377	0.396
	192	0.428	0.425	0.424	0.422	0.436	0.437	0.438	0.434	0.439	0.435	0.439	0.435	0.437	0.425
	336	0.466	0.445	0.467	0.441	0.473	0.443	0.473	0.455	0.481	0.453	0.484	0.455	0.486	0.449
	720	0.453	0.453	0.465	0.463	0.474	0.466	0.523	0.487	0.516	0.482	0.542	0.510	0.488	0.467
	Avg	0.430	0.429	0.431	0.429	0.438	0.434	0.452	0.443	0.453	0.442	0.460	0.449	0.447	0.434
ECL	96	0.145	0.238	0.151	0.245	0.152	0.247	0.166	0.263	0.158	0.253	0.168	0.266	0.161	0.258
	192	0.162	0.255	0.166	0.256	0.166	0.257	0.174	0.267	0.170	0.263	0.218	0.313	0.174	0.269
	336	0.176	0.270	0.181	0.274	0.183	0.278	0.188	0.280	0.190	0.286	0.197	0.291	0.194	0.290
	720	0.211	0.300	0.213	0.304	0.216	0.304	0.232	0.318	0.229	0.316	0.240	0.322	0.235	0.319
	Avg	0.173	0.266	0.178	0.270	0.179	0.272	0.190	0.282	0.187	0.280	0.206	0.298	0.191	0.284
Weather	96	0.172	0.212	0.171	0.208	0.174	0.213	0.174	0.214	0.173	0.214	0.173	0.213	0.180	0.220
	192	0.218	0.255	0.219	0.253	0.219	0.254	0.220	0.256	0.225	0.260	0.220	0.255	0.222	0.258
	336	0.277	0.297	0.277	0.295	0.278	0.296	0.280	0.298	0.280	0.299	0.281	0.296	0.283	0.301
	720	0.352	0.347	0.353	0.346	0.354	0.347	0.354	0.347	0.355	0.348	0.369	0.355	0.358	0.348
	Avg	0.255	0.277	0.255	0.276	0.256	0.277	0.257	0.279	0.258	0.280	0.261	0.280	0.261	0.282



(a) ETTm1 (b) ETTm2 (c) ETTh1 (d) ETTh2

Figure 6: Performance of different forecast models with and without DistDF. The forecast errors are averaged over forecast lengths and the error bars represent 50% confidence intervals.

Table 10: Varying input sequence length results on the Weather dataset.

Models			**DistDF**		TimeBridge		**DistDF**		PatchTST	
Metrics			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Historical sequence length	96	96	0.164	0.209	0.168	0.211	0.179	0.220	0.189	0.230
		192	0.212	0.252	0.214	0.254	0.222	0.257	0.228	0.262
		336	0.270	0.295	0.273	0.297	0.278	0.298	0.288	0.305
		720	0.348	0.345	0.353	0.347	0.354	0.348	0.362	0.354
		Avg	0.248	0.275	0.252	0.277	0.258	0.281	0.267	0.288
	192	96	0.160	0.207	0.163	0.210	0.157	0.203	0.163	0.209
		192	0.202	0.244	0.205	0.248	0.202	0.244	0.207	0.249
		336	0.260	0.290	0.259	0.288	0.258	0.285	0.268	0.293
		720	0.335	0.342	0.338	0.344	0.335	0.338	0.511	0.451
		Avg	0.239	0.271	0.241	0.273	0.238	0.267	0.287	0.301
	336	96	0.155	0.206	0.156	0.206	0.153	0.204	0.158	0.208
		192	0.198	0.244	0.199	0.245	0.200	0.249	0.235	0.291
		336	0.245	0.283	0.259	0.294	0.250	0.285	0.252	0.287
		720	0.325	0.337	0.323	0.335	0.323	0.337	0.326	0.336
		Avg	0.231	0.267	0.234	0.270	0.232	0.269	0.243	0.280
	720	96	0.147	0.198	0.148	0.201	0.149	0.204	0.153	0.205
		192	0.197	0.247	0.203	0.253	0.196	0.247	0.205	0.254
		336	0.240	0.279	0.239	0.278	0.247	0.291	0.248	0.288
		720	0.319	0.339	0.329	0.346	0.313	0.333	0.317	0.339
		Avg	0.226	0.266	0.230	0.269	0.226	0.269	0.231	0.272

D.4 Generalization Studies

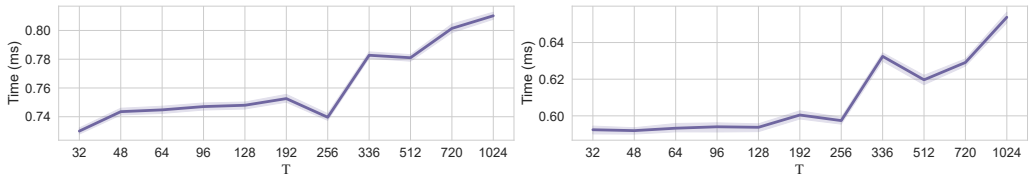
Additional experimental results of varying forecast models are available in Fig. 6, where four forecast models are involved on four datasets.

D.5 Case Study with PatchTST of Varying Historical Lengths

Additional experimental results of varying historical lengths are available in Table 10, complementing the fixed length of 96 used in the main text. The forecast models selected include TimeBridge (Liu et al., 2025) which is the recent state-of-the-art forecast model, and PatchTST (Nie et al., 2023) which is known to require large historical lengths. The results demonstrate that DistDF consistently improves both forecast models across different historical sequence lengths.

Table 11: Experimental results ($\text{mean}_{\pm\text{std}}$) with varying seeds (2021-2025).

Dataset	ECL				Weather			
Models	**DistDF**		DF		**DistDF**		DF	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	$0.138_{\pm 0.001}$	$0.236_{\pm 0.001}$	$0.141_{\pm 0.001}$	$0.239_{\pm 0.001}$	$0.167_{\pm 0.003}$	$0.209_{\pm 0.001}$	$0.169_{\pm 0.001}$	$0.212_{\pm 0.001}$
192	$0.159_{\pm 0.001}$	$0.257_{\pm 0.001}$	$0.161_{\pm 0.000}$	$0.258_{\pm 0.001}$	$0.213_{\pm 0.001}$	$0.253_{\pm 0.001}$	$0.215_{\pm 0.001}$	$0.254_{\pm 0.001}$
336	$0.179_{\pm 0.001}$	$0.272_{\pm 0.001}$	$0.183_{\pm 0.002}$	$0.279_{\pm 0.002}$	$0.271_{\pm 0.002}$	$0.296_{\pm 0.002}$	$0.272_{\pm 0.001}$	$0.296_{\pm 0.001}$
720	$0.210_{\pm 0.001}$	$0.301_{\pm 0.001}$	$0.221_{\pm 0.005}$	$0.311_{\pm 0.004}$	$0.349_{\pm 0.002}$	$0.347_{\pm 0.002}$	$0.352_{\pm 0.002}$	$0.348_{\pm 0.001}$
Avg	$0.172_{\pm 0.000}$	$0.266_{\pm 0.001}$	$0.177_{\pm 0.002}$	$0.272_{\pm 0.001}$	$0.250_{\pm 0.001}$	$0.276_{\pm 0.001}$	$0.252_{\pm 0.001}$	$0.277_{\pm 0.000}$



(a) Running time in the forward phase. (b) Running time in the backward phase.

Figure 7: Running time (ms) with varying forecast horizons.

D.6 Random Seed Sensitivity

Additional experimental results of random seed sensitivity are available in Table 11, where we report the mean and standard deviation of results obtained from experiments conducted with five different random seeds (2021, 2022, 2023, 2024, and 2025). The results indicate minimal sensitivity of the proposed method to random initialization, as most averaged standard deviations remain below 0.005.

D.7 Complexity

Additional experimental results of the running time of DistDF are available in Fig. 7. The batch size and dimension are set to 128 and 21, respectively. As the forecast horizon T increases, the running time for both forward and backward passes generally rises, with some fluctuations. This trend is expected, since T affects the size of the matrices involved in computing the joint-distribution Wasserstein discrepancy in (5). Nevertheless, the running time remains below 1 ms even when T increased to 1024. Furthermore, DistDF's additional computations occur exclusively during training and are completely isolated from the inference stage.

As a result, *DistDF introduces no additional complexity to model inference, and the extra computational cost during training is negligible.*

D.8 Joint-distribution Discrepancy in Varying Settings

Additional experimental results of joint distribution discrepancy are available for different learning objectives in Table 12 and α values in Table 13 and Table 14, as a supplement to Table 2, Table 5 and Table 6. The joint distribution discrepancy, denoted as Disc, is evaluated on the test set to compare the discrepancy between (X, Y) and $(X, \hat{Y})$.

D.9 Utility to Improve Recent Forecasting Models

Additional experimental results demonstrating utility for improving recent forecast architectures are available in Table 15. We select TQNet (Lin et al., 2025), TimeBridge (Liu et al., 2025), and FredFormer (Piao et al., 2024) as testbeds due to their recency and competitive performance.

Table 12: Joint-distribution discrepancy of different objectives for time-series forecasting.

Loss		**DistDF**	Time-o1	FreDF	Koopman	Dilate	LDTW	Soft-DTW	DTW	DF
TimeBridge	ETTm1	**0.230**	0.231	0.231	0.271	0.231	0.231	0.238	0.237	0.232
	ETTh1	**0.326**	0.331	0.330	0.350	0.352	0.352	0.340	0.344	0.332
	ECL	**0.129**	0.135	0.137	0.139	0.136	0.139	0.133	0.140	0.136
	Weather	**0.147**	0.148	0.149	0.157	0.148	0.148	0.153	0.150	0.148
Fredformer	ETTm1	**0.227**	0.228	0.231	0.232	0.233	0.233	0.240	0.239	0.232
	ETTh1	**0.324**	0.325	0.333	0.349	0.349	0.350	0.356	0.355	0.342
	ECL	**0.130**	0.133	0.134	0.142	0.140	0.144	0.153	0.151	0.143
	Weather	0.148	**0.148**	0.149	0.150	0.150	0.152	0.152	0.152	0.152

Note: **Bold** and underlined denote best and second-best Disc results, respectively. The reported results are averaged over forecast horizons: T=96, 192, 336, and 720. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

Table 13: Varying α results where Timebridge acts as the forecasting model.

α	ETTh2			ECL			Weather		
	MSE	MAE	Disc	MSE	MAE	Disc	MSE	MAE	Disc
0	0.377	0.403	0.292	0.176	0.271	0.136	0.252	0.277	0.148
0.001	0.378	0.402	0.292	0.172	0.267	0.130	0.250	0.276	0.148
0.002	0.377	0.402	0.291	0.173	0.267	**0.130**	0.250	0.276	0.148
0.005	0.376	0.401	0.291	**0.172**	**0.267**	0.130	0.250	**0.276**	0.148
0.01	0.376	0.400	0.291	0.172	0.267	0.130	**0.249**	0.276	**0.146**
0.02	0.376	0.400	0.291	0.174	0.269	0.133	0.249	0.276	0.147
0.05	**0.375**	0.399	**0.290**	0.174	0.268	0.132	0.251	0.278	0.147
0.1	0.375	**0.399**	0.291	0.174	0.269	0.132	0.254	0.280	0.148
0.2	0.376	0.399	0.291	0.177	0.270	0.134	0.254	0.280	0.148
0.5	0.378	0.400	0.294	0.186	0.277	0.140	0.256	0.281	0.149
1	0.381	0.402	0.296	0.197	0.282	0.147	0.260	0.283	0.150

Note: **Bold** and underlined denote the best and second-best results. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

D.10 Convergence Analysis

Additional experimental results on the convergence of the BW discrepancy are available in Fig. 8. The BW objective consistently exhibits a monotonic decrease throughout the training process and reaches a plateau after several epochs, thereby empirically validating the convergence of its optimization. In addition, we examine the evolution of MAE and MSE on the validation set. A significant positive correlation is observed between the dynamics of the BW loss and both forecasting metrics (MAE and MSE). It implies that minimizing the BW discrepancy effectively improves these forecasting metrics.

D.11 Autoregression-based Forecasting Performance

Additional experimental results under the autoregression-based forecasting are available in Table 16.

D.12 Probabilistic Forecasting Performance

Additional experimental results under the probabilistic forecasting setting are available in Table 17, where we select D3U (Li et al., 2025b), the state-of-the-art probabilistic forecasting framework as the testbed.

D.13 Multi-scale Forecasting Performance

Additional experimental results under the multi-scale forecasting setting are available in Table 18, where we select TimeMixer (Wang et al., 2024) and SCINet (Liu et al., 2022) as the testbeds.

E Statement on the Use of Large Language Models (LLMs)

In accordance with the conference guidelines, we disclose our use of Large Language Models (LLMs) in the preparation of this paper as follows:

Table 14: Varying α results where Fredformer acts as the forecasting model.

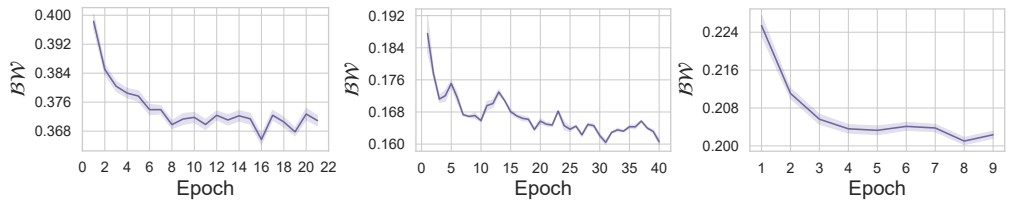
α	ETTh2			ECL			Weather		
	MSE	MAE	Disc	MSE	MAE	Disc	MSE	MAE	Disc
0	0.377	0.402	0.293	0.191	0.284	0.143	0.261	0.282	0.152
0.001	0.371	0.397	0.287	0.175	0.268	0.132	**0.255**	**0.278**	**0.148**
0.002	0.372	0.398	0.289	**0.175**	**0.267**	**0.131**	0.256	0.278	0.149
0.005	0.372	0.398	0.288	0.182	0.275	0.137	0.256	0.279	0.149
0.01	0.370	0.397	0.285	0.183	0.275	0.137	0.257	0.279	0.150
0.02	**0.369**	**0.395**	0.286	0.182	0.275	0.136	0.258	0.280	0.149
0.05	0.370	0.396	**0.285**	0.187	0.279	0.141	0.259	0.281	0.150
0.1	0.371	0.397	0.288	0.196	0.287	0.148	0.261	0.283	0.151
0.2	0.372	0.398	0.290	0.209	0.298	0.158	0.263	0.285	0.152
0.5	0.376	0.399	0.292	0.230	0.317	0.171	0.266	0.287	0.153
1	0.386	0.406	0.299	0.239	0.326	0.177	0.268	0.290	0.154

Note: **Bold** and underlined denote the best and second-best results. When metric values coincide up to three decimal places, **Bold** indicates the numerically superior result based on full precision.

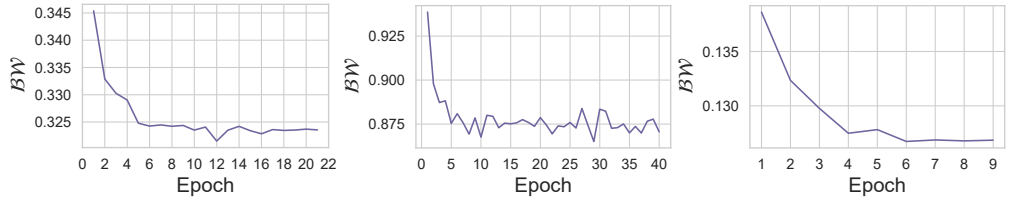
Table 15: The performance comparison of DF and DistDF on different forecast models.

Models		TQNet		TQNet†		TimeBridge		TimeBridge†		Fredformer		Fredformer†		iTransformer		iTransformer†		FreTS		FreTS†	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.372	0.391	0.372	0.391	0.373	0.395	0.372	0.392	0.377	0.396	0.373	0.393	0.385	0.405	0.383	0.403	0.398	0.409	0.399	0.409
	192	0.430	0.424	0.430	0.422	0.428	0.426	0.424	0.429	0.437	0.425	0.428	0.425	0.440	0.437	0.438	0.434	0.451	0.442	0.457	0.447
	336	0.486	0.454	0.472	0.444	0.471	0.451	0.467	0.450	0.486	0.449	0.466	0.445	0.480	0.457	0.476	0.455	0.501	0.472	0.504	0.474
	720	0.507	0.486	0.477	0.468	0.495	0.487	0.472	0.471	0.488	0.467	0.453	0.453	0.504	0.492	0.492	0.483	0.608	0.571	0.557	0.537
	Avg	0.449	0.439	0.438	0.431	0.442	0.440	0.434	0.436	0.447	0.434	0.430	0.429	0.452	0.448	0.447	0.444	0.489	0.474	0.479	0.467
ETTh2	96	0.293	0.343	0.289	0.339	0.294	0.344	0.289	0.338	0.293	0.344	0.287	0.336	0.301	0.349	0.296	0.347	0.315	0.374	0.311	0.369
	192	0.364	0.390	0.362	0.388	0.371	0.394	0.369	0.390	0.372	0.391	0.358	0.381	0.383	0.397	0.375	0.397	0.466	0.467	0.418	0.433
	336	0.411	0.424	0.410	0.424	0.421	0.429	0.415	0.426	0.420	0.433	0.408	0.421	0.425	0.432	0.421	0.434	0.522	0.502	0.521	0.505
	720	0.430	0.444	0.426	0.443	0.423	0.443	0.420	0.438	0.421	0.439	0.416	0.435	0.436	0.448	0.423	0.441	0.792	0.643	0.613	0.560
	Avg	0.375	0.400	0.371	0.399	0.377	0.403	0.373	0.398	0.377	0.402	0.367	0.393	0.386	0.407	0.379	0.405	0.524	0.496	0.466	0.467
ETTm1	96	0.310	0.352	0.311	0.351	0.323	0.361	0.319	0.358	0.326	0.361	0.316	0.357	0.338	0.372	0.334	0.372	0.342	0.375	0.335	0.371
	192	0.356	0.377	0.353	0.377	0.366	0.385	0.363	0.383	0.365	0.382	0.358	0.380	0.382	0.396	0.381	0.397	0.385	0.400	0.379	0.393
	336	0.388	0.400	0.387	0.400	0.398	0.408	0.394	0.405	0.396	0.404	0.392	0.404	0.427	0.424	0.415	0.418	0.416	0.421	0.408	0.415
	720	0.450	0.437	0.449	0.436	0.461	0.445	0.455	0.442	0.459	0.444	0.448	0.437	0.496	0.463	0.485	0.454	0.513	0.489	0.479	0.456
	Avg	0.376	0.391	0.375	0.391	0.387	0.400	0.383	0.397	0.387	0.398	0.378	0.394	0.411	0.414	0.404	0.410	0.414	0.421	0.400	0.409
ETTm2	96	0.175	0.256	0.171	0.254	0.177	0.259	0.176	0.256	0.177	0.260	0.174	0.256	0.182	0.265	0.181	0.263	0.188	0.279	0.185	0.275
	192	0.243	0.300	0.234	0.295	0.243	0.303	0.241	0.300	0.242	0.300	0.239	0.298	0.257	0.315	0.249	0.307	0.264	0.329	0.253	0.318
	336	0.297	0.336	0.292	0.333	0.303	0.343	0.302	0.340	0.302	0.340	0.300	0.338	0.320	0.354	0.311	0.347	0.322	0.369	0.338	0.386
	720	0.394	0.393	0.390	0.390	0.401	0.399	0.403	0.397	0.399	0.397	0.397	0.394	0.423	0.411	0.414	0.404	0.489	0.482	0.449	0.453
	Avg	0.277	0.321	0.272	0.318	0.281	0.326	0.280	0.323	0.280	0.324	0.277	0.321	0.295	0.336	0.289	0.330	0.316	0.365	0.306	0.358
ECL	96	0.143	0.237	0.139	0.233	0.142	0.239	0.137	0.235	0.161	0.258	0.145	0.238	0.150	0.242	0.148	0.239	0.180	0.266	0.179	0.266
	192	0.161	0.252	0.157	0.249	0.161	0.257	0.159	0.257	0.174	0.269	0.162	0.255	0.168	0.259	0.163	0.253	0.184	0.272	0.183	0.271
	336	0.178	0.270	0.174	0.267	0.182	0.278	0.178	0.272	0.194	0.290	0.176	0.270	0.182	0.274	0.176	0.270	0.199	0.290	0.199	0.288
	720	0.218	0.303	0.212	0.298	0.217	0.309	0.212	0.302	0.235	0.319	0.211	0.300	0.214	0.304	0.209	0.298	0.234	0.322	0.235	0.322
	Avg	0.175	0.265	0.171	0.262	0.176	0.271	0.172	0.267	0.191	0.284	0.173	0.266	0.179	0.270	0.174	0.265	0.199	0.288	0.199	0.287
Weather	96	0.160	0.203	0.160	0.202	0.168	0.211	0.164	0.209	0.180	0.220	0.172	0.212	0.171	0.210	0.174	0.214	0.174	0.228	0.173	0.229
	192	0.210	0.247	0.208	0.246	0.214	0.254	0.212	0.252	0.222	0.258	0.218	0.255	0.246	0.278	0.223	0.256	0.213	0.266	0.212	0.264
	336	0.267	0.289	0.264	0.287	0.273	0.297	0.270	0.295	0.283	0.301	0.277	0.297	0.296	0.313	0.280	0.299	0.270	0.316	0.263	0.305
	720	0.346	0.342	0.344	0.342	0.353	0.347	0.348	0.345	0.358	0.348	0.352	0.347	0.362	0.353	0.357	0.350	0.337	0.362	0.331	0.355
	Avg	0.246	0.270	0.244	0.269	0.252	0.277	0.248	0.275	0.261	0.282	0.255	0.277	0.269	0.289	0.258	0.280	0.249	0.293	0.245	0.288

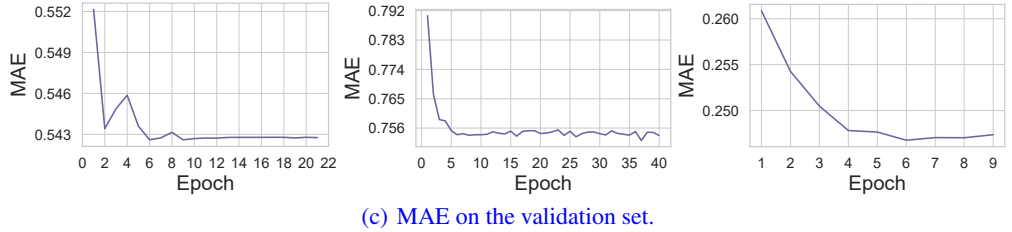
Note: The length of history window is set to 96 for all baselines. `Avg` indicates the results averaged over forecasting lengths: T=96, 192, 336 and 720. † marks the forecasting model trained via DistDF.



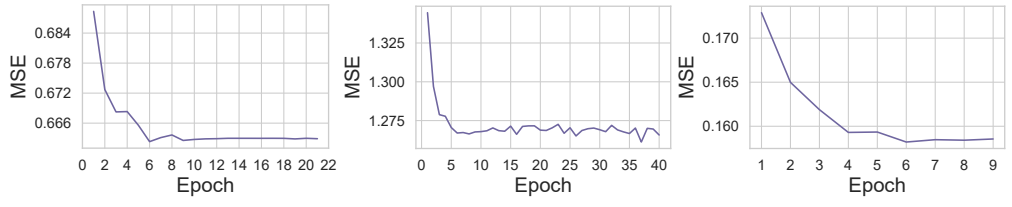
(a) Disc on the training set.



(b) Disc on the validation set.



(c) MAE on the validation set.



(d) MSE on the validation set.

Figure 8: Evolution of training objectives and validation metrics across four datasets: ETTm1, ETTh1, and ECL (from left to right).

Table 16: The performance comparison of DF and DistDF on the autoregressive forecasting setting.

Models		TimeBridge		TimeBridge†		Fredformer		Fredformer†	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.405	0.402	0.395	0.391	0.391	0.396	0.386	0.390
	192	0.467	0.438	0.419	0.408	0.494	0.449	0.493	0.446
	336	0.518	0.467	0.460	0.437	0.572	0.500	0.579	0.486
	720	0.725	0.514	0.527	0.478	1.821	0.837	0.833	0.563
	Avg	0.528	0.455	0.450	0.428	0.820	0.546	0.573	0.471
Weather	96	0.527	0.343	0.241	0.275	0.241	0.267	0.211	0.245
	192	1.165	0.494	0.303	0.320	0.306	0.318	0.274	0.292
	336	4.826	0.749	0.371	0.365	0.330	0.331	0.312	0.322
	720	9.363	1.374	0.461	0.421	0.433	0.406	0.407	0.380
	Avg	3.970	0.740	0.344	0.345	0.327	0.330	0.301	0.310

Note: The length of history window is set to 96 for all baselines. `Avg` indicates the results averaged over forecasting lengths: T=96, 192, 336 and 720. † marks the forecasting model trained via DistDF.

Table 17: The performance comparison of DF and DistDF on the probabilistic forecasting task.

Models		D3U				D3U†			
Metrics		MSE	MAE	CRPS	$CRPS_{sum}$	MSE	MAE	CRPS	$CRPS_{sum}$
ETTm1	96	0.317	0.357	0.263	0.723	0.316	0.357	0.265	0.720
	192	0.361	0.383	0.285	0.749	0.360	0.383	0.282	0.747
	336	0.394	0.404	0.299	0.742	0.390	0.402	0.298	0.731
	720	0.460	0.437	0.325	0.892	0.453	0.435	0.328	0.849
	Avg	0.383	0.395	0.293	0.776	0.380	0.394	0.293	0.762
Weather	96	0.176	0.240	0.174	0.179	0.173	0.225	0.171	0.173
	192	0.223	0.271	0.205	0.234	0.217	0.265	0.198	0.210
	336	0.279	0.309	0.233	0.269	0.278	0.310	0.233	0.260
	720	0.359	0.361	0.273	0.419	0.353	0.360	0.269	0.378
	Avg	0.259	0.295	0.221	0.275	0.255	0.290	0.218	0.255

Note: The length of history window is set to 96 for all baselines. `Avg` indicates the results averaged over forecasting lengths: T=96, 192, 336 and 720. † marks the forecasting model trained via DistDF.

Table 18: The performance comparison of DF and DistDF on the multi-scale architectures.

Models		TimeMixer		TimeMixer†		SCINet		SCINet†	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.329	0.369	0.326	0.369	0.325	0.365	0.319	0.359
	192	0.371	0.391	0.373	0.392	0.383	0.397	0.367	0.385
	336	0.427	0.425	0.412	0.423	0.436	0.424	0.403	0.406
	720	0.564	0.506	0.491	0.459	0.528	0.476	0.469	0.444
	Avg	0.422	0.423	0.401	0.411	0.418	0.416	0.389	0.399
ETTh1	96	0.419	0.426	0.400	0.410	0.409	0.415	0.397	0.405
	192	0.464	0.451	0.439	0.436	0.457	0.441	0.448	0.434
	336	0.509	0.472	0.485	0.450	0.499	0.461	0.491	0.455
	720	0.614	0.553	0.501	0.486	0.505	0.482	0.501	0.479
	Avg	0.501	0.476	0.456	0.446	0.467	0.450	0.459	0.443
ECL	96	0.159	0.260	0.145	0.242	0.146	0.248	0.141	0.242
	192	0.161	0.258	0.159	0.256	0.167	0.266	0.159	0.257
	336	0.173	0.272	0.176	0.272	0.179	0.280	0.177	0.277
	720	0.212	0.302	0.207	0.298	0.202	0.298	0.197	0.294
	Avg	0.176	0.273	0.172	0.267	0.173	0.273	0.169	0.268
Weather	96	0.173	0.220	0.168	0.217	0.160	0.208	0.158	0.207
	192	0.213	0.254	0.212	0.253	0.214	0.257	0.211	0.254
	336	0.286	0.306	0.273	0.298	0.276	0.300	0.271	0.298
	720	0.377	0.362	0.354	0.352	0.362	0.356	0.359	0.351
	Avg	0.262	0.285	0.252	0.280	0.253	0.280	0.250	0.278

Note: The length of history window is set to 96 for all baselines. `Avg` indicates the results averaged over forecasting lengths: T=96, 192, 336 and 720. † marks the forecasting model trained via DistDF.

We used LLMs (specifically, OpenAI GPT-4.1, GPT-5 and Google Gemini 2.5) *solely for checking grammar errors and improving the readability of the manuscript*. The LLMs *were not involved in research ideation, the development of research contributions, experiment design, data analysis, or interpretation of results*. All substantive content and scientific claims were created entirely by the authors. The authors have reviewed all LLM-assisted text to ensure accuracy and originality, and take full responsibility for the contents of the paper. The LLMs are not listed as an author.