DEEP TIME-SERIES FORECASTING NEEDS KERNELIZED MOMENT BALANCING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep time-series forecasting can be formulated as a distribution balancing problem aimed at aligning the distribution of the forecasts and ground truths. According to Imbens' criterion, balancing two distributions requires that their first moments be identical with respect to any balancing function. We demonstrate that existing objectives enforce moment matching only for one or two predefined balancing functions, thus failing to achieve full distribution balance. To address this limitation, we propose direct forecasting with kernelized moment balancing (KMB-DF). Unlike existing objectives, KMB-DF adaptively selects the most informative balancing functions from a reproducing kernel Hilbert space (RKHS) to enforce sufficient distribution balancing. A tractable and differentiable formulation is derived to enable estimation from empirical samples and seamless integration into gradient-based training pipelines. Extensive experiments across multiple models and datasets show that KMB-DF consistently improves forecasting accuracy and achieves state-of-the-art performance. Code is available at https://anonymous.4open.science/r/KMB-DF-403C.

1 INTRODUCTION

Deep time-series forecasting aims to predict future values from historical observations using neural networks, which supports numerical applications (Wang et al., 2024b), such as asset pricing in finance (Li et al., 2025), user visit prediction in e-commerce (Chen et al., 2023), and weather prediction in meteorology (Wu et al., 2023b). Currently, the progress in this field primarily surrounds two problems (Wang et al., 2025a; 2026a): *(1) the devise of neural architectures serving as the forecast models, and (2) the design of learning objectives driving model training.* Both are essential for improving forecasting performance (Qiu et al., 2025).

The devise of neural architectures has been widely studied. The key challenge arises from the input autocorrelation, where different steps in input sequence are correlated. To address this challenge, various architectures have been proposed (Chen et al., 2025; Lin et al., 2025). A significant advancement lies in Transformer-based models, which employ self-attention mechanisms to model input autocorrelation while scaling effectively to complex tasks (Lin et al., 2024; Liu et al., 2024a; Nie et al., 2023; Jin et al., 2024). Other progresses include recurrent neural networks (Kong et al., 2025), convolutional neural networks (Luo & Wang, 2024), and linear models (Yue et al., 2025; Zeng et al., 2023), each with unique inductive biases and strengths in modeling input autocorrelation.

The design of learning objectives has also garnered significant attention (Qiu et al., 2025; Kudrat et al., 2025). The key challenge arises from the label autocorrelation, where different steps in label sequence are correlated, which renders standard objectives like mean squared error (MSE) biased. To address this, distribution balancing methods train forecast models by balancing the distributions of forecast and label sequences, ensuring unbiased training despite label autocorrelation (Wang et al., 2026b). However, ideal distribution balance requires identical first moments with respect to any balancing function (Imbens & Rubin, 2015). We demonstrate that existing objectives enforce this equivalence for only one or two predefined functions (Wang et al., 2025a;b; Qiu et al., 2025), thus failing to achieve the full distribution balance necessary for unbiased forecast model training.

To address the limitation of existing objectives, we propose direct forecasting with kernelized moment balancing (KMB-DF). The core strategy is to adaptively select the most informative balancing functions from a reproducing kernel Hilbert space (RKHS) and align first moments with these

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

functions to define the learning objective. Leveraging the rich capacity of RKHS associated with universal kernels, KMB-DF enforces sufficient distribution balance with theoretical guarantees. Furthermore, the objective formulation is tractable and differentiable, enabling efficient estimation from finite samples and seamless integration into gradient-based training. Experimental results confirm that KMB-DF outperforms existing learning objectives and achieves state-of-the-art performance.

Contributions. The main contributions can be summarized as follows. ❶ **We identify a critical limitation of existing learning objectives for time-series forecasting**: they enforce first-moment equivalence for only one or two balancing functions, failing to achieve full distribution balance and thus unbiased training. ❷ **We propose KMB-DF, a learning objective that aligns first moments with most informative balancing functions.** This enforces sufficient distribution balance with theoretical guarantees while permitting gradient-based optimization. ❸ **We evaluate KMB-DF through extensive experiments**, demonstrating that it consistently outperforms existing objectives, generalizes across various forecast models, and achieves state-of-the-art performance.

2 PRELIMINARIES

2.1 PROBLEM DEFINITION

In this section, we introduce the deep time-series forecasting problem. We use uppercase bold letters (*e.g.*, $\mathbf{X}$) to denote matrices, lowercase bold letters (*e.g.*, $\mathbf{x}$) to denote vectors, and lowercase normal letters (*e.g.*, x) to denote scalars. A time-series consists of a sequence of chronologically ordered observations, denoted as $\mathbf{s} = \{s_1, s_2, ..., s_{\mathrm{M}}\} \in \mathbb{R}^{\mathrm{M}}$, where M is the total number of observations. Here we adopt the univariate setting, which generalizes naturally to the multivariate setting.

At a given time step n, the task definition requires the following components: (1) the *history sequence* $\mathbf{x} = [s_{n-\mathrm{H}+1}, \ldots, s_n] \in \mathbb{R}^{\mathrm{H}}$, where H is the history length; (2) the *label sequence* $\mathbf{y} = [s_{n+1}, \ldots, s_{n+\mathrm{T}}] \in \mathbb{R}^{\mathrm{T}}$, where T is the forecast horizon; (3) the *forecast model* g, which is a neural network mapping $\mathbf{x}$ to a forecast sequence $\hat{\mathbf{y}}$. The task is to train a neural network g by optimizing an appropriate learning objective, such that for any given $\mathbf{x}$, the forecast $\hat{\mathbf{y}}$ accurately approximates the ground-truth label sequence $\mathbf{y}$ (Wang et al., 2025a; 2024b).

2.2 LEARNING OBJECTIVES FOR DEEP TIME-SERIES FORECASTING

The learning objective plays a central role in deep time-series forecasting, because it governs how forecasting models are optimized. A widespread objective is mean squared error (MSE) (Yi et al., 2023; Liu et al., 2024a; Yue et al., 2025), measuring the point-wise divergence between $\mathbf{y}$ and $\hat{\mathbf{y}}$:

$$\mathcal{E}_{\mathrm{MSE}} = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 = \sum_{t=1}^{\mathrm{T}} (y_t - \hat{y}_t)^2 . \tag{1}$$

While widely applied in modern literature, $\mathcal{E}_{\mathrm{MSE}}$ has a critical limitation: it treats future observations as conditionally independent, ignoring the label autocorrelation structure where y_t depends on its predecessors $\mathbf{y}_{<t}$ (Zeng et al., 2023). Consequently, $\mathcal{E}_{\mathrm{MSE}}$ constitutes a biased objective. This phenomenon, termed autocorrelation bias, is formalized in Lemma 2.1.

Lemma 2.1 (Autocorrelation bias)**.** *Let $\mathbf{y} \in \mathbb{R}^{\mathrm{T}}$ be a label sequence with conditional covariance $\mathbf{\Sigma} \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$. $\mathcal{E}_{\mathrm{MSE}}$ is biased against the likelihood of $\mathbf{y}$ unless $\mathbf{\Sigma}$ is identity, i.e., steps in $\mathbf{y}$ are conditionally independent given $\mathbf{x}$ (Wang et al., 2025a).*

To accommodate label autocorrelation, recent work has developed alternative learning objectives along two main lines. The first applies a transform function ϕ to obtain latent components, $\phi(\hat{\mathbf{y}})$ and $\phi(\mathbf{y})$, and aligns them component-wise (Wang et al., 2025b; Qiu et al., 2025; Wang et al., 2025a). The second line seeks to align the shapes of $\mathbf{y}$ and $\hat{\mathbf{y}}$, motivated by the observation that label autocorrelation is reflected in the temporal structure of the label sequence. This line can be traced back to dynamic time warping (DTW), which measures shape dissimilarity through the minimum-cost alignment of two sequences, but its non-differentiability limits its use in training neural networks. To address this, SoftDTW (Cuturi & Blondel, 2017) provides a differentiable relaxation and has been applied across domains (Song et al., 2022; Liao et al., 2025). Building on SoftDTW, Le Guen & Thome (2019) incorporate a temporal distortion index to penalize lag mismatches, while Le Guen &

Thome (2023) extend the approach to probabilistic forecasting. *Despite the empirical effectiveness of the works described above, they fail to guarantee theoretical unbiasedness as learning objectives under label autocorrelation.* For transform-based objectives, Lemma 2.1 establishes unbiasedness only when the resulting components are conditionally decorrelated, a condition that existing transform functions do not guarantee (Wang et al., 2026a). For shape-based objectives, they do not provide theoretical guarantee of unbiasedness under autocorrelated labels[1].

2.3 Distribution balancing and Imbens' criterion

Suppose $\mathbb{P}_1$ and $\mathbb{P}_2$ are two distributions, ideal distribution balance is attained when the two distributions are indistinguishable with respect to all relevant statistics. This condition is characterized by Imbens' criterion (Imbens & Rubin, 2015) as:

$$\forall \phi : \mathbb{E}_{z\in\mathbb{P}_1}\left[\phi(z)\right] = \mathbb{E}_{z\in\mathbb{P}_2}\left[\phi(z)\right], \tag{2}$$

where ϕ is an arbitrary balancing function. This equality is required to hold for any choice of ϕ, which ensures that $\mathbb{P}_1$ and $\mathbb{P}_2$ have equivalent first moment under any transformation. This criterion is widely used to assess distribution balancing in diverse fields, such as causal inference (Imai & Ratkovic, 2014; Fong et al., 2018) and trustworthy machine learning (Li et al., 2023).

3 Methodology

3.1 Motivation

The training of deep time-series forecast models can be interpreted as a distribution balancing problem (Wang et al., 2026b). The forecast model is considered accurately trained if $\mathbb{P}(\mathbf{y}|\mathbf{x})$ and $\mathbb{P}(\hat{\mathbf{y}}|\mathbf{x})$ are balanced, as it correctly captures the underlying generation process of the label sequence. To achieve distribution balancing, likelihood maximization offers a practical way, exemplified by $\mathcal{E}_{\text{MSE}}$ and its variants (Wang et al., 2025b;a; 2026a). However, they suffer from autocorrelation bias in Lemma 2.1, thereby failing to ensure distribution balancing and theoretical unbiasedness.

Imbens' criterion characterizes distributional balance through Eq. (2), providing a formal basis for assessing the limitations of existing learning objectives. As shown in Table 1, these objectives seek to balance distributions with respect to a restricted set of balancing functions ϕ. For example, $\mathcal{E}_{\text{MSE}}$ seeks to satisfy Eq. (2) with the identity map, whereas FreDF uses the Fourier transform. However, each objective considers only **one or two prespecified** balancing functions, while Imbens' criterion requires Eq. (2) to hold for **all** admissible ϕ. Consequently, optimizing these objectives is insufficient to guarantee full distribution balance for training forecasting models.

Table 1: The choice of ϕ in recent learning objectives for deep time-series forecasting.

Method	Venue	Choice of ϕ	# ϕ
FreDF	ICLR-25	Discrete Fourier transform.	1
Time-o1	NeurIPS-25	Principal component analysis.	1
DBLoss	NeurIPS-25	Trend-seasonal EMA decomposition.	2
QDF	ICLR-26	Meta-learned linear mapping.	1
DistDF	ICLR-26	Mean and variance statistics.	2

The promise of distribution balancing, together with the limitations of existing learning objectives, motivates objectives that explicitly target distribution balance as defined by Imbens' criterion. This raises three research questions: *(1) How can Imbens' criterion be used to formulate a learning objective for training forecasting models? (2) Can such an objective incorporate a broader class of balancing functions? (3) Does it improve forecasting performance in practice?*

3.2 Distribution balancing with specified ϕ

In this section, we develop a distribution balancing strategy to train forecast models. Notably, directly balancing the conditional distributions $\mathbb{P}(\mathbf{y}|\mathbf{x})$ and $\mathbb{P}(\hat{\mathbf{y}}|\mathbf{x})$ is intractable given finite-sample

[1] As a growing literature, there are other learning objectives for training deep forecast models (Xiong et al., 2025; Kudrat et al., 2025; Wang et al., 2024a; Wu et al., 2020). While they improve upon $\mathcal{E}_{\text{MSE}}$, they are not explicitly designed to accommodate label autocorrelation (Rasul et al., 2021; Pan et al., 2025) central to this paper. Thus, we list them for completeness but omit further discussion for brevity.

time-series datasets[2]. We instead balance the joint distributions $\mathbb{P}(\mathbf{x}, \mathbf{y})$ and $\mathbb{P}(\mathbf{x}, \hat{\mathbf{y}})$, which is effective for two reasons (Wang et al., 2026b): (i) $\mathbb{P}(\mathbf{x}, \mathbf{y}) = \mathbb{P}(\mathbf{x}, \hat{\mathbf{y}})$ implies $\mathbb{P}(\mathbf{y}|\mathbf{x}) = \mathbb{P}(\hat{\mathbf{y}}|\mathbf{x})$ for all $\mathbf{x}$, thus ensuring effective training of forecast models; (ii) joint balancing is tractable since the joint distributions can be estimated and balanced using all dataset samples.

Let $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ and $\hat{\mathbf{z}} = (\mathbf{x}, \hat{\mathbf{y}})$ denote the joint random variables. According to Imbens' criterion in Eq.(2), the two joint distributions are balanced if we have

$$\forall \phi : \mathbb{E}_{\mathbf{z} \in \mathbb{P}(\mathbf{z})} \left[\phi(\mathbf{z})\right] = \mathbb{E}_{\hat{\mathbf{z}} \in \mathbb{P}(\hat{\mathbf{z}})} \left[\phi(\hat{\mathbf{z}})\right]. \tag{3}$$

Given a finite time-series dataset $\{\mathbf{z}_n = (\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{\mathrm{N}}$ and corresponding forecasts $\{\hat{\mathbf{z}}_n = (\mathbf{x}_n, \hat{\mathbf{y}}_n)\}_{n=1}^{\mathrm{N}}$ where $\hat{\mathbf{y}}_n = g_\theta(\mathbf{x}_n)$. One can attempt to enforce Eq.(3) over a chosen set of K balancing functions $\{\phi_k\}_{k=1}^{\mathrm{K}}$. This yields the constrained learning problem as follow:

$$\min_\theta \sum_{n=1}^{\mathrm{N}} \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_2^2, \quad \text{s.t.} \sum_{n=1}^{\mathrm{N}} \phi_k(\mathbf{z}_n) = \sum_{n=1}^{\mathrm{N}} \phi_k(\hat{\mathbf{z}}_n), \forall k \in [\mathrm{K}] \tag{4}$$

where each constraint enforces balance with respect to certain balancing function ϕ_k. In contrast to the objectives in Table 1, the formulation in Eq.(4) increases the number of balancing functions to K. As K grows, the constraints in Eq.(4) increasingly guarantee the Imbens' criterion, thereby promoting distribution balance between $\mathbb{P}(\mathbf{z})$ and $\mathbb{P}(\hat{\mathbf{z}})$.

However, the formulation Eq.(4) relies on a predefined set of balancing functions $\{\phi_k\}_{k=1}^{\mathrm{K}}$. In practice, selecting a finite set of balancing functions that sufficiently depicts distribution discrepancies is difficult. To guarantee Eq.(3), one would need to specify infinitely many balancing constraints, rendering Eq.(4) computationally intractable. Therefore, it is essential to develop tractable relaxations that selectively enforce balance over the most informative balancing functions.

3.3 Kernel function, universal property, & kernelized soft-margin balancing

In this section, we leverage reproducing kernel Hilbert space (RKHS) theory to construct a rich, data-adaptive, and theoretically grounded set of balancing functions and derive a tractable learning problem, thereby overcoming the reliance on a predefined finite set of balancing functions. We begin by clarifying key properties of kernel functions in Definitions 3.1 and 3.2.

Definition 3.1 (Kernel function)**.** Let $\mathcal{Z}$ be a non-empty set. A function $\mathcal{K} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a kernel function if there exist a Hilbert space $\mathcal{H}$ and a feature map $\psi : \mathcal{Z} \to \mathcal{H}$ such that $\forall\, \mathbf{z}, \mathbf{z}' \in \mathcal{Z}$, $\mathcal{K}(\mathbf{z}, \mathbf{z}') := \langle \psi(\mathbf{z}), \psi(\mathbf{z}') \rangle_{\mathcal{H}}$.

Definition 3.2 (Universal kernel)**.** For $\mathcal{Z}$ a compact Hausdorff space, a universal kernel ensures that any continuous function $e : \mathcal{Z} \to \mathbb{R}$ can be approximated arbitrarily well within its RKHS. Specifically, for any $\epsilon > 0$, there exists $f \in \mathcal{H}$ such that: $\sup_{\mathbf{z} \in \mathcal{Z}} |f(\mathbf{z}) - e(\mathbf{z})| \leq \epsilon$.

The exponential kernel, $\mathcal{K}(\mathbf{z}, \mathbf{z}') = \exp(-\frac{|\mathbf{z}-\mathbf{z}'|}{2\sigma^2})$, is universal in the sense of Definition 3.2. It implies that functions in the RKHS of $\mathcal{K}$, defined as $\mathcal{H} = \mathrm{span}\{\mathcal{K}(\cdot, \mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}\}$, can approximate any continuous balancing function. Motivated by this property, it is natural to instantiate the balancing functions as exponential kernel functions anchored by $\{\mathbf{z}_k\}_{k=1}^{\mathrm{K}}$:

$$\phi_k(\cdot) = \mathcal{K}(\cdot, \mathbf{z}_k), \quad \text{i.e.}, \forall \mathbf{z}, \phi_k(\mathbf{z}) = \mathcal{K}(\mathbf{z}, \mathbf{z}_k),$$

which yields K balancing functions. On this basis, substituting $\phi_k(\cdot) = \mathcal{K}(\cdot, \mathbf{z}_k)$ into Eq.(4) yields the following kernelized constraints that are directly computable from the data:

$$\min_\theta \sum_{n=1}^{\mathrm{N}} \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_2^2, \quad \text{s.t.} \sum_{n=1}^{\mathrm{N}} \mathcal{K}(\mathbf{z}_n, \mathbf{z}_k) = \sum_{n=1}^{\mathrm{N}} \mathcal{K}(\hat{\mathbf{z}}_n, \mathbf{z}_k), \forall k \in [\mathrm{K}] \tag{5}$$

One could directly set $\mathrm{K} = \mathrm{N}$, *i.e.*, using balancing functions anchored at all samples. However, the resulting N constraints are computationally prohibitive and can over-constrain large time-series datasets. To obtain a tractable formulation, two refinements to Eq.(5) are introduced as follows.

[2]Specifically, for any fixed $\mathbf{x}$, the dataset contains only one associated label sequence, and the model produces only one forecast sequence. Thus, only one sample can be observed from each of $\mathbb{P}(\mathbf{y}|\mathbf{x})$ and $\mathbb{P}(\hat{\mathbf{y}}|\mathbf{x})$, which is insufficient to characterize—and therefore balance—these conditional distributions.

Algorithm 1 The workflow of KMB-DF.

Input: $\mathbf{X}$: history sequences, $\mathbf{Y}$: label sequences.
Parameter: α: the strength of penalty, K: the number of balancing functions, C: the tolerated imbalance, g_θ: the forecast model with parameters θ.
Output: $\mathcal{L}_{\mathrm{KMB-DF}}$: the obtained learning objective.

1: $\hat{\mathbf{Y}} \leftarrow g_\theta(\mathbf{X})$ ▷ Forecast sequence generation
2: $\mathbf{Z} = \mathrm{concat}(\mathbf{X}, \mathbf{Y}), \hat{\mathbf{Z}} = \mathrm{concat}(\mathbf{X}, \hat{\mathbf{Y}})$ ▷ Extended sequence construction
3: $\delta_k = \sum_{n=1}^{\mathrm{N}} [\mathcal{K}(\mathbf{z}_n, \mathbf{z}_k) - \mathcal{K}(\hat{\mathbf{z}}_n, \mathbf{z}_k)], k = 1, ..., \mathrm{N}$ ▷ Informativeness score computation
4: $n_1, ..., n_{\mathrm{K}} \leftarrow \arg \mathrm{TopK}_k \delta_k$ ▷ Balancing function selection
5: $\xi_{n_k} = [-\mathrm{C} - \delta_{n_k}]_+ + [\delta_{n_k} + \mathrm{C}]_+, k = 1, ..., \mathrm{K}$ ▷ Penalty computation
6: $\mathcal{E}_{\mathrm{KMB-DF}} \leftarrow \alpha \sum_{k=1}^{\mathrm{K}} \xi_{n_k} + (1-\alpha) \sum_{n=1}^{\mathrm{N}} \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_2^2$ ▷ Learning objective computation

- First, we introduce a selective balancing mechanism. Noticing that different balancing functions have different abilities to detect imbalance, we quantify the informativeness score of ϕ_k as:

$$\delta_k = \sum_{n=1}^{\mathrm{N}} \mathcal{K}(\mathbf{z}_n, \mathbf{z}_k) - \sum_{n=1}^{\mathrm{N}} \mathcal{K}(\hat{\mathbf{z}}_n, \mathbf{z}_k), \quad \forall k \in [\mathrm{N}] \tag{6}$$

  where larger $|\delta_k|$ signifies greater imbalance. We select only K ($\mathrm{K} < \mathrm{N}$) balancing functions with the largest $|\delta_k|$, and denote their associated anchor samples as $\{\mathbf{z}_{n_k}\}_{k=1}^{\mathrm{K}}$. By prioritizing the most informative balancing functions, we reduce the number of constraints while maintaining effective distribution balance. This strategy is theoretically grounded in Theorem 3.3, which guarantees distribution alignment between $\mathbb{P}(\mathbf{z})$ and $\mathbb{P}(\hat{\mathbf{z}})$ provided that the optimal discrimination function lies within the linear span of the selected balancing functions $\{\mathcal{K}(\cdot, \mathbf{z}_{n_k})\}_{k=1}^{\mathrm{K}}$.
- Second, we introduce a soft-margin relaxation mechanism. Since the equality constraints in Eq.(5) might be overly rigid—particularly for time-series data, where outliers or abrupt shocks may induce large but spurious imbalances—we replace each equality constraint with a pair of inequality constraints that allow bounded violations, and penalize these violations in the objective.

Theorem 3.3. *Suppose $\mathcal{H}$ is the RKHS of a exponential kernel $\mathcal{K}$, $f^* \in \mathcal{H}$ is the discrimination function that maximizes the discrepancy between two distributions $\mathbb{P}$ and $\mathbb{Q}$. If f^* lies within the linear span of a finite kernel function set: $f^* \in \{\mathcal{K}(\cdot, \mathbf{z}_k)\}_{k=1}^{\mathrm{K}}$; then $\mathbb{P}$ and $\mathbb{Q}$ are balanced if $\forall k \in [\mathrm{K}] : \mathbb{E}_{\mathbf{z} \in \mathbb{P}}[\mathcal{K}(\mathbf{z}, \mathbf{z}_k)] = \mathbb{E}_{\hat{\mathbf{z}} \in \mathbb{Q}}[\mathcal{K}(\hat{\mathbf{z}}, \mathbf{z}_k)]$.*

The two efforts collectively yield the following kernelized soft-margin balancing problem:

$$\begin{aligned} \min_{\theta, \boldsymbol{\varepsilon}} \quad & \sum_{n=1}^{\mathrm{N}} \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_2^2 + \kappa \sum_{k=1}^{\mathrm{K}} \varepsilon_k \\ \text{s.t.} \quad & \left| \sum_{n=1}^{\mathrm{N}} [\mathcal{K}(\mathbf{z}_n, \mathbf{z}_{n_k}) - \mathcal{K}(\hat{\mathbf{z}}_n, \mathbf{z}_{n_k})] \right| \leq C + \varepsilon_k, \quad k = 1, \ldots, \mathrm{K}, \\ & \varepsilon_k \geq 0, \quad k = 1, \ldots, \mathrm{K}, \end{aligned} \tag{7}$$

where $C \geq 0$ specifies tolerated imbalance, $\kappa > 0$ controls the penalty for exceeding this tolerance, and ε_k are slack variables. This formulation preserves the balancing principle underlying Imbens' criterion while offering a computationally tractable objective given large-scale time-series data.

The formulation in Eq. (7) matches moments for K adaptively selected balancing functions. This provides a tractable approximation to Imbens' criterion, though balancing a finite set does not generally guarantee balance over all admissible functions. *Nevertheless, relative to existing objectives that employ at most two fixed balancing functions, it improves by incorporating a broader, data-adaptive set of balancing functions*—a significant step forward. Moreover, Theorem 3.3 demonstrate that ideal distribution balancing can be achieved when f^* lies in the span of the selected functions.

3.4 Model implementation

In this section, we introduce the implementation of KMB-DF, which adapts the kernelized soft-margin balancing problem to train a deep forecast model g_θ, as detailed in Algorithm 1.

Given a set of history sequences $\mathbf{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{H}}$ and corresponding label sequences $\mathbf{Y} \in \mathbb{R}^{\mathrm{N}\times\mathrm{T}}$, the forecast model generates the forecast sequences (step 1). To perform joint distribution balancing, we construct the extended sequences (step 2), where $\mathrm{concat}(\cdot,\cdot)$ denotes concatenation along the time dimension. Then, we compute informativeness scores for all balancing functions and retain the top K functions with largest $|\delta_k|$ (steps 3-4). The final learning objective is:

$$\begin{aligned} \mathcal{E}_{\mathrm{KMB-DF}} &= \alpha \sum_{k=1}^{\mathrm{K}} \xi_{n_k} + (1-\alpha) \sum_{n=1}^{\mathrm{N}} \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|_2^2, \\ \text{where } \xi_{n_k} &= [-\mathrm{C} - \delta_{n_k}]_+ + [\delta_{n_k} + \mathrm{C}]_+, \delta_{n_k} = \sum_{n=1}^{\mathrm{N}} \left[\mathcal{K}(\mathbf{z}_n, \mathbf{z}_{n_k}) - \mathcal{K}(\hat{\mathbf{z}}_n, \mathbf{z}_{n_k})\right], \end{aligned} \tag{8}$$

where $[\cdot]_+ = \max(\cdot, 0)$, $\alpha \in [0,1]$ controls the strength of penalty for violating balancing constraints, and C controls the tolerance margin for imbalance. This unconstrained objective is directly obtained from Eq.(7) by introducing the hinge penalties ξ_k, enabling gradient-based optimization (step 6). By iterative minimization, KMB-DF improves forecast performance by effectively balancing joint distributions while preserving the benefits of the canonical DF framework (Zeng et al., 2023), such as efficient multi-task training and inference. Moreover, as a model-agnostic learning objective, KMB-DF can be integrated into various forecast models to improve performance.

4 Experiments

In this section, we conduct an empirical investigation to demonstrate the utility of KMB-DF in training forecast models across five main aspects as follows:

1. **Performance:** *Does KMB-DF perform well?* In Section 4.2, we compare KMB-DF with previously developed learning objectives for training forecast models.
2. **Gain:** *Why does it work?* In section 4.3, we perform an ablative study, dissecting the individual components and clarifying their contributions to forecast accuracy.
3. **Generality:** *Does it support other models and kernels?* In Section 4.4, we analyze the impact of kernel selection, and in Section 4.5, we examine its compatibility with various models.
4. **Sensitivity:** *Is it sensitive to hyperparameters?* In Section 4.6, we analyze the sensitivity to the hyperparameters α, K and C, showing stable performance across a broad parameter range.
5. **Efficiency:** *Is it computationally efficient?* In Appendix, we evaluate the running cost of KMB-DF across different scenarios.

4.1 Experimental Setup

Datasets. We evaluate our methods using several standard public benchmarks for long-term time-series forecasting, following Wu et al. (2023a). Specifically, we use the ETT (four subsets), ECL, and Weather (Liu et al., 2024a) datasets. Additionally, we introduce M5 dataset (Makridakis et al., 2022), which is notably challenging. All datasets are split chronologically into training, validation, and test sets. Comprehensive dataset statistics are presented in Appendix C.1.

Baselines. Since this paper focuses on the devise of learning objectives, we select competitive learning objectives tailored for training forecast models as baselines following codebase of Wang et al. (2026b): ❶ **shape-alignment objectives:** GDTW (Liu et al., 2021), Dilate (Le Guen & Thome, 2019), and Soft-DTW (Cuturi & Blondel, 2017); ❷ **likelihood maximation objectives:** QDF (Wang et al., 2026a), Time-o1 (Wang et al., 2025a), Koopman (Lange et al., 2021), FreDF (Wang et al., 2025b), and MSE; ❸ **component-wise alignment objectives:** DBLoss (Qiu et al., 2025); ❹ **distribution balancing objectives:** DistDF (Wang et al., 2026b).

Implementation. We employ CFPT (Kou et al., 2025) as the default forecast model in comparing learning objectives. To ensure fair comparison, the drop-last trick is disabled as recommended in Qiu et al. (2024). All objectives are trained with the Adam optimizer (Kingma & Ba, 2015). When integrating KMB-DF to train a forecast model, we retain all hyperparameters from the public benchmarks (Kou et al., 2025), only tuning α, the learning rate, the tolerated imbalance C and the number of balancing functions K. Experiments are run on Intel(R) Xeon(R) Platinum 8463B CPUs with 32 NVIDIA RTX H800 GPUs. More implementation details are provided in Appendix C.

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

Loss	**KMB-DF**		QDF		DistDF		Time-o1		FreDF		DBLoss		Koopman		GDTW		Dilate		Soft-DTW		MSE	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.372**	**0.391**	0.375	0.393	0.375	0.393	0.378	0.393	0.376	0.392	0.379	0.392	0.378	0.395	0.398	0.412	0.379	0.396	0.394	0.403	0.378	0.394
ETTm2	**0.265**	**0.312**	0.268	0.315	0.269	0.315	0.271	0.314	0.271	0.313	0.273	0.314	0.274	0.319	0.289	0.332	0.273	0.317	0.291	0.330	0.271	0.317
ETTh1	**0.424**	**0.426**	0.434	0.429	0.434	0.428	0.430	0.429	0.434	0.428	0.436	0.429	0.437	0.431	0.452	0.444	0.441	0.434	0.461	0.445	0.436	0.430
ETTh2	**0.361**	**0.391**	0.367	0.396	0.365	0.395	0.367	0.394	0.368	0.394	0.371	0.394	0.368	0.397	0.386	0.409	0.371	0.398	0.393	0.409	0.372	0.398
ECL	**0.163**	0.258	0.164	0.259	0.164	0.259	0.165	0.259	0.166	**0.258**	0.167	0.259	0.165	0.260	0.861	0.755	0.165	0.260	0.165	0.261	0.165	0.260
Weather	**0.239**	**0.265**	0.241	0.267	0.241	0.268	0.240	0.266	0.240	0.265	0.245	0.266	0.246	0.272	0.248	0.276	0.246	0.273	0.262	0.281	0.241	0.267
M5	**0.152**	**0.309**	0.156	0.313	0.156	0.313	0.161	0.319	0.168	0.327	0.162	0.321	0.158	0.317	0.160	0.318	0.159	0.317	0.162	0.322	0.168	0.327

Note: We fix the input length as 96 following (Liu et al., 2024a). **Bold** and underlined denote best and second-best results, respectively. *Avg* indicates average results over forecast horizons: T=96, 192, 336 and 720.

4.2 Overall Performance

Table 2 presents a comprehensive comparison between KMB-DF and competitive learning objectives. We have the following observations: ❶ **The naive MSE objective exhibits suboptimal performance.** For example, it yields MSE/MAE values of 0.378/0.394 on ETTm1, 0.165/0.260 on ECL, and 0.168/0.327 on M5, all of which are higher than those of KMB-DF. This result is consistent with MSE optimizing each forecast step independently, without explicitly accounting for the autocorrelation among future labels described in Section 2.2. ❷ **Existing alternative objectives offer limited improvements over MSE.** QDF and DistDF reduce the MSE from 0.378 to 0.375 on ETTm1 and from 0.372 to 0.365 on ETTh2, respectively, while FreDF matches the best MAE on ECL and Weather. In contrast, shape-alignment objectives can yield substantially larger errors on some datasets: GDTW reports an MSE of 0.861 and an MAE of 0.755 on ECL. By aligning only a few predefined transformed components or statistics, these objectives cannot consistently achieve distribution balance, which leads to suboptimal performance. ❸ **KMB-DF achieves the best overall results among the compared objectives.** It obtains the lowest MSE and MAE on almost all datasets, while tying the best MAE on ECL with FreDF. These results demonstrate the effectiveness of adaptively selected kernelized balancing functions for reducing distributional discrepancies.

Showcases. We qualitatively compare the forecast sequences produced by models trained with KMB-DF and MSE. Figure 1 presents a snapshot from ETTm2 under lookback window $H = 96$. Although the MSE-trained model captures overall periodic pattern, its prediction deviates noticeably around the transition between steps 200 and 300 and overestimates the subsequent peak. In contrast, KMB-DF follows these local variations more closely. By adaptively selecting kernelized balancing functions to align the joint distributions of labels and forecasts, KMB-DF enables the model to better capture the temporal dependencies within the label sequence.

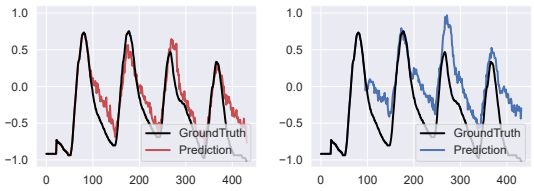


Figure 1: The forecast sequence of KMB-DF (in red) and MSE (in blue).

4.3 Ablation Studies

In this section, we dissect the contributions of the soft-margin relaxation and the selective balancing mechanism inherent in KMB-DF. Table 3 examines the performance of variants by selectively enabling these components. The main findings are as follows: ❶ **Soft-margin relaxation improves forecasting performance.** KMB-DF† adds slack variables to DF while retaining all balancing functions. It reduces the average MSE from 0.378 to 0.374 on ETTm1 and from 0.436 to 0.430 on ETTh1, confirming the benefit of allowing bounded moment mismatches in the balancing objective. ❷ **Selective balancing also benefits forecasting performance.** KMB-DF‡ uses the informativeness score in Eq. (6) while enforcing strict constraints. It reduces the average MSE by 0.79% on ETTm1 and 2.06% on ETTh1, demonstrating the effectiveness of selecting balancing functions rather than balancing all candidate functions uniformly. ❸ **The two ingredients yield complementary gains.** This is supported by the leading performance of the full KMB-DF across all datasets, forecasting horizons, and metrics. For example, it reduces the average MSE by 1.59% on ETTm1 and 2.75% on ETTh1 relative to DF. This demonstrates their complementary benefits for forecasting.

Table 3: Ablation study results.

Model	Soft-Margin	Selective	Data	T=96		T=192		T=336		T=720		Avg	
				MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DF	✗	✗	ETTm1	0.321	0.358	0.357	0.382	0.387	0.401	0.446	0.435	0.378	0.394
			ETTh1	0.373	0.391	0.427	0.421	0.466	0.441	0.477	0.468	0.436	0.430
			ECL	0.137	0.232	0.153	0.247	0.168	0.266	0.199	0.294	0.165	0.260
			Weather	0.156	0.201	0.205	0.245	0.261	0.286	0.343	0.339	0.241	0.267
KMB-DF†	✓	✗	ETTm1	0.318	0.356	0.355	0.381	0.383	0.400	0.442	0.433	0.374	0.392
			ETTh1	0.372	0.391	0.423	0.421	0.462	0.443	0.462	0.469	0.430	0.431
			ECL	0.137	0.232	0.153	0.247	0.168	0.265	0.201	0.294	0.165	0.259
			Weather	0.154	0.199	0.204	0.242	0.261	0.286	0.342	0.341	0.240	0.267
KMB-DF‡	✗	✓	ETTm1	0.318	0.356	0.356	0.380	0.384	0.401	0.442	0.433	0.375	0.393
			ETTh1	0.372	**0.389**	0.425	**0.420**	**0.458**	0.439	0.455	0.457	0.427	0.426
			ECL	0.137	0.232	0.153	0.247	0.168	0.266	0.200	0.294	0.165	0.260
			Weather	0.153	0.198	0.205	0.244	0.261	0.285	0.341	0.338	0.240	0.266
KMB-DF	✓	✓	ETTm1	**0.313**	**0.354**	**0.351**	**0.379**	**0.381**	**0.400**	**0.442**	**0.433**	**0.372**	**0.391**
			ETTh1	**0.367**	0.390	**0.422**	0.420	0.460	**0.438**	**0.447**	**0.455**	**0.424**	**0.426**
			ECL	**0.136**	**0.231**	**0.152**	**0.245**	**0.168**	**0.265**	**0.197**	**0.291**	**0.163**	**0.258**
			Weather	**0.152**	**0.196**	**0.204**	**0.242**	**0.260**	**0.284**	**0.339**	**0.337**	**0.239**	**0.265**

Note: **Bold** and underlined denote best and second-best results, respectively.

4.4 Kernel Function Selection Studies

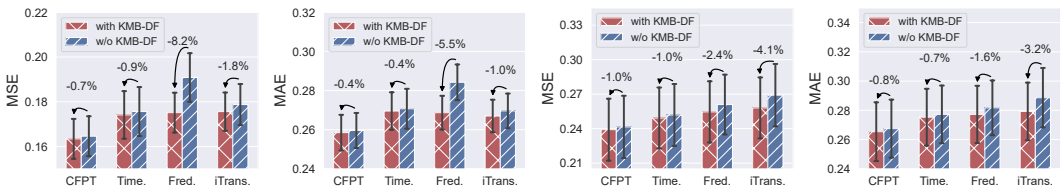
Table 4: Varying kernel function results.

	ETTh1				ETTm1			
Kernel	MSE	ΔMSE	MAE	ΔMAE	MSE	ΔMSE	MAE	ΔMAE
DF	0.436	-	0.430	-	0.378	-	0.394	-
Linear	0.435	0.3%↓	0.428	0.4%↓	0.375	0.7%↓	0.393	0.3%↓
Polynomial	0.434	0.4%↓	0.430	0.1%↓	0.375	0.7%↓	0.393	0.3%↓
Sigmoid	0.433	0.6%↓	0.430	0.0%↑	0.375	0.9%↓	0.392	0.5%↓
Gaussian	0.434	0.5%↓	0.430	0.1%↓	0.374	1.0%↓	0.392	0.4%↓
Exponential	0.424	2.7%↓	0.426	1.0%↓	0.372	1.6%↓	0.391	0.7%↓

In this section, we evaluate the impact of the kernel function choice $\mathcal{K}(\cdot,\cdot)$ in instantiating the balancing constraints. We compare the exponential kernel against Linear, Polynomial, Sigmoid, and Gaussian kernels, as detailed in Table 4. We make the following observations: **❶ All tested kernels reduce MSE relative to DF on both datasets.** For example, the linear kernel reduces the MSE from 0.378 to 0.375 on ETTm1. This result suggests that kernelized moment-balancing constraints complement the point-wise MSE objective by accounting for dependencies among future labels induced by label auto-correlation. **❷ The exponential kernel yields the lowest errors among the tested kernels.** It reduces the MSE by 2.7% on ETTh1 relative to DF, and also gives the lowest MAE on both datasets. These results demonstrate that the exponential kernel is an effective choice for adaptively balancing nonlinear distributional discrepancies between labels and forecasts.

4.5 Generalization Studies

In this section, we assess the generalizability of KMB-DF by integrating it into four representative state-of-the-art forecast models: CFPT (Kou et al., 2025), TimeBridge (Liu et al., 2025b), Fredformer (Piao et al., 2024), and iTransformer (Liu et al., 2024a). The results are presented in Figure 2 with the following observations. **❶ KMB-DF improves the performance of backbones developed in different periods.** As illustrated by the ECL results, KMB-DF improves both earlier backbones, including Fredformer and iTransformer, and more recent backbones, including TimeBridge and CFPT. **❷ KMB-DF improves the performance of backbones with different architectural complexity.** As illustrated by the Weather results, it improves both the linear CFPT model and the more expressive Transformer-based models. These results demonstrate that KMB-DF is a general backbone-agnostic learning objective rather than an enhancement specialized for a par-



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

Figure 2: Improvement of KMB-DF 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.

ticular backbone or dataset. By adaptively selecting kernelized balancing functions, it provides an objective-level complement to point-wise losses that overlook label autocorrelation.

4.6 HYPERPARAMETER SENSITIVITY

In this section, we examine the sensitivity of KMB-DF to the penalty strength α, the tolerated imbalance margin C, and the number of selected balancing functions K. The results are presented in Tables 5, 6, and 7, respectively, with the following observations.

Table 5: Varying α results.

	ETTh1				ETTm1			
α	MSE	ΔMSE	MAE	ΔMAE	MSE	ΔMSE	MAE	ΔMAE
DF	0.436	-	0.430	-	0.378	-	0.394	-
0.1	0.431	1.2%↓	0.429	0.4%↓	0.374	1.0%↓	0.392	0.4%↓
0.3	0.431	1.2%↓	0.429	0.4%↓	0.373	1.2%↓	0.393	0.2%↓
0.5	0.433	0.7%↓	0.430	0.1%↓	0.374	1.1%↓	0.392	0.5%↓
0.7	0.426	2.3%↓	0.426	0.9%↓	0.374	1.1%↓	0.392	0.5%↓
0.9	0.431	1.1%↓	0.429	0.3%↓	0.373	1.3%↓	0.392	0.6%↓

Penalty strength α. ❶ **All nonzero values of α improve performance over DF.** On both datasets, every tested value achieves lower MSE than DF. For example, $\alpha = 0.1$ reduces the MSE from 0.436 to 0.431 on ETTh1. These results demonstrate the benefit of the kernelized moment-balancing penalty across a broad range of weights. ❷ **Intermediate values of α achieve the lowest errors.** In particular, $\alpha = 0.7$ gives the lowest MSE and MAE on ETTh1, while $\alpha = 0.9$ attains best performance on ETTm1. This suggests that the balancing penalty provides complementary information to the point-wise MSE objective.

Table 6: Varying C results.

	ETTh1				ETTm1			
C	MSE	ΔMSE	MAE	ΔMAE	MSE	ΔMSE	MAE	ΔMAE
DF	0.436	-	0.430	-	0.378	-	0.394	-
0.0005	0.430	1.4%↓	0.429	0.3%↓	0.374	1.0%↓	0.392	0.4%↓
0.001	0.424	2.6%↓	0.426	0.9%↓	0.374	1.0%↓	0.392	0.4%↓
0.005	0.429	1.5%↓	0.429	0.3%↓	0.373	1.3%↓	0.392	0.6%↓
0.01	0.430	1.4%↓	0.428	0.5%↓	0.373	1.3%↓	0.392	0.6%↓
0.05	0.434	0.5%↓	0.430	0.1%↓	0.374	1.0%↓	0.392	0.4%↓

Tolerated imbalance margin C. ❶ **All tested values of C improve performance over DF.** Each margin reduces MSE on both datasets. For example, $C = 0.0005$ reduces the MSE from 0.436 to 0.430, demonstrating that soft-margin relaxation benefits forecasting performance across a broad range of margins. ❷ **Smaller values of C generally achieve better performance.** For example, $C = 0.001$ achieves the largest MSE reduction of 2.6%. In contrast, increasing C to 0.05 yields only a 0.5% reduction. Thus, relatively small tolerated mismatches provide effective default choices.

Table 7: Varying K results.

	ETTh1				ETTm1			
K	MSE	ΔMSE	MAE	ΔMAE	MSE	ΔMSE	MAE	ΔMAE
DF	0.436	-	0.430	-	0.378	-	0.394	-
1	0.432	0.8%↓	0.429	0.3%↓	0.374	1.1%↓	0.392	0.5%↓
2	0.431	1.1%↓	0.429	0.4%↓	0.374	1.1%↓	0.392	0.5%↓
3	0.427	2.1%↓	0.426	0.9%↓	0.374	1.1%↓	0.392	0.4%↓
4	0.432	0.9%↓	0.429	0.2%↓	0.374	0.9%↓	0.393	0.3%↓
5	0.426	2.4%↓	0.427	0.9%↓	0.374	1.1%↓	0.392	0.4%↓

Number of balancing functions K. ❶ **Using one adaptively selected balancing function improves performance over DF.** With $K = 1$, the MSE decreases from 0.436 to 0.432 on ETTh1 and from 0.378 to 0.374 on ETTm1. This demonstrates that even a single function selected by the proposed adaptive search provides a useful complement to the point-wise MSE objective. ❷ **A small number of adaptively selected functions achieves near-optimal performance.** On ETTh1, $K = 3$ achieves the lowest MAE, while $K = 5$ gives the lowest MSE; on ETTm1, all tested values yield similar errors. Thus, $K = 3$ is an effective default that maintains a small selection size without extensive tuning.

5 CONCLUSION

In this work, we identify a fundamental limitation in existing deep time-series forecasting objectives: they fail to satisfy Imbens' criterion for true distribution balance, as they enforce moment matching only on a limited set of predefined functions. To address this, we propose Direct Forecasting with Kernelized Moment Balancing (KMB-DF) to ensure sufficient distribution alignment. Instead of relying on finite statistics, KMB-DF leverages reproducing kernel Hilbert spaces (RKHS) to adaptively select infinite-dimensional balancing functions. We theoretically justify balancing joint distributions as a tractable proxy for aligning conditional distributions. Experiments demonstrate that KMB-DF consistently achieves state-of-the-art performance by effectively mitigating autocorrelation bias.

Limitations. While KMB-DF enforces global alignment through kernelized moment matching, it does not dictate sample-wise temporal correspondence. Consequently, KMB-DF serves as a regularizer rather than a standalone objective. It requires a point-wise loss (*e.g.*, MSE) to anchor element-wise alignment, correcting distributional discrepancies while maintaining temporal precision.

AI USE STATEMENT

We used generative AI tools for language polishing, formatting and coding assistance, as well as preliminary brainstorming and literature assistance. Generative AI tools were not used to formulate the core research method, conduct experiments, perform the empirical or theoretical analyses, or determine the scientific conclusions. All AI-assisted text, ideas, and code were critically reviewed and verified by the authors; the core methodology, experiments, analyses, and final conclusions were developed and validated by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Khaled Alkilane, Yihang He, and Der-Horng Lee. Mixmamba: Time series modeling with adaptive expertise. *Inf. Fusion*, 112:102589, 2024.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *Proc. Int. Conf. Learn. Represent.*, 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.*, pp. 4523–4529, 2023.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

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

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.

Christian Fong, Chad Hazlett, and Kosuke Imai. Covariate balancing propensity score for a continuous treatment: application to the efficacy of political advertisements. *Ann. Appl. Stat.*, 12(1): 156–177, 2018.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(1):723–773, 2012.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Proc. Conf. Lang. Model.*, 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

Kosuke Imai and Marc Ratkovic. Covariate balancing propensity score. *J. R. Stat. Soc. Series. B. Stat. Methodol.*, 76(1):243–263, 2014.

Guido W Imbens and Donald B Rubin. Causal inference in statistics, social, and biomedical sciences. *New York*, 517, 2015.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *Proc. Int. Conf. Learn. Represent.*, 2024.

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

Yaxuan Kong, Zepu Wang, Yuqi Nie, Tian Zhou, Stefan Zohren, Yuxuan Liang, Peng Sun, and Qingsong Wen. Unlocking the power of lstm for long term time series forecasting. In *Proc. AAAI Conf. Artif. Intell.*, volume 39, pp. 11968–11976, 2025.

Feifei Kou, Jiahao Wang, Lei Shi, Yuhan Yao, Yawen Li, Suguo Zhu, Zhongbao Zhang, and Junping Du. Cfpt: Empowering time series forecasting through cross-frequency interaction and periodic-aware timestamp modeling. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proc. Int. Conf. Mach. Learn.*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31634–31647. PMLR, 13–19 Jul 2025.

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.*, 2025.

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. *Proc. Adv. Neural Inf. Process. Syst.*, 32, 2019.

Vincent Le Guen and Nicolas Thome. Deep time series forecasting with shape and temporal criteria. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):342–355, 2023.

Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, Peng Wu, and Peng Cui. Propensity matters: measuring and enhancing balancing for recommendation. In *Proc. Int. Conf. Mach. Learn.*, volume 202, pp. 20182–20194, 2023.

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.*, volume 2025, pp. 39490–39524, 2025.

Zijun Liao, Li Zheng, Xinru Xie, Hairui Wu, and Fangjia Liu. Real-time bidirectional haptic data forecasting in teleoperated telemedicine systems. *IEEE Transactions on Medical Robotics and Bionics*, 2025.

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, pp. 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.

Bingyi Liu, Lingtian Yuan, Xun Shao, Enshu Wang, Zhenchang Xia, Weizhen Han, and Celimuge Wu. Transformer-based generative adversarial network for traffic forecasting. *IEEE Trans. Consum. Electron.*, 2025a.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proc. Int. Conf. Mach. Learn.*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840. PMLR, 13–19 Jul 2025b.

Xiang Liu, Naiqi Li, and Shu-Tao Xia. Gdtw: A novel differentiable dtw loss for time series tasks. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2860–2864. IEEE, 2021.

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.*, 2024a.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Proc. Adv. Neural Inf. Process. Syst.*, 37:122154–122184, 2024b.

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

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *Int. J. Forecast.*, 38(4):1346–1364, 2022.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

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.

Licheng Pan, Zhichao Chen, Haoxuan Li, Guangyi Liu, Zhijian Xu, Zhaoran Liu, Hao Wang, and Ying Wei. Mixture of low rank adaptation with partial parameter sharing for time series forecasting. *arXiv preprint arXiv:2505.17872*, 2025.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp. 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.*, pp. 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.*, 2025.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proc. Int. Conf. Mach. Learn.*, pp. 8857–8868. PMLR, 2021.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast*, 36(3):1181–1191, 2020.

Cao Song, Minghui Lu, Yuqing Wang, and Wenkai Lu. A dynamic time warping loss-based closed-loop cnn for seismic impedance inversion. *IEEE Trans. Geosci. Remote Sens.*, 60:1–13, 2022.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Proc. Adv. Neural Inf. Process. Syst.*, 34:24804–24816, 2021.

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.*, pp. 1–9, 2025a.

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.*, pp. 1–9, 2025b.

Hao Wang, Licheng Pan, Yuan Lu, Zhichao Chen, Tianqiao Liu, Shuting He, Zhixuan Chu, Qingsong Wen, Haoxuan Li, and Zhouchen Lin. Quadratic direct forecast for training multi-step time-series forecast models. In *Proc. Int. Conf. Learn. Represent.*, pp. 1–9, 2026a.

Hao Wang, Licheng Pan, Yuan Lu, Zhixuan Chu, Xiaoxi Li, Shuting He, Zhichao Chen, Haoxuan Li, Qingsong Wen, and Zhouchen Lin. Distdf: time-series forecasting needs joint-distribution wasserstein alignment. In *Proc. Int. Conf. Learn. Represent.*, pp. 1–9, 2026b.

Haoxin Wang, Bixiong Li, Songhai Fan, Yuankai Wu, and Xianggen Liu. Timesql: Improving multivariate time series forecasting with multi-scale patching and smooth quadratic loss. *Inf. Sci*, 671:120652, 2024a.

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.*, 2023.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024b.

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.*, 2023a.

Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nat. Mach. Intell.*, pp. 1–10, 2023b.

Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. *Proc. Adv. Neural Inf. Process. Syst.*, 33:17105–17115, 2020.

Qi Xiong, Kai Tang, Minbo Ma, Ji Zhang, Jie Xu, and Tianrui Li. Modeling temporal dependencies within the target for long-term time series forecasting. *IEEE Trans. Knowl. Data Eng.*, 2025.

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.*, 2023.

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.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. Int. Conf. Mach. Learn.*, 2022.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Proc. Adv. Neural Inf. Process. Syst.*, 36:43322–43355, 2023.

A Extended Related Work

This section provides an extended review of deep time-series forecasting literature through the lens of autocorrelation modeling. As delineated in the Introduction, existing research primarily addresses two fundamental challenges: (1) devising neural architectures to model autocorrelation in history sequences, and (2) designing learning objectives to accommodate autocorrelation in label sequences.

A.1 Neural Architectures for History Autocorrelation

Recent advancements in neural architectures focus on capturing temporal dependencies within the historical input to generate latent representations for forecasting. These approaches can be broadly categorized into non-Transformer and Transformer-based models.

Non-Transformer Models. This category includes Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Dense Neural Networks (DNNs). ❶ RNNs naturally model sequential autocorrelation via hidden states. While traditional RNNs (*e.g.*, LSTM (Hochreiter & Schmidhuber, 1997), GRU (Cho et al., 2014)) suffer from efficiency bottlenecks, recent State Space Models (SSMs) like Mamba (Gu & Dao, 2023) and its variants (*e.g.*, MixMamba (Alkilane et al., 2024)) have emerged to efficiently model long-range dependencies with linear complexity. ❷ CNNs capture local autocorrelation through convolutional kernels. To expand the receptive field for long-term forecasting, recent works employ large-kernel designs (*e.g.*, ModernTCN (Luo & Wang, 2024)) or operate in the frequency domain (*e.g.*, TimesNet (Wu et al., 2023a), MICN (Wang et al., 2023)). ❸ DNNs utilizing simple linear layers have shown competitive performance by treating history sequences as feature vectors. Representative models like DLinear (Zeng et al., 2023) and TiDE (Das et al., 2023) demonstrate that Multi-Layer Perceptrons (MLPs) can effectively model autocorrelation when combined with decomposition or channel-independence strategies.

Transformer-based Models. Transformers utilize self-attention mechanisms to model high-order autocorrelation by computing pairwise similarities across time steps. ❶ Standard Self-Attention Models often employ tokenization to incorporate local context. For instance, PatchTST (Nie et al., 2023) segments series into patches to retain semantic meaning. A growing trend involves adapting Large Language Models (LLMs) for time-series (*e.g.*, Time-LLM (Jin et al., 2024), GPT4TS (Zhou et al., 2023)) via reprogramming or fine-tuning to leverage their pre-trained reasoning capabilities. ❷ Modified Attention Models refine the attention mechanism to reduce complexity or enhance feature extraction. Notable approaches include frequency-domain attention (*e.g.*, FedFormer (Zhou et al., 2022), FreDF (Wang et al., 2025b)) and inverted attention mechanisms that model correlations across variates (*e.g.*, iTransformer (Liu et al., 2024a)).

A.2 Learning Objectives for Label Autocorrelation

While architectures focus on inputs, learning objectives aim to mitigate the bias arising from label autocorrelation, where future values are statistically dependent on their predecessors. Standard objectives like Mean Squared Error (MSE) assume conditional independence, leading to suboptimal training. Recent objectives mainly address this issue via likelihood estimation, shape alignment, conditional generation, and distribution balancing.

Likelihood Estimation. These methods aim to refine the negative log-likelihood (NLL) estimation. ❶ Label Transformation: To eliminate autocorrelation bias, methods like FreDF (Wang et al., 2025b) and Time-o1 (Wang et al., 2025a) transform labels into frequency or principal component domains, respectively, aiming to whiten the label distribution before applying point-wise losses. ❶ Covariance Modeling: Approaches such as QDF (Wang et al., 2026a) explicitly model the conditional covariance matrix to approximate the true NLL, often utilizing meta-learning or auxiliary networks.

Shape Alignment. Recognizing that autocorrelation manifests in morphological shapes, these objectives measure geometric dissimilarity. SoftDTW (Cuturi & Blondel, 2017) provides a differentiable relaxation of Dynamic Time Warping. Variants like Dilate (Le Guen & Thome, 2019) and

STRIPE (Le Guen & Thome, 2023) further incorporate temporal distortion penalties to better align forecast shapes with ground truths.

Conditional Generation. This paradigm reframes forecasting as a generative task to implicitly capture label autocorrelation. Diffusion models (*e.g.*, TimeGrad (Rasul et al., 2021), CSDI (Tashiro et al., 2021)) generate forecasts by learning the conditional distribution $\mathbb{P}(\mathbf{y}|\mathbf{x})$ via iterative denoising. Autoregressive models (*e.g.*, DeepAR (Salinas et al., 2020), AutoTimes (Liu et al., 2024b)) generate predictions step-by-step to explicitly condition on previous outputs, though often suffering from error accumulation.

Distribution Balancing. Most relevant to our work, these methods view forecasting as aligning the distribution of forecasts with ground truths. ❶ Adversarial Training: Methods like AST (Wu et al., 2020) and variants (Liu et al., 2025a) employ GAN-based discriminators to distinguish between predicted and real sequences, enforcing distributional alignment implicitly. ❷ Discrepancy Minimization: Recent works like DistDF (Wang et al., 2026b) explicitly minimize distributional discrepancies (*e.g.*, joint-distribution Wasserstein distance).

While distribution balancing methods are theoretically appealing, existing approaches typically enforce moment matching only for limited or predefined functions (*e.g.*, first/second moments in DistDF or specific discriminator architectures in AST). As discussed in Section 3.1, they fail to satisfy Imbens' criterion, which requires equivalence across **any** balancing function. Our proposed KMB-DF addresses this gap by leveraging the Reproducing Kernel Hilbert Space (RKHS) to adaptively select infinite dimensional balancing functions, ensuring sufficient distribution alignment.

B Theoretical Justification

In this section, we delve into the theoretical underpinnings of the framework. By exploiting the universality, injectivity, and reproducing properties of kernel functions, we derive formal guarantees for the proposed method and demonstrate how these properties translate into effective balancing.

Lemma B.1 (Autocorrelation bias, Lemma 2.1 in the main text)**.** *Let $\mathbf{y} \in \mathbb{R}^{\mathrm{T}}$ be a univariate label sequence with conditional covariance $\mathbf{\Sigma} \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$. $\mathcal{E}_{\mathrm{MSE}}$ in Eq. (1) is biased against the likelihood of $\mathbf{y}$ unless $\mathbf{\Sigma}$ is diagonal, i.e., different steps in $\mathbf{y}$ are conditionally decorrelated given $\mathbf{x}$.*

Proof. The proof can be found in Theorem 3.1 of Wang et al. (2025a). □

Lemma B.2 (Representer theorem)**.** *Suppose $h(\|f\|) : \mathbb{R}_+ \to \mathbb{R}$ is a non-decreasing function. The minimizer of an empirical risk functional regularized by $h(\|f\|)$ admits the form: $f^*(\cdot) = \sum_{n=1}^{\mathrm{N}} \alpha_n \mathcal{K}(\cdot, \mathbf{z}_n)$, where $\mathcal{K}$ is the kernel function associated with the RKHS $\mathcal{H}$.*

Proof. The proof can be found in Theorem 6.11 of Mohri et al. (2018). □

Theorem B.3. *Suppose $\mathcal{H}$ is the RKHS of a exponential kernel $\mathcal{K}$, $f^* \in \mathcal{H}$ is the discrimination function that maximizes the discrepancy between two distributions $\mathbb{P}$ and $\mathbb{Q}$. If f^* lies within the linear span of a finite kernel function set: $f^* \in \{\mathcal{K}(\cdot, \mathbf{z}_k)\}_{k=1}^{\mathrm{K}}$; then $\mathbb{P}$ and $\mathbb{Q}$ are balanced if $\forall k \in [\mathrm{K}] : \mathbb{E}_{\mathbf{z}\in\mathbb{P}}[\mathcal{K}(\mathbf{z}, \mathbf{z}_k)] = \mathbb{E}_{\hat{\mathbf{z}}\in\mathbb{Q}}[\mathcal{K}(\hat{\mathbf{z}}, \mathbf{z}_k)]$.*

Proof. We start by clarifying the maximum mean discrepancy (MMD) as our discrepancy measure between two distributions. Specifically, given a universal kernel $\mathcal{K}$ (*e.g.*, exponential kernel), the MMD is defiend as:

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}) = \sup_{f\in\mathcal{H}, \|f\|_{\mathcal{H}}\leq 1} \left|\mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[f(\mathbf{z})] - \mathbb{E}_{\hat{\mathbf{z}}\sim\mathbb{Q}}[f(\hat{\mathbf{z}})]\right|^2. \tag{9}$$

Let f^* be the discrimination function that maximizes the mean discrepancy between the two distributions $\mathbb{P}$ and $\mathbb{Q}$, i.e., the function that achieves the supremum in the MMD^2 definition. By Lemma B.2, functions in $\mathcal{H}$ can be represented as linear combinations of kernel functions. On the

Table 8: Dataset description.

Dataset	D	Forecast length	Train / validation / test	Frequency	Domain
ETTh1	7	96, 192, 336, 720	8545/2881/2881	Hourly	Electricity
ETTh2	7	96, 192, 336, 720	8545/2881/2881	Hourly	Electricity
ETTm1	7	96, 192, 336, 720	34465/11521/11521	15min	Electricity
ETTm2	7	96, 192, 336, 720	34465/11521/11521	15min	Electricity
ECL	321	96, 192, 336, 720	18317/2633/5261	Hourly	Electricity
Weather	21	96, 192, 336, 720	36792/5271/10540	10min	Weather
M5	10	8, 12, 20, 28	1782/21/21	Daily	Sale

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., 2023a).

basis of this, by the assumption that f^* lies within the linear span of a finite kernel function set, there exist $\alpha_1, \ldots, \alpha_{\mathrm{K}}$ such that:

$$f^*(\cdot) = \sum_{k=1}^{\mathrm{K}} \alpha_k \mathcal{K}(\cdot, \mathbf{z}_k). \tag{10}$$

On this basis, we calculate the informativeness score where f^* serves as the balancing function:

$$\begin{aligned}
\delta(f^*) &= \mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[f^*(\mathbf{z})] - \mathbb{E}_{\hat{\mathbf{z}}\sim\mathbb{Q}}[f^*(\hat{\mathbf{z}})] \\
&= \mathbb{E}_{\mathbf{z}\sim\mathbb{P}}\left[\sum_{k=1}^{\mathrm{K}} \alpha_k \mathcal{K}(\mathbf{z}, \mathbf{z}_k)\right] - \mathbb{E}_{\hat{\mathbf{z}}\sim\mathbb{Q}}\left[\sum_{k=1}^{\mathrm{K}} \alpha_k \mathcal{K}(\hat{\mathbf{z}}, \mathbf{z}_k)\right] \\
&\overset{(a)}{=} \sum_{k=1}^{\mathrm{K}} \alpha_k \left(\mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[\mathcal{K}(\mathbf{z}, \mathbf{z}_k)] - \mathbb{E}_{\hat{\mathbf{z}}\sim\mathbb{Q}}[\mathcal{K}(\hat{\mathbf{z}}, \mathbf{z}_k)]\right) \\
&\overset{(b)}{=} 0,
\end{aligned} \tag{11}$$

where (a) exploits the linearity of the expectation operator to swap the summation and expectation; (b) applies the assumption that $\mathbb{E}_{\mathbf{z}\in\mathbb{P}}[\mathcal{K}(\mathbf{z}, \mathbf{z}_k)] = \mathbb{E}_{\hat{\mathbf{z}}\in\mathbb{Q}}[\mathcal{K}(\hat{\mathbf{z}}, \mathbf{z}_k)]$ for all $k = 1, \ldots, \mathrm{K}$.

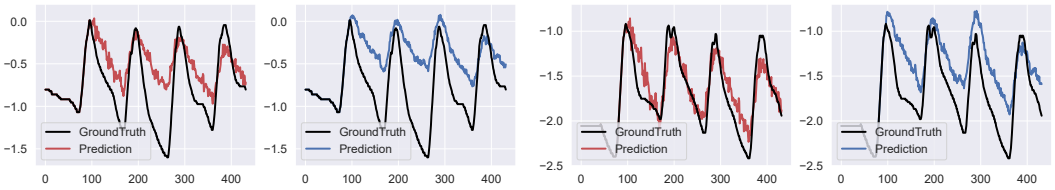
Since f^* is the function that maximizes the mean discrepancy between $\mathbb{P}$ and $\mathbb{Q}$, the maximum mean discrepancy is zero, *i.e.*, $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = 0$. Moreover, since $\mathcal{K}$ is a universal kernel, $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}) = 0$ implies $\mathbb{P} = \mathbb{Q}$ (Gretton et al., 2012), which completes the proof. □

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 8.

- **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).
- **ECL** (Wu et al., 2021): Features the hourly electricity consumption of 321 clients.
- **Weather** (Wu et al., 2021): Comprises 21 meteorological variables from the Max Planck Biogeochemistry Institute's weather station, captured every 10 minutes throughout 2020.
- **M5** (Makridakis et al., 2022): Comprises 3049 individual products from 3 categories and 7 departments, sold in 10 stores in 3 states.



(a) ETTm2 snapshot 1. (b) ETTm2 snapshot 2.

Figure 3: The forecast sequences generated with DF and KMB-DF. The forecast length is set to 336 and the experiment is conducted on ETTm2.

Following established protocols (Qiu et al., 2024; Liu et al., 2024a), all datasets are chronologically partitioned into training, validation, and test sets. For the ETT, Weather, and ECL, and datasets, we use a fixed history sequence length of 96 and evaluate performance across four prediction horizons with lengths of 96, 192, 336, and 720. For the M5 datasets, we also use an historical length of 96 but evaluate on shorter prediction horizons of 8, 12, 20, and 28 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 ensure a rigorous and fair comparison, we strictly adhere to standard evaluation protocols established in recent literature (Wu et al., 2023a; Liu et al., 2024a). We reproduce all baseline methods using their official implementations, primarily integrating codes from the DistDF (Wang et al., 2026b) and CFPT (Kou et al., 2025) repositories. All models are optimized using the Adam optimizer (Kingma & Ba, 2015) with a fixed batch size of 32. Following Qiu et al. (2024), we disable the "drop-last" operation in data loaders during the testing phase to prevent data leakage and ensure metrics are calculated over the complete test set. To prevent overfitting, we employ an early stopping mechanism that terminates training if the validation loss fails to improve for 15 consecutive epochs. The initial learning rate is tuned for each dataset-model combination via a grid search over $\{5 \times 10^{-3}, 2 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 2 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}\}$, selecting the value that yields the minimal MSE on the validation set. All experiments are conducted on a computational cluster equipped with Intel(R) Xeon(R) Platinum 8463B CPUs and NVIDIA RTX H800 GPUs.

When integrating KMB-DF with forecast backbones (*e.g.*, CFPT, TimeBridge), we explicitly retain the original models' architectural hyperparameters as reported in their respective benchmarks to ensure that performance gains are attributed solely to the proposed objective. Consequently, our hyperparameter tuning is exclusively focused on the components of KMB-DF: the penalty strength α, the tolerated imbalance margin C, and the number of balancing functions K. Specifically, α is tuned within the interval $(0, 1]$ to balance the trade-off between the standard forecasting loss and the moment matching penalty. The margin C is selected from the set $\{0.0005, 0.001, 0.005, 0.01, 0.05\}$ to control the strictness of distribution alignment, while the number of balancing functions K is searched within the integer range $[1, 6]$. We utilize the exponential kernel as the default instantiation for the balancing functions.

D More Experimental Results

D.1 Overall performance

Additional experimental results of overall performance are available in Table 9, where the performance given different T is reported.

D.2 Showcase

Additional experimental results of showcases on ETTm2 dataset are available in Figure 3.

Table 9: 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 for ETT, ECL, and Weather; T=8, 12, 20 and 28 for M5.

Loss		**KMB-DF (Ours)**		QDF (2025)		DistDF (2025)		Time-o1 (2025)		FreDF (2025)		DBLoss (2025)		Koopman (2021)		GDTW (2021)		Dilate (2019)		Soft-DTW (2017)		MSE (2002)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.315**	**0.352**	0.318	0.359	0.318	0.355	0.321	0.353	0.320	0.354	0.318	0.354	0.326	0.363	0.324	0.362	0.323	0.361	0.322	0.359	0.321	0.358
	192	**0.351**	**0.379**	0.355	0.379	0.356	0.380	0.358	0.380	0.356	0.379	0.359	0.379	0.359	0.383	0.368	0.392	0.363	0.385	0.366	0.385	0.357	0.382
	336	**0.381**	**0.400**	0.384	0.401	0.383	0.401	0.386	0.403	0.386	0.401	0.389	0.401	0.383	0.401	0.405	0.421	0.384	0.401	0.399	0.409	0.387	0.401
	720	**0.442**	**0.433**	0.443	0.435	0.444	0.434	0.447	0.436	0.444	0.434	0.450	0.434	0.446	0.436	0.496	0.473	0.446	0.437	0.490	0.461	0.446	0.435
	Avg	**0.372**	**0.391**	0.375	0.393	0.375	0.393	0.378	0.393	0.376	0.392	0.379	0.392	0.378	0.395	0.398	0.412	0.379	0.396	0.394	0.403	0.378	0.394
ETTm2	96	**0.166**	**0.249**	0.168	0.250	0.168	0.251	0.173	0.251	0.174	0.251	0.175	0.250	0.182	0.262	0.176	0.257	0.172	0.253	0.176	0.256	0.175	0.256
	192	**0.227**	**0.288**	0.232	0.292	0.232	0.293	0.233	0.291	0.233	0.290	0.233	0.289	0.235	0.294	0.243	0.305	0.236	0.296	0.243	0.299	0.234	0.295
	336	**0.284**	0.327	0.288	0.329	0.289	0.330	0.290	0.329	0.290	**0.327**	0.294	0.329	0.291	0.331	0.303	0.344	0.290	0.331	0.306	0.342	0.289	0.330
	720	**0.382**	0.385	0.386	0.387	0.385	0.387	0.386	**0.385**	0.386	0.386	0.391	0.387	0.388	0.390	0.434	0.419	0.393	0.390	0.438	0.422	0.385	0.389
	Avg	**0.265**	**0.312**	0.268	0.315	0.269	0.315	0.271	0.314	0.271	0.313	0.273	0.314	0.274	0.319	0.289	0.332	0.273	0.317	0.291	0.330	0.271	0.317
ETTh1	96	**0.367**	0.390	0.373	0.392	0.373	0.391	0.369	0.392	0.373	**0.389**	0.375	0.391	0.374	0.393	0.377	0.395	0.376	0.394	0.377	0.394	0.373	0.391
	192	**0.422**	0.420	0.426	0.421	0.426	0.420	0.424	0.424	0.424	**0.418**	0.428	0.420	0.426	0.424	0.434	0.427	0.430	0.423	0.436	0.427	0.427	0.421
	336	**0.460**	**0.438**	0.464	0.441	0.468	0.441	0.464	0.439	0.465	0.439	0.466	0.441	0.466	0.440	0.486	0.457	0.470	0.442	0.479	0.451	0.466	0.441
	720	**0.447**	**0.455**	0.473	0.464	0.467	0.462	0.462	0.460	0.472	0.468	0.474	0.462	0.483	0.469	0.512	0.497	0.489	0.478	0.551	0.510	0.477	0.468
	Avg	**0.424**	**0.426**	0.434	0.429	0.434	0.428	0.430	0.429	0.434	0.428	0.436	0.429	0.437	0.431	0.452	0.444	0.441	0.434	0.461	0.445	0.436	0.430
ETTh2	96	**0.280**	**0.331**	0.288	0.337	0.287	0.337	0.286	0.336	0.289	0.337	0.288	0.335	0.287	0.338	0.289	0.340	0.287	0.338	0.292	0.342	0.287	0.337
	192	**0.358**	**0.381**	0.367	0.390	0.364	0.391	0.367	0.384	0.365	0.382	0.367	0.385	0.365	0.390	0.377	0.399	0.368	0.389	0.380	0.398	0.368	0.390
	336	**0.411**	**0.425**	0.412	0.427	0.412	0.426	0.414	0.427	0.414	0.427	0.420	0.427	0.413	0.429	0.422	0.432	0.413	0.427	0.455	0.443	0.416	0.428
	720	**0.396**	**0.426**	0.401	0.428	0.398	0.427	0.402	0.428	0.406	0.430	0.411	0.430	0.405	0.430	0.457	0.463	0.415	0.437	0.445	0.453	0.415	0.436
	Avg	**0.361**	**0.391**	0.367	0.396	0.365	0.395	0.367	0.394	0.368	0.394	0.371	0.394	0.368	0.397	0.386	0.409	0.371	0.398	0.393	0.409	0.372	0.398
ECL	96	**0.136**	**0.231**	0.137	0.232	0.137	0.232	0.141	0.235	0.139	0.232	0.141	0.233	0.138	0.232	0.882	0.777	0.137	0.232	0.136	0.232	0.137	0.232
	192	**0.152**	**0.245**	0.153	0.246	0.152	0.246	0.154	0.246	0.156	0.246	0.155	0.246	0.153	0.247	0.853	0.762	0.154	0.247	0.154	0.247	0.153	0.247
	336	**0.168**	0.265	0.168	0.264	0.168	0.265	0.168	0.263	0.169	**0.263**	0.170	0.265	0.169	0.267	0.809	0.711	0.169	0.267	0.169	0.267	0.168	0.266
	720	**0.197**	0.291	0.199	0.293	0.201	0.294	0.198	0.291	0.199	**0.290**	0.203	0.293	0.201	0.296	0.900	0.769	0.202	0.297	0.203	0.298	0.199	0.294
	Avg	**0.163**	0.258	0.164	0.259	0.164	0.259	0.165	0.259	0.166	**0.258**	0.167	0.259	0.165	0.260	0.861	0.755	0.165	0.260	0.165	0.261	0.165	0.260
Weather	96	**0.152**	**0.196**	0.155	0.200	0.155	0.200	0.153	0.197	0.154	0.197	0.160	0.198	0.173	0.215	0.158	0.206	0.173	0.215	0.177	0.220	0.156	0.201
	192	**0.204**	**0.242**	0.205	0.244	0.205	0.245	0.205	0.243	0.204	0.242	0.208	0.243	0.205	0.245	0.209	0.250	0.205	0.246	0.209	0.245	0.205	0.245
	336	**0.260**	**0.284**	0.261	0.285	0.261	0.286	0.261	0.284	0.261	0.284	0.267	0.286	0.262	0.286	0.270	0.296	0.262	0.287	0.278	0.296	0.261	0.286
	720	**0.339**	**0.337**	0.343	0.338	0.343	0.341	0.343	0.340	0.341	0.338	0.346	0.338	0.345	0.342	0.354	0.352	0.345	0.342	0.385	0.363	0.343	0.339
	Avg	**0.239**	**0.265**	0.241	0.267	0.241	0.268	0.240	0.266	0.240	0.265	0.245	0.266	0.246	0.272	0.248	0.276	0.246	0.273	0.262	0.281	0.241	0.267
M5	8	**0.150**	**0.307**	0.152	0.309	0.152	0.308	0.159	0.318	0.151	0.308	0.152	0.308	0.154	0.311	0.153	0.312	0.154	0.312	0.154	0.312	0.151	0.309
	12	**0.151**	**0.311**	0.152	0.311	0.154	0.311	0.164	0.325	0.155	0.313	0.155	0.312	0.156	0.314	0.160	0.319	0.161	0.319	0.160	0.319	0.157	0.315
	20	**0.153**	**0.311**	0.155	0.313	0.154	0.313	0.159	0.317	0.166	0.328	0.162	0.321	0.158	0.318	0.163	0.321	0.165	0.326	0.164	0.326	0.165	0.324
	28	**0.154**	**0.309**	0.165	0.320	0.162	0.322	0.162	0.316	0.199	0.361	0.181	0.343	0.164	0.323	0.164	0.319	0.156	0.311	0.169	0.332	0.199	0.361
	Avg	**0.152**	**0.309**	0.156	0.313	0.156	0.313	0.161	0.319	0.168	0.327	0.162	0.321	0.158	0.317	0.160	0.318	0.159	0.317	0.162	0.322	0.168	0.327
1st Count		**35**	**28**	0	0	0	0	0	**1**	0	**6**	0	0	0	0	0	0	0	0	0	0	0	0

D.3 Generalization Studies

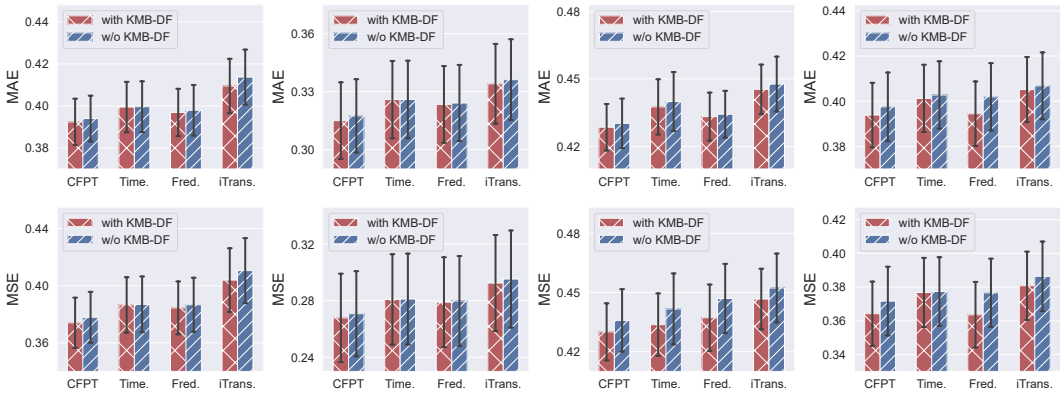
Additional experimental results of varying forecast models are available in Figure 4, where four forecast models are involved on four datasets.

D.4 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 CFPT (Kou 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 KMB-DF consistently improves both forecast models across different history sequence lengths.

D.5 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 (2022, 2023, 2024, 2025, and 2026). The results indicate minimal sensitivity of the proposed method to random initialization, as most averaged standard deviations remain below 0.005.



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

Figure 4: Performance of different forecast models with and without KMB-DF. 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		**KMB-DF**		CFPT		**KMB-DF**		PatchTST	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	96	**0.152**	**0.196**	0.156	0.201	0.178	0.219	0.189	0.230
	192	**0.204**	**0.242**	0.205	0.245	0.224	0.259	0.228	0.262
	336	**0.260**	**0.284**	0.261	0.286	0.280	0.299	0.288	0.305
	720	**0.339**	**0.337**	0.343	0.339	0.356	0.348	0.362	0.354
	Avg	**0.239**	**0.265**	0.241	0.267	0.260	0.281	0.267	0.288
192	96	**0.148**	**0.196**	0.150	0.198	0.159	0.204	0.163	0.209
	192	**0.193**	**0.238**	0.195	0.239	0.206	0.247	0.207	0.249
	336	**0.247**	**0.278**	0.250	0.282	0.260	0.289	0.268	0.293
	720	**0.325**	**0.331**	0.330	0.333	0.335	0.339	0.338	0.339
	Avg	**0.228**	**0.261**	0.231	0.263	0.240	0.270	0.244	0.273
336	96	**0.143**	**0.192**	0.145	0.195	0.153	0.202	0.158	0.208
	192	**0.188**	0.235	0.188	**0.235**	0.195	0.242	0.235	0.291
	336	**0.240**	**0.275**	0.240	0.277	0.249	0.283	0.252	0.287
	720	**0.317**	**0.328**	0.321	0.332	0.323	0.337	0.326	0.336
	Avg	**0.222**	**0.258**	0.224	0.260	0.230	0.266	0.243	0.280
720	96	**0.144**	**0.196**	0.148	0.200	0.150	0.201	0.153	0.205
	192	**0.189**	**0.238**	0.189	0.240	0.196	0.245	0.205	0.254
	336	0.239	**0.279**	**0.238**	0.279	0.244	0.285	0.248	0.288
	720	**0.307**	**0.327**	0.310	0.329	0.313	0.335	0.317	0.339
	Avg	**0.220**	**0.260**	0.221	0.262	0.226	0.266	0.231	0.272

D.6 Complexity Analysis

We empirically evaluate the computational overhead of KMB-DF, as illustrated in Table 12. The experiments are conducted with a batch size of 128, variable dimension $D = 21$, and the number of balancing functions $K = 3$. As the forecast horizon T extends from 96 to 720, the running time for both forward and backward passes exhibits a marginal upward trend. This is expected, as a larger T increases the length of the joint sequences $\mathbf{z}$, thereby scaling the kernel matrix calculations required for moment balancing. Accordingly, forward FLOPs increase from 19.68M to 83.57M, backward FLOPs from 1.29M to 5.48M, and GPU memory consumption from 103.36 to 391.13 MiB. Nevertheless, the running time remains low even at $T = 720$, with the forward and backward passes taking 0.721 ms and 0.308 ms, respectively. During inference, the KMB-DF objective is removed, ensuring that the model's inference speed is unaffected. Consequently, KMB-DF enhances forecasting performance with minimal training cost and *zero* additional inference latency.

Table 11: Experimental results ($\mathrm{mean}_{\pm\mathrm{std}}$) with varying seeds (2022-2026).

Dataset	ETTh1				ETTm1			
Models	**KMB-DF**		DF		**KMB-DF**		DF	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	$0.372_{\pm 0.002}$	$0.390_{\pm 0.001}$	$0.373_{\pm 0.000}$	$0.391_{\pm 0.000}$	$0.320_{\pm 0.004}$	$0.356_{\pm 0.003}$	$0.319_{\pm 0.004}$	$0.357_{\pm 0.002}$
192	$0.425_{\pm 0.001}$	$0.421_{\pm 0.001}$	$0.427_{\pm 0.000}$	$0.421_{\pm 0.000}$	$0.354_{\pm 0.002}$	$0.380_{\pm 0.001}$	$0.359_{\pm 0.005}$	$0.382_{\pm 0.001}$
336	$0.459_{\pm 0.001}$	$0.437_{\pm 0.001}$	$0.468_{\pm 0.002}$	$0.442_{\pm 0.001}$	$0.385_{\pm 0.002}$	$0.402_{\pm 0.001}$	$0.390_{\pm 0.002}$	$0.402_{\pm 0.001}$
720	$0.462_{\pm 0.013}$	$0.463_{\pm 0.006}$	$0.475_{\pm 0.003}$	$0.467_{\pm 0.002}$	$0.443_{\pm 0.001}$	$0.434_{\pm 0.000}$	$0.446_{\pm 0.000}$	$0.435_{\pm 0.000}$
Avg	$0.429_{\pm 0.004}$	$0.428_{\pm 0.002}$	$0.436_{\pm 0.001}$	$0.430_{\pm 0.001}$	$0.375_{\pm 0.002}$	$0.393_{\pm 0.001}$	$0.379_{\pm 0.002}$	$0.394_{\pm 0.001}$

Table 12: Computational complexity of KMB-DF with varying forecast horizons.

Forecast Length	96	128	192	256	336	512	720
Forward time (ms)	$0.670_{\pm 0.002}$	$0.672_{\pm 0.003}$	$0.679_{\pm 0.002}$	$0.686_{\pm 0.003}$	$0.689_{\pm 0.002}$	$0.686_{\pm 0.004}$	$0.721_{\pm 0.004}$
Backward time (ms)	$0.303_{\pm 0.000}$	$0.302_{\pm 0.000}$	$0.303_{\pm 0.000}$	$0.303_{\pm 0.000}$	$0.304_{\pm 0.000}$	$0.305_{\pm 0.000}$	$0.308_{\pm 0.000}$
Forward FLOPs	19677832	22954504	29507848	36061192	44252872	62274568	83572936
Backward FLOPs	1291078	1506118	1936198	2366278	2903878	4086598	5484358
GPU memory (MiB)	103.358	120.338	146.096	180.057	224.559	294.280	391.127