PROJECTION-BASED ALIGNMENT WITH A FRACTIONAL TRAJECTORY METRIC FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard MSE-based training inherently oversmooths forecasts by predicting conditional mean trajectories, failing to capture localized, high-frequency mean shifts. We address this limitation with PRODA, a model-agnostic objective that prevents oversmoothing by aligning the joint history–future and history–forecast distributions through the projection-based Cramér–Wold discrepancy. PRODA incorporates fractional differencing directly into the alignment objective, inducing a structured trajectory metric that attenuates low-frequency discrepancies and relatively emphasizes localized temporal variation. Theoretically, we show that the induced Cramér–Wold kernel remains characteristic under the fractional transformation and derive its spectral reweighting. This establishes that the geometry of distributional alignment can be reshaped without sacrificing distributional identifiability. Extensive experiments across multiple forecasting backbones, benchmark datasets, and zero-shot transfer settings demonstrate consistent improvements over standard training objectives and recent objective-centric baselines.

1 INTRODUCTION

In direct multi-step time-series forecasting, the prediction target is a future trajectory of which coordinates are generally dependent across forecasting horizons (Rangapuram et al., 2018; Lim et al., 2021). Most multi-step time-series forecasting models rely on MSE-based estimation to predict the conditional mean of the series pointwise (Wu et al., 2021; Nie et al., 2023; Zhang & Yan, 2023; Zeng et al., 2023). Consequently, models often fail to capture the underlying cross-horizon dependencies and sharp transitions inherent in the data, as MSE-trained models tend to suppress variations to avoid large pointwise errors induced by phase shifts (Le Guen & Thome, 2019; Wang et al., 2026a).

The oversmoothing issue in MSE-based time-series forecasting models stems from the optimization instability inherent in pointwise conditional mean estimation. Le Guen & Thome (2019) and Rahaman et al. (2019) have reported that even for expressive models capable of capturing high-frequency dynamics and sharp transitions, MSE-driven optimization often steers local minima toward overly smoothed mean trajectories.

Another limitation of conditional mean forecasting is poor extrapolation capability, especially when the future trajectory falls outside the input-output support of the training data. Several prior studies (Kim et al., 2021; Zeng et al., 2023; Kang et al., 2024) have highlighted the extrapolation limitations of neural network models, and these issues become particularly pronounced when estimating the conditional mean during abrupt trend shifts. Since conventional models merely track the pointwise conditional mean using raw observations as targets, avoiding extrapolation bias in out-of-distribution regimes remains fundamentally difficult (Fan et al., 2023; Liu et al., 2024a). These two issues motivate objectives that treat the predicted sequence as a unified trajectory rather than only through coordinate-wise deviations.

To faithfully capture the dynamic variability of time series, we consider a two-fold regularization framework. First, fractional differenced time series are introduced to capture not only the pointwise mean of predicted trajectories, but also trajectory-level variability. In particular, dynamically differenced sequences accentuate high-frequency fluctuations that are otherwise obscured by slowly varying low-frequency trends (De Prado, 2018; Le Guen & Thome, 2019). Fractional differencing is a canonical technique for handling non-stationary series that exhibit long-range dependence and sharp transitions (Granger & Joyeux, 1980).

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.

Second, rather than enforcing step-by-step pointwise matching, the temporal characteristics of these differenced series are preserved and aligned through their trajectory-level distributional properties. Specifically, we formulate a trajectory-level objective that quantifies the discrepancy between the joint distributions of history–future and history–forecast trajectories using the Cramér–Wold distance (Cramér & Wold, 1936), seamlessly integrating it as a regularization loss into the overall training framework. The resulting Cramér–Wold discrepancy enables nonparametric distribution comparison and admits a closed-form kernel representation suitable for efficient optimization (Knop et al., 2020).

Building on these two pillars, we propose **PRODA** (Projection-based Distribution Alignment), a model-agnostic objective that combines Cramér–Wold distribution alignment with a fractional trajectory metric. PRODA applies fractional differencing to history–future and history–forecast trajectories and aligns their transformed distributions. We show that the induced Cramér–Wold kernel remains characteristic and that the resulting metric admits a frequency-selective interpretation, attenuating slowly varying discrepancies while relatively emphasizing higher-frequency variation. Across multiple forecasting backbones and benchmark datasets, PRODA consistently improves forecasting accuracy and also exhibits strong zero-shot transfer performance.

Our contributions are summarized as follows:

- We propose PRODA, a model-agnostic forecasting objective that aligns history–future and history–forecast trajectory distributions through the Cramér–Wold discrepancy, with fractional differencing incorporated as a structured trajectory metric.
- We theoretically characterize the resulting alignment geometry, showing that the induced Cramér–Wold kernel remains characteristic and that the fractional trajectory metric admits an explicit frequency-domain interpretation.
- We conduct extensive experiments across multiple forecasting backbones, benchmark datasets, objective-centric baselines, and zero-shot settings, demonstrating consistent improvements in forecasting performance.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents PRODA, its fractional trajectory metric, and theoretical properties. Section 4 reports experimental results across multiple forecasting settings. Section 5 concludes the paper.

2 RELATED WORK

Objective design for time-series forecasting. Most time-series forecasting models are trained with pointwise regression losses, particularly MSE (Wu et al., 2021; Nie et al., 2023; Liu et al., 2024b; Zeng et al., 2023). While statistically natural for estimating conditional means, such objectives do not explicitly account for the joint structure of multi-step future trajectories. This has motivated structure-aware objectives that incorporate temporal alignment, shape, frequency-domain representations, or decomposed components (Cuturi & Blondel, 2017; Le Guen & Thome, 2019; Park et al., 2025; Qiu et al., 2026; Kudrat et al., 2025). More recent work considers distributional agreement between observed and forecast trajectories. DistDF (Wang et al., 2026a) aligns selected distributional statistics, whereas KMB-DF (Pan et al., 2026) formulates forecasting as joint-distribution balancing using kernelized moment conditions. These methods consider trajectory-level distribution comparison, with different choices of distributional discrepancy and trajectory geometry.

Differencing for time-series modeling. Differencing is a classical operation in time-series modeling for removing persistent level and trend components and obtaining a more stationary representation of the series. Integer differencing takes discrete orders, whereas fractional differencing extends the differencing order to real values, allowing it to vary continuously (Granger & Joyeux, 1980; Hosking, 1981). This construction forms the basis of fractionally integrated and autoregressive fractionally integrated moving average (ARFIMA) models, which accommodate long-range dependence while attenuating low-frequency components less aggressively than conventional integer differencing (Baillie, 1996; Robinson, 2003). Accordingly, fractional differencing has been studied as a statistical transformation for controlling persistence, long memory, and non-stationarity (Geweke & Porter-Hudak, 1983; Beran, 1995; Furuoka et al., 2024). In this work, we employ the fractional operator within a trajectory alignment objective, where it induces a structured metric on trajectory differences.

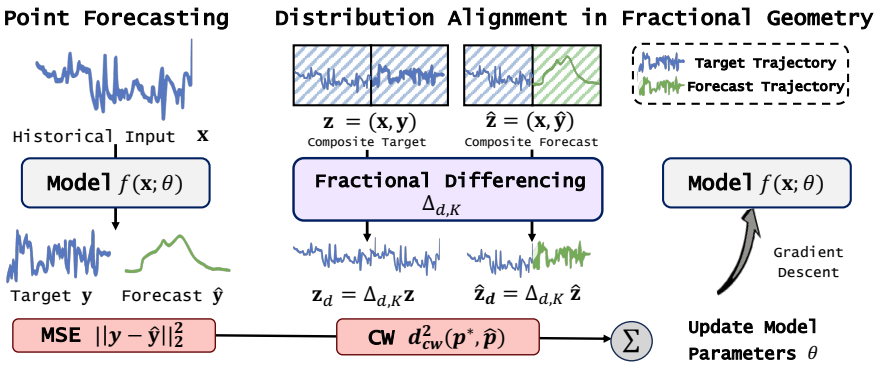


Figure 1: Overview of our proposed method PRODA.

3 PROPOSED METHOD

In this section, we introduce PRODA, a projection-based alignment method for time-series forecasting with a fractional trajectory metric. The overall procedure is illustrated in Figure 1.

3.1 PROBLEM SETUP

Throughout this paper, the direct multi-step forecasting setting commonly used in long-horizon benchmarks is adopted (Nie et al., 2023; Liu et al., 2024b). Let $\mathbf{z} = (z_1, \ldots, z_N) \in \mathbb{R}^N$ be a trajectory of length $N = L + F$, and let p^* denote the ground-truth distribution induced by $\mathbf{z}$. The trajectory $\mathbf{z}$ is decomposed into historical observations $\mathbf{x}$ and future targets $\mathbf{y}$ as $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^L \times \mathbb{R}^F$, where $\mathbf{x} = (z_1, \ldots, z_L)$ and $\mathbf{y} = (z_{L+1}, \ldots, z_N)$.

A forecasting model $f(\cdot; \theta) : \mathbb{R}^L \to \mathbb{R}^F$ produces the forecast $\hat{\mathbf{y}} = (\hat{z}_{L+1}, \cdots, \hat{z}_N) = f(\mathbf{x}; \theta)$. To formulate trajectory-level distribution alignment, history–forecast joint trajectories are defined as $\hat{\mathbf{z}} = (\mathbf{x}, \hat{\mathbf{y}}) \in \mathbb{R}^N$. We then define $\hat{p}$ as the distribution induced by $\hat{\mathbf{z}}$.

3.2 CRAMÉR–WOLD ALIGNMENT AND ITS GEOMETRY

Trajectory-level distribution alignment requires a discrepancy that can distinguish high-dimensional joint distributions without assuming a specific parametric family. The Cramér–Wold discrepancy satisfies this requirement by comparing multivariate distributions through their one-dimensional linear projections (Knop et al., 2020). By the Cramér–Wold device (Cramér & Wold, 1936) (Theorem 1 in Appendix A.1), equality of all one-dimensional projected distributions is equivalent to equality of the original multivariate distributions.

For a collection $\mathbf{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^n \subset \mathbb{R}^N$, the density estimator along the projected direction $\nu \in \mathbb{S}^{N-1}$ is defined as $\mathrm{sm}_\kappa(\nu^\top \mathbf{Z}) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\nu^\top \mathbf{z}^{(i)}, \kappa)$, where $\mathcal{N}(\mu, \kappa)$ denotes a Gaussian probability density function with mean μ and variance $\kappa > 0$.

Definition 1 (Cramér–Wold discrepancy (Knop et al., 2020))**.** *Given two distributions $p^*(\mathbf{z})$ and $\hat{p}(\hat{\mathbf{z}})$, where $\mathbf{z}, \hat{\mathbf{z}} \in \mathbb{R}^N$, let $\mathbf{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^n$ and $\hat{\mathbf{Z}} = \{\hat{\mathbf{z}}^{(i)}\}_{i=1}^n$ denote empirical samples from the respective distributions. The squared Cramér–Wold discrepancy is defined as*

$$d^2_{\mathrm{cw},\kappa}\left(p^*(\mathbf{z}), \hat{p}(\hat{\mathbf{z}})\right) := \int_{\mathbb{S}^{N-1}} \left\| \mathrm{sm}_\kappa(\nu^\top \mathbf{Z}) - \mathrm{sm}_\kappa(\nu^\top \hat{\mathbf{Z}}) \right\|^2_{L^2(\mathbb{R})} d\sigma_N(\nu), \tag{1}$$

where $\|\cdot\|_{L_2(\mathbb{R})}$ denotes the standard norm on the space of square-integrable functions $L_2(\mathbb{R})$ and σ_N denotes the normalized uniform surface measure on $\mathbb{S}^{N-1}$.

The Cramér–Wold discrepancy admits a closed-form kernel representation whose pairwise interaction depends on the Euclidean distance between trajectory samples (Knop et al., 2020). In particular, the associated kernel can be written in the radial form in equation 3 (see Appendix A.5 for the closed-form expression of equation 1). Thus, Cramér–Wold alignment inherits the Euclidean geometry of the trajectory representation space.

3.3 Fractional Trajectory Metric

However, in non-stationary time series, slowly varying level and trend components can dominate Euclidean pairwise distances and obscure localized temporal discrepancies (Liu et al., 2023; Cortes et al., 2025). To introduce temporal structure into trajectory comparison, the fractional differencing operator $(1-B)^d$ is applied to the joint trajectories. For differencing order $d > 0$,

$$\mathbf{z}_d = (1-B)^d \mathbf{z}, \quad \hat{\mathbf{z}}_d = (1-B)^d \hat{\mathbf{z}},$$

where $(1-B)^d = \sum_{k=0}^{\infty} (-1)^k \binom{d}{k} B^k$ and B is the backshift operator.

For a finite trajectory, we consider a truncated fractional differencing operator with maximum lag $K < N$. Let the K-truncated fractionally differenced sequence $\mathbf{z}_{d,K} = (z_{d,K,1}, \ldots, z_{d,K,N})^\top \in \mathbb{R}^N$, whose t-th element is given by $z_{d,K,t} = \sum_{k=0}^{\min(t-1,K)} (-1)^k \binom{d}{k} z_{t-k}, t \in \{1, \ldots, N\}$. In matrix form, this linear operation is represented by $\mathbf{z}_{d,K} = \Delta_{d,K}\mathbf{z}$ and $\hat{\mathbf{z}}_{d,K} = \Delta_{d,K}\hat{\mathbf{z}}$, where $\Delta_{d,K} \in \mathbb{R}^{N \times N}$ is a lower triangular banded Toeplitz matrix given by $[\Delta_{d,K}]_{i,j} = (-1)^{i-j}\binom{d}{i-j}$, if $0 \le i - j \le K$, and 0 otherwise. A detailed matrix representation is provided in Appendix A.2.

Under this linear mapping, the similarity between two trajectories is naturally evaluated in the transformed feature space. The Euclidean distance in the transformed space is therefore

$$\|\Delta_{d,K}(\mathbf{z} - \mathbf{z}')\|_2^2 = (\mathbf{z} - \mathbf{z}')^\top \Delta_{d,K}^\top \Delta_{d,K}(\mathbf{z} - \mathbf{z}').$$

Hence, fractional differencing induces the trajectory metric $\Delta_{d,K}^\top \Delta_{d,K}$, under which pairwise trajectory differences are evaluated. Figure 2 visualizes the induced metric for $N = 96$, $d = 0.2$, and $K = 70$. Its banded structure highlights the temporal coupling induced by fractional differencing. The matrix magnitudes are normalized and power-scaled for visualization.

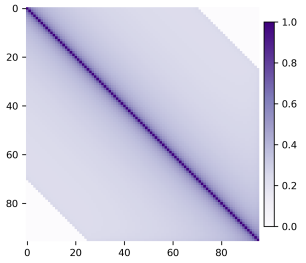


Figure 2: Visualization of the induced trajectory metric.

Objective. The objective function consists of L_2 loss and the squared Cramér–Wold discrepancy:

$$\min_{\theta} \quad (1-\gamma) \cdot \mathbb{E}_{p^*(\mathbf{x},\mathbf{y})} \left[\|\mathbf{y} - f(\mathbf{x};\theta)\|_2^2 \right] + \gamma \cdot d_{\mathrm{cw},\kappa}^2(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K})), \tag{2}$$

where $\gamma \in [0,1]$ is a tuning parameter balancing pointwise accuracy and trajectory-level alignment.

The objective in equation 2 combines two complementary forms of regularization. First, fractional differencing induces a frequency-weighted trajectory geometry, attenuating slowly varying discrepancies while relatively emphasizing higher-frequency variations, whereas the Cramér–Wold term constrains joint distributional structure beyond the conditional mean, including higher-order moments and cross-temporal dependence. Second, its closed-form kernel representation provides a kernel-depth interpretation (Chen et al., 2008), encouraging predicted trajectories toward high-depth regions of the observed trajectory distribution while discouraging excessive concentration among predictions. A detailed interpretation is provided in Appendix A.5.

3.4 Theoretical Analysis

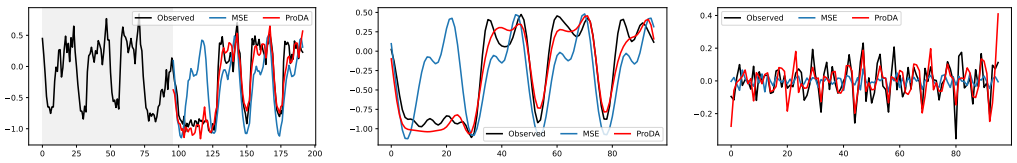
In this subsection, we examine the theoretical properties of the fractional trajectory metric, focusing on its induced kernel and spectral interpretation.

Proposition 1 (Fractional-differencing-induced Cramér–Wold kernel)**.** *Let the characteristic radial kernel associated with the Cramér–Wold discrepancy with smoothing parameter $\kappa > 0$ be*

$$k_{\mathrm{cw},\kappa}(\mathbf{z}, \mathbf{z}') = \frac{1}{2\sqrt{\pi\kappa}} \varphi_N \left(\frac{(\mathbf{z} - \mathbf{z}')^\top \mathbf{I}_N \, (\mathbf{z} - \mathbf{z}')}{4\kappa} \right), \tag{3}$$

where φ_N is its radial profile and $\mathbf{I}_N$ the identity matrix. Let $k_{d,\kappa}(\mathbf{z}, \mathbf{z}') \coloneqq k_{\mathrm{cw},\kappa}(\Delta_{d,K}\mathbf{z}, \Delta_{d,K}\mathbf{z}')$.

1. *The kernel $k_{d,\kappa}(\mathbf{z}, \mathbf{z}')$ is positive definite and characteristic on the original domain space of $\mathbf{z}$.*
2. *The Cramér–Wold discrepancy in the differenced time-series space is equivalent to the kernel MMD in the original time-series space, $d_{\mathrm{cw},\kappa}^2(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K})) = \mathrm{MMD}_{k_{d,\kappa}}^2(p^*(\mathbf{z}), \hat{p}(\hat{\mathbf{z}}))$.*



(a) Original forecast trajectories (b) Low-frequency components (c) High-frequency components

Figure 3: Frequency-decomposed forecasting visualization on the ECL dataset using iTransformer.

Proposition 1 shows that fractional differencing induces a temporally structured geometry for Cramér–Wold distribution alignment in which the pairwise distance underlying the induced kernel is determined by the quadratic form

$$k_{d,\kappa}(\mathbf{z}, \mathbf{z}') = \frac{1}{2\sqrt{\pi\kappa}}\, \varphi_N\left(\frac{(\mathbf{z}-\mathbf{z}')^\top \Delta_{d,K}^\top \Delta_{d,K}(\mathbf{z}-\mathbf{z}')}{4\kappa}\right).$$

This perspective reveals that the choice of the differencing order d is essentially equivalent to selecting an anisotropic Mahalanobis metric $\mathbf{W}^{(d)}$.

Proposition 2 (Spectral characterization of the induced metric)**.** *Let the frequency response of the truncated fractional differencing operator $\Delta_{d,K}$ with $d > 0$ be $H_{d,K}(\omega) = \sum_{k=0}^{K}(-1)^k \binom{d}{k} \exp(-i\omega k)$, where $K < N$ is the maximum lag and $\omega \in [-\pi, \pi]$ denotes the angular frequency. Suppose that the trajectory discrepancy is square summable, $\sum_{l=1}^{\infty} |z_l - \hat{z}_l|^2 < \infty$. Then,*

$$\|\mathbf{z}_{d,K} - \hat{\mathbf{z}}_{d,K}\|_2^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_{d,K}(\omega)|^2 \left\| \sum_{l=1}^{N} (z_l - \hat{z}_l) \exp(-i\omega l) \right\|_2^2 d\omega + R_{N,K} \tag{4}$$

$$\to \frac{1}{2\pi}\int_{-\pi}^{\pi} \left(2\left|\sin\frac{\omega}{2}\right|\right)^{2d} \left\| \sum_{l=1}^{\infty} (z_l - \hat{z}_l) \exp(-i\omega l) \right\|_2^2 d\omega \quad \text{as } K \to \infty, \tag{5}$$

where $R_{N,K}$ denotes the finite-boundary remainder induced by trajectory truncation.

Proposition 2 provides a frequency-domain interpretation of the induced trajectory metric. As shown in equation 4, the squared magnitude response $|H_{d,K}(\omega)|^2$ determines how trajectory discrepancies at different frequencies contribute to the transformed distance. In the limit equation 5, this weighting converges to $(2|\sin(\omega/2)|)^{2d}$ as $K \to \infty$, revealing the frequency-selective structure induced by fractional differencing.

In addition, Figure 4 shows the theoretical spectral weight $(2|\sin(\omega/2)|)^{2d}$ for various d, demonstrating that larger d imposes stronger non-uniform spectral weighting across frequencies. Consequently, when optimizing $\hat{\mathbf{z}}_{d,K}$ via this induced metric, larger values of d assign relatively higher weights to signal reconstruction in high-frequency bands and backpropagate larger gradients accordingly. This enables the forecasting model to faithfully recover sharp signal transitions and fine-grained localized fluctuations. In contrast, smaller values of d constrain the model to prioritize smoothly varying, long-term global dynamics and macro-trends.

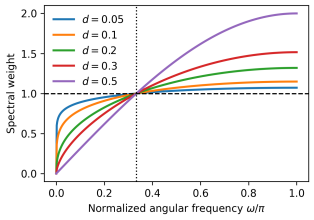


Figure 4: Theoretical spectral weight.

Spectral behavior of the learned forecasts. Figure 3 provides a qualitative view of how the frequency-selective geometry is reflected in the learned forecasts. The original trajectories in Figure 3a show that ProDA follows the target trajectory more closely than standard MSE training. To examine this behavior across frequency ranges, the target and predicted trajectories are decomposed into low- and high-frequency components using the cutoff $\omega/\pi = 1/3$, consistent with the spectral analysis. As shown in Figure 3b, ProDA more closely preserves the slowly varying trajectory structure, while Figure 3c shows that localized high-frequency variations are also retained. This behavior is consistent with the frequency-selective weighting induced by fractional differencing, which relatively emphasizes higher-frequency discrepancies while still preserving low-frequency trajectory structure.

Table 1: **Main results** for long-term multivariate forecasting are reported across forecasting horizons $\{96, 192, 336, 720\}$. The better results are highlighted in **bold**.

model f		iTransformer				PatchTST				TimeMixer				DLinear			
loss		Original		+ ProDA		Original		+ ProDA		Original		+ ProDA		Original		+ ProDA	
metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.393**	**0.416**	0.402	0.418	**0.380**	**0.398**	0.386	0.407	0.416	0.421	**0.408**	**0.418**	0.390	0.405	**0.370**	**0.392**
	192	0.448	**0.449**	**0.442**	0.455	0.429	0.432	**0.417**	**0.429**	0.460	0.446	**0.459**	**0.462**	0.435	0.430	**0.404**	**0.412**
	336	0.475	**0.459**	**0.452**	0.462	0.483	0.464	**0.436**	**0.450**	0.521	0.480	**0.463**	**0.470**	0.478	0.455	**0.432**	**0.433**
	720	**0.493**	0.492	0.494	**0.490**	0.496	0.491	**0.456**	**0.476**	0.563	0.526	**0.474**	**0.484**	0.502	0.500	**0.454**	**0.471**
	Avg.	0.452	**0.454**	**0.447**	0.456	0.447	0.446	**0.424**	**0.441**	0.490	0.468	**0.451**	**0.458**	0.451	0.447	**0.415**	**0.427**
ETTh2	96	0.332	0.376	**0.316**	**0.363**	**0.291**	**0.343**	0.311	0.366	0.325	0.363	**0.294**	**0.359**	0.319	0.376	**0.280**	**0.349**
	192	**0.394**	**0.416**	0.396	0.426	0.374	**0.397**	**0.372**	0.411	0.431	0.427	**0.379**	**0.417**	0.456	0.460	**0.383**	**0.420**
	336	0.420	0.442	**0.416**	**0.438**	0.423	0.436	**0.395**	**0.426**	0.440	0.448	**0.379**	**0.420**	0.543	0.515	**0.416**	**0.444**
	720	0.437	0.462	**0.435**	**0.450**	0.429	**0.449**	**0.428**	0.454	0.430	0.448	**0.399**	**0.440**	0.787	0.636	**0.642**	**0.581**
	Avg.	0.396	0.424	**0.391**	**0.419**	0.379	**0.406**	**0.378**	0.414	0.406	0.422	**0.363**	**0.409**	0.526	0.497	**0.430**	**0.449**
ETTm1	96	0.331	0.375	**0.324**	**0.372**	0.336	0.366	**0.300**	**0.356**	0.327	0.366	**0.315**	**0.364**	0.345	0.372	**0.299**	**0.342**
	192	0.367	0.396	**0.367**	**0.395**	0.368	0.389	**0.342**	**0.377**	0.370	0.397	**0.370**	**0.394**	0.380	0.390	**0.334**	**0.364**
	336	0.409	0.421	**0.404**	**0.416**	0.397	0.410	**0.382**	**0.408**	0.414	0.418	**0.385**	**0.396**	0.413	0.415	**0.365**	**0.383**
	720	0.466	0.453	**0.452**	**0.446**	0.455	0.444	**0.430**	**0.430**	0.475	0.455	**0.432**	**0.424**	0.476	0.457	**0.418**	**0.414**
	Avg.	0.393	0.411	**0.387**	**0.407**	0.389	0.402	**0.363**	**0.393**	0.397	0.409	**0.377**	**0.395**	0.404	0.408	**0.354**	**0.376**
ETTm2	96	**0.207**	**0.298**	**0.207**	**0.298**	0.177	0.259	**0.173**	**0.222**	0.186	0.270	**0.174**	**0.264**	0.184	0.276	**0.167**	**0.260**
	192	**0.268**	0.338	**0.268**	**0.336**	0.245	0.306	**0.228**	**0.301**	0.253	0.311	**0.224**	**0.298**	0.255	0.329	**0.231**	**0.308**
	336	0.317	0.368	**0.310**	**0.358**	0.306	0.345	**0.303**	**0.337**	0.365	0.369	**0.296**	**0.345**	0.362	0.410	**0.291**	**0.351**
	720	0.404	0.413	**0.402**	**0.411**	0.404	0.403	**0.386**	**0.402**	0.425	0.408	**0.383**	**0.407**	0.486	0.484	**0.386**	**0.409**
	Avg.	0.299	0.354	**0.297**	**0.351**	0.283	0.328	**0.273**	**0.326**	0.307	0.340	**0.270**	**0.328**	0.322	0.375	**0.269**	**0.332**
Weather	96	0.186	0.237	**0.169**	**0.222**	0.152	0.213	**0.150**	**0.200**	0.153	0.210	**0.149**	**0.204**	0.178	0.241	**0.170**	**0.227**
	192	0.217	0.260	**0.216**	**0.264**	0.206	0.253	**0.197**	**0.246**	0.213	0.263	**0.208**	**0.259**	0.217	0.278	**0.214**	**0.264**
	336	0.281	0.306	**0.263**	**0.297**	0.255	0.295	**0.248**	**0.285**	0.280	0.302	**0.272**	**0.301**	0.265	0.317	**0.255**	**0.299**
	720	0.343	0.347	**0.327**	**0.345**	0.321	0.344	**0.317**	**0.335**	**0.318**	0.342	**0.318**	**0.333**	0.327	0.365	**0.321**	**0.353**
	Avg.	0.257	0.287	**0.244**	**0.282**	0.234	0.276	**0.228**	**0.267**	0.241	0.279	**0.237**	**0.274**	0.247	0.300	**0.240**	**0.286**
ECL	96	**0.135**	0.234	**0.135**	**0.232**	0.151	0.250	**0.142**	**0.243**	0.158	0.261	**0.149**	**0.250**	0.145	0.237	**0.140**	**0.236**
	192	**0.159**	**0.253**	**0.159**	0.257	0.160	0.268	**0.157**	**0.254**	0.179	0.282	**0.163**	**0.262**	0.154	0.251	**0.153**	**0.248**
	336	0.172	0.269	**0.171**	**0.267**	0.178	0.275	**0.173**	**0.270**	0.201	0.305	**0.180**	**0.279**	0.176	0.266	**0.169**	**0.264**
	720	0.204	0.297	**0.203**	**0.293**	0.221	0.308	**0.212**	**0.299**	**0.209**	**0.307**	0.219	0.310	**0.204**	0.309	**0.204**	**0.295**
	Avg.	0.168	**0.262**	**0.167**	**0.262**	0.178	0.273	**0.171**	**0.267**	0.187	0.289	**0.178**	**0.275**	0.170	0.266	**0.166**	**0.261**
Exchange	96	0.099	0.227	**0.096**	**0.224**	0.088	0.209	**0.086**	**0.210**	0.097	0.218	**0.093**	**0.214**	**0.079**	0.199	**0.079**	**0.198**
	192	0.200	0.323	**0.182**	**0.307**	0.200	0.319	**0.169**	**0.295**	0.242	0.354	**0.181**	**0.302**	0.154	0.290	**0.150**	**0.284**
	336	0.332	0.425	**0.331**	**0.417**	0.344	0.429	**0.308**	**0.408**	0.370	0.443	**0.358**	**0.437**	0.279	0.399	**0.259**	**0.385**
	720	0.842	0.698	**0.826**	**0.689**	0.950	0.738	**0.808**	**0.679**	0.941	0.731	**0.758**	**0.653**	0.722	**0.651**	**0.680**	0.658
	Avg.	0.368	0.418	**0.359**	**0.409**	0.395	0.424	**0.343**	**0.398**	0.413	0.437	**0.348**	**0.401**	0.308	0.385	**0.292**	**0.381**
PEMS08	12	0.091	0.198	**0.086**	**0.192**	0.100	0.213	**0.099**	**0.209**	0.083	0.192	0.085	**0.194**	**0.112**	**0.222**	**0.112**	0.223
	24	0.123	0.227	**0.118**	**0.222**	0.168	0.280	**0.156**	**0.267**	**0.128**	**0.244**	0.133	0.251	**0.195**	0.300	**0.195**	**0.298**
	48	0.286	0.368	**0.254**	**0.337**	0.266	0.347	**0.260**	**0.337**	**0.229**	**0.333**	0.235	0.340	0.382	0.424	**0.381**	**0.421**
	96	0.443	0.437	**0.394**	**0.410**	0.439	0.440	**0.432**	**0.436**	0.415	0.449	**0.399**	**0.438**	**0.644**	**0.538**	0.659	0.534
	Avg.	0.236	0.307	**0.213**	**0.290**	0.243	0.320	**0.237**	**0.312**	0.214	0.305	**0.213**	**0.305**	**0.333**	0.371	0.336	**0.369**

4 EXPERIMENTS

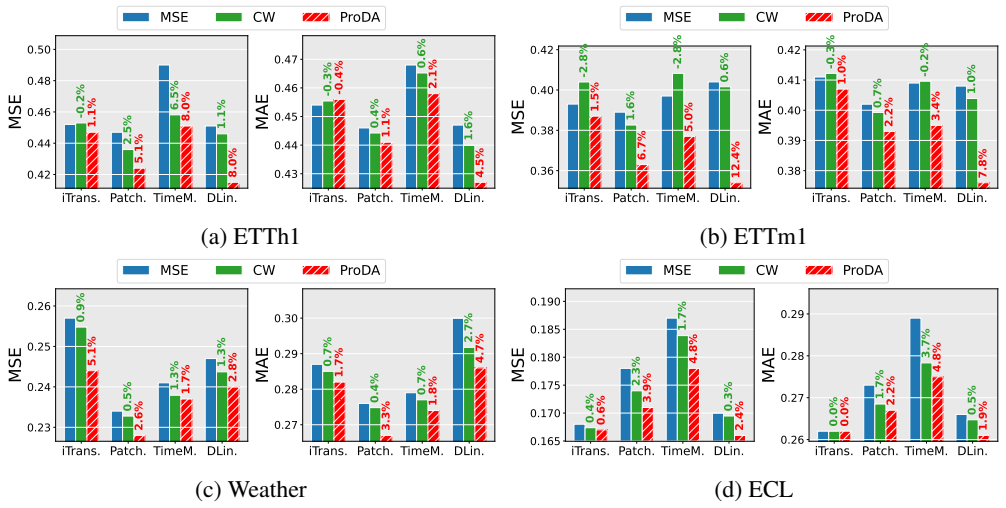
4.1 SETUP

Datasets. Our experiments are conducted on eight widely used forecasting benchmarks: ETT (ETTh1, ETTh2, ETTm1, and ETTm2), Weather, Electricity (ECL), Exchange, and PEMS08. Dataset statistics and detailed descriptions are provided in Table 5 of Appendix B.2.

Baselines. Four representative time-series forecasting backbones are considered: iTransformer (Liu et al., 2024b), PatchTST (Nie et al., 2023), TimeMixer (Wang et al., 2024), and DLinear (Zeng et al., 2023). For comparison with recent objective-design methods, Koopman (Lange et al., 2021), AliO (Park et al., 2025), PSLoss (Kudrat et al., 2025), Time-o1 (Wang et al., 2026b), DBLoss (Qiu et al., 2026), FreDF (Wang et al., 2025), and DistDF (Wang et al., 2026a) are additionally evaluated, with MSE serving as the standard training objective. Detailed descriptions of the forecasting backbones and comparison objectives are provided in Appendix B.3 and Appendix B.4, respectively.

Table 2: **Comparison of other objectives** on three datasets. The means across forecasting horizons $\{96, 192, 336, 720\}$ are reported. The best result is **bolded**, and the second best is underlined.

objectives		**ProDA**		DistDF		FreDF		DBLoss		Time-o1		PSLoss		AliO		Koopman	
metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
iTransformer	ETTh1	0.447	0.456	0.458	0.457	**0.444**	**0.445**	0.453	0.450	0.456	0.454	0.445	0.447	0.462	0.459	0.460	0.458
	Weather	**0.244**	0.282	0.259	0.282	0.255	0.278	0.250	**0.271**	0.257	0.281	0.249	0.272	0.258	0.281	0.259	0.282
	ECL	**0.167**	**0.262**	0.177	0.273	0.184	0.280	0.179	0.276	0.176	0.272	0.186	0.282	0.180	0.277	0.178	0.275
PatchTST	ETTh1	**0.424**	0.441	0.437	0.441	0.450	0.451	0.435	**0.436**	0.436	0.443	0.443	0.440	0.441	0.444	0.443	0.445
	Weather	**0.228**	**0.267**	0.240	0.276	0.246	0.281	0.235	0.270	0.243	0.278	0.248	0.283	0.241	0.275	0.239	0.272
	ECL	**0.171**	**0.267**	0.181	0.278	0.186	0.285	0.177	0.273	0.180	0.277	0.188	0.287	0.183	0.280	0.181	0.278
TimeMixer	ETTh1	**0.451**	**0.458**	0.479	0.463	0.456	0.459	0.475	0.461	0.473	0.460	0.472	0.460	0.479	0.463	0.483	0.465
	Weather	**0.237**	**0.274**	0.251	0.283	0.255	0.285	0.246	0.275	0.251	0.281	0.251	0.283	0.249	0.280	0.248	0.281
	ECL	**0.178**	**0.275**	0.183	0.282	0.180	0.278	0.184	0.283	0.186	0.284	0.181	0.281	0.184	0.283	0.187	0.286
DLinear	ETTh1	**0.415**	**0.427**	0.452	0.448	0.438	0.436	0.426	0.436	0.423	0.444	0.424	0.438	0.454	0.450	0.450	0.446
	Weather	**0.240**	0.286	0.247	0.300	0.243	0.296	0.245	0.288	0.244	0.293	0.244	**0.284**	0.247	0.300	0.249	0.298
	ECL	**0.166**	**0.261**	0.170	0.266	0.170	0.262	**0.166**	0.262	0.170	0.264	0.171	0.262	0.170	0.266	0.171	0.267



(a) ETTh1 (b) ETTm1

(c) Weather (d) ECL

Figure 5: **Ablation studies** on the ETTh1, ETTm1, Weather, and ECL datasets. 'CW' denotes the variant without fractional differencing.

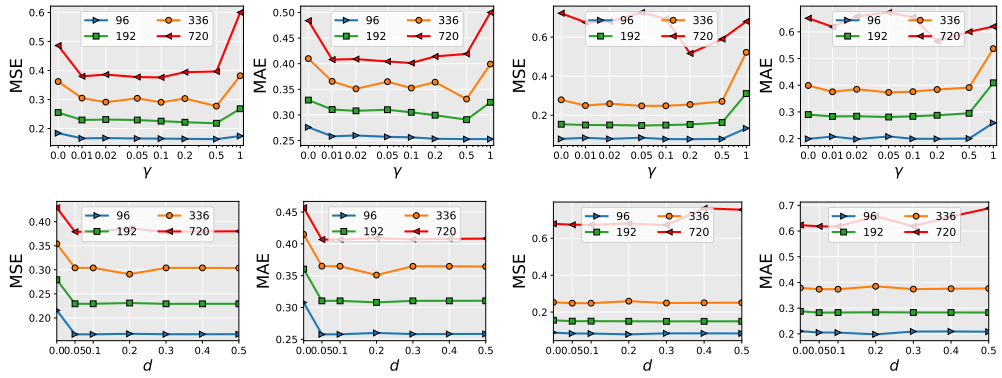
Implementation details. For consistency with previous work, we adopt MSE and Mean Absolute Error (MAE) as evaluation metrics. We consider forecasting horizons $F \in \{96, 192, 336, 720\}$ across all datasets, with the exception of PEMS08, for which we use $F \in \{12, 24, 48, 96\}$. We utilize the comprehensive time series forecasting benchmark TFB (Qiu et al., 2024) for unified evaluation. All experiments are conducted on an NVIDIA RTX A6000 GPU using implementations in `PyTorch`. The training process employs the Adam optimizer (Kingma & Ba, 2015). The initial batch size is set to 512, and is adaptively reduced by half (down to a minimum of 64) if out-of-memory (OOM) issues occur. The detailed experimental settings are provided in Appendix B.5.

4.2 Results

Main results. As shown in Table 1, ProDA provides consistent improvements across different forecasting backbones and datasets. Across the 32 backbone–dataset combinations, ProDA improves average MSE in 31 cases and average MAE in 28 cases, with two additional MAE ties. The only exception in average MSE is DLinear on PEMS08, where MSE slightly increases from 0.333 to 0.336 while MAE improves from 0.371 to 0.369. For example, on ETTh2 with DLinear, ProDA reduces average MSE from 0.526 to 0.430 and MSE at the longest horizon from 0.787 to 0.642. Overall, across all 128 horizon-level MSE comparisons, ProDA achieves 107 wins, 10 losses, and 11 ties, with a one-sided sign test yielding p-value < 0.001.

Table 3: **Zero-shot forecasting performance** on ETT datasets. The results are for baseline models trained with and without PRODA for zero-shot forecasting. The means across forecasting horizons {96, 192, 336, 720} are reported. The better results are highlighted in **bold**.

model		iTransformer				PatchTST				TimeMixer				DLinear			
loss		Original		+ PRODA		Original		+ PRODA		Original		+ PRODA		Original		+ PRODA	
source	target	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	ETTh2	0.380	0.401	**0.366**	**0.400**	0.383	0.403	**0.364**	**0.402**	0.387	0.408	**0.359**	**0.397**	0.464	0.467	**0.437**	**0.450**
	ETTm1	**0.889**	**0.601**	0.907	0.612	0.873	0.604	**0.810**	**0.598**	0.796	**0.578**	**0.794**	0.593	0.755	0.578	**0.731**	**0.569**
	ETTm2	0.327	**0.367**	**0.321**	0.369	0.319	**0.361**	**0.310**	0.362	0.314	**0.357**	**0.309**	0.361	0.398	0.442	**0.383**	**0.427**
ETTh2	ETTh1	**0.658**	**0.564**	0.670	0.575	**0.627**	**0.545**	0.665	0.569	1.122	0.689	**0.725**	**0.596**	0.484	0.475	**0.450**	**0.458**
	ETTm1	**1.008**	**0.640**	1.012	0.646	0.936	0.612	**0.744**	**0.551**	2.209	0.860	**0.899**	**0.601**	0.792	0.592	**0.731**	**0.567**
	ETTm2	0.336	**0.375**	**0.331**	**0.375**	0.331	0.369	**0.295**	**0.353**	0.386	0.399	**0.304**	**0.358**	0.464	0.476	**0.409**	**0.438**
ETTm1	ETTh1	0.907	0.642	**0.646**	**0.551**	0.678	0.566	**0.523**	**0.483**	1.037	0.719	**0.645**	**0.558**	0.587	0.518	**0.470**	**0.466**
	ETTh2	0.508	0.475	**0.415**	**0.435**	0.449	0.445	**0.396**	**0.420**	0.513	0.484	**0.380**	**0.419**	0.475	0.477	**0.409**	**0.438**
	ETTm2	0.316	0.348	**0.278**	**0.330**	0.296	0.334	**0.270**	**0.323**	0.311	0.346	**0.265**	**0.321**	0.347	0.395	**0.307**	**0.363**
ETTm2	ETTh1	**0.911**	**0.644**	0.999	0.680	**0.665**	**0.568**	**0.665**	**0.568**	1.656	0.753	**0.904**	**0.668**	0.591	0.522	**0.477**	**0.473**
	ETTh2	0.455	**0.452**	**0.438**	0.453	0.411	0.425	**0.379**	**0.414**	0.429	0.439	**0.393**	**0.429**	0.440	0.452	**0.369**	**0.409**
	ETTm1	0.707	0.543	**0.577**	**0.511**	0.626	0.517	**0.487**	**0.454**	0.770	0.553	**0.492**	**0.461**	0.510	0.464	**0.405**	**0.413**



(a) ETTm2 (b) Exchange

Figure 6: **Hyperparameter sensitivity analyses** of γ (top) and d (bottom) with DLinear.

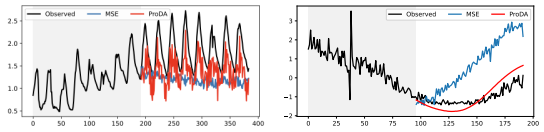
Comparison of other objectives. Table 2 compares PRODA with recent objective-centric baselines across four forecasting backbones and three datasets. Across the 12 backbone–dataset settings, PRODA achieves the best average MSE in 11 settings, including one tie, and the best average MAE in 8 settings. In particular, compared with DistDF, which aligns trajectory distributions using a Wasserstein-based discrepancy, PRODA improves average MSE in all 12 settings and average MAE in 10 settings, with two ties. Across all 84 pairwise MSE comparisons against the seven competing objectives, PRODA records 81 wins, 1 tie, and 2 losses, yielding a one-sided sign test p-value < 0.001. These results further support the model-agnostic applicability of PRODA, showing that its advantage over alternative objective designs is maintained across different forecasting architectures. Detailed horizon-wise comparisons for each backbone are provided in Appendix C.1.

Ablation studies. As shown in Figure 5, Cramér–Wold alignment alone provides effective trajectory-level distributional regularization beyond MSE, while the full PRODA further performs this regularization under the fractional-differencing-induced trajectory metric, generally yielding larger and more consistent gains across datasets and backbones. For example, on ETTm1 with DLinear, PRODA achieves a 12.4% reduction in MSE and a 7.8% reduction in MAE over the MSE baseline. These results suggest that the induced metric provides additional benefit by reshaping the geometry of trajectory-level distributional regularization. Additional ablation results are provided in Appendix C.2.

Zero-shot forecasting performance. Table 3 shows that the benefit of PRODA extends beyond in-domain training to zero-shot transfer across the ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2). Across the 48 source–target–backbone comparisons in the average results, PRODA improves MSE

Table 4: **Comparison of alternative discrepancies** for trajectory-level alignment using DLinear. The best result is **bolded**, and the second best is underlined.

discrepancy		PRODA		MMD@Linear		MMD@RBF		KL	
	metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear	ETTh1	**0.415**	**0.427**	**0.415**	0.430	0.417	0.429	0.418	0.430
	Weather	**0.240**	**0.266**	0.243	0.268	0.242	0.266	0.243	0.267
	ECL	**0.166**	**0.261**	0.173	0.269	0.172	0.266	0.176	0.270



(a) ETTh2 (b) PEMS08

Figure 7: **Forecasting visualization** of PRODA using iTransformer.

in 42 cases and MAE in 36 cases, with DLinear improving all 12 transfer pairs on both metrics and TimeMixer improving all 12 pairs in MSE. This suggests that projection-based distribution alignment improves cross-dataset generalization under source–target distribution shifts without target-specific fine-tuning. Detailed source–target and horizon-wise results are provided in Appendix C.3.

Hyperparameter sensitivity analyses. Figure 6 shows that PRODA is reasonably robust to the choices of γ and d on both ETTm2 and Exchange. For γ, small to moderate values generally perform well, while $\gamma = 1$ leads to clear degradation. It suggests that Cramér–Wold alignment is more effective as a regularizer alongside MSE than as a standalone objective. In contrast, performance remains stable across a broad range of d, indicating robustness to the choice of fractional order. These results suggest that the benefit of the induced fractional trajectory metric does not rely on a narrowly tuned spectral geometry. Additional sensitivity results on other datasets are provided in Appendix C.4.

Comparison of alternative discrepancies. Table 4 compares the Cramér–Wold discrepancy with alternative distribution alignment objectives, including MMD with linear and RBF kernels and KL divergence. PRODA shows consistently competitive performance across the three datasets and both evaluation metrics. The differences are modest on ETTh1 and Weather but more pronounced on ECL, where PRODA improves MSE from 0.172 to 0.166 and MAE from 0.266 to 0.261 over MMD@RBF. These results empirically suggest that projection-based distribution comparison across multiple directions can be effective for joint trajectory alignment.

Forecasting visualization. Figure 7 provides qualitative examples from the iTransformer model. Consistent with the frequency-selective interpretation of the induced trajectory metric, PRODA better preserves both localized temporal variations and the overall trajectory structure than MSE training. On ETTh2 in Figure 7a, PRODA more closely follows short-term fluctuations and oscillatory patterns, including peak–trough variations that are smoothed or underestimated by the MSE baseline. On PEMS08 in Figure 7b, it also better preserves the slowly varying level and trend, while MSE exhibits substantial drift over the forecasting horizon. These examples suggest that the relative emphasis on higher-frequency discrepancies does not discard low-frequency structure, but instead improves trajectory reconstruction across temporal scales. Further visualizations are provided in Appendix C.5.

5 CONCLUSION AND LIMITATIONS

We introduced PRODA, a model-agnostic objective for time-series forecasting that aligns history–future and history–forecast trajectory distributions through the Cramér–Wold discrepancy. By incorporating fractional differencing into the alignment objective, PRODA induces a structured trajectory metric that reweights trajectory discrepancies across temporal frequencies. We theoretically showed that the induced Cramér–Wold kernel remains characteristic and characterized the resulting metric through its frequency-selective spectral weighting. Empirically, PRODA consistently improves forecasting performance across diverse backbones and datasets, compares favorably with recent objective-centric baselines, and shows consistent gains in zero-shot transfer across ETT variants.

However, several limitations remain. PRODA is applied variable-wise in multivariate forecasting, without explicitly modeling cross-variate dependencies, and its pairwise Cramér–Wold computation can increase cost on large-scale datasets. Future work includes scalable cross-variate alignment, more efficient pairwise computation, and evaluation under broader distribution shifts.

AI USE STATEMENT

Generative AI tools were used to aid language editing and improve the clarity of the manuscript. They were also used to assist with mathematical derivations and proofs, including checking intermediate arguments and suggesting possible proof steps. All AI-assisted mathematical content was independently examined and verified by the authors. The authors made all final decisions regarding the claims, proofs, and presentation and take full responsibility for the content of this work.

ETHICS STATEMENT

This work focuses on methodological advances in time-series forecasting and uses publicly available benchmark datasets. It does not involve human subjects, personally identifiable information, or the collection of sensitive data. We are not aware of any direct ethical concerns specific to the proposed methodology beyond those generally associated with the deployment of forecasting models in downstream applications.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed objective, fractional trajectory metric, and training procedure in the main text. The appendix includes complete proofs of the theoretical results, implementation and hyperparameter details, dataset and experimental settings, additional objective comparisons, ablation studies, repeated experiments, and zero-shot evaluations. All experiments are conducted using publicly available forecasting benchmarks and established backbone implementations. Source code and configuration files required to reproduce the experiments are provided in the supplementary material.

REFERENCES

Richard T. Baillie. Long memory processes and fractional integration in econometrics. *Journal of Econometrics*, 73(1):5–59, 1996. ISSN 0304-4076. doi: https://doi.org/10.1016/0304-4076(95)01732-1. URL https://www.sciencedirect.com/science/article/pii/0304407695017321.

Jan Beran. Maximum likelihood estimation of the differencing parameter for invertible short and long memory autoregressive integrated moving average models. *Journal of the royal statistical society series b-methodological*, 57:659–672, 1995. URL https://api.semanticscholar.org/CorpusID:118142388.

Yixin Chen, Xin Dang, Hanxiang Peng, and Henry L Bart. Outlier detection with the kernelized spatial depth function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2): 288–305, 2008.

Adrien Cortes, Remi Rehm, and Victor Letzelter. Winner-takes-all for multivariate probabilistic time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4QcFfTu6UT.

H. Cramér and H. Wold. Some theorems on distribution functions. *Journal of the London Mathematical Society*, s1-11(4):290–294, 1936. doi: https://doi.org/10.1112/jlms/s1-11.4.290. URL https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/jlms/s1-11.4.290.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 894–903. JMLR.org, 2017.

Marcos Lopez De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational*

Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i6.25914. URL https://doi.org/10.1609/aaai.v37i6.25914.

Fumitaka Furuoka, Luis Alberiko Gil-Alana, OlaOluwa Simon Yaya, Elayaraja Aruchunan, and Ahamuefula Ephraim Ogbonna. A new fractional integration approach based on neural network nonlinearity with an application to testing unemployment hysteresis. *Empirical Economics*, pp. 1–29, 2024. URL https://api.semanticscholar.org/CorpusID:266695730.

John Geweke and Susan Porter-Hudak. The estimation and application of long memory time series models. *Journal of Time Series Analysis*, 4(4):221–238, 1983. doi: https://doi.org/10.1111/j.1467-9892.1983.tb00371.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9892.1983.tb00371.x.

C. W. J. Granger and Roselyne Joyeux. An introduction to long-memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1(1):15–29, 1980. doi: https://doi.org/10.1111/j.1467-9892.1980.tb00297.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9892.1980.tb00297.x.

J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981. ISSN 00063444. URL http://www.jstor.org/stable/2335817.

Katie Kang, Amrith Setlur, Claire Tomlin, and Sergey Levine. Deep neural networks tend to extrapolate predictably. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ljwoQ3cvQh.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Szymon Knop, Przemysław Spurek, Jacek Tabor, Igor Podolak, Marcin Mazur, and Stanisław Jastrzebski. Cramer-wold auto-encoder. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=p1KkW2kgDp.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pp. 95–104, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210006. URL https://doi.org/10.1145/3209978.3210006.

Henning Lange, Steven L. Brunton, and J. Nathan Kutz. From fourier to koopman: spectral methods for long-term time series prediction. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/466accbac9a66b805ba50e42ad715740-Paper.pdf.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/6775a0635c302542da2c32aa19d86be0-Paper.pdf.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.03.012. URL https://www.sciencedirect.com/science/article/pii/S0169207021000637.

Haoxin Liu, Harshavardhan Kamarthi, Lingkai Kong, Zhiyuan Zhao, Chao Zhang, and B. Aditya Prakash. Time-series forecasting for out-of-distribution generalization using invariant learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31312–31325. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/liu24ae.html.

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 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=AyajSjTAzmg.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=A4zzxu82a7.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=JePfAI8fah.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Licheng Pan, Hao Wang, Hao Wang, Yuqi Li, Qingsong Wen, Xiaoxi Li, Zhichao Chen, Hao-Xuan Li, Zhixuan Chu, and Yuan Lu. Deep time-series forecasting needs kernelized moment balancing. *ArXiv*, abs/2602.00717, 2026. URL https://api.semanticscholar.org/CorpusID:285270716.

Kwangryeol Park, Jaeho Kim, and Seulki Lee. Alio: Output alignment matters in long-term time series forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=AuOZDp4gy7.

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. *Proc. VLDB Endow.*, 17(9):2363–2377, May 2024. ISSN 2150-8097. doi: 10.14778/3665844.3665863. URL https://doi.org/10.14778/3665844.3665863.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=SbhBIkiRLT.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, pp. 5301–5310, 2019.

Syama Sundar Rangapuram, Matthias Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 7796–7805, Red Hook, NY, USA, 2018. Curran Associates Inc.

Peter M Robinson (ed.). *Time Series With Long Memory*. Oxford University Press, 06 2003. ISBN 9780199257294. doi: 10.1093/oso/9780199257294.001.0001. URL https://doi.org/10.1093/oso/9780199257294.001.0001.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

Eric Wang, Licheng Pan, Yuan Lu, Zhixuan Chu, Xiaoxi Li, Shuting He, Zi Ciu Chan, Qingsong Wen, Haoxuan Li, and Zhouchen Lin. DistDF: Time-series forecasting needs joint-distribution wasserstein alignment. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=VrdLwUmzBy.

Hao Wang, Lichen 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4A9IdSa1ul.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026b. URL https://openreview.net/forum?id=RxWILaXuhb.

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

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, May 2021. doi: 10.1609/aaai.v35i12.17325. URL https://ojs.aaai.org/index.php/AAAI/article/view/17325.

A Proof/Mathematical Derivation

A.1 Cramér–Wold Device

Theorem 1 (Cramér–Wold Device (Cramér & Wold, 1936)). *Let $\{(\mathbf{z}^{(i)})\}$ be a random sequence of random vectors in $\mathbb{R}^H (H > 1)$. Then $\mathbf{z}^{(i)}$ converges in distribution to a random vector $\mathbf{z}$ if and only if:*

$$\nu^\top \mathbf{z}^{(i)} \xrightarrow{d} \nu^\top \mathbf{z}, \quad \forall \nu \in \mathbb{R}^H. \tag{6}$$

Proof. Please refer to the proof of Theorem I in Cramér & Wold (1936). □

Cramér–Wold device says that a multivariate distribution is fully characterized by its one-dimensional linear projections along all directions, thereby reducing a high-dimensional comparison problem to an aggregation of one-dimensional comparisons (Cramér & Wold, 1936). Accordingly, we compare forecast and ground-truth trajectories using a discrepancy between joint history–future distributions, defined over all possible directions of projection.

A.2 Invertibility of truncated fractional differencing operator

Lemma 1. *For a finite trajectory of length N, the truncated fractional differencing operator $\Delta_{d,K}$ is invertible.*

Proof. For a finite trajectory, let $\mathbf{z}_{d,K} = \Delta_{d,K}\mathbf{z}$ and $\hat{\mathbf{z}}_{d,K} = \Delta_{d,K}\hat{\mathbf{z}}$, where $\Delta_{d,K} \in \mathbb{R}^{N\times N}$ is the truncated fractional differencing operator:

$$\Delta_{d,K} = \begin{bmatrix} \delta_0 & 0 & 0 & \cdots & 0 \\ \delta_1 & \delta_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \delta_K & \cdots & \delta_1 & \delta_0 & \ddots \\ 0 & \delta_K & \cdots & \delta_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \qquad \delta_k = (-1)^k \binom{d}{k}, \tag{7}$$

with entries $[\Delta_{d,K}]_{i,j} = \delta_{i-j}$ if $0 \le i - j \le K$, and 0 otherwise. Since $\delta_0 = \binom{d}{0} = 1$, $\Delta_{d,K}$ is a lower-triangular matrix whose diagonal entries are all equal to one. Therefore, $\det(\Delta_{d,K}) = \prod_{h=1}^{N} [\Delta_{d,K}]_{hh} = 1 \neq 0$. Hence, $\Delta_{d,K}$ is invertible. □

A.3 Proof of Proposition 1

Proof. By Lemma 1, $\mathbf{z}_d = \Delta_d \mathbf{z}$, $\hat{\mathbf{z}}_d = \Delta_d \hat{\mathbf{z}}$. Since $k_{\mathrm{CW},\kappa}$ is positive definite, the composed kernel $k_{d,\kappa}(\mathbf{z}, \mathbf{z}') \coloneqq k_{\mathrm{CW},\kappa}(\Delta_d \mathbf{z}, \Delta_d \mathbf{z}')$ is also positive definite.

The squared MMD associated with $k_{d,\kappa}$ is

$$\begin{aligned} \mathrm{MMD}^2_{k_{d,\kappa}}(p^*(\mathbf{z}), \hat{p}(\hat{\mathbf{z}})) &= \iint k_{d,\kappa}(\mathbf{z}, \mathbf{z}') p^*(\mathbf{z}) p^*(\mathbf{z}') d\mathbf{z} d\mathbf{z}' + \iint k_{d,\kappa}(\hat{\mathbf{z}}, \hat{\mathbf{z}}') \hat{p}(\hat{\mathbf{z}}) \hat{p}(\hat{\mathbf{z}}') d\hat{\mathbf{z}} d\hat{\mathbf{z}}' \\ &\quad - 2 \iint k_{d,\kappa}(\mathbf{z}, \hat{\mathbf{z}}') p^*(\mathbf{z}) \hat{p}(\hat{\mathbf{z}}') d\mathbf{z} d\hat{\mathbf{z}} \qquad (8) \\ &= \iint k_{\mathrm{CW},\kappa}(\mathbf{z}, \mathbf{z}') p^*(\mathbf{z}) p^*(\mathbf{z}') d\mathbf{z} d\mathbf{z}' + \iint k_{\mathrm{CW},\kappa}(\hat{\mathbf{z}}, \hat{\mathbf{z}}') \hat{p}(\hat{\mathbf{z}}) \hat{p}(\hat{\mathbf{z}}') d\hat{\mathbf{z}} d\hat{\mathbf{z}}' \\ &\quad - 2 \iint k_{\mathrm{CW},\kappa}(\mathbf{z}, \hat{\mathbf{z}}') p^*(\mathbf{z}) \hat{p}(\hat{\mathbf{z}}') d\mathbf{z} d\hat{\mathbf{z}} \qquad (9) \\ &= \mathrm{MMD}^2_{k_{\mathrm{CW},\kappa}}(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K})) \qquad (10) \\ &= d^2_{\mathrm{cw}}(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K})). \qquad (11) \end{aligned}$$

Since $k_{\mathrm{CW},\kappa}$ is characteristic,

$$\mathrm{MMD}^2_{k_{\mathrm{CW},\kappa}}\left(p^*(\Delta_{d,K}\mathbf{z}), \hat{p}(\Delta_{d,K}\hat{\mathbf{z}})\right) = 0 \iff p^*(\Delta_{d,K}\mathbf{z}) = \hat{p}(\Delta_{d,K}\hat{\mathbf{z}}) \tag{12}$$

$$\iff p^*(\mathbf{z}) = \hat{p}(\hat{\mathbf{z}}) \quad (\because \text{Lemma 1}). \tag{13}$$

Therefore,

$$d^2_{\mathrm{cw}}(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K})) = 0 \iff p^*(\mathbf{z}) = \hat{p}(\hat{\mathbf{z}}), \tag{14}$$

and hence $k_{d,\kappa}$ is characteristic on the original trajectory space. □

A.4 Proof of Proposition 2

Proof.

$$\left\|\mathbf{z}_{d,K} - \hat{\mathbf{z}}_{d,K}\right\|_2^2 = \left\|\Delta_{d,K}(\mathbf{z} - \hat{\mathbf{z}})\right\|_2^2 \tag{15}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left\|\mathcal{F}\{\Delta_{d,K}(\mathbf{z} - \hat{\mathbf{z}})\}(\omega)\right\|_2^2 d\omega \tag{16}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left\|\sum_{k=0}^{K}(-1)^k\binom{d}{k}\mathcal{F}\{B^k(\mathbf{z} - \hat{\mathbf{z}})\}(\omega)\right\|_2^2 d\omega \tag{17}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left\|\sum_{k=0}^{K}(-1)^k\binom{d}{k}\exp(-i\omega k)\sum_{l=1}^{N-k}(z_l - \hat{z}_l)\exp(-i\omega l)\right\|_2^2 d\omega \tag{18}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \Bigg\| \underbrace{H_{d,K}(\omega)\sum_{l=1}^{N}(z_l - \hat{z}_l)\exp(-i\omega l)}_{=A_1} - \underbrace{\sum_{k=0}^{K}(-1)^k\binom{d}{k}\exp(-i\omega k)\sum_{l=N-k+1}^{N}(z_l - \hat{z}_l)\exp(-i\omega l)}_{=A_2}\Bigg\|_2^2 d\omega \tag{19}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left\|H_{d,K}(\omega)\sum_{l=1}^{N}(z_l - \hat{z}_l)\exp(-i\omega l)\right\|_2^2 d\omega - \frac{1}{2\pi}\left\|\sum_{k=0}^{K}(-1)^k\binom{d}{k}\exp(-i\omega k)\sum_{l=N-k+1}^{N}(z_l - \hat{z}_l)\exp(-i\omega l)\right\|_{L_2}^2 \tag{20}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_{d,K}(\omega)|^2 \left\|\sum_{l=1}^{N}(z_l - \hat{z}_l)\exp(-i\omega l)\right\|_2^2 d\omega + R_{N,K}, \tag{21}$$

where $\mathcal{F}\{\cdot\}$ denotes the discrete-time Fourier transform of the zero-extended finite trajectory, and $R_{N,K} \le 0$ denotes the finite-boundary remainder induced by trajectory truncation.

Note that, in equation 19, $A_1 - A_2$ and A_2 correspond to the retained trajectory and the discarded convolution tail, respectively, whose time-domain supports are disjoint. Hence, by Parseval's identity, $\frac{1}{2\pi}\int_{-\pi}^{\pi}(A_1 - A_2)\overline{A_2}d\omega = 0$. Consequently, $\frac{1}{2\pi}\int_{-\pi}^{\pi}\|A_1 - A_2\|^2 d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\|A_1\|^2 d\omega - \frac{1}{2\pi}\int_{-\pi}^{\pi}\|A_2\|^2 d\omega$.

Because $d > 0$, the fractional-differencing coefficients $a_k = (-1)^k\binom{d}{k}$ are absolutely summable $\sum_{k=0}^{\infty}|a_k| < \infty$. Hence, $\sup_{\omega\in[-\pi,\pi]}|H_{d,K}(\omega) - H_d(\omega)| \le \sum_{k=K+1}^{\infty}|a_k| \to 0$, where $H_d(\omega) = \sum_{k=0}^{\infty} a_k \exp(-i\omega k) = (1 - \exp(-i\omega))^d$. Therefore,

$$|H_{d,K}(\omega)|^2 \to |1 - \exp(-i\omega)|^{2d} = \left(2\left|\sin\frac{\omega}{2}\right|\right)^{2d} \tag{22}$$

uniformly in ω.

Next, since $\sum_{l=1}^{\infty}|z_l-\hat{z}_l|^2<\infty$ and $K\to\infty$ with $N>K$, the boundary tail satisfies $\sum_{l=N-K+1}^{\infty}|z_l-\hat{z}_l|^2\to 0$. Together with the absolute summability of the fractional-differencing coefficients,

$$|R_{N,K}| = \frac{1}{2\pi}\left\|\sum_{k=0}^{K}(-1)^k\binom{d}{k}\exp(-i\omega k)\sum_{l=N-k+1}^{N}(z_l-\hat{z}_l)\exp(-i\omega l)\right\|_{L^2}^2 \tag{23}$$

$$\leq\left[\sum_{k=0}^{K}\left|(-1)^k\binom{d}{k}\right|\left(\sum_{l=N-k+1}^{N}(z_l-\hat{z}_l)^2\right)^{1/2}\right]^2 \tag{24}$$

$$\leq\left(\sum_{k=0}^{\infty}\left|(-1)^k\binom{d}{k}\right|\right)^2\sum_{l=N-K+1}^{\infty}|z_l-\hat{z}_l|^2\to 0, \tag{25}$$

where the first inequality follows from the triangle inequality and Parseval's identity. Hence, the finite-boundary remainder vanishes, $R_{N,K}\to 0$. Moreover, square summability implies that in $L^2([-\pi,\pi])$, $\sum_{l=1}^{N}(z_l-\hat{z}_l)\exp(-i\omega l)\to\sum_{l=1}^{\infty}(z_l-\hat{z}_l)\exp(-i\omega l)$.

Since $H_{d,K}$ converges uniformly to H_d and $R_{N,K}\to 0$, it follows that

$$\lim_{K\to\infty}\|\mathbf{z}_{d,K}-\hat{\mathbf{z}}_{d,K}\|_2^2=\lim_{K\to\infty}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi}|H_{d,K}(\omega)|^2\left|\sum_{l=1}^{N}(z_l-\hat{z}_l)\exp(-i\omega l)\right|^2 d\omega+R_{N,K}\right] \tag{26}$$

$$=\frac{1}{2\pi}\int_{-\pi}^{\pi}\left(2\left|\sin\frac{\omega}{2}\right|\right)^{2d}\left|\sum_{l=1}^{\infty}(z_l-\hat{z}_l)\exp(-i\omega l)\right|^2 d\omega. \tag{27}$$

Thus, the induced trajectory metric converges to a frequency-weighted quadratic form with weight $(2|\sin(\omega/2)|)^{2d}$. □

Remark 1. *For* $0<\omega_1<\omega_2\leq\pi$, $\left(\frac{\sin(\omega_2/2)}{\sin(\omega_1/2)}\right)^{2d}>1$. *Thus, higher frequencies receive greater relative weight than lower frequencies.*

A.5 Closed-form expression of Cramér–Wold discrepancy

For a smoothing parameter $\kappa>0$, chosen according to Silverman's rule of thumb as $\kappa=(4/3n)^{2/5}$ (Silverman, 1986), the empirical Cramér–Wold discrepancy admits the following closed-form expression (Knop et al., 2020):

$$d^2_{\mathrm{cw},\kappa}(p^*(\mathbf{z}_{d,K}),\hat{p}(\hat{\mathbf{z}}_{d,K}))=\frac{1}{2n^2\sqrt{\pi\kappa}}\Bigg[\sum_{i,j=1}^{n}\varphi_N\left(\frac{\|\mathbf{z}_{d,K}^{(i)}-\mathbf{z}_{d,K}^{\prime(j)}\|_2^2}{4\kappa}\right) \tag{28}$$

$$+\sum_{i,j=1}^{n}\varphi_N\left(\frac{\|\hat{\mathbf{z}}_{d,K}^{(i)}-\hat{\mathbf{z}}_{d,K}^{\prime(j)}\|_2^2}{4\kappa}\right)-2\sum_{i,j=1}^{n}\varphi_N\left(\frac{\|\mathbf{z}_{d,K}^{(i)}-\hat{\mathbf{z}}_{d,K}^{(j)}\|_2^2}{4\kappa}\right)\Bigg].$$

For sufficiently large N, $\varphi_N(s)\approx(1+4s/(2N-3))^{-1/2}$, which allows $d^2_{\mathrm{cw},\kappa}$ to be computed efficiently using only pairwise squared Euclidean distances.

Interpretation. Based on the closed-form expression above, $d^2_{\mathrm{cw},\kappa}(p^*(\mathbf{z}_{d,K}),\hat{p}(\hat{\mathbf{z}}_{d,K}))$ can be decomposed into two terms: (i) kernelized data depth maximization (equation 30) and (ii) a diversity-preserving regularizer (equation 31).

$$d^2_{\mathrm{cw},\kappa}\left(p^*(\mathbf{z}_{d,K}),\hat{p}(\hat{\mathbf{z}}_{d,K})\right)=C_0+\frac{1}{2\sqrt{\pi\kappa}}\left[R(\hat{\mathbf{Z}})-2D(\hat{\mathbf{Z}},\mathbf{Z})\right], \tag{29}$$

where $C_0 = \frac{1}{2n^2\sqrt{\pi\kappa}} \sum_{i,j=1}^{n} \varphi_N \left(\frac{\|\mathbf{z}_{d,K}^{(i)} - \mathbf{z}_{d,K}'^{(j)}\|_2^2}{4\kappa} \right)$,

$$D(\hat{\mathbf{Z}}, \mathbf{Z}) := \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \varphi_N \left(\frac{\|\mathbf{z}_{d,K}^{(i)} - \hat{\mathbf{z}}_{d,K}^{(j)}\|_2^2}{4\kappa} \right), \tag{30}$$

$$R(\hat{\mathbf{Z}}) := \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \varphi_N \left(\frac{\|\hat{\mathbf{z}}_{d,K}^{(i)} - \hat{\mathbf{z}}_{d,K}'^{(j)}\|_2^2}{4\kappa} \right). \tag{31}$$

(i) Here, $D(\hat{\mathbf{Z}}, \mathbf{Z})$ in equation 30 quantifies the kernelized data depth of the predicted trajectories relative to the ground-truth data. Since the kernel value increases as the pairwise distance decreases, maximizing $D(\hat{\mathbf{Z}}, \mathbf{Z})$ encourages the predictions to move toward regions that are densely supported by the observed trajectory distribution.

(ii) In contrast, $R(\hat{\mathbf{Z}})$ in equation 31 measures the mutual proximity within the ensemble of predictions. Because $R(\hat{\mathbf{Z}})$ enters the objective with a positive sign, minimizing the Cramér–Wold discrepancy discourages excessive concentration of predictions and thereby acts as a diversity-preserving regularizer.

Consequently, training $\hat{\mathbf{z}}$ by minimizing $d_{\mathrm{cw},\kappa}^2 \left(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K}) \right)$ encourages the trajectories to reach a deeper kernel depth within the training data while imposing a regularizer that maintains trajectory diversity.

B REPRODUCIBILITY

B.1 ALGORITHM

The overall training pipeline is summarized in Algorithm 1.

Algorithm 1 Training Algorithm

Require: $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$, $f(\cdot;\theta)$, γ, κ
Ensure: Updated model parameters θ

for $i = 1, \ldots, n$ **do**
  $\hat{\mathbf{y}}^{(i)} \leftarrow f(\mathbf{x}^{(i)};\theta)$
  $\mathbf{z}^{(i)} \leftarrow \mathrm{concat}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, $\hat{\mathbf{z}}^{(i)} \leftarrow \mathrm{concat}(\mathbf{x}^{(i)}, \hat{\mathbf{y}}^{(i)})$
  $\mathbf{z}_{d,K}^{(i)} \leftarrow \Delta_{d,K}\mathbf{z}^{(i)}$, $\hat{\mathbf{z}}_{d,K}^{(i)} \leftarrow \Delta_{d,K}\,\hat{\mathbf{z}}^{(i)}$
end for
$\mathbf{z}_{d,K} \leftarrow \{\mathbf{z}_{d,K}^{(i)}\}_{i=1}^{n}$, $\hat{\mathbf{z}}^{(i)} \leftarrow \{\hat{\mathbf{z}}_{d,K}^{(i)}\}_{i=1}^{n}$
$\mathcal{L}_{\mathrm{cw}} \leftarrow d^2_{\mathrm{cw},\kappa}(p^*(\mathbf{z}_{d,K}), \hat{p}(\hat{\mathbf{z}}_{d,K}))$
$\mathcal{L}_{\mathrm{MSE}} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}\|_2^2$
$\mathcal{L} \leftarrow (1-\gamma)\cdot\mathcal{L}_{\mathrm{MSE}} + \gamma\cdot\mathcal{L}_{\mathrm{CW}}$
Update θ using gradient descent.

We compute the Cramér–Wold discrepancy in closed form using the smoothing parameter $\kappa = (4/3n)^{2/5}$, chosen by Silverman's rule of thumb (Silverman, 1986). This closed-form computation (Appendix A.5) is particularly well suited to deep forecasting, as it avoids explicit sampling over projection directions and enables efficient computation via pairwise distance operations.

B.2 DATASETS

We evaluate our method PRODA on a diverse collection of benchmark forecasting datasets spanning energy, weather, finance, and traffic domains.

- ETT (Electricity Transformer Temperature) (Zhou et al., 2021): The ETT dataset contains seven transformer-related variables collected from two regions between July 2016 and July 2018. Variants with suffix "h" correspond to hourly observations, while "m" denotes 15-minute intervals. The suffixes "1" and "2" indicate different geographical regions.
- Weather (Wu et al., 2021): The Weather dataset contains 21 meteorological variables recorded every 10 minutes during 2020 at the Max Planck Biogeochemistry Institute weather station.
- ECL (Electricity Consuming Load) (Li et al., 2019): The ECL dataset consists of hourly electricity consumption measurements from 321 clients between 2012 and 2014.
- Exchange (Lai et al., 2018): The Exchange dataset contains daily exchange rates from 1990 to 2016 for eight countries, including Australia, the United Kingdom, Canada, Switzerland, China, Japan, New Zealand, and Singapore.
- PEMS (LIU et al., 2022): The PEMS dataset contains traffic flow measurements from the California highway system aggregated at 5-minute intervals. We use the PEMS08 subset in our experiments.

We follow the standard preprocessing and train-validation-test split protocols used in prior forecasting studies (Zhou et al., 2021; Wu et al., 2021; Liu et al., 2024b). Specifically, the ETT datasets are split into training, validation, and test sets using a 12/4/4-month partition, while all other datasets follow a 7:1:2 ratio. Additional dataset statistics are summarized in Table 5.

B.3 BASELINE MODELS

We evaluate PRODA using four representative time-series forecasting models spanning Transformer- and MLP-based architectures. For all experiments, we utilize the official implementations.

Table 5: Summary of dataset statistics and forecasting settings.

Dataset	#Variables	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	15 min	Electricity
ETTm2	7	96, 192, 336, 720	34465 / 11521 / 11521	15 min	Electricity
Weather	21	96, 192, 336, 720	36792 / 5271 / 10540	10 min	Weather
ECL	321	96, 192, 336, 720	18317 / 2633 / 5261	Hourly	Electricity
Exchange	8	96, 192, 336, 720	5120 / 665 / 1422	1 day	Exchange Rates
PEMS08	170	12, 24, 48, 96	10690 / 3548 / 265	5min	Transportation

- iTransformer (Liu et al., 2024b)[1]: iTransformer inverts the conventional Transformer representation by treating each variable as a token, with its historical observations embedded as the token representation. Self-attention is then applied across variates to capture multivariate correlations, while feed-forward networks learn nonlinear representations of individual time series.
- PatchTST (Nie et al., 2023)[2]: PatchTST segments each univariate time series into temporal patches and treats these patches as input tokens to a Transformer encoder, reducing the effective sequence length while preserving local temporal information. It adopts a channel-independent design in which the same embedding and Transformer weights are shared across variables, enabling efficient learning from long look-back windows.
- TimeMixer (Wang et al., 2024)[3]: TimeMixer is an MLP-based forecasting architecture that models time series through decomposable multiscale mixing. Its Past-Decomposable-Mixing block separately mixes seasonal and trend components across multiple temporal scales, while Future-Multipredictor-Mixing combines predictions from different scales to exploit their complementary forecasting information.
- DLinear (Zeng et al., 2023)[4]: DLinear provides a lightweight linear forecasting baseline based on explicit series decomposition. It separates the input sequence into trend and seasonal components using a moving-average decomposition, applies an independent linear layer to each component, and sums the resulting forecasts to obtain the final prediction.

B.4 Comparison Objective Functions

We compare ProDA with recent objective-centric baselines, each of which imposes a different structural view on forecast trajectories. We use the official implementations of all objectives and follow the hyperparameter settings recommended by the original authors.

- AliO (Park et al., 2025)[5]: AliO addresses output inconsistency by aligning predictions for overlapping future timestamps generated from temporally shifted input windows. It performs alignment in both the time and frequency domains and employs regression pulling to encourage the aligned predictions to move toward the ground truth without compromising pointwise forecasting accuracy.
- PSLoss (Kudrat et al., 2025)[6]: PSLoss moves beyond pointwise errors by partitioning the forecast horizon into local patches and comparing their structural statistics. Specifically, it measures discrepancies in patch-level mean, variance, and correlation, and combines these structural terms with the conventional pointwise loss to preserve local temporal patterns.
- Time-o1 (Wang et al., 2026b)[7]: Time-o1 introduces transformed label alignment by constructing a data-dependent transformation from the training targets and comparing predictions and labels in the transformed space. Its implementation derives a PCA basis from the training label trajectories and combines the resulting transformed-space discrepancy with the conventional temporal loss.

[1] https://github.com/thuml/iTransformer
[2] https://github.com/yuqinie98/PatchTST
[3] https://github.com/kwuking/TimeMixer
[4] https://github.com/cure-lab/LTSF-Linear
[5] https://github.com/eai-lab/AliO
[6] https://github.com/Dilfiraa/PS_Loss
[7] https://github.com/Master-PLC/Time-o1

- DBLoss (Qiu et al., 2026)[8]: DBLoss explicitly incorporates trend and seasonal structure into the training objective. It decomposes predictions and targets within the forecasting horizon using exponential moving averages, computes separate losses for the resulting trend and seasonal components, and combines them through weighted aggregation.
- FreDF (Wang et al., 2025) [9]: FreDF complements temporal-domain supervision with frequency-domain alignment between predictions and targets. It applies the Fourier transform along the forecasting horizon and penalizes discrepancies between their complex-valued spectral representations, thereby encouraging preservation of frequency-domain characteristics that may be overlooked by pointwise losses.
- DistDF (Wang et al., 2026a) [10]: DistDF formulates forecasting as joint-distribution alignment between predicted and observed future trajectories using a Wasserstein-based discrepancy. It constructs pairwise costs between forecast and target trajectories and solves an optimal transport problem to regularize their empirical distributions, enabling distribution-level comparison beyond sample-wise regression errors.

B.5 Implementation Details

- Our implementation code is provided in the supplementary materials.
- All experiments are conducted on an NVIDIA RTX A6000 GPU using implementations in `PyTorch`.
- Training is conducted using the Adam optimizer (Kingma & Ba, 2015) with an initial learning rate of 0.01. We also employ the `ReduceLROnPlateau` scheduler from `PyTorch`, with a patience of 5 and a reduction factor of 0.1.
- All methods are trained for 100 epochs with a batch size of 512, halved adaptively (to a minimum of 64) if out-of-memory (OOM) issues occur.
- For all methods, the stopping criterion is the validation loss.
- We follow the comprehensive time series forecasting benchmark TFB (Qiu et al., 2024)[11] to set the historical input length H.

Table 6: Hyperparameter search space for each neural network architecture. Bold values denote the configurations used in our experiments.

Hyperparameter	iTransformer (Liu et al., 2024b)	PatchTST (Nie et al., 2023)	TimeMixer (Wang et al., 2024)	DLinear (Zeng et al., 2023)
`d_model`	{**128**, 256, 512}	{**32**, 64, 128}	{**16**, 32, 64}	-
`d_ff`	{**128**, 256, 512}	{**128**, 256, 512}	{**32**, 64, 128}	-
`e_layers`	{**2**, 3, 4}	{2, **3**, 4}	{**2**, 3, 4}	-
`moving_avg`	-	-	**25**	**25**

Table 7: Hyperparameter search space for ProDA.

Hyperparameter	ProDA
γ	$\{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$
d	$\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$
K	$\min\{L-1, \lfloor H/2 \rfloor\}$

[8] https://github.com/decisionintelligence/DBLoss
[9] https://github.com/Master-PLC/FreDF
[10] https://github.com/Master-PLC/DistDF
[11] https://github.com/decisionintelligence/TFB

C Additional Experimental Results

C.1 Comparison of other objectives

Table 8: Comparison between ProDA and other loss functions. The baseline model is iTransformer and we report the results on three datasets. The best result is **bolded**, and the second best is underlined.

Dataset		ETTh1					Weather					ECL				
Forecast length		96	192	336	720	Avg.	96	192	336	720	Avg.	96	192	336	720	Avg.
Original (MSE)	MSE	0.393	0.448	0.475	0.493	0.452	0.186	0.217	0.281	0.343	0.257	0.135	0.159	0.172	0.204	0.168
	MAE	0.416	0.449	0.459	0.492	0.454	0.237	0.260	0.306	0.347	0.287	0.234	0.253	0.269	0.297	0.262
Koopman	MSE	0.399	0.452	0.486	0.503	0.460	0.177	0.224	0.280	0.354	0.259	0.145	0.169	0.181	0.216	0.178
	MAE	0.416	0.450	0.466	0.501	0.458	0.219	0.258	0.301	0.350	0.282	0.246	0.265	0.278	0.309	0.275
AliO	MSE	0.398	0.455	0.490	0.506	0.462	0.177	0.222	0.279	0.352	0.258	0.147	0.171	0.184	0.219	0.180
	MAE	0.417	0.452	0.469	0.499	0.459	0.218	0.257	0.299	0.349	0.281	0.248	0.267	0.281	0.312	0.277
PSLoss	MSE	0.389	0.437	0.471	**0.482**	0.445	0.170	0.217	0.271	0.339	0.249	0.151	0.175	0.190	0.228	0.186
	MAE	0.409	0.440	0.456	**0.481**	0.447	0.205	0.249	**0.291**	**0.343**	0.272	0.252	0.271	0.286	0.319	0.282
Time-o1	MSE	0.394	0.450	0.483	0.496	0.456	0.180	0.223	0.278	0.347	0.257	0.143	0.166	0.179	0.214	0.176
	MAE	0.413	0.447	0.465	0.491	0.454	0.220	0.257	0.300	0.348	0.281	0.243	0.263	0.276	0.305	0.272
DBLoss	MSE	0.396	0.444	0.479	0.491	0.453	**0.167**	**0.215**	0.274	0.344	0.250	0.146	0.170	0.182	0.217	0.179
	MAE	0.411	0.443	0.461	0.485	0.450	**0.202**	**0.247**	0.292	0.344	**0.271**	0.247	0.266	0.279	0.310	0.276
FreDF	MSE	**0.386**	**0.435**	0.468	0.486	**0.444**	0.175	0.221	0.276	0.346	0.255	0.149	0.174	0.188	0.225	0.184
	MAE	**0.406**	**0.437**	**0.452**	0.483	**0.445**	0.214	0.254	0.296	0.347	0.278	0.250	0.269	0.284	0.316	0.280
DistDF	MSE	0.397	0.453	0.482	0.500	0.458	0.178	0.224	0.280	0.355	0.259	0.144	0.168	0.180	0.215	0.177
	MAE	0.415	0.450	0.466	0.497	0.457	0.218	0.258	0.301	0.351	0.282	0.245	0.264	0.277	0.307	0.273
ProDA (Ours)	MSE	0.402	0.442	**0.452**	0.494	0.447	0.169	0.216	**0.263**	**0.327**	**0.244**	**0.135**	**0.159**	**0.171**	**0.203**	**0.167**
	MAE	0.418	0.455	0.462	0.490	0.456	0.222	0.264	0.297	0.345	0.282	**0.232**	**0.257**	**0.267**	**0.293**	**0.262**

Table 9: Comparison between ProDA and other loss functions. The baseline model is PatchTST and we report the results on three datasets. The best result is **bolded**, and the second best is underlined.

Dataset		ETTh1					Weather					ECL				
Forecast length		96	192	336	720	Avg.	96	192	336	720	Avg.	96	192	336	720	Avg.
Koopman	MSE	0.381	0.431	0.472	0.489	0.443	0.158	0.207	0.261	0.329	0.239	0.154	0.164	0.181	0.225	0.181
	MAE	0.399	0.433	0.460	0.487	0.445	0.207	0.250	0.291	0.341	0.272	0.253	0.270	0.279	0.311	0.278
AliO	MSE	0.379	0.430	0.469	0.486	0.441	0.160	0.209	0.263	0.332	0.241	0.156	0.165	0.183	0.227	0.183
	MAE	0.399	0.432	0.458	0.486	0.444	0.210	0.252	0.293	0.343	0.275	0.255	0.272	0.281	0.313	0.280
PSLoss	MSE	0.390	0.445	0.465	0.472	0.443	0.166	0.214	0.270	0.340	0.248	0.162	0.170	0.188	0.233	0.188
	MAE	0.399	0.433	0.453	**0.473**	0.440	0.220	0.260	0.300	0.350	0.283	0.262	0.279	0.288	0.320	0.287
Time-o1	MSE	0.377	0.428	0.461	0.477	0.436	0.162	0.211	0.265	0.334	0.243	0.152	0.163	0.180	0.223	0.180
	MAE	0.398	0.431	0.459	0.483	0.443	0.214	0.255	0.296	0.345	0.278	0.251	0.269	0.278	0.309	0.277
DBLoss	MSE	**0.370**	0.423	0.468	0.479	0.435	0.154	0.202	0.257	0.326	0.235	0.149	0.161	0.177	0.220	0.177
	MAE	**0.393**	**0.424**	0.454	**0.473**	**0.436**	0.204	0.248	0.289	0.338	0.270	0.248	0.263	0.274	0.306	0.273
FreDF	MSE	0.392	0.438	0.475	0.493	0.450	0.164	0.212	0.268	0.338	0.246	0.160	0.168	0.186	0.231	0.186
	MAE	0.405	0.441	0.465	0.491	0.451	0.218	0.258	0.298	0.348	0.281	0.259	0.276	0.286	0.318	0.285
DistDF	MSE	0.378	0.428	0.460	0.480	0.437	0.159	0.208	0.262	0.331	0.240	0.153	0.164	0.181	0.224	0.181
	MAE	0.399	0.431	0.456	0.476	0.441	0.211	0.253	0.294	0.344	0.276	0.252	0.270	0.279	0.310	0.278
ProDA (Ours)	MSE	0.386	**0.417**	**0.436**	**0.456**	**0.424**	**0.150**	**0.197**	**0.248**	**0.317**	**0.228**	**0.142**	**0.157**	**0.173**	**0.212**	**0.171**
	MAE	0.407	0.429	**0.450**	0.476	0.441	**0.200**	**0.246**	**0.285**	**0.335**	**0.267**	**0.243**	**0.254**	**0.270**	**0.299**	**0.267**

Table 10: Comparison between PRODA and other loss functions. The baseline model is TimeMixer and we report the results on three datasets. The best result is **bolded**, and the second best is underlined.

Dataset		ETTh1					Weather					ECL				
Forecast length		96	192	336	720	Avg.	96	192	336	720	Avg.	96	192	336	720	Avg.
Koopman	MSE	0.416	0.462	0.505	0.548	0.483	0.158	0.217	0.279	0.337	0.248	0.154	0.175	0.194	0.224	0.187
	MAE	0.421	0.446	0.476	0.515	0.465	0.212	0.262	0.302	0.346	0.281	0.255	0.277	0.296	0.314	0.286
AliO	MSE	0.413	0.463	0.500	0.541	0.479	0.160	0.218	0.276	0.340	0.249	0.151	0.177	0.185	0.222	0.184
	MAE	0.420	0.447	0.474	0.511	0.463	0.211	0.261	0.300	0.347	0.280	0.252	0.279	0.289	0.313	0.283
PSLoss	MSE	0.407	0.472	0.488	0.522	0.472	0.166	0.220	0.277	0.342	0.251	0.156	0.169	0.187	0.213	0.181
	MAE	0.412	0.455	0.468	0.503	0.460	0.218	0.265	0.298	0.349	0.283	0.258	0.270	0.290	0.304	0.281
Time-o1	MSE	0.410	0.454	0.493	0.536	0.473	0.162	0.219	0.278	0.344	0.251	0.150	0.173	0.189	0.232	0.186
	MAE	0.418	0.444	0.466	0.510	0.460	0.213	0.261	0.303	0.348	0.281	**0.248**	0.274	0.291	0.323	0.284
DBLoss	MSE	0.412	0.476	0.496	0.515	0.475	0.157	0.216	0.274	0.336	0.246	0.155	0.167	0.193	0.221	0.184
	MAE	0.419	0.452	0.472	0.500	0.461	0.206	**0.254**	**0.295**	0.342	0.275	0.256	0.268	0.294	0.312	0.283
FreDF	MSE	**0.388**	**0.438**	0.468	0.530	0.456	0.165	0.221	0.286	0.348	0.255	0.151	**0.161**	0.198	**0.211**	0.180
	MAE	**0.402**	**0.428**	**0.446**	0.560	0.459	0.216	0.264	0.307	0.353	0.285	0.251	**0.259**	0.299	**0.303**	0.278
DistDF	MSE	0.413	0.462	0.498	0.544	0.479	0.161	0.221	0.281	0.341	0.251	0.152	0.176	0.184	0.220	0.183
	MAE	0.419	0.448	0.473	0.513	0.463	0.214	0.266	0.303	0.350	0.283	0.253	0.276	0.287	0.311	0.282
PRODA (Ours)	MSE	0.408	0.459	**0.463**	**0.474**	**0.451**	**0.149**	**0.208**	**0.272**	**0.318**	**0.237**	**0.149**	0.163	**0.180**	0.219	**0.178**
	MAE	0.418	0.462	0.470	**0.484**	**0.458**	**0.204**	0.259	0.301	**0.333**	**0.274**	0.250	0.262	**0.279**	0.310	**0.275**

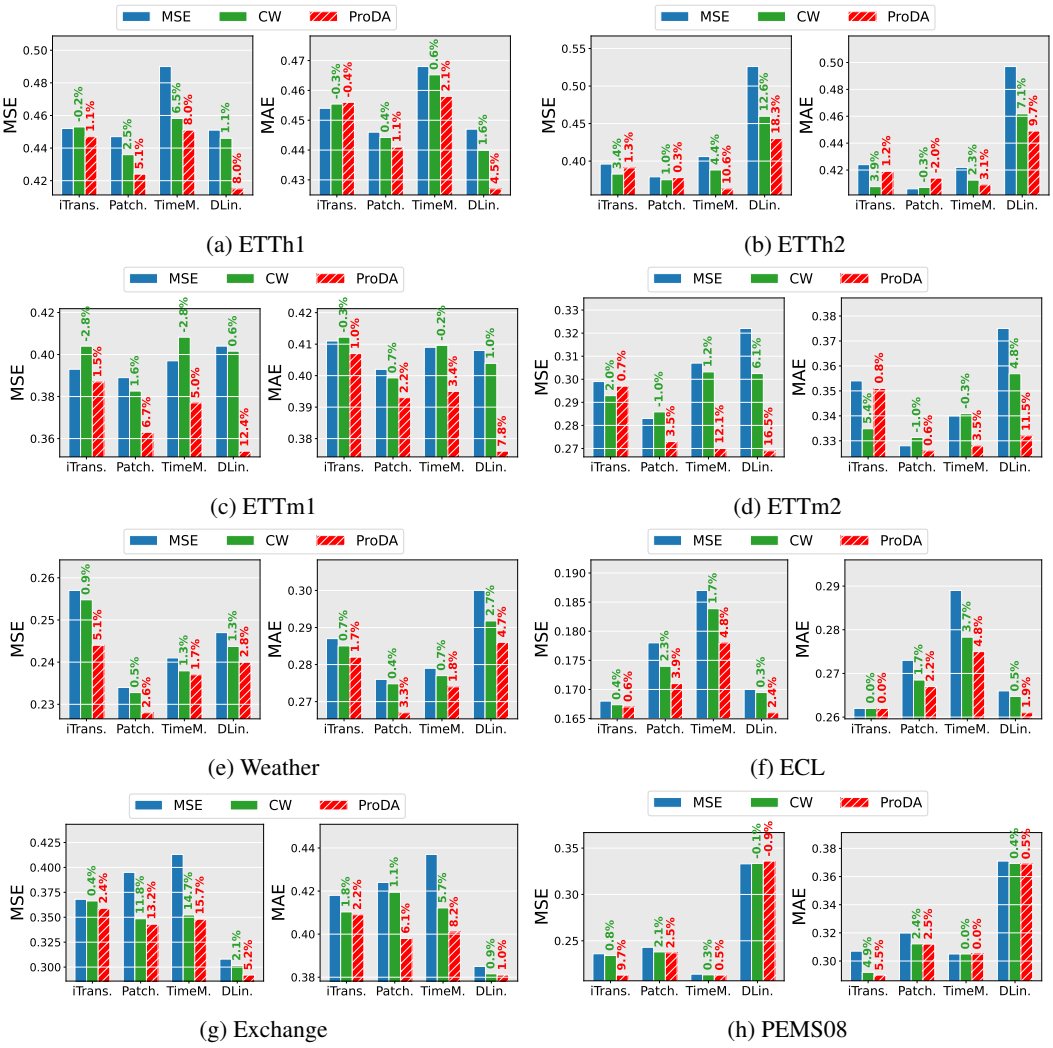
Table 11: Comparison between PRODA and other loss functions. The baseline model is DLinear and we report the result on three datasets. The best result is **bolded**, and the second best is underlined.

Dataset		ETTh1					Weather					ECL				
Forecast length		96	192	336	720	Avg.	96	192	336	720	Avg.	96	192	336	720	Avg.
Koopman	MSE	0.389	0.434	0.477	0.499	0.450	0.179	0.218	0.266	0.331	0.249	0.146	0.155	0.177	0.207	0.171
	MAE	0.403	0.429	0.454	0.497	0.446	0.240	0.276	0.314	0.362	0.298	0.239	0.252	0.267	0.311	0.267
AliO	MSE	0.388	0.439	0.482	0.507	0.454	0.177	0.218	0.263	0.329	0.247	0.145	0.154	0.176	0.205	0.170
	MAE	0.403	0.435	0.460	0.504	0.450	0.238	0.280	0.314	0.369	0.300	0.238	0.250	0.267	0.310	0.266
PSLoss	MSE	0.374	0.421	0.451	**0.451**	0.424	0.180	0.217	0.260	**0.321**	0.244	0.146	0.155	0.177	0.205	0.171
	MAE	0.392	0.423	0.449	0.488	0.438	**0.226**	0.265	**0.299**	**0.348**	**0.284**	0.237	0.249	0.265	0.305	0.262
Time-o1	MSE	**0.368**	0.420	0.444	0.458	0.423	0.176	0.215	0.261	0.325	0.244	0.145	0.154	0.176	0.204	0.170
	MAE	0.393	0.428	0.455	0.500	0.444	0.232	0.271	0.310	0.361	0.293	0.237	0.249	0.265	0.309	0.264
DBLoss	MSE	0.381	0.425	0.446	0.452	0.426	0.181	0.216	0.259	0.322	0.245	0.143	**0.153**	0.177	0.204	**0.166**
	MAE	0.392	0.422	0.446	0.485	0.436	0.236	0.265	0.303	0.350	0.288	0.237	**0.248**	0.265	0.304	0.262
FreDF	MSE	0.369	0.428	0.470	0.485	0.438	0.176	0.215	0.261	0.322	0.243	0.146	0.154	0.176	0.205	0.170
	MAE	**0.390**	0.420	0.446	0.486	0.436	0.231	0.272	0.314	0.366	0.296	0.238	**0.248**	0.265	0.305	0.262
DistDF	MSE	0.390	0.435	0.478	0.504	0.452	0.178	0.217	0.265	0.328	0.247	0.145	0.154	0.176	0.205	0.170
	MAE	0.405	0.430	0.455	0.501	0.448	0.240	0.278	0.316	0.365	0.300	0.237	0.251	0.266	0.310	0.266
PRODA (Ours)	MSE	0.370	**0.404**	**0.432**	0.454	**0.415**	**0.170**	**0.214**	**0.255**	**0.321**	**0.240**	**0.140**	**0.153**	**0.169**	**0.204**	**0.166**
	MAE	0.392	**0.412**	**0.433**	**0.471**	**0.427**	**0.227**	0.264	**0.299**	0.353	0.286	**0.236**	**0.248**	**0.264**	**0.295**	**0.261**

Results. Tables 8–11 provide detailed horizon-wise comparisons with recent forecasting objectives. Across the 12 backbone–dataset settings, PRODA achieves the best or tied-best average MSE in 11 settings and average MAE in 8 settings. At the individual forecasting horizons, it obtains the best or tied-best MSE in 36 of 48 settings, including 28 of 32 settings on Weather and ECL. For PatchTST, TimeMixer, and DLinear, PRODA achieves the best or tied-best average MSE on all three datasets, whereas ETTh1 with iTransformer shows more competitive results against FreDF and PSLoss. These results demonstrate that the improvements of PRODA remain consistent across different forecasting architectures and prediction horizons.

C.2 Ablation Studies

In this subsection, we provide the detailed ablation study results for the eight datasets we used.



(a) ETTh1 (b) ETTh2

(c) ETTm1 (d) ETTm2

(e) Weather (f) ECL

(g) Exchange (h) PEMS08

Figure 8: Ablation study of the components of ProDA across the eight datasets.

Results. Across the eight datasets, the full ProDA objective improves average MSE over plain MSE training in 31 of 32 dataset–backbone cases and improves or matches average MAE in 30 of 32 cases, demonstrating the benefit of trajectory-level distributional regularization beyond pointwise objective. Compared with Cramér–Wold regularization alone, the full objective yields larger MSE gains in 28 of 32 cases and larger or equal MAE gains in 28 of 32 cases. This indicates that imposing the distributional regularizer under the fractional-differencing-induced trajectory metric provides additional benefit beyond Cramér–Wold alignment alone. The largest improvements are observed with DLinear, reaching 18.3% in MSE on ETTh2 and 11.5% in MAE on ETTm2. Overall, the regularization effect is particularly pronounced for simpler forecasting backbones and on datasets exhibiting stronger temporal variation, including ETTh2, ETTm2, and Exchange.

C.3 ZERO-SHOT FORECASTING PERFORMANCE

In this subsection, we provide the detailed zero-shot forecasting performance in the ETT datasets.

Table 12: Zero-shot forecasting results on ETT datasets.

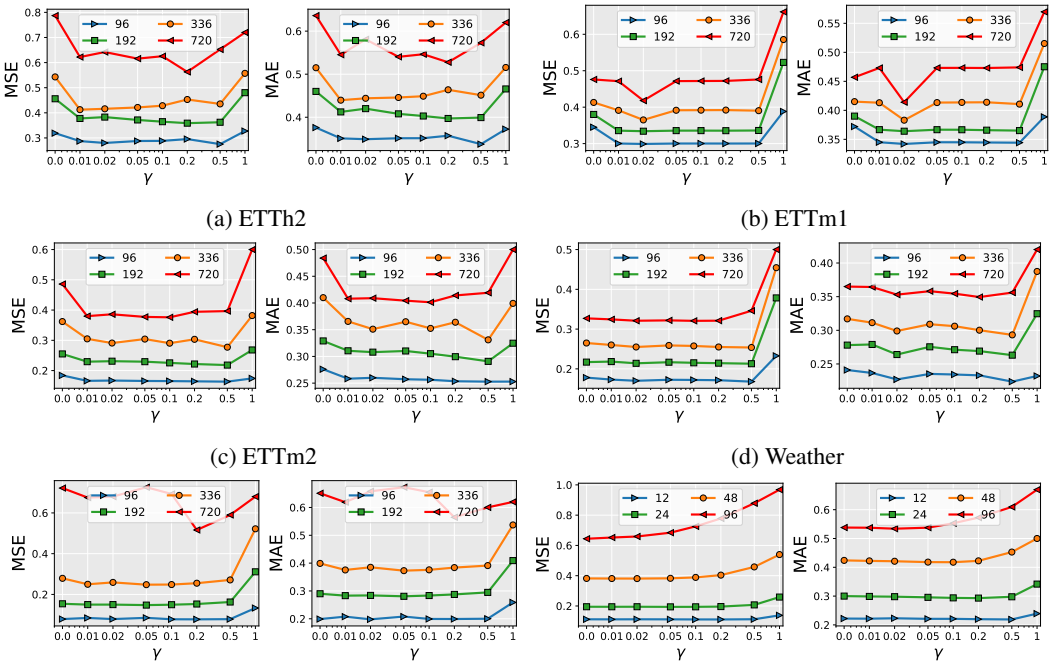
model		iTransformer				PatchTST				TimeMixer				DLinear			
loss		Original		+ PRODA		Original		+ PRODA		Original		+ PRODA		Original		+ PRODA	
metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Forecasting horizon = 96																	
ETTh1	ETTh2	0.297	0.344	0.297	0.344	0.301	0.345	0.301	0.345	0.303	0.353	0.287	0.345	0.321	0.378	0.306	0.370
	ETTm1	0.931	0.601	0.931	0.601	0.885	0.591	0.886	0.592	0.773	0.558	0.739	0.548	0.732	0.552	0.718	0.548
	ETTm2	0.229	0.313	0.229	0.313	0.217	0.303	0.217	0.303	0.216	0.302	0.206	0.298	0.247	0.351	0.240	0.345
ETTh2	ETTh1	0.590	0.528	0.584	0.521	0.501	0.466	0.570	0.518	0.821	0.589	0.619	0.531	0.422	0.427	0.417	0.427
	ETTm1	1.139	0.669	1.055	0.640	0.928	0.591	0.736	0.534	0.987	0.604	0.788	0.556	0.775	0.561	0.760	0.556
	ETTm2	0.241	0.326	0.241	0.323	0.225	0.309	0.203	0.295	0.225	0.308	0.204	0.297	0.249	0.351	0.224	0.327
ETTm1	ETTh1	0.778	0.580	0.668	0.553	0.617	0.515	0.542	0.486	0.903	0.667	0.599	0.538	0.582	0.500	0.471	0.449
	ETTh2	0.364	0.398	0.333	0.384	0.358	0.386	0.349	0.386	0.372	0.403	0.325	0.385	0.356	0.405	0.311	0.369
	ETTm2	0.205	0.281	0.184	0.270	0.198	0.270	0.182	0.265	0.201	0.278	0.180	0.265	0.215	0.306	0.185	0.277
ETTm2	ETTh1	0.670	0.547	0.836	0.615	0.556	0.506	0.587	0.520	0.859	0.617	0.744	0.603	0.564	0.492	0.434	0.441
	ETTh2	0.360	0.388	0.364	0.409	0.332	0.367	0.295	0.357	0.354	0.387	0.351	0.392	0.312	0.366	0.294	0.356
	ETTm1	0.599	0.486	0.467	0.447	0.535	0.451	0.424	0.420	0.519	0.463	0.425	0.420	0.489	0.442	0.360	0.389
Forecasting horizon = 192																	
ETTh1	ETTh2	0.376	0.392	0.358	0.395	0.381	0.394	0.347	0.389	0.384	0.397	0.346	0.384	0.423	0.441	0.379	0.414
	ETTm1	0.874	0.592	0.930	0.619	0.855	0.596	0.756	0.578	0.791	0.557	0.745	0.560	0.759	0.573	0.716	0.553
	ETTm2	0.289	0.346	0.281	0.352	0.277	0.337	0.275	0.345	0.277	0.336	0.262	0.332	0.333	0.408	0.299	0.380
ETTh2	ETTh1	0.648	0.555	0.623	0.556	0.609	0.537	0.528	0.491	1.404	0.764	0.741	0.613	0.473	0.463	0.437	0.441
	ETTm1	1.012	0.636	1.005	0.645	0.965	0.613	0.742	0.544	2.836	0.953	1.044	0.640	0.804	0.590	0.717	0.553
	ETTm2	0.299	0.355	0.287	0.356	0.293	0.349	0.260	0.333	0.362	0.381	0.273	0.342	0.371	0.433	0.309	0.388
ETTm1	ETTh1	0.738	0.570	0.606	0.525	0.695	0.571	0.528	0.478	0.877	0.632	0.630	0.538	0.578	0.504	0.481	0.460
	ETTh2	0.454	0.442	0.395	0.417	0.457	0.440	0.402	0.419	0.545	0.487	0.375	0.411	0.449	0.456	0.388	0.419
	ETTm2	0.261	0.315	0.240	0.308	0.258	0.312	0.235	0.300	0.282	0.334	0.229	0.299	0.286	0.357	0.254	0.331
ETTm2	ETTh1	0.855	0.619	1.011	0.684	0.641	0.553	0.542	0.495	0.785	0.593	0.739	0.577	0.592	0.511	0.483	0.469
	ETTh2	0.435	0.437	0.422	0.441	0.414	0.418	0.389	0.415	0.416	0.423	0.381	0.415	0.410	0.425	0.360	0.395
	ETTm1	0.650	0.517	0.560	0.505	0.579	0.488	0.499	0.467	0.551	0.474	0.461	0.446	0.491	0.449	0.397	0.407
Forecasting horizon = 336																	
ETTh1	ETTh2	0.422	0.425	0.383	0.418	0.424	0.431	0.384	0.421	0.426	0.433	0.382	0.415	0.493	0.486	0.451	0.461
	ETTm1	0.922	0.614	0.935	0.630	0.855	0.608	0.796	0.607	0.799	0.587	0.761	0.581	0.756	0.580	0.733	0.571
	ETTm2	0.351	0.382	0.336	0.385	0.337	0.372	0.331	0.377	0.332	0.368	0.319	0.366	0.409	0.454	0.386	0.434
ETTh2	ETTh1	0.674	0.567	0.761	0.624	0.722	0.595	0.875	0.665	0.763	0.593	0.715	0.576	0.502	0.480	0.454	0.457
	ETTm1	0.967	0.630	1.076	0.673	0.950	0.621	0.748	0.554	1.542	0.766	0.757	0.572	0.781	0.592	0.704	0.559
	ETTm2	0.355	0.385	0.348	0.388	0.359	0.388	0.319	0.368	0.380	0.396	0.321	0.368	0.464	0.488	0.396	0.442
ETTm1	ETTh1	0.960	0.664	0.644	0.553	0.669	0.565	0.498	0.468	0.957	0.711	0.646	0.576	0.598	0.522	0.446	0.454
	ETTh2	0.573	0.509	0.455	0.456	0.485	0.467	0.383	0.415	0.562	0.519	0.398	0.431	0.510	0.498	0.404	0.434
	ETTm2	0.342	0.367	0.302	0.345	0.314	0.347	0.287	0.335	0.330	0.359	0.280	0.331	0.368	0.414	0.297	0.359
ETTm2	ETTh1	1.131	0.721	1.080	0.706	0.664	0.568	0.588	0.547	0.907	0.651	1.078	0.742	0.612	0.531	0.512	0.487
	ETTh2	0.530	0.496	0.468	0.468	0.446	0.450	0.404	0.428	0.441	0.449	0.439	0.457	0.497	0.488	0.405	0.432
	ETTm1	0.768	0.572	0.603	0.524	0.621	0.531	0.468	0.437	0.708	0.547	0.531	0.480	0.512	0.470	0.416	0.420
Forecasting horizon = 720																	
ETTh1	ETTh2	0.426	0.441	0.426	0.442	0.427	0.442	0.426	0.452	0.434	0.448	0.421	0.445	0.618	0.563	0.611	0.557
	ETTm1	0.830	0.596	0.832	0.597	0.897	0.618	0.802	0.616	0.822	0.611	0.930	0.682	0.775	0.607	0.758	0.603
	ETTm2	0.438	0.425	0.439	0.426	0.444	0.431	0.415	0.424	0.430	0.420	0.449	0.446	0.604	0.553	0.605	0.548
ETTh2	ETTh1	0.722	0.606	0.712	0.601	0.674	0.581	0.687	0.603	1.498	0.809	0.826	0.662	0.541	0.532	0.490	0.508
	ETTm1	0.913	0.627	0.910	0.627	0.899	0.624	0.752	0.570	3.472	1.117	1.005	0.635	0.808	0.626	0.742	0.599
	ETTm2	0.449	0.433	0.449	0.433	0.446	0.431	0.399	0.415	0.577	0.512	0.419	0.424	0.773	0.630	0.708	0.593
ETTm1	ETTh1	1.152	0.752	0.668	0.575	0.731	0.615	0.524	0.500	1.413	0.864	0.703	0.582	0.590	0.547	0.482	0.499
	ETTh2	0.642	0.552	0.478	0.482	0.496	0.486	0.448	0.462	0.574	0.527	0.423	0.450	0.585	0.550	0.533	0.529
	ETTm2	0.455	0.429	0.387	0.397	0.417	0.406	0.378	0.391	0.431	0.411	0.370	0.388	0.521	0.502	0.492	0.486
ETTm2	ETTh1	0.989	0.687	1.071	0.717	0.800	0.644	0.944	0.709	4.074	1.151	1.053	0.750	0.597	0.551	0.478	0.493
	ETTh2	0.496	0.488	0.499	0.493	0.452	0.463	0.426	0.455	0.506	0.496	0.401	0.452	0.541	0.527	0.416	0.454
	ETTm1	0.809	0.598	0.679	0.570	0.769	0.599	0.559	0.491	1.302	0.730	0.551	0.498	0.548	0.495	0.446	0.438

Results. The detailed horizon-wise results reinforce the average trends reported in the main paper: across all 192 source–target–horizon–baseline model comparisons, PRODA improves MSE in 163 cases and MAE in 148 cases. The gains are especially consistent for DLinear and TimeMixer; DLinear improves MSE in 47 of 48 detailed comparisons and improves or matches MAE in all 48, while

TimeMixer improves MSE in 45 of 48 and MAE in 42 of 48. The improvement rate is highest at horizon 192 for MSE, where PRODA improves 46 of 48 comparisons, and remains strong even at horizon 720, where it still improves 37 of 48 MSE comparisons and 36 of 48 MAE comparisons. These detailed results show that the transfer benefit of PRODA is not confined to a small subset of forecast lengths or source–target pairs.

C.4 HYPERPARAMETER SENSITIVITY ANALYSES

In this subsection, we provide the hyperparameter sensitivity across six datasets, including ETTh2, ETTm1, ETTm2, Weather, Exchange, and PEMS08.

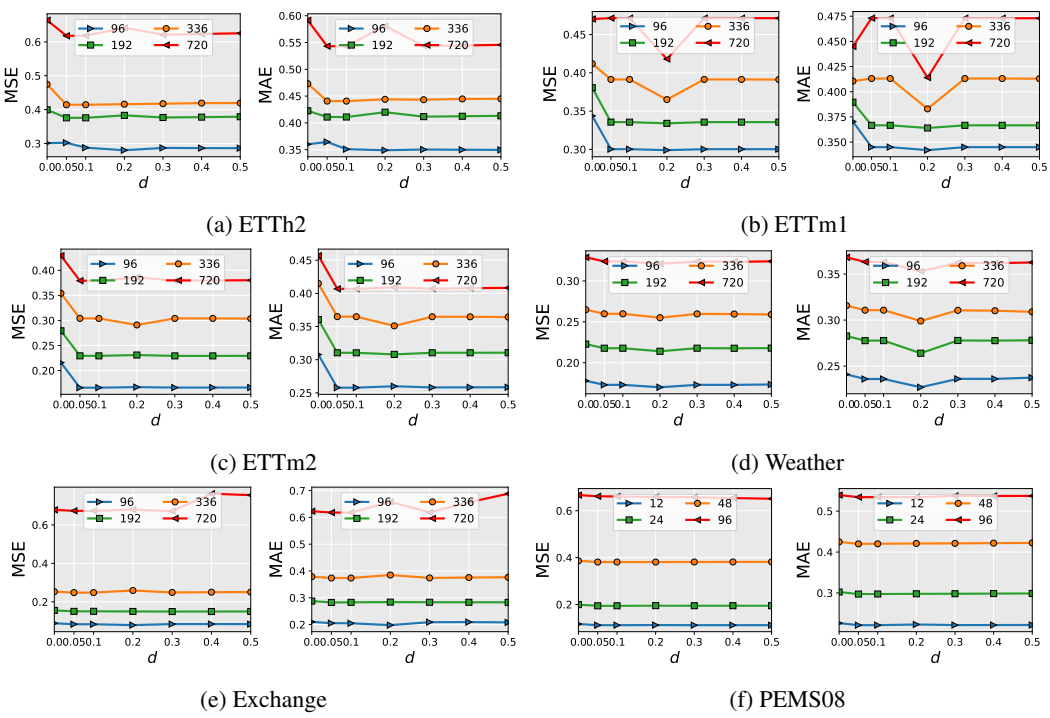


(a) ETTh2 (b) ETTm1

(c) ETTm2 (d) Weather

(e) Exchange (f) PEMS08

Figure 9: Sensitivity analyses of the hyperparameter γ of PRODA in the DLinear model.



(a) ETTh2 (b) ETTm1

(c) ETTm2 (d) Weather

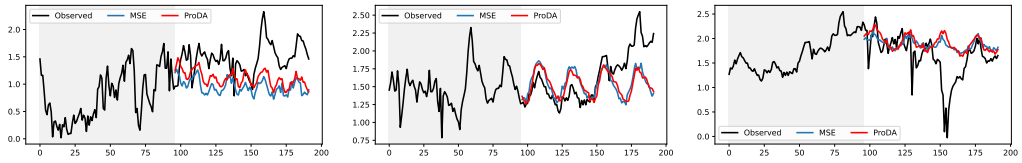
(e) Exchange (f) PEMS08

Figure 10: Sensitivity analyses of the hyperparameter differencing order d of PRODA in the DLinear model.

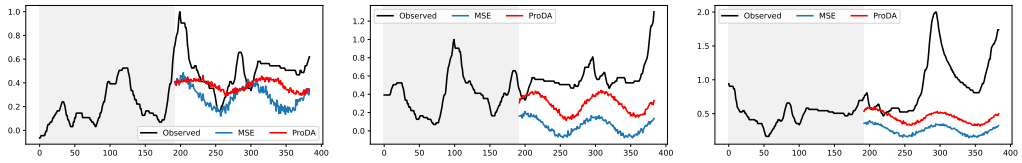
Results. The supplementary sensitivity plots further support the robustness of PRODA. For the balancing parameter γ in equation 2, most datasets attain their best or near-best performance in a moderate range roughly between 0.01 and 0.2, whereas the extreme choice $\gamma = 1$ often causes a visible degradation, especially at the longest horizons on ETTh2, ETTm1, ETTm2, Weather, and PEMS08. For the differencing order d, the curves are generally flatter: values in the range $[0.05, 0.3]$ are consistently competitive across all datasets, with ETTm1 showing its clearest optimum around $d = 0.2$ and Exchange remaining almost unchanged across a wider range. Overall, these supplementary panels indicate that the method does not depend on highly fragile hyperparameter tuning, although moderate values are consistently safer than the extremes.

C.5 FORECASTING VISUALIZATION

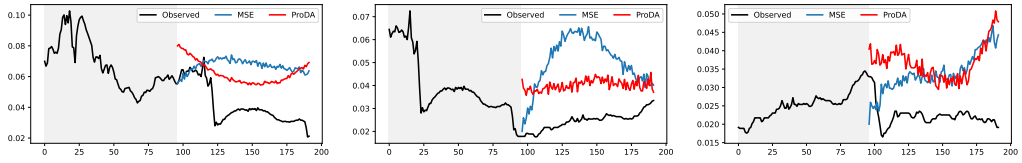
In this section, we visualize forecasting samples from the ETTh1, ETTm2, Weather, ECL, and PEMS08 datasets using iTransformer as the baseline model. Compared with standard pointwise training, predictions generated with PRODA show better alignment with the ground-truth trajectories, capturing temporal directionality, fluctuation dynamics, and overall trends.



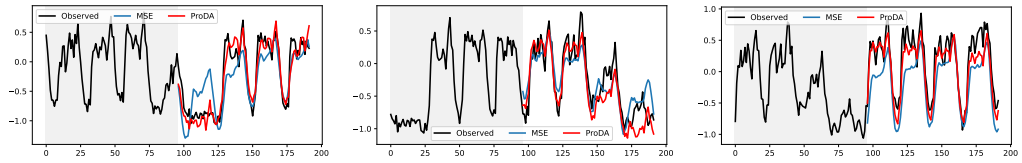
(a) ETTh1



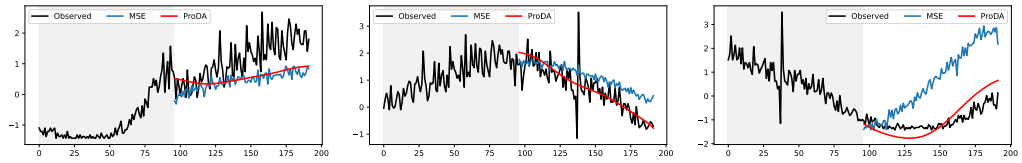
(b) ETTm2



(c) Weather



(d) ECL



(e) PEMS08

Figure 11: Forecasting visualization of PRODA in the iTransformer model.

Results. As shown in Figure 11, the qualitative differences are most pronounced on ECL and PEMS08. On ECL, PRODA more closely follows the repeated peak–trough variations of the observed trajectories, preserving oscillatory patterns that are often attenuated or shifted under MSE training. On PEMS08, PRODA better captures the overall direction and curvature of the future trajectory, whereas the MSE baseline can exhibit substantial drift, particularly over longer forecasting intervals. For ETTh1, ETTm2, and Weather, the differences are more sample-dependent, although PRODA often provides closer alignment with local variations or trajectory levels. Overall, these examples are consistent with the frequency-selective interpretation of PRODA, showing improved preservation of localized variations while maintaining broader trajectory structure.

C.6 Computational Cost

Table 13: Training time (s/epoch) comparison. Δ denotes the training time increment of PRODA over MSE.

dataset	ETTh1			ETTh2			ETTm1			ETTm2			Weather			Exchange			ECL			PEMS08		
metric	MSE	PRODA	Δ	MSE	PRODA	Δ	MSE	PRODA	Δ	MSE	PRODA	Δ	MSE	PRODA	Δ	MSE	PRODA	Δ	MSE	PRODA	Δ	MSE	PRODA	Δ
iTransformer																								
96 (12)	1.48	1.64	0.16	1.49	1.62	0.13	3.82	3.94	0.13	3.83	4.12	0.29	18.45	21.86	3.41	1.36	1.45	0.08	17.72	26.01	8.30	14.72	16.31	1.59
192 (24)	1.52	1.82	0.30	1.52	1.80	0.28	4.06	4.25	0.19	4.09	4.14	0.04	18.85	23.11	4.26	1.36	1.47	0.11	21.11	33.85	12.74	14.97	15.91	0.94
336 (48)	1.57	1.81	0.24	1.58	1.76	0.18	4.41	4.49	0.08	4.37	4.59	0.21	19.26	22.56	3.30	1.39	1.60	0.21	26.34	44.64	18.29	15.33	16.15	0.82
720 (96)	1.67	1.74	0.07	1.67	1.76	0.09	5.40	5.78	0.38	5.29	5.35	0.06	20.98	25.36	4.38	1.40	1.54	0.14	39.98	52.38	12.40	16.16	18.31	2.16
Avg.	1.56	1.75	0.19	1.56	1.73	0.17	4.42	4.62	0.20	4.39	4.55	0.15	19.39	23.22	3.84	1.38	1.51	0.14	26.29	39.22	12.93	15.30	16.67	1.38
DLinear																								
96 (12)	1.34	1.58	0.23	1.34	1.64	0.30	3.04	5.56	2.51	3.02	3.46	0.43	17.73	21.95	4.22	1.25	1.55	0.30	9.67	18.89	9.22	12.71	15.34	2.63
192 (24)	1.37	1.60	0.23	1.37	1.63	0.25	2.94	4.02	1.07	3.29	3.69	0.40	18.15	21.30	3.16	1.27	1.36	0.09	12.87	25.01	12.14	12.91	15.85	2.94
336 (48)	1.42	1.62	0.20	1.43	1.63	0.21	3.73	3.92	0.19	3.60	4.01	0.41	18.64	22.50	3.86	1.29	1.39	0.10	18.38	36.54	18.16	13.37	15.89	2.52
720 (96)	1.53	1.70	0.17	1.52	1.71	0.19	4.67	4.84	0.16	4.68	4.70	0.02	20.32	23.31	2.98	1.33	1.44	0.12	30.36	43.40	13.04	14.10	16.20	2.09
Avg.	1.42	1.63	0.21	1.42	1.65	0.24	3.60	4.58	0.98	3.65	3.96	0.32	18.71	22.27	3.56	1.29	1.44	0.15	17.82	30.96	13.14	13.27	15.82	2.54

Results. Table 13 reports the training time per epoch across forecasting horizons. Overall, PRODA introduces modest overhead on most datasets for both iTransformer and DLinear. The average increase is below 0.25 seconds on ETTh1, ETTh2, and Exchange, and remains relatively small on ETTm1 and ETTm2. Larger overheads are observed on Weather and ECL, especially for ECL, where the average increase reaches 12.93 and 13.14 seconds for iTransformer and DLinear, respectively. Nevertheless, the results show that PRODA remains computationally practical while adding manageable training cost.

C.7 Repeated Experiments

Table 14: Long-term forecasting performance over the repeated experiments on ECL and PEMS08. The means and their standard errors are reported across 5 random seeds.

dataset	model f	Original		+PRODA	
		MSE	MAE	MSE	MAE
ECL	iTransformer	$0.168_{\pm 0.002}$	$0.262_{\pm 0.002}$	$\mathbf{0.167}_{\pm 0.001}$	$\mathbf{0.262}_{\pm 0.001}$
	PatchTST	$0.178_{\pm 0.003}$	$0.273_{\pm 0.003}$	$\mathbf{0.171}_{\pm 0.002}$	$\mathbf{0.267}_{\pm 0.002}$
	TimeMixer	$0.187_{\pm 0.003}$	$0.289_{\pm 0.004}$	$\mathbf{0.178}_{\pm 0.002}$	$\mathbf{0.275}_{\pm 0.003}$
	DLinear	$0.170_{\pm 0.002}$	$0.266_{\pm 0.002}$	$\mathbf{0.166}_{\pm 0.001}$	$\mathbf{0.261}_{\pm 0.002}$
PEMS08	iTransformer	$0.236_{\pm 0.004}$	$0.307_{\pm 0.003}$	$\mathbf{0.213}_{\pm 0.003}$	$\mathbf{0.290}_{\pm 0.003}$
	PatchTST	$0.243_{\pm 0.004}$	$0.320_{\pm 0.004}$	$\mathbf{0.237}_{\pm 0.003}$	$\mathbf{0.312}_{\pm 0.003}$
	TimeMixer	$0.214_{\pm 0.003}$	$0.305_{\pm 0.003}$	$\mathbf{0.213}_{\pm 0.002}$	$\mathbf{0.305}_{\pm 0.002}$
	DLinear	$\mathbf{0.333}_{\pm 0.005}$	$0.371_{\pm 0.004}$	$0.336_{\pm 0.004}$	$\mathbf{0.369}_{\pm 0.003}$

Results. Table 14 summarizes repeated runs on ECL and PEMS08. PRODA consistently improves or matches the original objective across all backbones on ECL and shows a similar trend on PEMS08, except for a slight MSE degradation with DLinear. Overall, the results are consistent with the main experiments, suggesting that the observed improvements are not specific to a single run.