Beyond MSE: Ordinal Cross-Entropy for Probabilistic Time Series Forecasting

Jieting Wang[1,2]*, **Huimei Shi**[1], **Feijiang Li**[1,2], **Xiaolei Shang**[1]

[1]Institute of Big Data Science and Industry, Shanxi University
[2]Key Laboratory of Evolutionary Science Intelligence of Shanxi Province, Taiyuan, China
jtwang@sxu.edu.cn

Abstract

Time series forecasting is an important task that involves analyzing temporal dependencies and underlying patterns (such as trends, cyclicality, and seasonality) in historical data to predict future values or trends. Current deep learning-based forecasting models primarily employ Mean Squared Error (MSE) loss functions for regression modeling. Despite enabling direct value prediction, this method offers no uncertainty estimation and exhibits poor outlier robustness. To address these limitations, we propose OCE-TS, a novel ordinal classification approach for time series forecasting that replaces MSE with Ordinal Cross-Entropy (OCE) loss, preserving prediction order while quantifying uncertainty through probability output. Specifically, OCE-TS begins by discretizing observed values into ordered intervals and deriving their probabilities via a parametric distribution as supervision signals. Using a simple linear model, we then predict probability distributions for each timestep. The OCE loss is computed between the cumulative distributions of predicted and ground-truth probabilities, explicitly preserving ordinal relationships among forecasted values. Through theoretical analysis using influence functions, we establish that cross-entropy (CE) loss exhibits superior stability and outlier robustness compared to MSE loss. Empirically, we compared OCE-TS with five baseline models—Autoformer, DLinear, iTransformer, TimeXer, and TimeBridge—on seven public time series datasets. Using MSE and Mean Absolute Error (MAE) as evaluation metrics, the results demonstrate that OCE-TS consistently outperforms benchmark models.

Code — https://github.com/Shi-hm/OCE-TS

Introduction

Long-term time series forecasting plays a vital role in practical applications such as industrial monitoring, traffic management, and financial risk control. Traditional forecasting methods including Autoregressive Integrated Moving Average (ARIMA) (Box et al. 2015), Error-Trend-Seasonal (ETS) models (Hyndman et al. 2008) typically require manual feature engineering and make certain assumptions about data patterns.

Recent advances in deep learning have significantly improved time series forecasting performance. Modern architectures can automatically learn complex temporal features and long-term dependencies, achieving state-of-the-art performance in various forecasting tasks. Particularly for long-term forecasting tasks, current research demonstrates that Transformer-based models (Zhou et al. 2021; Wu et al. 2021), modern Multilayer Perceptron (MLP) architectures (Zeng et al. 2023), and enhanced Long Short-Term Memory (LSTM) networks can all effectively extract multi-scale features from time series through end-to-end training. These developments have been further validated in comprehensive benchmarks , establishing neural methods as the new paradigm in time series forecasting.

Nevertheless, the predominant use of MSE loss in neural network-based forecasting introduces fundamental constraints: the regression framework provides no inherent uncertainty quantification, and the quadratic loss term amplifies the influence of anomalous observations.

Researchers have developed specialized approaches that each target specific aspects of the problem in regression. For uncertainty quantification, probabilistic methods like Gaussian Processes offer principled Bayesian uncertainty estimates but struggle with computational scalability, while quantile regression provides prediction intervals but may violate natural ordering constraints. Deep generative models can capture complex distributions but require substantial training data and careful tuning. On the robustness front, both Huber loss and correntropy-based methods (Liu, Pokharel, and Principe 2007) can effectively mitigate outlier sensitivity while lacking probabilistic uncertainty quantification. The adaptive weighting scheme (Kendall, Gal, and Cipolla 2018), despite enabling dynamic sample weighting, requires joint optimization of multiple interacting hyperparameters that critically determine its performance. However, no existing method simultaneously addresses uncertainty quantification and robustness.

To this end, we propose OCE-TS as an integrated solution combining probabilistic uncertainty quantification with robust outlier handling while preserving ordinal relationships and maintaining computational efficiency. Since time series values exhibit meaningful ordinal structure, the proposed OCE-TS method reformulates the regression task as an ordinal classification problem. Ordinal classification (Gutiérrez et al. 2016) is a specialized machine learning task designed to handle categorical labels with inherent ordinal relationships. Its core objective is to predict discrete yet ordered cat-

egories (e.g., user ratings 1-5 stars) while preserving the sequential relationships between categories. Through ordinal classification, we obtain probabilistic outputs for each predicted value, naturally quantifying prediction uncertainty.

To optimize the ordinal classification model, OCE-TS uses ordinal cross-entropy (OCE) loss (Niu et al. 2016). The OCE loss modifies conventional cross-entropy through order-preserving mechanisms. By evaluating discrepancies in cumulative distribution functions (CDFs) of predicted versus true labels, it guarantees that predictions adhere to the inherent hierarchy of ordinal categories.

The main contributions of this paper are as follows:

- We propose OCE-TS, a novel approach that combines linear model with ordinal cross-entropy to solve time series value prediction problems while preserving temporal ordering constraints.
- We compare MSE (regression) and cross-entropy (classification) loss stability via influence function analysis, demonstrating that classification achieves superior stability when predicted probabilities are uniformly distributed and feature representations exhibit isotropic property;
- We verify OCE-TS's effectiveness across multiple benchmarks, which provides empirical guarantees for a new learning paradigm for time series forecasting.

Proofs and supplementary experiments are provided in the Appendix[1].

Related Work

Building upon ordinal classification for time series forecasting, our work intersects with three key domains: distributional regression (already discussed in Introduction), time series forecast, and ordinal classification. We now survey the latter two research areas.

Long-term time series forecasting

Long-term time series forecasting remains highly challenging due to extended prediction horizons, error accumulation, and distribution drift. Recent research primarily focuses on four key directions. First, for network architecture innovation, research emphasizes enhanced long-term dependency and periodicity modeling, Transformers (Vaswani et al. 2017) and their variants (e.g., Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021)) have become predominant. Meanwhile, lightweight models (e.g., DLinear (Zeng et al. 2023), LightTS (Campos et al. 2023)) achieve comparable performance with lower computational costs in specific scenarios. Second, for time series representation learning and knowledge distillation, self-supervised learning (e.g., TS2Vec) and knowledge distillation methods (e.g., TCN-Distill and TimeNet) are leveraged to enhance generalization capability and robustness while reducing dependence on labeled data. Third, for variable and feature hierarchical modeling, methods explicitly capture inter-variable dependencies and feature importance, primarily through cross-variable attention mechanisms (e.g., MTGNN (Wu et al. 2020)), learnable variable selection modules, and adaptive feature transformations to improve multivariate data modeling. Finally, for large models and zero-shot inference, research explores large language models (LLMs (Gruver et al. 2023)) for contextual learning (e.g., Time-LLM (Jin et al. 2024)), enabling zero-shot forecasting and cross-domain transfer through time series tokenization and prompt tuning techniques.

Ordinal Classification Task

Ordinal classification faces challenges including strong label-order dependency, ambiguous category intervals, and limited model adaptability. Existing research primarily focuses on three directions. First, loss function design, where order-aware loss functions (e.g., threshold-based (Niu et al. 2016) and margin-based (Pitawela, Carneiro, and Chen 2025) losses) are introduced to explicitly model ordinal relationships; second, model architecture improvements, including approaches like Support Vector Ordinal Regression (SVOR) (Gu et al. 2015) and COrrelation ALignment (CORAL) (Shi, Cao, and Raschka 2023), which address label ordering via explicit threshold partitioning and implicit ordinal encoding, while attention mechanisms (Vaswani et al. 2017) and Bayesian models demonstrate strong performance in capturing long-range dependencies and modeling uncertainty; and finally, probabilistic ordinal modeling, which adopts frameworks like ordered Probit/Logit (Liu, Wang, and Kong 2019) and cumulative link models (Gutiérrez et al. 2016) to characterize the conditional probability distribution over ordered categories, incorporating monotonicity constraints to reflect the inherent ordinal structure.

Time Series Forecasting Problem

Time series forecasting aims to predict future values based on historical observations. Given a lookback window $\boldsymbol{X}_t = (X_t, X_{t-1}, \ldots, X_{t-w+1})^\top \in \mathbb{R}^w$ of w historical observations, we predict the H-step future trajectory $\boldsymbol{Y} = (Y_{t+1}, \ldots, Y_{t+H})^\top \in \mathbb{R}^H$ through $\hat{\boldsymbol{Y}} = g(\boldsymbol{X}_t)$, where $g : \mathbb{R}^w \to \mathbb{R}^H$ is the forecasting function. For multivariate series with M-dimensional observations $\boldsymbol{x}_t \in \mathbb{R}^M$, this extends to predicting $\boldsymbol{Y} = \{\boldsymbol{y}_{t+1}, \ldots, \boldsymbol{y}_{t+H}\} \in \mathbb{R}^{H\times M}$ from input $\boldsymbol{X}_t = \{\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_{t-w+1}\} \in \mathbb{R}^{w\times M}$.

The forecasting quality is evaluated via:

$$\text{MSE} = \frac{1}{H}\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_F^2 = \frac{1}{H}\sum_{j=1}^{H}\|\boldsymbol{y}_{t+j} - \hat{\boldsymbol{y}}_{t+j}\|_2^2. \quad (1)$$

Minimizing MSE corresponds to fitting the conditional expectation of the true data, ensuring unbiased predictions. However, MSE only supports point estimates and fails to capture uncertainty, motivating reformulations of the forecasting task as a probability prediction problem.

Regression-to-Classification Framework

The core idea of transforming regression into classification involves shifting from predicting point estimates $\hat{Y}_{t+j}$

[1]The Appendix is in (Wang et al. 2025)

through MSE minimization to learning complete predictive distributions $\boldsymbol{\pi}(\hat{Y}_{t+j})$, where $\hat{Y}_{t+j} \in \mathbb{R}$ represents the original continuous predicted variable and $\boldsymbol{\pi}(\hat{Y}_{t+j}) \in \mathbb{R}^K$ denotes a discrete probability distribution vector satisfying $\sum_{k=1}^{K} \pi_k = 1$. The probabilistic regression framework transforms continuous prediction into a distribution classification problem through three key operations.

First, the true distribution is constructed by discretizing the ground truth $Y_{t+j} \in \mathbb{R}$ into bin probabilities $p_k = F_Y(u_k) - F_Y(\ell_k)$ for $k = 1, ..., K$, where F_Y is the empirical cumulative distribution function (CDF) and $\mathcal{B}_k = [\ell_k, u_k)$ forms a partition of the target value space.

The predictive model g_θ with parameters θ then learns to output a probability distribution $\boldsymbol{\pi}(\boldsymbol{X}_t; \theta) = (\pi_1, ..., \pi_K)$, where each $\pi_k = \mathbb{P}(\hat{Y}_{t+j} \in \mathcal{B}_k | \boldsymbol{X}_t)$ represents the predicted probability for bin k. This model is trained by minimizing the distributional discrepancy $\mathcal{L}(\theta) = D\big(\boldsymbol{p}(Y_{t+j}), \boldsymbol{\pi}(\boldsymbol{X}_t; \theta)\big)$ through the optimization objective $\theta^* = \arg\min_\theta \mathbb{E}_{\boldsymbol{X}_t, Y_{t+j}} \big[D\big(\boldsymbol{p}(Y_{t+j}), \boldsymbol{\pi}(\boldsymbol{X}_t; \theta)\big)\big]$, where D measures the divergence (e.g., cross-entropy) between the true distribution $\boldsymbol{p}$ and predicted distribution $\boldsymbol{\pi}$.

Finally, continuous point estimates $\hat{Y}_{t+j}$ are recovered via $\hat{Y}_{t+j} = \mathcal{G}\big((v_k, \pi_k(\boldsymbol{X}_t; \theta^*))_{k=1}^K\big)$, where $\mathcal{G}$ transforms the discrete probability distribution back to the continuous space using representative values v_k for each bin $\mathcal{B}_k$.

Ordinal Cross-Entropy Loss

This section presents the definition of OCE, followed by a comparative example illustrating the fundamental difference with standard cross-entropy (CE) and a novel influence function analysis that contrasts their sensitivity characteristics with MSE. See Appendix A.1 for the definition of CE.

Definition 1 (Ordinal Cross-Entropy Loss)**.** (Hu et al. 2010) Let $Y \in \{1, \ldots, K\}$ be the true class label and $\hat{Y}$ the predicted class. Given the predicted probability distribution $\boldsymbol{q} = [q_1, \ldots, q_K]$ where $q_k = \mathbb{P}(\hat{Y} = k)$ and the true probability distribution $\boldsymbol{p} = [p_1, \ldots, p_K]$. We define the cumulative probabilities as: $\mathbb{P}_{\text{pred}}(\hat{Y} \le k) = \sum_{i=1}^{k} q_i$, $\mathbb{P}_{\text{pred}}(\hat{Y} > k) = 1 - \mathbb{P}_{\text{pred}}(\hat{Y} \le k)$. Let $\mathbb{P}_{\text{true}}(Y \le k)$ represent the ground-truth cumulative distribution. For hard labels: $\mathbb{P}_{\text{true}}(Y \le k) = \mathbb{I}\{Y \le k\}$, where $\mathbb{I}$ is the indictor function. The ordinal cross-entropy loss is then:

$$\mathcal{L}_{\text{OCE}}(Y, \hat{Y}) = -\sum_{k=1}^{K-1} \Big[\mathbb{P}_{\text{true}}(Y \le k) \log \mathbb{P}_{\text{pred}}(\hat{Y} \le k) + \mathbb{P}_{\text{true}}(Y > k) \log \mathbb{P}_{\text{pred}}(\hat{Y} > k)\Big]. \tag{2}$$

Comparative Analysis of OCE and CE

Table 1 compares standard cross-entropy and ordinal cross-entropy (OCE) performance across various prediction scenarios. In Case 1, where the true distribution is [0.8,0.1,0.1], although predicted distribution B achieves a lower CE value (0.449 vs 0.518), its OCE loss is higher (1.528 vs 1.396). This indicates that CE may underestimate the ordinal errors in predicted distribution B. Similarly, in Case 2 with a true distribution of [0.2,0.1,0.7], while predicted distribution A shows a slightly better CE value (0.604 vs 0.624), its OCE loss (2.029) is higher, further validating the limitations of CE in evaluating ordinal relationships.

Scenario	Type	Distribution $\boldsymbol{p}$	CE	OCE
Case 1	True	$[0.8, 0.1, 0.1]$	–	–
	Pred A	$[0.3, 0.5, 0.2]$	0.518	1.396
	Pred B	$[0.4, 0.1, 0.5]$	0.449	1.528
Case 2	True	$[0.2, 0.1, 0.7]$	–	–
	Pred A	$[0.6, 0.2, 0.2]$	0.604	2.029
	Pred B	$[0.3, 0.5, 0.2]$	0.624	1.720

Table 1: Comparison of CE and OCE

Notably, the OCE loss shows greater sensitivity to deviations in class ordering between predicted and true distributions. Therefore, it is better suited for classification tasks emphasizing ordinal relationships, providing a more accurate reflection of the model's ordering performance.

Comparative Analysis of OCE and MSE

This subsection presents a comparative influence function analysis of MSE and CE losses, focusing on their differential sensitivity to feature scaling characteristics and prediction residuals. The analysis framework applies uniformly to all cross-entropy-type losses (including both CE and OCE), as they share identical functional forms - differing only in their probability computation methods (plain softmax for CE versus cumulative probabilities for OCE).

Influence functions are widely used in robustness analysis, outlier detection, and training sample evaluation.Existing research focuses on computing IF (Schioppa et al. 2022) and using them for sample selection (Pruthi et al. 2020), whereas this paper analyzes the stability of MSE and CE losses through influence functions. We analyze and compare MSE's and CE's influence functions via simple linear models. We consider the standard linear regression model $y = \boldsymbol{x}^\top \boldsymbol{\theta} + \epsilon$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional feature vector, $\boldsymbol{\theta} \in \mathbb{R}^d$ is the d-dimensional parameter vector, $\epsilon \in \mathbb{R}$ is the scalar noise term, and $y \in \mathbb{R}$ is the scalar response. The mean squared error loss function for a data point $\boldsymbol{z} = (\boldsymbol{x}, y)$ is given by: $L(\boldsymbol{z}, \boldsymbol{\theta}) = \frac{1}{2}(y - \boldsymbol{x}^\top \boldsymbol{\theta})^2 \in \mathbb{R}$.

Consider a K-class classification problem with a parametric softmax model. Let $\boldsymbol{x} \in \mathbb{R}^d$ be an input feature vector and $y \in \{1, ..., K\}$ its corresponding class label. The model's predicted probabilities are given by:

$$p(y|\boldsymbol{x}; \boldsymbol{\beta}) = \sigma(\boldsymbol{x}^\top \boldsymbol{\beta})_y = \frac{e^{\boldsymbol{x}^\top \boldsymbol{\beta}_y}}{\sum_{k=1}^{K} e^{\boldsymbol{x}^\top \boldsymbol{\beta}_k}}, \tag{3}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K] \in \mathbb{R}^{d \times K}$ contains the model parameters. The cross-entropy loss for a single observation $\boldsymbol{z} = (\boldsymbol{x}, y)$ is: $L(\boldsymbol{z}, \boldsymbol{\beta}) = -\sum_{k=1}^{K} \mathbb{I}_{y=k} \log \sigma(\boldsymbol{x}^\top \boldsymbol{\beta})_k$.

Influence Function (IF) measures the sensitivity of model parameters or outputs to a single training sample. Its basic

form is given by:

$$\mathrm{IF}(\boldsymbol{z}) = -\boldsymbol{H}_{\boldsymbol{\theta}}^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}), \tag{4}$$

where $\boldsymbol{H}_{\boldsymbol{\theta}}$ denotes the Hessian matrix evaluated at the optimal parameters $\boldsymbol{\theta}$, and $\nabla_{\boldsymbol{\theta}} \mathcal{L}$ represents the gradient of the loss function with respect to the model parameters.

The influence function for MSE can be derived as:

$$\mathrm{IF}_{\mathrm{MSE}}(\boldsymbol{z}; \boldsymbol{\theta}) = \left(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]\right)^{-1} \boldsymbol{x}(y - \boldsymbol{x}^\top \boldsymbol{\theta}) \in \mathbb{R}^d, \tag{5}$$

which quantifies the d-dimensional effect of individual data points on parameter estimates.

The influence function for CE can be derived as:

$$\mathrm{IF}_{\mathrm{CE}}(\boldsymbol{z}; \boldsymbol{\beta}) = -\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top \otimes \boldsymbol{P}\right]\right)^{-1} (\boldsymbol{x} \otimes (\sigma(\boldsymbol{x}^\top \boldsymbol{\beta}) - \mathbf{e}_y)), \tag{6}$$

where $\boldsymbol{P} = \mathrm{diag}(\sigma(\boldsymbol{x}^\top \boldsymbol{\beta})) - \sigma(\boldsymbol{x}^\top \boldsymbol{\beta})\sigma(\boldsymbol{x}^\top \boldsymbol{\beta})^\top$ is the softmax output covariance matrix ($K \times K$), $\mathbf{e}_y \in \mathbb{R}^K$ is the one-hot encoded true label, $\otimes$ represents the Kronecker product.

As shown in Equations (5) and (6), the cross-entropy loss alleviates the influence of the covariance matrix and the residuals through transformations involving the $\boldsymbol{P}$ matrix and the sigmoid function, respectively. Below, we present the specific ratio between the two effects:

Theorem 1 (Influence Function Growth Rate Comparison)**.** Suppose that the covariance matrix $\boldsymbol{\Sigma}_X = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is positive definite with $\lambda_{\min}(\boldsymbol{\Sigma}_X) > 0$ and the expected Softmax matrix $\boldsymbol{P}$ is positive definite with $\lambda_{\min}(\boldsymbol{P}) > 0$. Assume the input features have finite second moments ($\mathbb{E}[\|\boldsymbol{x}\|_2^2] < \infty$) and the sample size exceeds the feature dimension ($n > d$). For non-degenerate predictions where the MSE residual is non-zero ($y - \boldsymbol{x}^\top \boldsymbol{\theta} \neq 0$) and the CE probability deviation is non-trivial ($\boldsymbol{\sigma} - \boldsymbol{e}_y \neq \mathbf{0}$), the influence function ratio $R = \|\mathrm{IF}_{\mathrm{CE}}\|_2 / \|\mathrm{IF}_{\mathrm{MSE}}\|_2$ satisfies the two-sided bound:

$$\frac{\|\boldsymbol{\sigma} - \boldsymbol{e}_y\|_2}{\kappa_2(\boldsymbol{\Sigma}_X)\lambda_{\max}(\boldsymbol{P})|y - \boldsymbol{x}^\top \boldsymbol{\theta}|} \leq R \leq \frac{\sqrt{2}\kappa_2(\boldsymbol{\Sigma}_X)}{\lambda_{\min}(\boldsymbol{P})|y - \boldsymbol{x}^\top \boldsymbol{\theta}|}, \tag{7}$$

where $\kappa_2(\boldsymbol{\Sigma}_X) = \|\boldsymbol{\Sigma}_X\|_2 \|\boldsymbol{\Sigma}_X^{-1}\|_2$ represents the spectral condition number.

From Theorem 1, we can observe that the CE loss becomes more stable than MSE when the ratio of influence functions satisfies $R \leq 1$. This stability condition holds when the data covariance matrix is well-conditioned and model predictions avoid extreme confidence.The proof and additional discussion are provided in Appendix A.6.

Methodology

We propose the Ordinal Cross-Entropy Time Series Forecasting (OCE-TS) method, a novel regression-to-classification framework that transforms continuous forecasting into an ordinal probability estimation problem. As shown in Figure 1, the approach consists of three key components: Target-to-Probability Transformation (TPT) that discretizes continuous values into probability distributions across ordered bins; Deep Ordinal Classifier Training using a neural architecture with ordinal cross-entropy loss to preserve the inherent ordering of bins while learning temporal patterns; and Probability-to-Value Reconstruction (PVR)

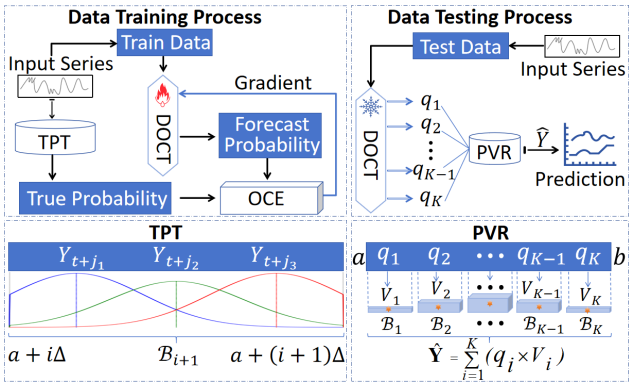


Figure 1: The Framework of Ordinal Cross-Entropy Loss-Based Time Series Forecasting (OCE-TS)

that converts the predicted class probabilities back to continuous values.

Given a true value $Y_{t+j} \in \mathbb{R}$ confined to the interval $[a, b]$, we construct its discrete probability distribution over K bins as follows. The target interval is divided into K equal-width bins, where each bin $\mathcal{B}_k$, $k = 1, \ldots, K$ has lower and upper bounds defined by: $\mathcal{B}_k = [\ell_k, u_k)$, where $\ell_k = a + (k-1)\Delta$, $u_k = a + k\Delta$, and $\Delta = (b-a)/K$ for $k = 1, \ldots, K$. Δ represents the uniform bin width, ensuring complete coverage of $[a, b]$ without overlap.

For each observation Y_{t+j}, we center a truncated Gaussian distribution at $y_c = Y_{t+j}$:

$$p(y) = \begin{cases} \frac{1}{Z\sigma\sqrt{2\pi}} \exp\left(-\frac{(y-y_c)^2}{2\sigma^2}\right) & y \in [a, b], \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The normalization factor Z ensures unit probability mass within $[a, b]$:

$$Z = \frac{1}{2}\left[\mathrm{erf}\left(\frac{b - y_c}{\sigma\sqrt{2}}\right) - \mathrm{erf}\left(\frac{a - y_c}{\sigma\sqrt{2}}\right)\right]. \tag{9}$$

Here, $\mathrm{erf}(\cdot)$ denotes the Gauss error function, and σ^2 controls the dispersion of the distribution.

The probability mass of Y_{t+j} for each bin $\mathcal{B}_k$ is obtained by integrating the CDF:

$$p_k^{(j)} = \mathbb{P}(Y_{t+j} \in \mathcal{B}_k) \tag{10}$$

$$= \frac{1}{2Z}\left[\mathrm{erf}\left(\frac{u_k - y_c}{\sigma\sqrt{2}}\right) - \mathrm{erf}\left(\frac{\ell_k - y_c}{\sigma\sqrt{2}}\right)\right]. \tag{11}$$

This formulation guarantees normalization and non-negativity $\forall k$: $\sum_{k=1}^{K} p_k^{(j)} = 1, \quad p_k^{(j)} \geq 0$.

Each observation Y_{t+j} is mapped to a K-dimensional probability vector: $\boldsymbol{p}^{(j)} = \left[p_1^{(j)}, p_2^{(j)}, \ldots, p_K^{(j)}\right]^T, \quad p_k^{(j)} = \mathbb{P}(Y_{t+j} \in \mathcal{B}_k)$, where the vector components sum to unity ($\|\boldsymbol{p}^{(j)}\|_1 = 1$) by construction.

Deep Ordinal Classifier Training

To obtain predictive probability distributions, this study adopts a probabilistic forecasting framework based on the

Decomposition-Linear (DLinear) Model. DLinear (Zeng et al. 2023) is a time series forecasting model that employs series decomposition and multi-layer linear mapping. Given an input window $\boldsymbol{X}_t \in \mathbb{R}^{w\times M}$ with w time steps and M features, the model first decomposes the series into trend and residual components through moving average smoothing:

$$\boldsymbol{T}_t = \mathrm{MA}_l(\boldsymbol{X}_t), \quad \boldsymbol{S}_t = \boldsymbol{X}_t - \boldsymbol{T}_t, \tag{12}$$

where $\mathrm{MA}_l(\cdot)$ is the moving average smoothing with window size l, and $\boldsymbol{S}_t$ captures residual variations. This decomposition separates temporal patterns at different scales.

The model processes each component through dedicated linear projections:

$$\tilde{\boldsymbol{Y}} = \boldsymbol{W}_t\boldsymbol{T}_t^\top + \boldsymbol{W}_s\boldsymbol{S}_t^\top \in \mathbb{R}^{H\times M}, \tag{13}$$

where $\boldsymbol{W}_t, \boldsymbol{W}_s \in \mathbb{R}^{H\times w}$ are learnable projection matrices. For each prediction horizon j, the scalar output $\tilde{Y}_{t+j}$ is transformed into bin probabilities:

$$\boldsymbol{q} = \mathrm{softmax}(\boldsymbol{W}_o\tilde{Y}_{t+j} + \boldsymbol{b}_o), \tag{14}$$

with parameters $\boldsymbol{W}_o \in \mathbb{R}^{K\times M}$ and $\boldsymbol{b}_o \in \mathbb{R}^K$ mapping to K ordinal bins. The complete parameter set $\theta = \{\boldsymbol{W}_t, \boldsymbol{W}_s, \boldsymbol{W}_o, \boldsymbol{b}_o\}$ is optimized by minimizing:

$$\mathcal{L}(\theta) = \frac{1}{N}\frac{1}{H}\sum_{i=1}^{N}\sum_{j=1}^{H}\mathcal{L}_{\mathrm{OCE}}\left(\boldsymbol{p}(Y_{t_i+j}), \boldsymbol{q}(\boldsymbol{X}_{t_i};\theta)\right) \tag{15}$$

where $\mathcal{L}_{\mathrm{OCE}}$ is the ordinal cross-entropy loss (Eq. (2)), Y_{t_i+j} is the ground truth at $t_i + j$, N is the number of training windows, H the prediction horizon, and i, j index training samples and forecast steps, respectively.

The model parameters are optimized by minimizing the loss function $\mathcal{L}(\theta)$ to obtain the optimal parameters:

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta), \tag{16}$$

after obtaining the optimal parameters θ^*, the model's prediction can be expressed as:

$$q_t = \boldsymbol{q}(\boldsymbol{X}_t;\theta^*), \tag{17}$$

where q_t is the model's prediction at time t for Y_{t+j}.

Probability-to-Value Reconstruction

The final prediction $\hat{Y}_{t+j}$ is computed as the probability-weighted average of bin representatives $\{v_k\}_{k=1}^K$:

$$\hat{Y}_{t+j} = \mathcal{G}\big(\{(v_k, q_k)\}_{k=1}^K\big) = \sum_{k=1}^{K} v_k \cdot q_k, \tag{18}$$

where v_k is the predefined center of bin $\mathcal{B}_k$.

The above end-to-end probabilistic forecasting framework is trained using ordinal cross-entropy loss to preserve bin ordering while enabling both accurate point forecasts and uncertainty quantification.

Experimental Analysis

This section evaluates OCE-TS against five baselines on seven datasets. We analyze key factors like loss functions, parameters and distribution assumptions to validate our method. Additional experimental results are provided in the Appendix C.

Experiments Settings

Dataset We conduct experiments on seven publicly available time-series datasets, including ETT (ETTh1, ETTh2, ETTm1, ETTm2), Exchange, ILI, and Weather. The detailed dataset descriptions are provided in Appendix C.1.

Baselines. We adopt five representative models for long-term time series forecasting as our baselines, including three Transformer-based models: Autoformer (Wu et al. 2021), iTransformer (Liu et al. 2023), TimeXer (Wang et al. 2024) and TimeBridge (Liu et al. 2024), as well as one MLP-based model: DLinear (Zeng et al. 2023).

Experimental Environment The experiments are conducted on a machine equipped with an NVIDIA GeForce RTX 4060 GPU, an Intel(R) Core(TM) i7-14700F CPU, and 16 GB of RAM. The implementation requires Python 3.8.20 and PyTorch 2.0.0 compiled with CUDA 11.8 support, leveraging GPU acceleration during model training.

Evaluation Metrics The MSE and MAE are employed to evaluate model performance in time series forecasting. Lower values indicate better predictive performance. The definition of MAE is provided in Appendix C.2.

Model Configuration and Training Details The input time series are normalized to $[-1, 1]$ or $[0, 1]$ during training and validation, and denormalized during testing for consistent metric evaluation. The support $[a, b]$ of the truncated Gaussian distribution is set to match the normalization range.Models are trained with a batch size of 32 for 15 epochs using the Adam optimizer (initial learning rate 0.005), employing dynamic learning rate adjustment based on validation performance, with an early stopping mechanism (patience=5 epochs). The input window size w is set to 336, and prediction lengths H are selected from $\{96, 192, 336, 720\}$. Details are provided in Appendix C.3.

Comparative Results

To verify the effectiveness of the proposed method, we conduct a systematic performance comparison between OCE-TS and five mainstream models trained with the traditional MSE loss function on multiple benchmark datasets.

Table 2 presents the comparative performance analysis. Given the relatively small size of the ILI dataset, its lookback window is configured as 104 timesteps with forecast horizons $\{24, 36, 48, 60\}$. Following conventional practice, other datasets adopt a fixed lookback window of 336 timesteps and forecast horizons $\{96, 192, 336, 720\}$. The experimental results show that OCE-TS achieves superior forecasting performance across most benchmark datasets.

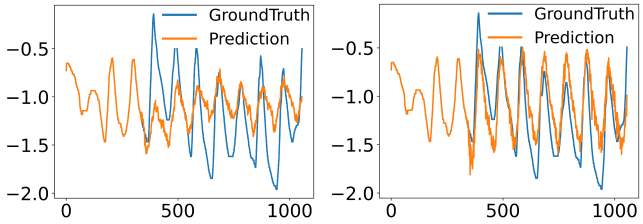
To intuitively demonstrate the model's fitting performance on two types of datasets, this paper visualizes the 720-step forecasting results by comparing the predicted values with the ground truth. Both the proposed method and the DLinear model use a 336-step historical sequence as input.

As shown in Figure 2, on the ETTm2 dataset, our method demonstrates superior ground truth tracking capability compared to DLinear, thereby validating the enhanced performance of our proposed approach.

Table 3 shows that OCE-TS achieves high accuracy with moderate cost, balancing efficiency and performance.

Dataset	H	MSE						MAE					
		Autoformer (2021)	Dlinear (2022)	Itransformer (2024)	Timexer (2024)	TimeBridge (2024)	Our	Autoformer (2021)	Dlinear (2022)	Itransformer (2024)	Timexer (2024)	TimeBridge (2024)	Our
ETTh1	96	0.453	0.375	0.386	0.377	0.358	**0.341**	0.453	0.399	0.405	0.397	**0.392**	0.395
	192	0.504	0.405	0.441	0.425	**0.388**	0.416	0.482	0.416	0.436	0.426	**0.411**	0.445
	336	0.505	0.439	0.487	0.457	**0.401**	0.491	0.484	0.443	0.458	0.441	**0.419**	0.491
	720	0.498	0.472	0.503	0.464	**0.447**	0.547	0.500	0.490	0.491	0.463	**0.458**	0.528
ETTh2	96	0.368	0.289	0.297	0.289	0.295	**0.197**	0.410	0.353	0.349	0.340	0.354	**0.313**
	192	0.422	0.383	0.380	0.370	0.351	**0.248**	0.434	0.418	0.400	0.391	0.389	**0.352**
	336	0.471	0.448	0.428	0.422	0.351	**0.313**	0.475	0.465	0.432	0.434	0.397	**0.395**
	720	0.474	0.605	0.427	0.429	0.388	**0.379**	0.484	0.551	0.445	0.445	0.436	**0.435**
ETTm1	96	0.481	0.299	0.334	0.309	0.297	**0.195**	0.463	0.343	0.368	0.352	0.353	**0.308**
	192	0.598	0.335	0.377	0.355	0.333	**0.279**	0.513	0.365	0.391	0.378	0.375	**0.363**
	336	0.579	0.369	0.426	0.387	**0.366**	0.372	0.509	**0.386**	0.420	0.399	0.399	0.407
	720	0.560	0.425	0.491	0.448	**0.414**	0.479	0.509	**0.421**	0.459	0.435	0.423	0.466
ETTm2	96	0.255	0.167	0.180	0.171	0.158	**0.118**	0.339	0.260	0.264	0.255	0.249	**0.230**
	192	0.281	0.224	0.250	0.238	0.215	**0.167**	0.340	0.303	0.309	0.300	0.291	**0.279**
	336	0.339	0.281	0.311	0.301	0.263	**0.211**	0.372	0.342	0.348	0.340	0.323	**0.319**
	720	0.422	0.397	0.412	0.401	0.348	**0.277**	0.419	0.421	0.407	0.397	0.376	**0.367**
Exchange	96	0.197	0.081	0.086	0.089	0.084	**0.055**	0.382	0.203	0.206	0.208	0.202	**0.148**
	192	0.300	0.157	0.177	0.196	0.189	**0.107**	0.369	0.293	0.299	0.313	0.310	**0.224**
	336	0.509	0.305	0.331	0.354	0.362	**0.199**	0.524	0.414	0.417	0.429	0.436	**0.322**
	720	1.447	0.643	0.847	0.894	0.900	**0.521**	0.941	0.601	0.691	0.708	0.717	**0.578**
ILI	24	3.483	2.215	2.695	1.971	2.055	**0.570**	1.287	1.081	1.703	0.868	0.890	**0.458**
	36	3.103	1.963	2.563	2.068	2.017	**0.749**	1.148	0.963	1.051	0.934	0.950	**0.558**
	48	2.669	2.130	2.567	1.925	1.898	**1.180**	1.085	1.024	1.048	0.884	0.905	**0.710**
	60	2.770	2.368	2.625	1.920	1.796	**1.282**	1.125	1.096	1.068	0.892	0.897	**0.738**
Weather	96	0.266	0.176	0.174	0.168	0.143	**0.107**	0.336	0.237	0.214	0.209	0.192	**0.144**
	192	0.307	0.220	0.221	0.220	0.185	**0.140**	0.367	0.282	0.254	0.254	0.235	**0.190**
	336	0.359	0.265	0.278	0.276	0.237	**0.169**	0.395	0.319	0.296	0.296	0.277	**0.228**
	720	0.419	0.323	0.358	0.353	0.307	**0.225**	0.428	0.362	0.347	0.347	0.330	**0.283**

Table 2: Comparison of Model Performances



(a) DLinear (ETTm2) (b) Ours (ETTm2)

Figure 2: Model Fitting Results of DLinear and Ours

models	Autoformer	Dlinear	Itransformer	Timexer	TimeBridge	Ours
ms/iter	123.3	19.3	29	17.2	21	34.9

Table 3: Per-Iteration Runtime by Model (ms)

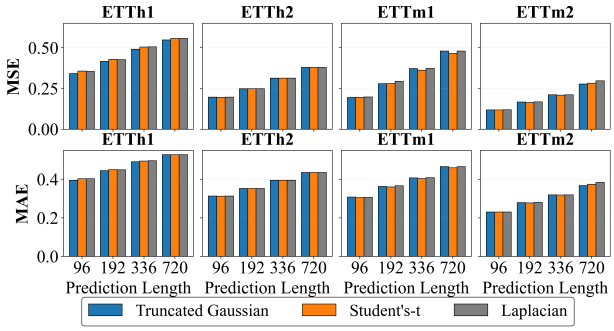


Figure 3: Distribution Comparison on ETT Datasets

Study of Different Probability Distributions

This section analyzes the impact of the probability distribution on the performance of the ETTh dataset. Distributions are defined in Appendix C.4.

As illustrated in Figure 3, the truncated Gaussian distribution achieves optimal performance on the ETTh1 dataset, demonstrating superior MSE and MAE metrics across all prediction horizons compared to both the Student's-t distribution and Laplacian distribution, with particularly notable advantages in long-term forecasting. However, the Student's-t distribution exhibits better performance on ETTm2, while all three distributions show comparable results on ETTh2 and ETTm1 without statistically significant differences. Hence, the selection of an appropriate probability distribution should be carefully determined according to specific task requirements and dataset characteristics.

Lookback Window Sizes

To evaluate the impact of lookback window size on forecasting performance, this study employs multi-scale sliding windows (48 / 96 / 192 / 336 / 504 / 720 timesteps) for 720-step-ahead prediction across multiple datasets, quantitatively analyzing the correlation between historical information scale and long-term forecasting accuracy.

The first row of Figure 4 presents the performance of ETTm1 and ETTh2 under different lookback window lengths, the experiments show that increasing the look-

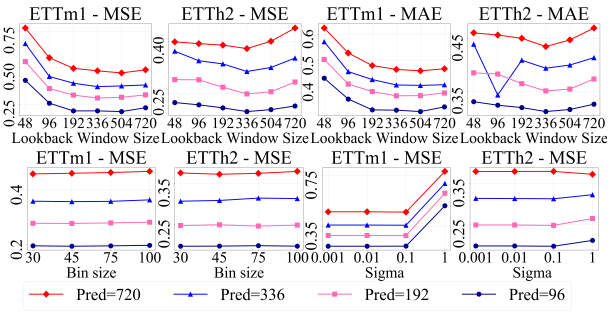


Figure 4: Lookback Window Sizes & Parameter Sensitivity

back window significantly improves prediction accuracy for ETTm1 but has limited benefits for ETTh2, indicating that the optimization effect of historical information length depends on dataset characteristics. ETTm1 achieves optimal performance with $\geq$336 steps, while ETTh2 performs best at 192 steps. Therefore, the selection of the lookback window should balance accuracy and efficiency, with adjustments tailored to different datasets. Please refer to Appendix C.5 for complete experimental results and details.

Parameter Sensitivity

The OCE-TS method involves two key parameters: the number of bins (bin = K) and the standard deviation (σ). In our experimental setup, we consider bin $\in \{30, 45, 75, 100\}$ and $\sigma \in \{0.001, 0.01, 0.1, 1\}$, while keeping all other settings consistent with the baseline configuration. To systematically analyze the influence of these parameters, we examine the effect of bin by fixing $\sigma = 0.01$, and analyze the impact of σ under a fixed bin of 100. Based on these results, we provide recommendations for parameter configuration.

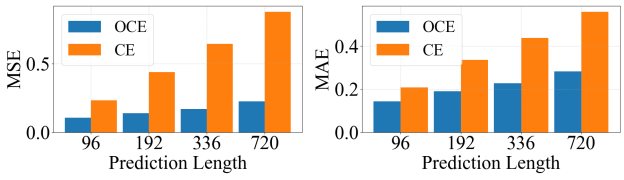
The second row of Figure 4 shows the variations in model performance with respect to σ values and binning parameters. The ETTm1 and ETTh2 datasets are insensitive to the number of bins but exhibit different levels of sensitivity to the standard deviation. Both datasets maintain stable performance across various bin settings; however, when $\sigma = 1$, the MSE on ETTm1 increases significantly, and ETTh2 also experiences a performance drop. Considering both stability and accuracy, we recommend setting the number of bins to 100 and the standard deviation σ to 0.01. Please refer to Appendix C.6 for complete experimental results and details.

Comparison with CE

This section compares the performance between the OCE loss and the CE loss. See Appendix C.7 for complete experimental results and details. Experimental results in Figure 5 validate that OCE loss achieves superior metric performance (MAE / MSE) over CE loss on the Weather dataset.

Model Performance under Different SNRs

This experiment compares the OCE-TS method with DLinear under varying signal-to-noise ratios (SNRs) ($-3\,\mathrm{dB}$, $0\,\mathrm{dB}$, $3\,\mathrm{dB}$, $10\,\mathrm{dB}$, and $20\,\mathrm{dB}$), where additive Gaussian white noise is injected into inputs to evaluate robustness



(a) MSE (CE vs. OCE) (b) MAE (CE vs. OCE)

Figure 5: The Comparison between CE Loss and OCE Loss

on the ETTm2 and Weather datasets. Please refer to Appendix C.8 for complete experimental results and details.

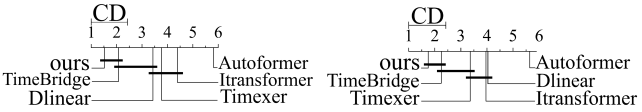
Dataset	-3dB		0dB		3dB		10dB		20dB	
	MSE	Ours	MSE	Ours	MSE	Ours	MSE	Ours	MSE	Ours
ETTm2	0.440	**0.275**	0.431	**0.274**	0.425	**0.273**	0.420	**0.277**	0.420	**0.280**
Weather	0.342	**0.227**	0.339	**0.225**	0.338	**0.224**	0.337	**0.223**	0.338	**0.224**

Table 4: Comparison Under Different Noise Levels

The experimental results show our method outperforms DLinear under varying SNRs with minimal noise impact, demonstrating its robustness.

Significance Analysis

To evaluate the statistical significance of performance differences among the compared algorithms, we conduct a critical difference (CD) analysis based on both MAE and MSE metrics. The x-axis of Figure 6 represents the average rankings, and algorithms connected by horizontal lines indicate no statistically significant differences. Our approach exhibits superiority, with its leftmost position on the CD axis confirming statistically significant advantages over counterparts. Please refer to Appendix C.9 for more details.



(a) MSE (b) MAE

Figure 6: Critical Difference Analysis

Conclusion

Temporal sequence prediction plays a crucial role across various domains, yet existing models generally suffer from two major limitations: insufficient uncertainty quantification and weak robustness. To address this, this paper proposes the OCE-TS framework, which synergistically integrates regression-as-classification with ordinal cross-entropy optimization to achieve more accurate predictive distribution modeling. Experiments on multiple benchmark datasets demonstrate that this approach significantly outperforms baseline models in both prediction accuracy and stability. Future work will explore more expressive distribution modeling techniques to capture complex uncertainty structures.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. T2495254, 62306170, U24A20253, 62136005, 62476160, 62441239), the Science and Technology Major Project of Shanxi (No. 202201020101006), the Special Fund for Science and Technology Innovation Teams of Shanxi Province (No. 202304051001001).

References

Box, G. E. P.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time Series Analysis: Forecasting and Control*. Hoboken, NJ: John Wiley & Sons, 5 edition.

Campos, D.; Zhang, M.; Yang, B.; Chen, H.; Wang, S.; Xu, L.; and Li, L. 2023. LightTS: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2): 1–27.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large Language Models Are Zero-Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*, volume 36.

Gu, B.; Sheng, V. S.; Tay, K. Y.; Romano, W.; and Li, S. 2015. Incremental Support Vector Learning for Ordinal Regression. *IEEE Transactions on Neural Networks and Learning Systems*, 26(7): 1403–1416.

Gutiérrez, P. A.; Perez-Ortiz, M.; Sanchez-Monedero, J.; Fernández-Navarro, F.; and Hervás-Martínez, C. 2016. Ordinal Regression Methods: Survey and Experimental Study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1): 127–146.

Hu, Q.; Guo, M.; Yu, D.; and Li, Q. 2010. Information Entropy for Ordinal Classification. *Science China Information Sciences*, 53(6): 1188–1200.

Hyndman, R. J.; Koehler, A. B.; Ord, K.; and Snyder, R. D. 2008. *Forecasting with Exponential Smoothing: The State Space Approach*.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; and Wen, Q. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations*.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7482–7491.

Liu, P.; Wu, B.; Hu, Y.; et al. 2024. TimeBridge: Non-Stationarity Matters for Long-term Time Series Forecasting. *arXiv preprint arXiv:2410.04442*.

Liu, W.; Pokharel, P. P.; and Principe, J. C. 2007. Correntropy: Properties and Applications in Non-Gaussian Signal Processing. *IEEE Transactions on Signal Processing*, 55(11): 5286–5298.

Liu, Y.; Hu, T.; Zhang, H.; Zhang, S.; Qin, Z.; Xie, Y.; Wu, Q.; Zhou, H.; and Zhang, W. 2023. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. *arXiv preprint arXiv:2310.06625*.

Liu, Y.; Wang, F.; and Kong, A. W. K. 2019. Probabilistic deep ordinal regression based on Gaussian processes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5301–5309.

Niu, Z.; Zhou, M.; Wang, L.; and Gao, Q. 2016. Ordinal Regression with Multiple Output CNN for Age Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4920–4928.

Pitawela, D.; Carneiro, G.; and Chen, H.-T. 2025. CLOC: Contrastive Learning for Ordinal Classification with Multi-Margin N-pair Loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15538–15548.

Pruthi, G.; Liu, F.; Kale, S.; et al. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930.

Schioppa, A.; Zablotskaia, P.; Vilar, D.; Chaudhary, V.; Guzmán, F.; and Schwenk, H. 2022. Scaling Up Influence Functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8179–8186.

Shi, X.; Cao, W.; and Raschka, S. 2023. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3): 941–955.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Wang, J.; Shi, H.; Li, F.; and Shang, X. 2025. Beyond MSE: Ordinal Cross-Entropy for Probabilistic Time Series Forecasting. arXiv:2511.10200.

Wang, Y.; Wu, H.; Dong, J.; Liu, G.; Liu, J.; Zhao, D.; and Tian, Q. 2024. Timexer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. *arXiv preprint arXiv:2402.19072*.

Wu, H.; Xu, J.; Wang, J.; Chen, Y.; Xue, Y.; Xue, Y.; and Tian, Q. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 753–763.

Zeng, A.; Chen, M.; Zhang, L.; Zhou, H.; Peng, J.; Zhang, S.; and Zhang, W. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11121–11128.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, Z.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.