Hierarchical Classification Auxiliary Network for Time Series Forecasting

Yanru Sun[1], **Zongxia Xie**[1]*, **Dongyue Chen**[1], **Emadeldeen Eldele**[2,3], **Qinghua Hu**[1]

[1] Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University, China
[2] Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
[3] Institute for InfoComm Research, Agency for Science, Technology and Research, Singapore
yanrusun@tju.edu.cn, caddiexie@hotmail.com, dyuechen@tju.edu.cn, emad0002@ntu.edu.sg, huqinghua@tju.edu.cn

Abstract

Deep learning has significantly advanced time series forecasting through its powerful capacity to capture sequence relationships. However, training these models with the Mean Square Error (MSE) loss often results in over-smooth predictions, making it challenging to handle the complexity and learn high-entropy features from time series data with high variability and unpredictability. In this work, we introduce a novel approach by tokenizing time series values to train forecasting models via cross-entropy loss, while considering the continuous nature of time series data. Specifically, we propose a **H**ierarchical **C**lassification **A**uxiliary **N**etwork, **HCAN**, a general model-agnostic component that can be integrated with any forecasting model. HCAN is based on a Hierarchy-Aware Attention module that integrates multi-granularity high-entropy features at different hierarchy levels. At each level, we assign a class label for timesteps to train an Uncertainty-Aware Classifier. This classifier mitigates the over-confidence in softmax loss via evidence theory. We also implement a Hierarchical Consistency Loss to maintain prediction consistency across hierarchy levels. Extensive experiments integrating HCAN with state-of-the-art forecasting models demonstrate substantial improvements over baselines on several real-world datasets.

Code — https://github.com/syrGitHub/HCAN

Introduction

Time series forecasting has received significant attention due to its wide-ranging social impact. Among existing approaches for time series forecasting, deep learning methods have emerged as significant contributors to this field (Zhou et al. 2021, 2022; Zeng et al. 2023; Ni et al. 2024). These methods showed a powerful capacity to capture sequence continuity features (Wen et al. 2023; Wang et al. 2024) and enhance forecasting performance in practical applications such as finance (Hou et al. 2022), weather forecasting (Lam et al. 2022), resource planning (Chen et al. 2021), and other domains (Shao et al. 2024; Wu et al. 2024).

Nevertheless, current time series forecasting methods relying on the Mean Square Error (MSE) loss for feature extraction can suffer inaccurate predictions. The main downside of the MSE loss is compressing the feature representation into a narrow space, limiting its ability to learn complexity and high-entropy feature representations, especially for those features that exhibit significant variability and unpredictability (Zhang et al. 2023; Pintea et al. 2023). Therefore, current methods often produce over-smooth predictions, leading to inaccuracies such as inflating wind speed estimates on sunny days when the actual wind speed is low, and underestimating wind speed on windy days when the actual wind speed is high. This weakness diminishes the utility of forecasting results for downstream applications, as shown in Figure 1a.

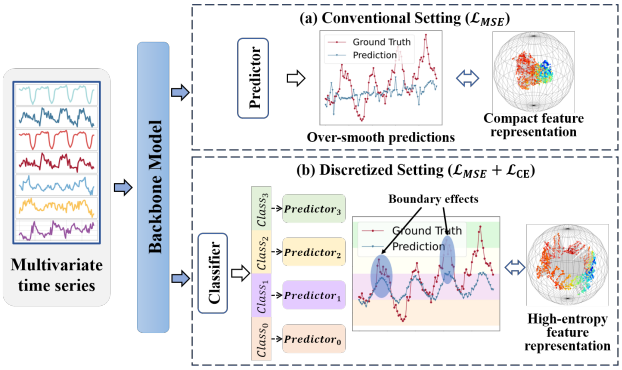


Figure 1: Comparison between Conventional and Discretized Settings for time series forecasting. (a) Conventional setting keeps features close together, producing over-smooth predictions; (b) Discretized setting spreads the features, resulting in a higher entropy feature space, but can misclassify inter-class boundary timesteps.

Recently, several studies have demonstrated the superiority of cross-entropy loss in capturing high-entropy feature representation from a mutual information perspective (Pintea et al. 2023; Zhang et al. 2023). Therefore, it has been successfully applied in various domains, such as depth estimation (Cao, Wu, and Shen 2017; Fu et al. 2018), age estimation (Rothe, Timofte, and Van Gool 2015; Shah et al. 2024), and crowd counting (Xiong and Yao 2022; Guo et al. 2024).

In this work, we reformulate time series forecasting as a classification problem. Specifically, we tokenize time series values into different categories based on their magnitude and leverage the cross-entropy loss to train a classifier on these

*Corresponding author.

tokenized values. For example, in Figure 1b, we employ quantization to convert the real values into four discrete intervals, where each interval is considered a separate class. In this way, we can generate predictions within the corresponding interval based on the output of the classifier.

However, the continuous nature of time series data makes it challenging to classify values near the inter-class boundaries accurately. This difficulty may result in sub-optimal relative improvements, as illustrated by the blue circle in Figure 1b, a phenomenon commonly referred to as the *boundary effects* (Liu, Zhang, and Duan 2020).

Therefore, we propose **H**ierarchical **C**lassification **A**uxiliary **N**etworks (**HCAN**), a novel model-agnostic component that can be integrated with any forecasting model. The architecture of HCAN is illustrated in Figure 2. In specific, we develop a Hierarchy-Aware Attention (HAA) module to incorporate multi-granular high-entropy features into the main features generated by the encoder network. For each hierarchy level, we propose an Uncertainty-Aware Classifier (UAC), combined with the evidence theory to mitigate the overconfident predictions and enhance the reliability of the features. Last, we propose a Hierarchical Consistency Loss (HCL) to ensure consistency of predicted values between hierarchies. In summary, our contributions are as follows:

- We reformulate forecasting as a hierarchical classification problem to introduce high-entropy feature representations, which helps to reduce over-smooth predictions.
- We propose HCAN, a hierarchy-aware attention module supported by uncertainty-aware classifiers and a consistency loss to alleviate issues caused by the boundary effects during the classification of timesteps.
- Extensive experiments conducted on real-world datasets show the effectiveness of integrating HCAN with various state-of-the-art methods.

Related Work

Time Series Forecasting

With the increased data availability and computational power, deep learning-based models have become an efficient solution to time series forecasting task (Qiu et al. 2024). In overall, based on the underlying network architecture, they can be categorized into models based on Recurrent neural networks (RNNs), Convolutional neural networks (CNNs), Transformer, and multi-layer perceptron (MLP). RNNs are traditionally utilized to capture temporal dependencies, yet they suffer from gradient vanishing and exploding problems. In addition, besides the sequential data processing, RNNs have short-term memory and may not be efficient in learning long-term dependencies. To overcome the limitations of RNNs, Transformer-based models have excelled recently (Zhou et al. 2021, 2022; Yu et al. 2023; Liu et al. 2023). Unlike RNNs, Transformers can process entire sequences simultaneously, benefiting from the parallel computations. In addition, Transformers handle long-range dependencies more effectively than RNNs (Nie et al. 2022).

On the other hand, recent studies have leveraged the robust abilities of CNNs to capture short-term patterns while attempting to enhance their capabilities for recognizing long-range dependencies (Liu et al. 2022; Eldele et al. 2024). Lastly, the recent development of MLP-based models has resulted in good performance with simple architectures (Zeng et al. 2023; Xu, Zeng, and Xu 2024).

Despite these advancements, these methods still struggle with capturing high-entropy feature representations due to their reliance on the MSE loss, which often leads to over-smooth predictions (Zhang et al. 2023). Differently, our proposed work aims to overcome this limitation and construct a complex and high-entropy feature space, thereby enhancing feature diversity and improving prediction accuracy.

Classification for Continuous Targets

Our approach draws inspiration from successful applications of classification in other domains, such as computer vision and pose estimation, where discretizing continuous targets has led to significant improvements (Rabanser et al. 2020; Gu, Yang, and Yao 2021). For instance, in-depth estimation tasks, classifying depth ranges has proven more effective than precise value prediction (Cao, Wu, and Shen 2017; Fu et al. 2018).

In the context of time series analysis, some recent works have explored limited categorization schemes. For example, DEMM (Wilson et al. 2022) and DEMMA (Wang and Gao 2023) propose frameworks that segment time series into three broad categories. Similarly, NEC+ (Li, Xu, and Anastasiu 2023) employs binary classification to distinguish between extreme and normal events.

Our work significantly extends and refines these initial explorations by introducing a comprehensive, multi-level classification framework specifically designed for time series forecasting. This novel approach achieves a balance between the simplification benefits of discretization and the need for nuanced, continuous predictions. In addition, it addresses key limitations of previous methods, such as the loss of granularity in predictions and the occurrence of boundary effects near class thresholds.

Methodology

Preliminaries

Given the historical time series data $X = \{x^i\}_{i=1}^N$ with N samples, where $x^i \in \mathbb{R}^{L\times D}$, the goal of time series forecasting is to predict horizon series $Y = \{y^i\}_{i=1}^N$, where $y^i \in \mathbb{R}^{T\times D}$. Here, L is the look-back window, T is the number of future timesteps, and D refers to the number of channels in the multivariate time series.

HCAN reformulates the forecasting task as a hierarchical classification task with 3 levels: the original series, coarse, and fine-grained. The number of categories at each level is $K_o = 1$, $K_c = 2$, $K_f = 4$. At each level, a discretizing mapping function converts the continuous target y^i into a categorical target k^i based on which interval $\mathcal{I}_k = (\rho_k^{\text{left}}, \rho_k^{\text{right}})$ the value y^i falls into. This interval $\mathcal{I}_k$ represents the range within which y^i is categorized. The detailed mapping process can be found in the Appendix. Subsequently, the relative forecasting target $\Delta y^i = y^i - \rho_k^{\text{left}}$ is computed as the offset

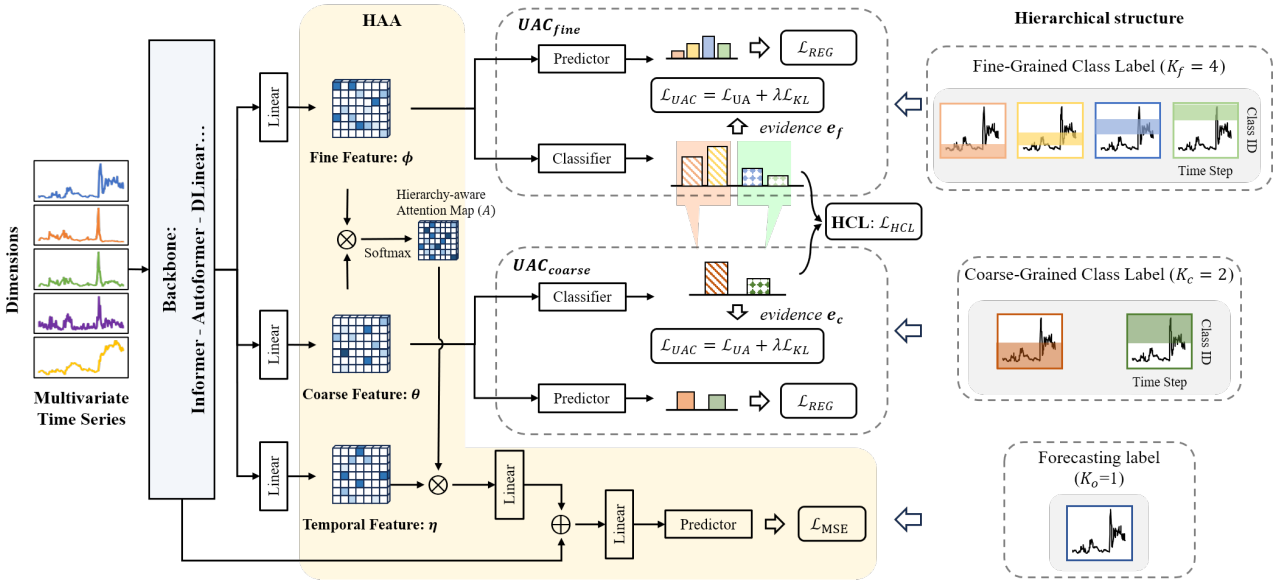


Figure 2: The structure of our proposed HCAN. From right to left, time series are first divided into fine-grained classes and coarse-grained classes to form category labels for *Hierarchical Classification*. According to these category labels, the *Uncertainty-Aware Classifier* (UAC) at each level obtains reliable multi-granularity high-entropy features using evidence theory. The *Hierarchical Consistency Loss* (HCL) ensures the consistency of values between hierarchies. Finally, the *Hierarchy-Aware Attention* (HAA) module integrated the multi-granularity features into the forecasting features obtained by the backbones.

of y^i from the lower bound ρ_k^{left} of the interval $\mathcal{I}_k$, where $\Delta y^i \in \mathbb{R}^{T\times D}$. Therefore, the new structure of the dataset becomes $D = \{x^i, y^i, \Delta y_c^i, k_c^i, \Delta y_f^i, k_f^i\}_{i=1}^N$ with N samples.

Hierarchical Classification Auxiliary Network

We propose a hierarchical structure that trains classifiers at the fine-grained and coarse-grained levels, each with a different number of classes, to obtain high-entropy features represented in multiple granularities. Specifically, the fine-grained feature is obtained from the hierarchy, which has a larger number of categories, providing the model with relatively precise quantification. Conversely, the coarse-grained feature, which corresponds to a hierarchy with fewer categories, aims to enhance classification accuracy, as shown in Figure 2.

To illustrate the workflow of our HCAN, we begin by extracting features $F \in \mathbb{R}^{D\times T}$ from the backbone model. Subsequently, we employ three distinct linear layers to generate three types of features: $\theta \in \mathbb{R}^{D\times M}$, $\phi \in \mathbb{R}^{D\times M}$, and $\eta \in \mathbb{R}^{D\times M}$, representing fine-grained, coarse-grained, and the original temporal features, respectively. Meanwhile, as depicted in the right-most part of Figure 2, we categorize the timesteps into fine-grained and coarse-grained classes based on their magnitude. Specifically, we define the boundary of each group by arranging the time series values in an ascending order and then dividing them based on the number of groups K (see the Appendix). This categorization forms a hierarchical structure and establishes the category labels.

These hierarchical categories are used as labels to train the Uncertainty-Aware Classifiers (UAC) at the coarse-grained and fine-grained levels. Through backpropagation, the UAC refines the features θ and ϕ, transforming them into high-entropy feature representations. The temporal feature η is tailored to capture the temporal characteristics of time series forecasting. Furthermore, we implement the Hierarchical Consistency Loss (HCL) to maintain consistency between the coarse-grained and fine-grained levels and to mitigate boundary effects. Finally, we combine θ, ϕ, and η with the initial forecasting features F through the Hierarchy-Aware Attention (HAA) module. In the subsequent sections, we provide a detailed description of these components.

Uncertainty-Aware Classification In our HCAN, we include a classifier at the coarse-grained and fine-grained levels to create the high entropy features. However, a key challenge is the high confidence often erroneously assigned to incorrect predictions by traditional softmax-based classifiers (Moon et al. 2020; Van Amersfoort et al. 2020). This issue becomes more obvious given our objective of classifying timesteps-level values into distinct classes. To address this issue and improve the robustness of classification across various hierarchical levels, we implement an evidence-based uncertainty estimation technique, which is meant to enhance the precision of uncertainty assessments. Moreover, we consider the case of challenging samples that are usually estimated with high uncertainty by the Evidential Deep Learning (EDL) methods (Han et al. 2022). To prioritize these samples, we propose a novel uncertainty-aware loss function. This loss increases the importance of these challenging samples in the learning process. Essentially, if the sample is hard to classify, it helps the model recognize its difficulty and pays more attention to it.

Our approach utilizes an evidence-based uncertainty estimation technique, leveraging the parameters of the Dirichlet

distribution, which is the conjugate prior of the categorical distribution. This method allows us to compute belief masses (b) for different categories and the overall uncertainty mass (u), derived from the evidence (e) collected from the data.

For the K-class classification problems, the softmax layer of a conventional neural network classifier is replaced with an activation function layer (*i.e.,* Softplus) to ensure non-negative outputs, which are then treated as evidence vectors $e \in \mathbb{R}_+^K$. These vectors are obtained by the classifier network based on the fine-grained feature θ or coarse-grained feature ϕ. Next, we use these evidence vectors to construct the parameters of the Dirichlet distribution, $i.e.$, $\alpha = e + 1$, and calculate the belief mass b_k and uncertainty u as:

$$b_k = \frac{e_k}{S} = \frac{\alpha_k - 1}{S} \quad \text{and} \quad u = \frac{K}{S}, \tag{1}$$

where $S = \sum_{i=1}^{K}(e_i+1) = \sum_{i=1}^{K}\alpha_i$ represents the Dirichlet strength. In addition, the sum of uncertainty mass u and belief mass b equals 1, $u + \sum_{k=1}^{K} b_k = 1$, where $u \geqslant 0$ and $b \geqslant 0$. Finally, the probability distribution p is calculated as $p_k = \frac{\alpha_k}{S}$.

According to Eq. 1, the more evidence observed for the k-th class, the greater the probability allocated to the k-th class. Conversely, the less total evidence observed, the greater the overall uncertainty. Therefore, we use the belief mass to calculate the class uncertainty for each instance. Specifically, for the i-th sample, we use $(1 - b^i)$ as class-level uncertainty, which is the uncertainty weight for categories during training. We define the uncertainty-aware (UA) coefficient as: $\omega^i = (1 - b^i) \bigodot o^i$, where $\bigodot$ means the Hadamard product.

Finally, the UAC loss is defined as:

$$\begin{aligned}\mathcal{L}_{UAC} &= \lambda_{UA}\mathcal{L}^i_{UA} + \lambda_{KL}\mathcal{L}^i_{KL} \\ &= \lambda_{UA}\sum_{k=1}^{K}\omega^i_k(\psi(S^i) - \psi(\alpha^i_k)) \\ &+ \lambda_{KL}KL[Dir(p^i|\widetilde{\alpha}^i)||Dir(p^i|1)],\end{aligned} \tag{2}$$

where $\psi(\cdot)$ is the digamma function, and $\lambda_{UA}, \lambda_{KL}$ are balance factors, and $\text{Dir}(p^i|1)$ approximates the uniform distribution. Notably, we make adjustments to the Dirichlet parameters α^i by $\widetilde{\alpha}^i = o^i + (1 - o^i) \bigodot \alpha^i$ to remove the non-misleading evidence.

By formalizing forecasting as a classification task, we introduce high entropy features into the forecasting feature space. At the same time, to encourage the continuity of the extracted features, we propose a relative prediction strategy, making predictions within each classification bin (Yu et al. 2021). We optimize using the MSE loss against the ground truth forecasting interval:

$$\mathcal{L}_{REG} = \sum_{k=1}^{K}\mathbb{I}(c_k = 1)(\Delta y_k - \Delta\hat{y}_k)^2, \tag{3}$$

where c_k and Δy_k denote the classification and relative prediction labels, respectively, and $\Delta\hat{y}_k$ is the relative prediction value obtained by the model.

The hierarchy loss is formulated across two layers with varying granularity as:

$$\mathcal{L}_{HIER} = \mathcal{L}^f_{UAC} + \alpha\mathcal{L}^f_{REG} + \mathcal{L}^c_{UAC} + \alpha\mathcal{L}^c_{REG}, \tag{4}$$

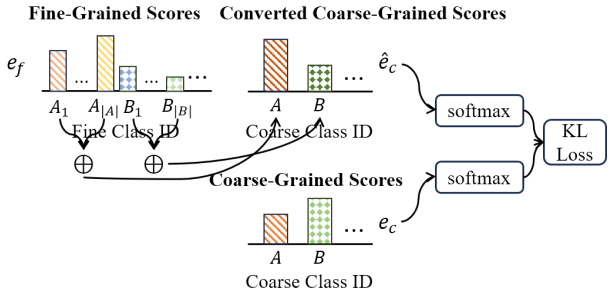


Figure 3: The hierarchical consistency loss between fine-grained and coarse-grained hierarchies encourages consistent predictions among them, alleviating the boundary effects. The e_f from the fine-grained classifier is converted to $\hat{e}_c$, which aligns with the coarse-grained classifier e_c. We minimize the KL divergence loss between their softmax outputs.

where α is the balance factor.

Hierarchical Consistency Loss Due to the continuous nature of time series data, directly classifying timestep values may result in misclassified values near the inter-class boundaries, known as *boundary effects*. Therefore, we propose the Hierarchical Consistency Loss (HCL), which aims to keep the values near the boundary of a fine-grained class within the correct coarse-grained category.

To reinforce this alignment between the hierarchical classifiers, we propose an HCL to penalize discrepancies between them. As illustrated in Figure 3, we minimize a symmetric version of the Kullback-Leibler (KL) divergence between the class distributions of the fine-grained and coarse-grained classifiers.

For each fine-grained category, represented by evidence $e_f = [e_f^{A_1}, ..., e_f^{A_{|A|}}, e_f^{B_1}, ..., e_f^{B_{|B|}}, ...]$, we first convert it to a coarse-grained category evidence $e_c = [e_c^A, e_c^B, ...]$. To align e_f and e_c, we average the e_f values that belong to the same coarse-grained class to produce the converted coarse-grained evidence:

$$\begin{aligned}\hat{e}_c &= [\hat{e}_c^A, \hat{e}_c^B, \dots] \\ &= [\frac{e_f^{A_1} + ... + e_f^{A_{|A|}}}{|A|}, \frac{e_f^{B_1} + ... + e_f^{B_{|B|}}}{|B|}, ...].\end{aligned} \tag{5}$$

The consistency loss for each coarse-grained class is then defined as a symmetric version of the KL divergence (equivalent to the Jensen-Shannon divergence) between e and $\hat{e}$:

$$\mathcal{L}_{HCL} = \frac{1}{2}D_{KL}(e_c||\hat{e}_c) + \frac{1}{2}D_{KL}(\hat{e}_c||e_c). \tag{6}$$

This approach ensures that our model's predictions remain consistent across different hierarchical levels, effectively alleviating boundary effects.

Hierarchy-Aware Attention To introduce the high-entropy feature into the forecasting features to alleviate the over-smooth predictions, and optimize the trade-off between fore-

Model	Informer	+HCAN	Autoformer	+HCAN	PatchTST	+HCAN	SCINet	+HCAN	Dlinear	+HCAN	iTransformer	+HCAN	FITS	+HCAN
Metric	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE
ETTh1	1.077	**0.897**	0.530	**0.462**	0.421	**0.396**	0.591	**0.536**	0.453	**0.428**	0.457	**0.451**	0.439	**0.436**
ETTh2	4.779	**2.359**	0.483	**0.406**	**0.342**	0.343	1.041	**0.820**	0.473	**0.411**	0.384	**0.375**	0.375	**0.368**
ETTm1	0.951	**0.717**	0.606	**0.540**	0.353	**0.350**	0.417	**0.390**	0.359	**0.344**	0.408	**0.403**	0.414	**0.405**
ETTm2	1.729	**0.981**	0.359	**0.303**	0.258	**0.250**	0.753	**0.685**	**0.287**	0.296	0.292	**0.285**	0.286	**0.280**
Weather	0.733	**0.370**	0.351	**0.303**	0.268	**0.254**	0.242	**0.225**	0.247	**0.237**	0.260	**0.250**	0.249	**0.248**
Exchange	1.726	**0.845**	0.525	**0.410**	0.516	**0.344**	0.844	**0.549**	0.369	**0.338**	**0.364**	0.395	**0.360**	0.426
ILI	2.889	**2.738**	5.012	**4.166**	1.516	**1.428**	3.277	**3.265**	2.347	**2.276**	2.767	**2.741**	3.680	**2.095**
Electricity	0.352	**0.337**	0.250	**0.236**	0.259	**0.233**	0.213	**0.209**	0.210	**0.208**	0.176	**0.167**	0.217	**0.216**
Traffic	0.853	**0.818**	0.651	**0.552**	0.490	**0.460**	0.612	**0.527**	0.625	**0.597**	0.422	**0.416**	0.642	**0.624**
Solar Wind	1.953	**1.025**	1.362	**1.057**	1.109	**0.948**	1.174	**1.091**	1.071	**1.019**	1.360	**1.028**	1.349	**1.239**

Table 1: Multivariate long sequence time-series forecasting results. We report the MSE of different prediction lengths. The look-up window is set to $L = 336$ for PatchTST, DLinear, and SCINet, and $L = 96$ for other models. The **best results** are highlighted in **bold**. Detailed results of all prediction lengths for MSE/MAE are provided in the Appendix.

casting features and high-entropy features at different granularities, we have developed the Hierarchy-Aware Attention (HAA) module.

Building on the feature architecture of Hierarchical Classification Auxiliary Network, we reshape $\phi \in \mathbb{R}^{H \times D}$ projections, allowing their dot-products to interact and generate the HAA map A of size $\mathbb{R}^{D \times D}$. This is combined with F through a residual connection to introduce high-entropy feature representations. The overall HAA process is defined as follows:

$$\begin{aligned} \hat{Y} &= W_f(W \cdot \text{Attention}(\theta, \phi, \eta) + F) + b, \\ \text{Attention}(\theta, \phi, \eta) &= \eta \cdot \text{Softmax}(\theta \cdot \phi), \end{aligned} \tag{7}$$

where W and W_f are linear layers, F is the backbone feature map, and $\hat{Y}$ is the prediction output. The MSE loss is optimized according to $\hat{Y}$ and the ground truth labels Y as:

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (Y^i - \hat{Y}^i)^2, \tag{8}$$

where N represents the number of samples.

To sum up, the overall training loss is defined as:

$$\mathcal{L} = \mathcal{L}_{HIER} + \beta \mathcal{L}_{HCL} + \gamma \mathcal{L}_{MSE}, \tag{9}$$

where β and γ are hyper-parameter loss weights chosen through grid search.

Experiments

In this section, we conduct extensive experiments to evaluate the performance of HCAN and further perform ablation studies to justify how each component contributes to the results. Further details about the experimental setup can be found in the Appendix.

Experimental Settings

Datasets. We ran our experiments on ten publicly available real-world multivariate time series datasets, namely: *ETT*, *Exchange-Rate*, *Weather*, *ILI*, *Electricity*, *Traffic*, and *Solar Wind*. We followed the standard protocol in the data preprocessing, where we split all datasets into training, validation, and testing in chronological order by a ratio of 6:2:2 for the ETT dataset and 7:1:2 for the other datasets (Zeng et al. 2023). See the Appendix for more details.

Backbone models. We experimented our HCAN on top of several state-of-the-art deep learning-based forecasting models. We selected these models with different architectures, where Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), PatchTST (Nie et al. 2022), and iTransformer (Liu et al. 2023) are Transformer-based models, SCINet (Liu et al. 2022) is a CNN-based model, while DLinear (Zeng et al. 2023) and FITS (Xu, Zeng, and Xu 2024) are MLP-based models. We evaluate their performance before and after including our HCAN in the multivariate and univariate settings. For the baselines, we re-run their codes in the same settings to ensure fairness and consistency.

Experiments details. Following previous works (Nie et al. 2022; Zeng et al. 2023), we used ADAM (Kingma and Ba 2014) as the default optimizer across all the experiments and reported the MSE and mean absolute error (MAE) as the evaluation metrics. A lower MSE/MAE value indicates a better performance. Detailed results for MSE/MAE are provided in the Appendix. We conducted the experiment for the same number of epochs as the baseline and the initial learning rate is chosen from {5e-3, 1e-3, 5e-4, 1e-4, 5e-5, 1e-5} through a grid search for different datasets. β was chosen from {1, 0.1, 0.01} and γ was chosen from {1, 0.1, 0.01} via grid search to obtain the best results. For HCAN parameters, we set $K_c = 2$ and $K_f = 4$. All the experiments were repeated five times with fixed random seeds, and we reported the average performance. HCAN was implemented by PyTorch (Paszke et al. 2019) and trained on a single NVIDIA RTX 3090 24GB GPU.

Main Results

Multivariate Forecasting Results. We present the multivariate forecasting results in Table 1. Notably, our proposed HCAN demonstrates a substantial impact on the performance of the baselines, as it boosts their forecasting results by a noticeable margin. This is evident in 66 out of 70 cases. For instance, HCAN achieves average performance gains of 9.1%, 35.5%, 10.2%, and 22.3% on the ETT dataset series. Similar improvements also observed on other datasets.

We attribute these performance enhancements to two primary aspects. First, HCAN incorporates a reliable hierarchi-

Model		Informer	+HCAN	Autoformer	+HCAN	PatchTST	+HCAN	SCINet	+HCAN	Dlinear	+HCAN	iTransformer	+HCAN	FITS	+HCAN
Metric		MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE
ETTh1	96	0.255	**0.121**	0.088	**0.082**	0.055	**0.055**	0.088	**0.068**	0.057	**0.053**	0.061	**0.060**	0.056	**0.054**
	192	0.283	**0.092**	0.108	**0.086**	**0.071**	0.072	0.105	**0.084**	0.077	**0.075**	0.073	**0.072**	0.075	**0.072**
	336	0.291	**0.088**	0.118	**0.091**	0.082	**0.078**	0.130	**0.094**	0.097	**0.088**	0.089	**0.087**	0.091	**0.089**
	720	0.256	**0.106**	0.138	**0.121**	0.086	**0.081**	0.214	**0.134**	0.168	**0.164**	**0.083**	0.105	0.104	**0.096**
ETTh2	96	0.302	**0.182**	0.169	**0.140**	0.129	**0.127**	0.130	**0.129**	0.133	**0.128**	0.135	**0.133**	0.125	**0.123**
	192	0.264	**0.206**	0.211	**0.179**	0.169	**0.162**	0.327	**0.169**	0.177	**0.174**	0.182	**0.178**	0.177	**0.174**
	336	0.324	**0.223**	0.255	**0.226**	0.187	**0.187**	**0.198**	0.220	**0.212**	0.225	0.218	**0.215**	0.222	**0.221**
	720	0.302	**0.249**	0.334	**0.292**	0.224	**0.201**	0.486	**0.221**	0.298	**0.259**	0.240	**0.238**	0.258	**0.255**
ETTm1	96	0.093	**0.046**	0.059	**0.047**	0.026	**0.024**	0.049	**0.029**	0.030	**0.026**	0.029	**0.028**	0.029	**0.027**
	192	0.232	**0.059**	0.081	**0.057**	0.039	**0.037**	0.077	**0.049**	0.044	**0.043**	0.049	**0.045**	0.043	**0.042**
	336	0.271	**0.108**	0.088	**0.072**	0.053	**0.050**	0.109	**0.089**	0.064	**0.059**	0.061	**0.060**	0.057	**0.056**
	720	0.464	**0.118**	0.122	**0.079**	0.074	**0.070**	0.139	**0.117**	**0.081**	0.082	0.083	**0.082**	0.079	**0.075**
ETTm2	96	0.092	**0.065**	0.127	**0.095**	0.065	**0.065**	0.079	**0.069**	0.064	**0.061**	0.069	**0.069**	0.070	**0.069**
	192	0.134	**0.107**	0.146	**0.123**	0.094	**0.091**	0.105	**0.094**	0.092	**0.087**	0.107	**0.106**	0.100	**0.098**
	336	0.178	**0.141**	0.217	**0.126**	0.120	**0.117**	0.130	**0.128**	0.129	**0.120**	0.144	**0.143**	0.128	**0.126**
	720	0.221	**0.156**	0.198	**0.184**	0.172	**0.169**	0.175	**0.155**	**0.176**	0.181	**0.185**	0.187	0.178	**0.176**
Solar Wind	96	1.443	**1.268**	2.316	**1.289**	1.021	**0.851**	1.518	**1.366**	1.316	**1.223**	1.727	**1.266**	1.669	**1.658**
	192	1.765	**1.581**	2.765	**1.590**	1.130	**1.030**	1.836	**1.723**	1.568	**1.549**	2.273	**1.568**	2.308	**2.280**
	336	1.849	**1.740**	2.783	**1.715**	1.137	**1.098**	1.853	**1.746**	1.686	**1.671**	2.370	**1.714**	2.355	**2.327**
	720	1.826	**1.694**	2.606	**1.701**	1.125	**1.041**	1.672	**1.547**	1.660	**1.654**	2.228	**1.679**	2.220	**2.189**

Table 2: Univariate long sequence time-series forecasting results on ETT full benchmark and Solar Wind dataset. We report the MSE of different prediction lengths $T \in \{96, 192, 336, 720\}$ for comparison. The look-up window is set to $L = 336$ for PatchTST, DLinear, and SCINet, and $L = 96$ for other models. The **best results** are highlighted in **bold**. Detailed results of all prediction lengths for MSE/MAE are provided in the Appendix.

Component					Weather				Solar Wind			
$\frac{UAC_{fine}}{\mathcal{L}_{UAC}}$	$\frac{}{\mathcal{L}_{REG}}$	Hierarchy	$\mathcal{L}_{HCL}$	HAA	96	192	336	720	96	192	336	720
-	-	-	-	-	0.352	0.636	0.680	1.265	1.710	1.991	1.958	2.154
✓	-	-	-	-	0.349	0.509	0.613	0.993	0.991	1.077	1.127	1.149
✓	✓	-	-	-	0.300	0.515	0.579	0.999	0.964	1.060	1.129	1.125
✓	✓	✓	-	-	0.322	0.406	0.580	0.961	0.948	1.048	1.099	1.109
✓	✓	✓	✓	-	0.295	0.345	0.395	0.614	0.935	1.038	1.097	1.083
✓	✓	✓	✓	✓	**0.291**	**0.306**	**0.369**	**0.513**	**0.920**	**1.027**	**1.087**	**1.065**

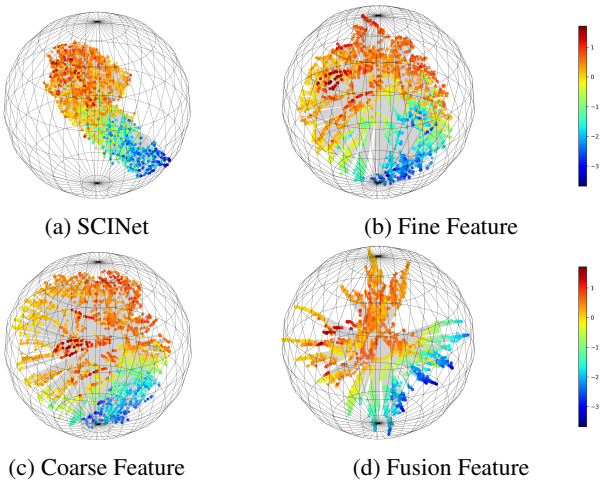
Table 3: Ablation study of the components of HCAN on the Weather and Solar Wind datasets using Informer as a backbone: Uncertainty-Aware Classification (UAC), Hierarchical Structure (Hierarchy), Hierarchical Consistency Loss ($\mathcal{L}_{HCL}$), and Hierarchy-Aware Attention (HAA). The results are in terms of MSE for different prediction lengths. The **best results** are highlighted in **bold.**

cal classification structure that captures high-entropy features, effectively alleviating the over-smooth predictions and reducing the boundary discontinuity typically associated with classification tasks. Second, the HAA mechanism enhances prediction accuracy by fusing features at different granular levels, thereby providing more reliable information for prediction. This attribute proves particularly advantageous in long-term forecasting scenarios, which inherently pose greater challenges as the forecast horizon extends. For example, as shown in the Appendix, when forecasting a length of 720 timesteps, the integration of HCAN with Autoformer leads to a significant reduction of 31.9% in MSE on the ETTh2 dataset and a reduction of 19.3% on the Exchange dataset. These results underscore the capability of HCAN to deliver stable and reliable predictions even in long-term forecasting scenarios.

Univariate Forecasting Results. We also report the univariate forecasting outcomes for the ETT and Solar Wind datasets in Table 2. Compared to the original performance of the baseline methods, incorporating our HCAN into these models yields an overall reduction of 23.0%, 35.8%, 7.5%, 12.6%, 2.5%, 22.8%, and 1.5% in the MSE results. These results validate the effectiveness of our proposed hierarchical structure in enhancing forecasting precision.

Ablation Study

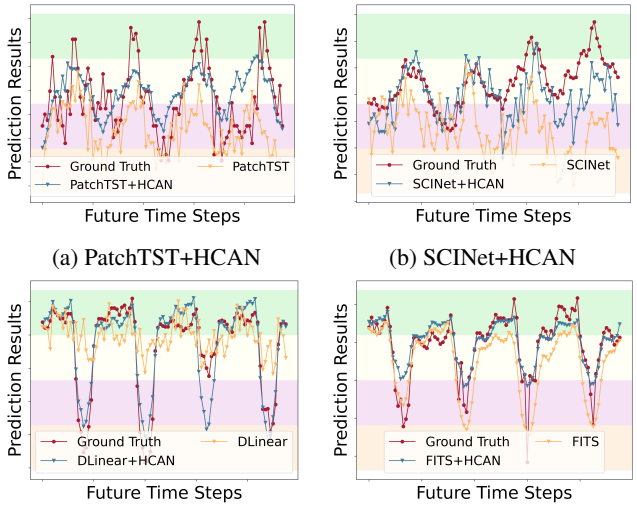
Table 3 presents an ablation study on the Weather and Solar Wind datasets to assess the effectiveness of each module in HCAN. Referring to Figure 2, we evaluate the following settings: **(1)** including the UAC with only the fine-grained classes ($\mathcal{L}_{UAC}$ alone) **(2)** with adding $\mathcal{L}_{REG}$ to the UAC module, i.e., $\mathcal{L}_{UAC} + \mathcal{L}_{REG}$ **(3)** with including the coarse-



(a) SCINet (b) Fine Feature

(c) Coarse Feature (d) Fusion Feature

Figure 4: t-SNE visualization of different features for SCINet on the ETTh1 dataset. (a) SCINet keeps features close together. (b)(c) Simply introducing classification spreads the features, obtaining a higher entropy feature space, while the ordinal relationship is lost. (d) By combining the classification features with the forecasting features, a high entropy and ordered feature representation is obtained. Features are coloured based on their predicted value.



(a) PatchTST+HCAN (b) SCINet+HCAN

(c) DLinear+HCAN (d) FITS+HCAN

Figure 5: The prediction results (Horizon = 96) of (a) PatchTST vs. PatchTST+HCAN, (b) SCINet vs. SCINet+HCAN, (c) DLinear vs. DLinear+HCAN, (d) FITS vs. FITS+HCAN, on randomly-selected sequences from the ETTh1 dataset.

grained classes and directly concatenating the multi-level features (Hierarchy) **(4)** with using $\mathcal{L}_{HCL}$ to keep consistency among hierarchy levels **(5)** with using the attention module for feature fusion instead of direct concatenation.

Impact of UAC. Initially, applying the UAC on the fine-grained features alone with $\mathcal{L}_{UAC}$ significantly enhances performance by creating a high-entropy feature space that enriches forecasting representations. Adding $\mathcal{L}_{REG}$ further improves performance by imposing relative forecasting constraints, ensuring feature continuity and coherence.

Impact of Hierarchy Structure. Implementing a hierarchical structure with two layers of UAC layers (by including the coarse-grained features) demonstrates the value of incorporating multi-granularity features, as indicated by performance gains in the ablation study.

Impact of HCL. Performance is further enhanced by integrating $\mathcal{L}_{HCL}$, which imposes a consistency constraint between hierarchies and effectively addresses boundary effects.

Impact of HAA. The best performance is observed when replacing direct concatenation with the HAA mechanism. This change indicates that different features contribute variably to forecasting outcomes, and simple concatenation can lead to sub-optimal results.

Qualitative Evaluation

High-entropy Feature Representation. The t-SNE visualization of the features from SCINet on the ETTh1 dataset is displayed in Figure 4. As depicted in Figure 4a, representations learned from the MSE loss exhibit lower diversity. Figures 4b and 4c illustrate that integrating classification indeed spreads features more broadly, yet it disrupts ordinality in feature space. Figure 4d shows how the HAA mechanism combines hierarchical features with the original features from the backbone model, effectively spreading the feature while maintaining ordinality. In conclusion, HCAN facilitates reliable high-entropy feature representations through hierarchical classification, significantly helping to alleviate over-smooth predictions.

Visualizations. To examine the quality of prediction results with and without our HCAN, Figure 5 presents this comparison on PatchTST, SCINe, DLinear, and FITS backbones on the ETTh1 dataset. Clearly, our HCAN yields more realistic predictions. This enhancement is largely regarded to the proposed hierarchical consistency loss (HCL), which notably improves performance at class boundaries. These results further validate the effectiveness of the high-entropy feature representations. Additionally, they demonstrate that HCL is effective in mitigating the boundary effects.

Conclusion

In this study, we addressed the issue of over-smooth predictions in time series forecasting by introducing a novel hierarchical classification from an entropy perspective. We proposed HCAN, a model-agnostic component that enhances forecasting by tokenizing output and integrating mutigranularity high-entropy feature representations through a hierarchical-aware attention module. The HCL loss further aids in mitigating boundary effects, promoting overall accuracy. Extensive experiments on benchmarking datasets demonstrate that HCAN substantially improves the performance of baseline forecasting models. Our results suggest that HCAN can serve as a foundation component in time series forecasting, providing deeper insights into the interplay between classification tasks and forecasting.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62376194, 61925602, U23B2049, 62406219, and 62436001 and in part by the China Postdoctoral Science Foundation - Tianjin Joint Support Program under Grant 2023T014TJ.

References

Cao, Y.; Wu, Z.; and Shen, C. 2017. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11): 3174–3182.

Chen, H.; Rossi, R. A.; Mahadik, K.; Kim, S.; and Eldardiry, H. 2021. Graph deep factors for forecasting with applications to cloud resource allocation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 106–116.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; and Li, X. 2024. TSLANet: Rethinking Transformers for Time Series Representation Learning. In *International Conference on Machine Learning*.

Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; and Tao, D. 2018. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2002–2011.

Gu, K.; Yang, L.; and Yao, A. 2021. Dive deeper into integral pose regression. In *International Conference on Learning Representations*.

Guo, Q.; Yuan, P.; Huang, X.; and Ye, Y. 2024. Consistency-constrained RGB-T crowd counting via mutual information maximization. *Complex & Intelligent Systems*, 1–22.

Han, Z.; Zhang, C.; Fu, H.; and Zhou, J. T. 2022. Trusted multi-view classification with dynamic evidential fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(2): 2551–2566.

Hou, M.; Xu, C.; Li, Z.; Liu, Y.; Liu, W.; Chen, E.; and Bian, J. 2022. Multi-Granularity Residual Learning with Confidence Estimation for Time Series Prediction. In *Proceedings of the ACM Web Conference 2022*, 112–121.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lam, R.; Sanchez-Gonzalez, A.; Willson, M.; Wirnsberger, P.; Fortunato, M.; Pritzel, A.; Ravuri, S.; Ewalds, T.; Alet, F.; Eaton-Rosen, Z.; et al. 2022. GraphCast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*.

Li, Y.; Xu, J.; and Anastasiu, D. C. 2023. An extreme-adaptive time series prediction model based on probability-enhanced lstm neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8684–8691.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, N.; Zhang, F.; and Duan, F. 2020. Facial age estimation using a multi-task network combining classification and regression. *IEEE Access*, 8: 92441–92451.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Moon, J.; Kim, J.; Shin, Y.; and Hwang, S. 2020. Confidence-aware learning for deep neural networks. In *international conference on machine learning*, 7034–7044. PMLR.

Ni, Z.; Yu, H.; Liu, S.; Li, J.; and Lin, W. 2024. Basisformer: Attention-based time series forecasting with learnable and interpretable basis. *Advances in Neural Information Processing Systems*, 36.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pintea, S. L.; Lin, Y.; Dijkstra, J.; and van Gemert, J. C. 2023. A step towards understanding why classification helps regression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19972–19981.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; et al. 2024. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*.

Rabanser, S.; Januschowski, T.; Flunkert, V.; Salinas, D.; and Gasthaus, J. 2020. The effectiveness of discretization in forecasting: An empirical study on neural time series models. *arXiv preprint arXiv:2005.10111*.

Rothe, R.; Timofte, R.; and Van Gool, L. 2015. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE international conference on computer vision workshops*, 10–15.

Shah, J.; Siddiquee, M. M. R.; Su, Y.; Wu, T.; and Li, B. 2024. Ordinal Classification with Distance Regularization for Robust Brain Age Prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 7882–7891.

Shao, Z.; Wang, F.; Xu, Y.; Wei, W.; Yu, C.; Zhang, Z.; Yao, D.; Sun, T.; Jin, G.; Cao, X.; et al. 2024. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Van Amersfoort, J.; Smith, L.; Teh, Y. W.; and Gal, Y. 2020. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, 9690–9700. PMLR.

Wang, J.; and Gao, Y. 2023. Generalized Mixture Model for Extreme Events Forecasting in Time Series Data. *arXiv preprint arXiv:2310.07435*.

Wang, X.; Zhou, T.; Wen, Q.; Gao, J.; Ding, B.; and Jin, R. 2024. CARD: Channel aligned robust blend transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2023. Transformers in Time Series: A Survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, 6778–6786. ijcai.org.

Wilson, T.; McDonald, A.; Galib, A. H.; Tan, P.-N.; and Luo, L. 2022. Beyond Point Prediction: Capturing Zero-Inflated & Heavy-Tailed Spatiotemporal Data with Deep Extreme Mixture Models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020–2028.

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

Wu, X.; Qiu, X.; Li, Z.; Wang, Y.; Hu, J.; Guo, C.; Xiong, H.; and Yang, B. 2024. CATCH: Channel-Aware multivariate Time Series Anomaly Detection via Frequency Patching. *arXiv preprint arXiv:2410.12261*.

Xiong, H.; and Yao, A. 2022. Discrete-constrained regression for local counting models. In *European Conference on Computer Vision*, 621–636. Springer.

Xu, Z.; Zeng, A.; and Xu, Q. 2024. FITS: Modeling Time Series with \$10k\$ Parameters. In *The Twelfth International Conference on Learning Representations*.

Yu, C.; Wang, F.; Shao, Z.; Sun, T.; Wu, L.; and Xu, Y. 2023. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, 3062–3072.

Yu, X.; Rao, Y.; Zhao, W.; Lu, J.; and Zhou, J. 2021. Group-aware contrastive regression for action quality assessment. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7919–7928.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, S.; Yang, L.; Mi, M. B.; Zheng, X.; and Yao, A. 2023. Improving Deep Regression with Ordinal Entropy. In *The Eleventh International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; 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.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.