
Article:
Li, D., Li, X., Gu, F. et al. (3 more authors) (2023) A universality–distinction mechanism-based multi-step sales forecasting for sales prediction and inventory optimization. Systems, 11 (6). 311. ISSN 2079-8954

https://doi.org/10.3390/systems11060311

---

systems

MDPI

Article

A Universality–Distinction Mechanism-Based Multi-Step Sales Forecasting for Sales Prediction and Inventory Optimization

Daifeng Li [1,*], Xin Li [1], Fengyun Gu [1], Ziyang Pan [1], Dingquan Chen [1] and Andrew Madden [2]

[1] School of Information Management, Sun Yat-sen University, Guangzhou 510275, China; lixin237@mail2.sysu.edu.cn (X.L.); gufy@mail2.sysu.edu.cn (F.G.); panzy25@mail2.sysu.edu.cn (Z.P.); chendq@mail.sysu.edu.cn (D.C.)

[2] Information School, University of Sheffield, Sheffield S10 2TN, UK; admadden@hotmail.com

* Correspondence: lidaifeng@mail.sysu.edu.cn

Abstract: Sales forecasting is a highly practical application of time series prediction. It is used to help enterprises identify and utilize information to reduce costs and maximize profits. For example, in numerous manufacturing enterprises, sales forecasting serves as a key indicator for inventory optimization and directly influences the level of cost savings. However, existing research methods mainly focus on detecting sequences and local correlations from multivariate time series (MTS), but seldom consider modeling the distinct information among the time series within MTS. The prediction accuracy of sales time series is significantly influenced by the dynamic and complex environment, so identifying the distinct signals between different time series within a sales MTS is more important. In order to extract more valuable information from sales series and to enhance the accuracy of sales prediction, we devised a universality–distinction mechanism (UDM) framework that can predict future multi-step sales. Universality represents the instinctive features of sequences and correlation patterns of sales with similar contexts. Distinction corresponds to the fluctuations in a specific time series due to complex or unobserved influencing factors. In the mechanism, a query-sparsity measurement (QSM)-based attention calculation method is proposed to improve the efficiency of the proposed model in processing large-scale sales MTS. In addition, to improve the specific decision-making scenario of inventory optimization and ensure stable accuracy in multi-step prediction, we use a joint Pin-DTW(Pinball loss and Dynamic Time Warping) loss function. Through experiments on the public Cainiao dataset, and via our cooperation with Galanz, we are able to demonstrate the effectiveness and practical value of the model. Compared with the best baseline, the improvements are 57.27%, 50.68%, and 35.26% on the Galanz dataset and 16.58%, 6.07%, and 5.27% on the Cainiao dataset, in terms of the MAE(Mean Absolute Error), MAPE(Mean Absolute Percentage Error), and RMSE(Root Mean Squared Error).

Keywords: time series; sales forecasting; deep learning; multi-step prediction

Citation: Li, D.; Li, X.; Gu, F.; Pan, Z.; Chen, D.; Madden, A. A Universality—Distinction Mechanism-Based Multi-Step Sales Forecasting for Sales Prediction and Inventory Optimization. *Systems* **2023**, *11*, 311. https://doi.org/10.3390/systems11060311

Academic Editor: Vladimír Bureš

Received: 10 April 2023
Revised: 12 June 2023
Accepted: 14 June 2023
Published: 19 June 2023

1. Introduction

Sales forecasting is an area of research with considerable practical significance due to its potential to improve commercial decision-making. However, the influences of external observable and unobservable factors, e.g., the weather, seasonal promotions, adjustments in sales strategies, etc., make forecasting particularly challenging. Such factors cause irregular fluctuations in sales, resulting in large deviations in sales forecasts. In many circumstances, even a slight reduction in such deviations can bring great benefits. For example, the optimization of sales forecasting for high-profit commodities can greatly reduce losses in profit caused by stock shortages. A second area in which there is scope for development, and where there appear to have been few studies, is that of specific enterprise decision-making scenarios, such as inventory optimization. In Galanz's e-commerce business scenario, the average profit per unit of goods surpasses its inventory management

cost. As a result, shortages lead to greater losses than excess inventory, which means that, with the same sales prediction deviation, predicting less than the actual sales may lead to greater losses than predicting more, which is one of the challenges of sales forecasting.

The problem of dealing with complex and diverse influencing factors is generally addressed by using a multi-step time series prediction based on multivariate time series (MTS). The specific models can be divided into traditional time series prediction methods and deep learning models. Figure 1 shows an actual sales sequence, together with the results of multi-step time series predictions made using, respectively, deep learning models (MLCNN [1]) and traditional methods (ES [2]). Both predictions deviate considerably from actual values when there are sudden peaks in sales.

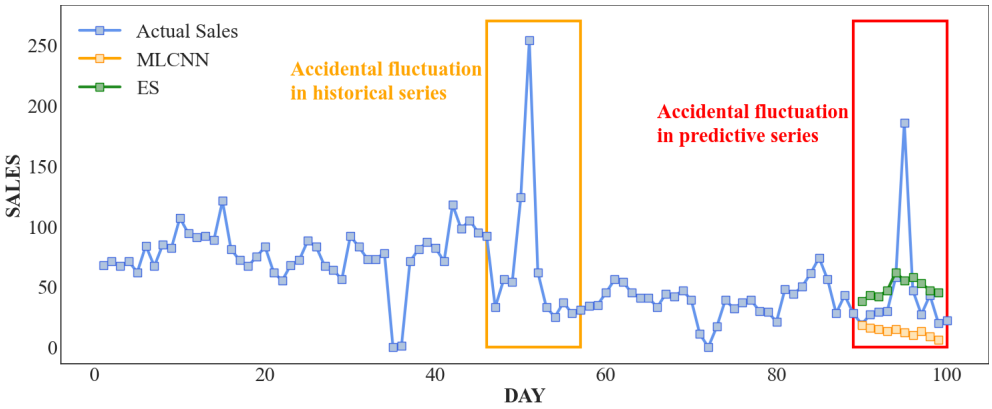

Figure 1. Sales sequence analysis: the blue line represents the sequence of real sales, the green line represents the prediction of the ES model, and the orange line represents MLCNN.

Existing research has proven that deep networks are capable of capturing valuable implicit information, and can be effective in predicting abnormal fluctuations in sales sequences. In the M4 time series forecasting competition, the winning method, [3], adopted a hybrid hierarchical prediction scheme, incorporating the standard exponential smoothing model (ES) into a common framework with long short-term memory (LSTM) networks; this approach resulted in better performance compared to traditional and machine learning methods. Traditional time series analysis methods and machine learning models, such as the autoregressive model (AR) [4], moving average (MA) [5], random forest (RF) [6], and XGBoost (XGB) [7], are widely used in sales forecasting. However, these methods are difficult to use when modeling large-scale MTS. For example, more than 10,000 time series need to be considered at the same time in traffic MTS prediction tasks. Sales forecasting is another complex and computationally intensive MTS task. It needs to consider various factors, including price, preferential strategy, and the sales of related commodities. The accuracy of these methods in capturing nonlinear correlation patterns needs to be improved. As is obvious from Figure 1, the exponential smoothing (ES) model fails to predict the nonlinear changes in a real sales sequence, leading to the accumulation of errors in the multi-step prediction. To some extent, deep learning models can solve the problems that traditional machine learning methods have with MTS prediction. They can fit each sequence independently and share general rules for different sequences. LSTNet [8] makes use of the advantages of CNN(Convolutional Neural Networks) and RNN(Recurrent Neural Networks) to capture the dependency patterns associated with different periods and it shares knowledge in multivariate time series (MTS). Deep learning models are also capable of automatically extracting features to reduce the work of artificial design features. Temporal fusion transformers [9] can automatically select relevant features and suppress unnecessary ones through a series of gating layers. Moreover, deep models improve the efficiency of dealing with long-period and large-scale MTS data [10].

Previous studies that applied deep learning models to fields such as traffic, finance, and industrial production, have shown promise, but there are deviations in the multi-step

sales forecasting tasks. Consequently, deep learning is a key research direction for MTS prediction and is the focus of this paper. However, as is evident from Figure 1, although the deep learning model is more effective at capturing fluctuations than the ES model, it does not accurately predict sudden large fluctuations, which is the first challenge in our research. Ad hoc marketing strategies or other unforeseen factors can lead to unusual fluctuations in sales series. Furthermore, compared to other time series forecasting tasks, such as transportation and finance, one main difference is that sales time series are more susceptible to external uncertainties that can dynamically alter the internal correlations between sales time series. These external uncertainties are critical factors that cannot be predetermined or observed in advance and may result from various causes, such as product reviews on social media, regional promotions, and the introduction of new competitive products. The differences were further verified in previous studies [1,10–12]. Deep learning models used to predict public MTS, such as transportation, energy, exchange rates, etc., often obtain good performances, and the improvements brought about by the new models are limited. The reason for this is that public MTS data exhibit more pronounced correlation patterns compared to sales time series, and these patterns are less influenced by external changes. Thus, the future trends of those MTS could be predicted more easily than those of sales time series.

Currently, few sales forecasting studies are based on specific decision-making scenarios. This is particularly important because the general model may not be directly applicable to specific scenarios, mainly due to issues such as mismatched decision objectives. Specific decision-making scenarios entail more constraints in mathematical expressions. This paper exemplifies the decision-making scenario of inventory optimization and emphasizes the integration of constraints into the deep learning model to achieve end-to-end efficient training and application. As mentioned above, one of the main targets of inventory optimization is to design a constraint that makes the predicted sales greater than the true sales, on the premise of minimizing prediction errors as much as possible, thereby reducing shortage risks. Modeling this constraint is the second challenge of our research since it requires finding an optimal balance between inventory shortages and excesses, considering time delays and abnormal fluctuations.

The two challenges mentioned above not only have theoretical values but also have practical values, which are essential in sales prediction. Many complex and changeable variables in the market will significantly influence the prediction results for products. For example, various online promotional activities will have huge impacts on sales fluctuations. Some products with dozens of daily sales may reach thousands of sales in one day due to temporary promotional activities. Of course, there are also many goods that are not sensitive to price fluctuations and are not significantly affected by external factors. This will make solving the target problem that is to be predicted more complex. Therefore, it is necessary to model and analyze the unique characteristics of each time series separately to better understand the fluctuation patterns of each time series. Due to the impact of fluctuations, there may be a significant deviation between the predicted sales and the true sales. This deviation can result in additional operational or cost losses in different decision-making scenarios. In the inventory optimization scenario, assume that, at time t, the task is to predict the sales at time t + 1. When the predicted sales at time t + 1 are much lower than the true value, the problem is even more serious because the enterprise will bear the losses caused by the inventory shortage. The inventory loss can be calculated as the product price $\times$ |true sales $-$ predicted sales| at time t + 1. Thereby, solving these two challenges is essential for enterprise decision-making, especially in the domain of inventory optimization. The improvement in sales prediction could help top manufacturing enterprises in saving hundreds of millions of dollars in average annual inventory costs.

To summarize, in the multi-step prediction of MTS, existing research methods are not adept at capturing non-linear changes or accurately predicting sudden fluctuations. This is because the current research mainly focuses on extracting sequence correlations be-

tween time series in an MTS, while seldom considering the differences between time series. To solve these problems, we designed a universality–distinction mechanism framework, which independently models the universality and distinction of the sales sequence. First, the universality mechanism can extract instinct features and common correlation patterns with a similar context from MTS. The instinct features are unique characteristics of each commodity, such as the sales range level and the distribution of sales numbers. Common correlation patterns within a similar context signify the general association between different types of time series, such as the correlations between the sales of a specific commodity and its promotional activities during a given time. Second, ad hoc marketing strategies or other unforeseen factors can lead to unusual fluctuations in sales series. A manual inventory strategy will incorporate the analyses of historical sales and current market conditions to formulate or adjust inventory plans. We devised a distinction extraction module that simulates manual inventory strategies to capture the sales fluctuations caused by these unexpected factors. The red box shown in Figure 1 is the prediction window and clearly shows the effects of existing models on predicting sudden high sales. The yellow box is part of a historical sales series that experienced similar fluctuations to those in the red box. The distinction extraction module captures the unique characteristics of a specific time series from its similar sub-sequences (ex: two windows in Figure 1) and improves the prediction of sudden fluctuations. By modeling the universality and distinction independently, the impacts of large fluctuations can be reduced and the deviations in the prediction of abnormal fluctuations can be minimized.

In addition, although the proposed universality and distinction mechanism can obtain more accurate representations of common and different data from time series in an MTS, we need to design an optimal loss function to adopt the information that was extracted for better sales predictions in a complex environment, such as the issues of shape distortion and time delays in multi-step forecasting. More importantly, the purpose of sales prediction is to realize inventory optimization. According to the investigation results, the cost of shortages is higher than the cost of excess inventory under the same conditions. Considering the characteristics of commodity inventory costs, as well as the issues of shape distortion and time delay in multi-step forecasting, we developed a loss function called Pin-DTW to improve predictive performance.

The main innovations of the model presented in this article are as follows.

- We propose a universality–distinction mechanism (UDM) framework, which consists of universality extraction and distinction-capturing components to improve the accuracy of predictions of multiple future steps.
- "Universality" refers to the inherent characteristics and common correlation patterns found in sales sequences with similar contexts. The shared knowledge is initially learned through a universality extraction component that ensures the overall prediction window's accuracy.
- "Distinction" refers to the process of identifying differences between time series in a sales MTS. To achieve this more efficiently, we propose an attention-based encoder–decoder framework with query-sparsity measurements, which enables us to capture distinct signals based on the states of future multi-step sales.
- We developed a novel loss function called Pin-DTW by jointly combining the pinball and DTW losses to enhance predictive performance. The DTW loss can make better use of the representations obtained from UDM to handle issues of time delay and shape distortion in future multi-step predictions. The pinball loss can be used to control the inventory shortage risk.

The purpose of our research was to design an end-to-end component that integrates a deep learning model and considers a specific decision-making scenario, which can be easily inserted into existing sales forecasting models or frameworks. This has the potential benefit of improving the overall prediction performance of the model in nonlinear relationship discovery and informing specific decision-making scenarios, resulting in cost savings and improved efficiency for enterprise production and sales. An example of our cooperation

with Galanz is described in detail in the experiments section. The source code and data are available at https://github.com/lx237/2023UDM (accessed on 11 February 2023).

2. Related Work

2.1. Time Series Prediction

Time series prediction involves a wide range of fields, including inventory management [13], macroeconomic forecast [14], natural phenomena observations [15], and medical and industrial detection [16]. Highly structured data have strong and complex dependencies among different time steps, and it is a great challenge to effectively model the complex dependencies. The VAE(Variational Auto-Encoder) provides flexible nonlinear mapping and effective inference capabilities [16]; it has been proposed that the VAE can be extended as a recurrent framework to model high-dimensional sequences. Aiming to predict sparse multivariate sequences, [17], a dynamic Gaussian-mixture-based deep generative model was devised, which can model the transitions of latent clusters of temporal features and the emissions of MTS using dynamic Gaussian mixture distributions. From the perspective of time series representation [18], a contrastive learning framework named TS-TCC(Time-Series representation learning framework via Temporal and Contextual Contrasting) was proposed. TS-TCC creates two views by applying strong and weak augmentations to learn robust representations of time series. Experiments have demonstrated the effectiveness of the TS-TCC framework for time series prediction, classification, and other downstream tasks.

Long sequence prediction is also a challenge in time series prediction, where the model is required to accurately capture the long-term dependencies between the input and output. Traditional time series analysis methods, such as the well-known autoregressive moving average (ARMA) and its variants, have proven to be effective in various real-world applications, but they cannot model nonlinear relationships. Yao Q et al. proposed a dual-stage attention-based recurrent neural network (DA-RNN) that can properly capture long-term dependencies and select relevant driving sequences to make predictions [11]. In addition, the informer, which is based on the transformer, can effectively capture the dependencies in long sequences. This enhances the capability for long time series prediction and effectively controls the time and space complexity of model training. The generative informer decoder can also avoid the diffusion of cumulative errors [10]. Farnoosh et al. proposed deep switching autoregressive factorization (DSARF), a deep generative model designed for spatiotemporal data; it has the ability to unravel recurring patterns in the data and perform robust short-term and long-term predictions [12].

Existing work generally reveals the potential trends and patterns from the perspective of time and features, but the efficiency of these models is not always sufficiently explained. A novel strategy called series saliency [19] was proposed for time series analysis and prediction, considering both accuracy and interpretability.

2.2. Sales Forecasting

Sales forecasting is an application that involves time series prediction, which is of practical significance and value to enterprises. In practical research, sales are related to many factors, such as marketing strategies, the weather, and holidays, the complexity of which determines the difficulty of sales forecasting. Aiming to fully capture the dynamic dependencies among multiple influential factors [20], a novel framework for sales prediction named TADA^{+} was proposed, which is enhanced by an online learning module used to carry out trend alignment with dual-attention and multitask RNNs. This is one application of deep learning models, and there are other types of predicting methods, which are classified as follows:

2.2.1. Machine Learning Methods

Hirche et al. [21] used weighted random forest (WRF) to predict under- and over-performing consumer-packaged goods of retail stores, including convenience stores, drug-

stores, food stores, liquor stores, and mass merchandise retail stores. Forecasting future sales changes in products holds great significance for retailing companies. Machine learning models and traditional time series models were both employed to analyze and predict Walmart sales, and the experiments showed that the former performed better [22]. Machine learning methods are widely used in measuring market performance for retail stores, and are also essential in facilitating the transformation from a traditional offline sales model to the B2C model. Brick-and-mortar retail has been hit harder than ever by the COVID-19 pandemic. In [23], the authors achieved this transformation by building a purchase prediction model with XGBoost and random forest. Accurately predicting sales is of high importance to improve the effectiveness of the supply chain. In inventory management, machine learning models, such as RF, XGB, and LGBM models, are used to extract knowledge from large amounts of historical data to predict future orders [24].

2.2.2. Deep Learning Models

Existing studies have applied deep learning models to sales forecast tasks. Reference [25] compares the performances of some deep learning models, including simple RNN, LSTM networks, bidirectional LSTM networks, encoder–decoder LSTM networks, and CNN, in the multi-step time series prediction task, and the bidirectional and encoder–decoder LSTM network provided the best performance in terms of accuracy. Reference [26] proved the effectiveness and robustness of LSTM in comparison to FF-recursive and FF-multi-output models in the multi-step prediction of noise-free, chaotic time series. Reference [27] collected and preprocessed the historical sales volumes and multi-channel online sentiment data to forecast the movement direction of car sales in Taiwan with a CNN-LSTM model. As one of the classic deep learning models, LSTM performed well in terms of advertising expenditures, sales, and demand forecasting [28]. Reference [29] used a deep learning method to predict new product sales in the fashion industry, which was compared with the linear regression, random forest, SVR(Support Vector Regression), and ANN(Artificial Neuronal Networks) models, and the evaluation results show that the deep learning model was no better than the use of single models (such as random forest). Reference [30] proposed a new deep neural framework for e-commerce sales prediction, named DSF(a novel deep neural framework for sales forecasting), which was applied to the Alibaba e-commerce dataset. DSF uses five kinds of features related to sales, including static and dynamic features (such as user behavior characteristics and promotional activities), to forecast sales, which can explicitly simulate the influence of competitive relations and improve model performance.

2.2.3. Integrated Models

Reference [31] put forward a meta-learning framework based on a dual-channel convolution neural network (DCCNN), which automatically learns feature representation in original time series data, and then links the feature representation with a set of weights. The weights are used in basic model combinations, such as random forest and GBRT, and finally find the best combination. A hybrid method composed of a linear model (ARIMA) and a non-linear model (LSTM) was employed to calculate a monthly sales quantity budget based on an enterprise's previous income data [32]. In the demand forecasting task for multi-channel fashion retailers [33], an integrated approach combining k-means clustering, extreme learning machines, and support vector regression was utilized to address challenges caused by the lack of historical data and product demand uncertainty.

Existing research has succeeded in the multi-step prediction of MTS and sales forecasting. However, research has mainly focused on modeling the correlation patterns between time series, and seldom considered how to model and capture the nonlinear dynamic changeable patterns and distinct fluctuation signals. In this research, we designed a universality–distinction mechanism framework to solve these problems to a certain extent. The universality extraction is used to capture linear and non-linear correlation patterns

from the MTS, and distinction capturing can capture distinct fluctuation signals of each time series based on the extracted correlation patterns.

3. Model

This section describes the proposed universality–distinction mechanism (UDM) framework in detail. UDM is a mechanism proposed to improve the performance of future sales predictions. According to previous studies [34,35] and the practical operating methods of e-commerce and manufacturing enterprises, inventory optimization is an objective of sales forecasting. Thus, in order to better optimize inventory, we firstly need to accurately predict future sales. The sales value predicted in this study is very important and is used as a reference for the minimum inventory level, which is provided for marketers to determine the final inventory level, in combination with marketing plans. As shown in Figure 2, first, the UDM framework starts with a convolutional component to encode the multivariate sales sequence and map the input to a higher dimension space. Then, the encoded sequence will be fed into a universality-extracting component to extract common knowledge. Next, the distinction-capture module is used to identify the differences between different prediction steps. Finally, the vector representations that consider the universality and distinctions are mapped to a one-dimensional space as the final output result. In addition, in order to ensure the accuracy and stability of the model, we devised a Pin-DTW loss function to minimize the shape and time delay loss by considering inventory shortage risks. We will introduce each component of our architecture and the loss function in the following subsections.

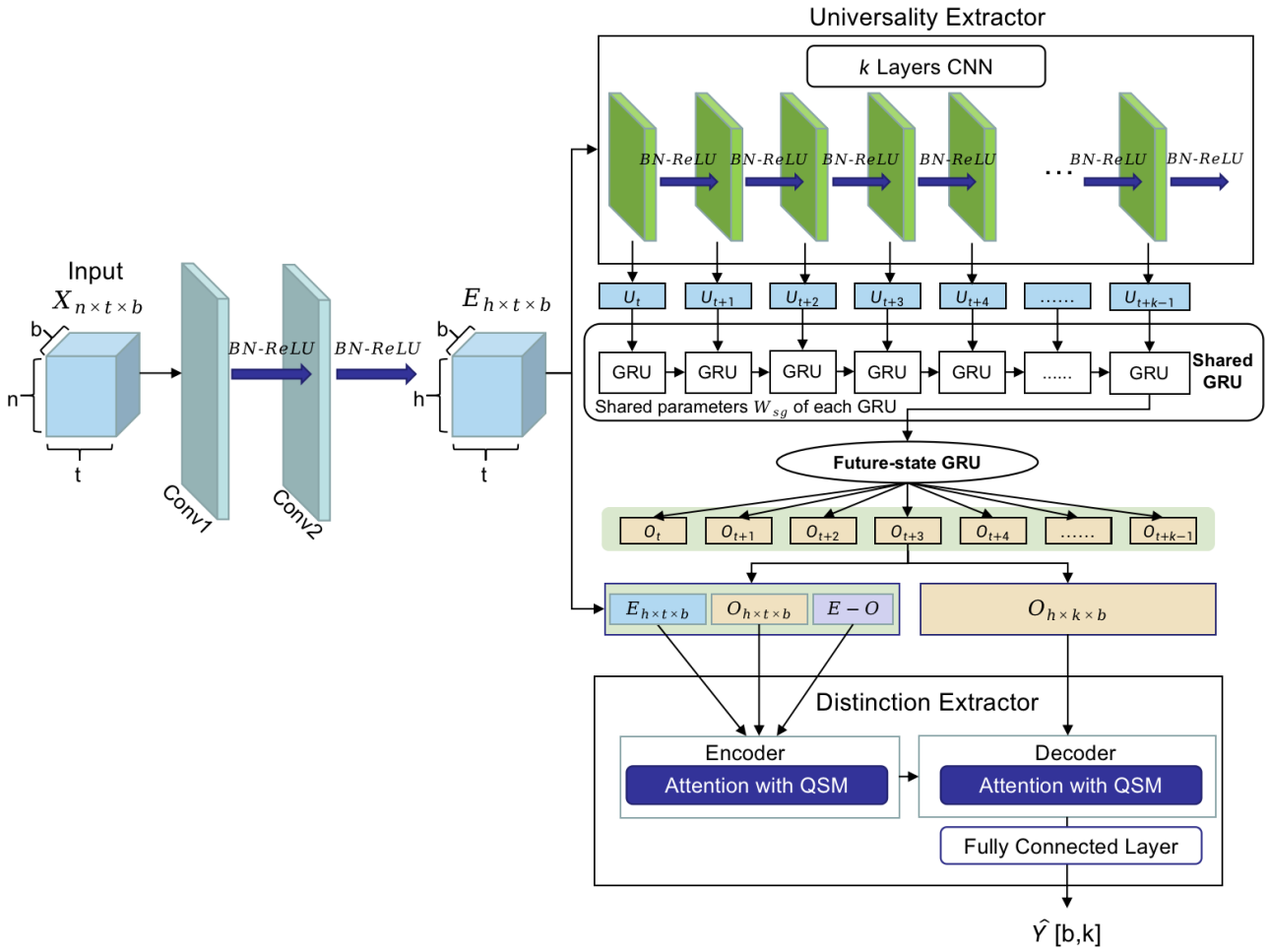

Figure 2. The framework of the universality–distinction mechanism.

3.1. Problem Statement

Assume that there are N products $X = \{X^1, \cdots, X^i, \cdots, X^N\}$ of different models in a warehouse. The sales time series of the ith product $X^i = X^i_{n\times t}$ through timespan $1 \sim t$ has n features. Features include the product type, shop discount, and discount rate. The main aim of this research is to predict the sales $Y_{t+1\sim t+k}$ of all N products at time $t+1 \sim t+k$, where $Y_{t+1} = \{Y^1_{t+1}, \cdots, Y^N_{t+1}\}$. The objective function can be described as follows: $\hat{Y}_{t+1} = UDM(X^{1\sim N}_{1\sim t})$. $X^{1\sim N}_{1\sim t}$ are divided into many batches, and each batch $X_{n\times t\times b}$ (n is the number of features, t is the length of the historical sales sequence, and b is the batch size) is individually input into the UDM. The input $X_{n\times t\times b}$ is first encoded by a convolution component, and the output is $E_{h\times t\times b}$. The output is then fed into a universality-extracting component to extract common correlation patterns $O_{h\times t\times b}$ and generate features $O_{h\times k\times b}$ that fuse future information for the final predictive task. Based on the encoded original input $E_{h\times t\times b}$ and the common correlation patterns $O_{h\times t\times b}$, we can obtain the matrix $Z_{h\times t\times b}$ with distinct fluctuation signals. These three matrices are mainly used to generate the final predictions $\hat{Y}_{bk}$ through an efficient attention mechanism in distinction capturing. Here, f^E and f^D refer to the encoder function with the self-attention mechanism and the decoder function with cross-attention mechanism, respectively. The prediction $\hat{Y}_{bk}$ represents the future k-step predicted values in a batch. Assuming that the real sales are from time $t+1 \sim t+k$ is $Y_{t+1} \sim Y_{t+k}$; the target of the prediction task is to minimize the deviation between these two sequences. We use a joint $L_{Pin-DTW}$ loss function, which consists of L_{Pin} and L_{DTW} to prevent a greater out-of-stock cost and align the ground-truth sequence and predicted value sequence. Some important variables and their explanations are listed in Table 1.

Table 1. Variable and explain.

Variable	Explain
$X_{n\times t\times b}$	The input time series
$E_{h\times t\times b}$	The matrix of the encoded input time series
$U_t, U_{t+1}, ..., U_{t+k-1}$	The construals derived from CNNs
$O_{h\times t\times b}$	The common correlation patterns
$Z_{h\times t\times b}$	The distinct fluctuation signals
f^{EE}, f^{EO}, f^{EZ}	Three encoder functions with self-attention for $E_{h\times t\times b}$, $O_{h\times t\times b}$ and $Z_{h\times t\times b}$
QSM	A measurement function used to select important $q(t)$ from matrix Q for the efficient attention calculation
f^{DE}, f^{DO}, f^{DZ}	Decoder functions with cross-attention
W_{sg}	Shared weights of the GRU(Gate Recurrent Unit) component
W_z, W_r	The parameters of the update gate and reset gate in the future-state GRU
$\hat{Y}_{b\times k}$	The k-step predictions for one batch
$L_{Pin-DTW}$	A joint L_{Pin} and L_{DTW} loss function
L_{Pin}	A loss function used to prevent higher out-of-stock costs
L_{DTW}	A loss function used to align two sequences neatly, reducing the influence of delays and fluctuations

3.2. Convolutional Component

For the input time series (TS) $X_{n\times t\times b}$, we adopted a two-layer convolutional network with batch normalization and *ReLU* activation functions, as shown in Figure 2, as the first part of UDM. For each filter, the kernel size is 1×1. Batch normalization helps to accelerate network convergence, and we assigned the *ReLU* activation function to add nonlinear factors to improve the network's expression ability. This convolutional component is used

to increase the input dimension and information interactions between different features. The input $X_{n\times t\times b}$ is encoded by the component as $E_{h\times t\times b}$(h is the hidden size):

$$E_{h\times t\times b} = ReLU(BN(Conv2(Conv1(X_{n\times t\times b})))), \tag{1}$$

where the $ReLU$ function is $ReLU(x) = max(0, x)$ and the encoded sequences $E_{h\times t\times b}$ are high-dimensional vectors with latent representations and abstract information.

3.3. Universality Extracting

The encoded sequences $E_{h\times t\times b}$ are fed into a universality-extracting component, which is a component used to extract the common temporal features and local correlation patterns and generate construals of multiple future k time steps. This module contains k layers, a convolutional neural network (CNN), a shared GRU, and a future-state GRU, which are described as follows. First, at the i-th CNN layer, where $i \leq k$, the CNN model is used to capture the local correlation patterns of the MTS at a future time step $t + i$, based on the correlation patterns captured from time $t + i - 1$. The correlation patterns can be seen as the construals of future k predictive steps, and are regarded as having a relatively universal law to describe non-linear correlations between different time series within MTS. In our model, we constructed seven different construals for seven predictive steps using a seven-layer CNN:

$$\begin{aligned} U_t &= \Psi_1(E_{h\times t\times b}), \\ U_{t+1} &= \Psi_2(U_t), \\ &\cdots\cdots \\ U_{t+k-1} &= \Psi_k(U_{t+k-2}), \end{aligned} \tag{2}$$

where $E_{h\times t\times b}$ is the matrix of the encoded sequence. $\Psi_i, i \in [0, k]$ are one-dimensional convolutional layers (Conv1D), where the kernel size is 3, stride is 1, and padding is 1. After the convolution operation, we apply the LeakyReLU function as an activation function. Moreover, we use the dropout operation to avoid overfitting. $U_t, U_{t+1}, ..., U_{t+i}, U_{t+k-1}$ are the construals extracted by the multi-layer CNNs described above.

3.3.1. Shared GRU

The construals $U_t, U_{t+1}, ..., U_{t+k-1}$ derived from CNNs of different steps are then individually fed into a shared GRU to share information and model the relations among multiple predictive steps. Similar to MLCNN [1], the shared component learns the states of MTS at a future time step $t + i$, where $i \leq k$, based on the correlations between different times in U_{t+i}; the formula could be described as follows:

$$\begin{aligned} &\text{For } i \text{ in range}(0, k) \\ &\quad O_{h\times t\times b} = \text{shared} - \text{GRU}(U_{t+i}, O_{h\times t\times b}|W_{sg}), \\ &\quad \hat{Y}_{t+i,b} = f^L(O_{h\times t\times b}), \end{aligned} \tag{3}$$

where $O_{h\times t\times b}$ is the sequence correlation representations of the MTS from time 0 to t in a batch b, and the length of the hidden state of the representation is h. The shared GRU is mainly based on a gated recurrent unit (GRU) component [36], which can model the sequence correlation-based hidden state $O[h, t', b]$ in $O_{h\times t\times b}$ at a different time t', where $t' \leq t$. Shared GRU has two parameters: the first parameter U_{t+i} is the local correlation matrix, which is related to the value at a future time step $t + i$. The second parameter $O_{h\times t\times b}$ indicates that the sequence correlation-based hidden states for a future time step $t + i$ accumulate based on the $O_{h\times t\times b}$ at time $t + i - 1$ (in Equation (3), the "For" loop is used to realize the accumulation). W_{sg} is the set of shared weights of the GRU component. For each GRU, $\hat{Y}_{t+i,b}$ will be predicted based on the shared parameters W_{sg}, and the predicted value will be used to update the W_{sg}, which will be taken as the initial parameters for the next

GRU with the input as U_{t+i+1}. For the $t^{'}$th hidden state $O_{h,t^{'},b}$ of GRU, the formula could be described as follows:

$$O[h, t^{'}, b] = \text{GRU}(O[h, t^{'} - 1, b], U_{t+i}(:, t^{'})), \tag{4}$$

where GRU is the recurrent unit of the GRU component; $O[h, t^{'}, b]$ is the $t^{'}$th sequence correlation-based hidden state of tensor $O_{h \times t \times b}$, and the length of the hidden state is h. $U_{t+i}(:, t^{'})$ represents the $t^{'}$th column of U_{t+i}; this is used to represent the local correlations at a time point $t^{'}$, and the local correlations will have a significant influence on the value in a future time step $t + i$. The formula indicates that the hidden state $O[h, t^{'}, b]$ is determined by its previous hidden state and the $t^{'}$th column of U_{t+i}.

3.3.2. Future-State GRU

The future-state GRU is designed to use each local correlation matrix U_{t+i}, where $i \leq k$ represents the future sequence correlation-based hidden state at each future time step $t + i$. Its main purpose is to obtain the initial representations at each future time step by integrating knowledge of instinct features and common correlation patterns from different construals. For example, many sales of a commodity at a future time step $t + i$ have a high probability of being correlated with the average sales number, which ranges from time 0 to t; this could be regarded as an intrinsic feature of the commodity. As introduced in the previous section, the construals are the set of local correlation matrices: $Ut + i$ (U_{t+i} has t rows; the $t^{'}$th row can be seen as the representation of the local correlation patterns between time series at time $t^{'}$), which is provided for all predictive steps to learn the universality of future steps. Assume that at time $t + i$, the hidden state of the future-state GRU at time $t + i$ is computed as follows :

$$\begin{aligned} z_{t+i} &= \sigma(W_z \cdot [O_{t+i-1}, U_{t+i}]), \\ r_{t+i} &= \sigma(W_r \cdot [O_{t+i-1}, U_{t+i}]), \\ \hat{O}_{t+i} &= \tanh(W \cdot [r_{t+i} * O_{t+i-1}, U_{t+i}]), \\ O_{t+i} &= (1 - z_{t+i}) * O_{t+i-1} + z_{t+i} * \hat{O}_{t+i}. \end{aligned} \tag{5}$$

where $*$ indicates Hadamard product.

This future-state GRU fuses future information by aggregating instinct features and correlation patterns from observable time series, which range from time 0 to t. Thus, this operation can produce fusion features $O_{h \times k \times b} = O_t, O_{t+1}, ..., O_{t+k-1}$ for the final predictive task. σ is the *sigmoid* function, and U_{t+1} is the construals at time $t + 1$. z is the update gate and r is the reset gate of GRU. W_z and W_r are the parameters of the update gate and reset gate, respectively.

3.4. Distinction Capturing

As introduced above, universality extraction can extract common correlation patterns from the representation $E[h, t, b]$ of MTS, and the outputs of universality extraction are $O[h, t, b]$ and $O[h, k, b]$. Distinction capturing is then designed to capture distinct fluctuation signals of each time series from MTS, and the captured fluctuation signals can help to attain fused knowledge from different construals to simulate the influences of changeable and complex environments, and then learn the distinctions of different future steps. Distinction capturing is mainly based on an encoder–decoder framework, and the mechanisms of both the encoder and decoder are described as follows:

3.4.1. Encoder Layer

The input Enc_x of distinction capturing consists of three parts: The original input $E[h, t, b]$ from the convolution component, the input $O[h, t, b]$ from universality extracting, and the input-containing distinct fluctuation signals $Z[h, t, b] = E[h, t, b] - O[h, t, b]$. All the

inputs are fed into an attention-based encoder layer with a query-sparsity measurement mechanism f^E to obtain the encoded output Enc_{out}. The formula is as follows:

$$Enc_{out} = [E_{out}, O_{out}, Z_{out}] = [f^{EE}(E_{h\times t\times b}), f^{EO}(O_{h\times t\times b}), f^{EZ}(Z_{h\times t\times b})], \tag{6}$$

where f^{EE}, f^{EO}, and f^{EZ} are the encoder functions used to transfer the$Enc_x = [E_{h\times t\times b}, O_{h\times t\times b}, Z_{h\times t\times b}]$ to the hidden representation Enc_{out}. Similar to the self-attention of the transformer [37], the proposed f^{EE}, f^{EO}, and f^{EZ} adopt a similar strategy to generate the corresponding query Q, key K, and value V for the self-attention calculation. Assume that, at time $t^{'}$, we define:

$$\begin{aligned} q_E(t^{'}) &= QE(E[h, t^{'}, b]), \\ q_O(t^{'}) &= QO(O[h, t^{'}, b]), \\ q_Z(t^{'}) &= QZ(Z[h, t^{'}, b]), \\ q(t^{''}) &\in \{q_E(t^{''}), q_O(t^{''}), q_Z(t^{''})\}, \end{aligned} \tag{7}$$

where QE, QO, and QZ are encoding functions from f^{EE}, f^{EO}, and f^{EZ}, respectively, used to represent the states of MTS at time $t^{'}$. The states are also defined as the query, indicating that the states are mainly used to find the most related "keys" from the MTS. For all $t^{'} \leq t$, the query matrices Q_E, Q_O, and Q_Z are defined as $[q_E(0); q_E(1); ...; q_E(t^{'}); ...; q_E(t)]$, $[q_O(0); q_O(1); ...; q_O(t^{'}); ...; q_O(t)]$ and $[q_Z(0); q_Z(1); ...; q_Z(t^{'}); ...; q_Z(t)]$.

According to the theory of self-attention in the transformer, the states of $q(t^{'})$ are influenced by the previous time series. Assume that, at time $t^{''}$, where $t^{''} \leq t^{'}$, the correlations between time $t^{'}$ and $t^{''}$ are evaluated based on $k(t^{''}) \in \{k_E(t^{''}), k_O(t^{''}), k_Z(t^{''})\}$:

$$\begin{aligned} k_E(t^{''}) &= KE(E[h, t^{''}, b]), \\ k_O(t^{''}) &= KO(O[h, t^{''}, b]), \\ k_Z(t^{''}) &= KZ(Z[h, t^{''}, b]), \\ k(t^{''}) &\in \{k_E(t^{''}), k_O(t^{''}), k_Z(t^{''})\}, \end{aligned} \tag{8}$$

where KE, KO, and KZ are encoding functions from f^{EE}, f^{EO}, and f^{EZ}, respectively, which are used to represent the unique characteristics of MTS at time $t^{''}$. Thus, assume $q(t^{'}) \in \{q_E(t^{'}), q_O(t^{'}), q_Z(t^{'})\}$; the correlations between $t^{'}$ and $t^{''}$ could be described as $q(t^{'}) \times k(t^{''})^T$, which indicates how much the value at time $t^{''}$ will influence the value at time $t^{'}$. For all $t^{''} \leq t$, the key matrices K_E, K_O, and K_Z are defined as $[k_E(0); k_E(1); ...; k_E(t^{''}); ...; k_E(t)]$, $[k_O(0); k_O(1); ...; k_O(t^{''}); ...; k_O(t)]$ and $[k_Z(0); k_Z(1); ...; k_Z(t^{''}); ...; k_Z(t)]$.

As introduced in [9,37], if the correlations between time $t^{'}$ and $t^{''}$ are high, we could use the value at time $t^{''}$ to calculate the attention weight of $t^{''}$ towards the target time $t^{'}$. Thus, the value function at time $t^{''}$ could be defined as follows:

$$\begin{aligned} v_E(t^{''}) &= VE(E[h, t^{''}, b]), \\ v_O(t^{''}) &= VO(O[h, t^{''}, b]), \\ v_Z(t^{''}) &= VZ(Z[h, t^{''}, b]), \\ v(t^{''}) &\in \{v_E(t^{''}), v_O(t^{''}), v_Z(t^{''})\}, \end{aligned} \tag{9}$$

where VE, VO, and VZ are encoding functions from f^{EE}, f^{EO}, and f^{EZ}, respectively, which represent the MTS values at time $t^{''}$. For all $t^{''} \leq t$, the value matrices V_E, V_O, and V_Z are defined as $[v_E(0); v_E(1); ...; v_E(t^{''}); ...; v_E(t)]$, $[v_O(0); v_O(1); ...; v_O(t^{''}); ...; v_O(t)]$ and $[v_Z(0); v_Z(1); ...; v_Z(t^{''}); ...; v_Z(t)]$. Similar to canonical self-attention, we use Q_E, K_E, and V_E to calculate the self-attention of f^{EE}; Q_O, K_O, and V_O to calculate the self-attention

of f^{EO}; and Q_Z, K_Z, and V_Z to calculate the self-attention of f^{EZ}. Finally, the encoding representations E_{out}, O_{out}, and Z_{out} of the input could be calculated by f^{EE}, f^{EO}, and f^{EZ}.

3.4.2. Query-Sparsity Measurement (QSM)-Based Attention

However, the time complexity and memory usage caused by the quadratic computation of canonical self-attention are $\mathcal{O}(L^2)$. According to existing research [10,38], the self-attention score forms a long-tail distribution, which means that only a few dot–product pairs contribute to most of the attention, while others generate trivial attention. Thus, on the basis of existing research on improving the transformer's efficiency [10,38,39], we propose a novel strategy to select the most important $q(t)$ from matrix Q based on the query-sparsity measurement (QSM). The main purpose of the QSM is to use a measurement function to select a few important $q(t)$ from Q for the attention calculation of those dominant $q(t^{'}) \times k(t^{''})$ pairs, and ignore the less important $q(t)$. This operation can enhance the efficiency of the model without compromising its performance, especially for MTS types, such as sales time series, because the MTS needs to process a large amount of spatiotemporal information at each time point.

In this research, we employ the Kullback–Leibler (KL) divergence to realize OSM; this can be used to distinguish the "important" queries. Assume that, at time $t^{'}$, the QSM could be represented as follows:

$$\begin{aligned} &\text{For } f^{EE}: QSM(q_E(t^{'}) = KL(q_E(t^{'})||K_E), \\ &\text{For } f^{EO}: QSM(q_O(t^{'}) = KL(q_O(t^{'})||K_O), \\ &\text{For } f^{EZ}: QSM(q_Z(t^{'}) = KL(q_Z(t^{'})||K_Z)), \end{aligned} \tag{10}$$

where $QSM(q(t^{'})$ indicates the important score of $q(t^{'})$, and $KL(q(t^{'})||K)$ can calculate the KL divergence between $q(t^{'})$ and $K = [k(0); k(1); ...; k(t)]$. A high QSM value means the current $q(t^{'})$ is more important. We can select the top u (u is a hyperparameter) $q(t^{'})$, $t' \in t$ to form a new matrix, Q_u, and the new matrix can be used to calculate the attention weights of each time point based on the self-attention mechanism. Since the sequence length of K is t, the time complexity could be reduced from $\mathcal{O}(t^2)$ to $\mathcal{O}(t \ln t)$.

3.4.3. Decoder Layer

The main task of the research is to predict the values of the future k time steps at the current time t. Thus, the main purpose of the decoder layer is to obtain the embedding representations at a future time step $t+1$, $t+2$, ..., $t+k$. As introduced in Section 3.2, the universality-extracting component can obtain the original representations $O[h,k,b]$ based on the detected correlation patterns. The output of the encoder layer (E_out, O_out, and Z_out) is fed into the decoder layer as input. The decoder layer can optimize $O[h,k,b]$ based on the input, to obtain distinct fluctuation signals, and the main function of the decoder layer is as follows:

$$Dec_{out} = \alpha \times f^{DE}(O[h,k,b], E_{out}) + \beta \times f^{DO}(O[h,k,b], O_{out}) + \gamma \times f^{DZ}(O[h,k,b], Z_{out}), \tag{11}$$

where f^{DE}, f^{DO} and f^{DZ} are decoder functions. α, β, and γ are weight parameters used to evaluate the importance of each decoder function in future predictions. Similar to the encoder functions f^E, all decoder functions are based on an attention-based decoder layer with a query-sparsity measurement mechanism f^D. The difference is that f^D adopts a cross-attention [37,40] mechanism to calculate the attentional relationships between two sequences. Take f^{DE} as an example; its query matrix Q is calculated based on $O[h,k,b]$. Its key and value matrix, K and V, are calculated based on the input E_{out}. Thus, for each $k^{'} \leq k$, the calculation process of the decoder's self-attention can be described as follows: $O[h,k^{'},b]$ will find a set of the most related time points $t^{'}$ from the input, E_{out}, O_{out}, and Z_{out}, based on the matching query Q from $O[h,k,b]$ and key K from the input, and the

values V of each input are used to calculate the attention weights. The calculations of Q, K, V in the decoder functions can be referred to using Equations (5)–(7). Finally, we can attain output $\hat{Y}$ after the Enc_{out} is fed into a fully connected layer, which is also the prediction of our model. The formula is shown as follows:

$$\hat{Y}_{b\times k} = f^L(Dec_{out}), \tag{12}$$

where f^L is the fully connected layer; $\hat{Y}_{b\times k}$ contains the predicted values at each time point $t + k^{'}$, where $k' \leq k$. $\hat{Y}_{b\times k}$ indicates the future predicted values at time $t + k$ in batch b.

3.5. Loss Function

UDM can identify the representations of common and different types of knowledge from sales MTS by adopting universality and distinction mechanisms. However, effectively leveraging the representations in specific sales prediction scenarios presents an additional challenge. In this research, we mainly discuss the use of UDM in inventory optimization scenarios. Based on our investigation and research on Galanz, inventory levels are often used by marketers as references to forecast future sales based on recent sales histories combined with future marketing plans. However, relying on the marketer's experience and traditional statistical learning methods to estimate sales often results in significant deviations. Our work uses models to achieve more accurate predictions to provide references for marketers. Setting inventory levels based on predicted sales, whether higher or lower than actual sales, can result in different cost losses. However, according to our findings, when the absolute value of predicted losses is the same or within the same range, underestimating sales will result in greater cost losses compared to overestimating the sales, because if the predicted sales are smaller than the true sales, there will be an inventory shortage risk. As discussed in previous sections, specific scenarios are usually represented as additional constraints in general scenarios. For the inventory optimization scenarios, the constraint aims to investigate the optimal value between the shortage and excess inventory. To achieve this target, we need to conduct the optimization by focusing on two aspects: (1) Simultaneously considering the shape distortion and time delay of multi-step predictions can further improve the prediction accuracy by better utilizing UDM representations. (2) When addressing the first aspect, we try to make the predicted value greater than the true value to reduce the risk of shortage. A joint Pin-DTW loss function is proposed to cope with the above problems. The DTW loss is used for the optimization of the first aspect and the pinball loss function is used for the optimization of the second aspect. We used the weight α to combine the pinball and DTW losses. For the k step-prediction task, $\hat{Y}_{1\sim k}$ is the prediction and $Y_{1\sim k}$ is the true value. The Pin-DTW loss function can be calculated as follows:

$$L_{Pin-DTW}(Y_{1\sim k}, \hat{Y}_{1\sim k}) = \alpha L_{Pin}(Y_{1\sim k}, \hat{Y}_{1\sim k}) + (1-\alpha) L_{DTW}(\hat{Y}_{1\sim k}, Y_{1\sim k}), \tag{13}$$

Pinball loss [41] is used for the quantile prediction, which is appropriate for the actual sales forecast scenario. When the predictions are smaller than the real sales, this will lead to out-of-stock costs. This will lead to overstock costs. Based on this investigative result, enterprises generally need to sacrifice inventory or out-of-stock costs to some extent to minimize the costs, which corresponds to our need to make forecasts higher or lower than the real demand. This objective can be transformed into a quantile prediction task, so we adopt the pinball loss function. Assuming that τ is the target quantile, $y_i \in Y_{1\sim k}$ is the actual value, and $\hat{y}_i \in \hat{Y}_{1\sim k}$ is the quantile prediction, the calculation formula of the pinball loss function is as follows:

$$L_{Pin}(Y_{1\sim k}, \hat{Y}_{1\sim k}) = \frac{1}{k}\sum_{i=1}^{k} L^i_{Pin}(y_i, \hat{y}_i), \tag{14}$$

$$L^{i}_{Pin}(y_i, \hat{y}_i) = \begin{cases} (y_i - \hat{y}_i)\tau, & if y_i \geq \hat{y}_i \\ (\hat{y}_i - y_i)(1-\tau), & if y_i > \hat{y}_i, \end{cases} \tag{15}$$

DTW(Dynamic Time Warping) [42] is a framework for multi-step forecasting that can be used to calculate the similarity between two time series of the same length, which can neatly align two sequences and reduce the influence of delay and fluctuation. The DTW loss function, which compares the prediction $\hat{Y}_{1\sim k}$ with the actual ground truth future trajectory $Y_{1\sim k} = (y_1, \ldots, y_k)$ of length k, is composed of two terms, which are balanced by the hyperparameter $\theta \in [0, 1]$. The calculations of $\mathcal{L}_{shape}$ and $\mathcal{L}_{temporal}$ are explained in $DILATE$ [42] in detail.

$$L_{DTW}(\hat{Y}_{1\sim k}, Y_{1\sim k}) = \theta \mathcal{L}_{shape}(\hat{Y}_{1\sim k}, Y_{1\sim k}) + (1-\theta)\mathcal{L}_{temporal}(\hat{Y}_{1\sim k}, Y_{1\sim k}). \tag{16}$$

4. Experiment

4.1. Dataset

Galanz: This time series dataset was collected from Galanz, one of China's leading home appliance enterprises. This includes the historical sales data of 583 products from 11 warehouses over a 2-year period. In addition, four other features could be utilized: product type, shop discount, performance discount, and discount rate.

Cainiao: This dataset is an official dataset provided by Aliyun for a specifically designed public algorithm competition. This contains the inventories of commodities in Cainiao's national and regional warehouses from 20,141,001 to 20,151,227. The dataset includes sales records of up to 200 products across 5 warehouses, as well as other features, such as product types, user visit records, visits to carts, and collections of user visits.

More information about these two datasets is shown in Table 2. For both datasets, each product was first grouped by warehouse, then by product type, to generate multivariate time series (MTS). For each MTS, training and testing samples were generated by dividing the whole series into a set of sub-series with the minimum length greater than 24 [1,30,43]. The sales of the last 1~7 time periods of each sub-series were taken as the prediction label, and other periods were taken as features. This operation can obtain 55,361 samples (GW1-N) from the Galanz and 74,595 samples (CW1-N) from Cainiao. To further assess the practical values of the proposed model, warehouse IDs were used to divide Galanz GW1-N into 11 groups (GW1~GW11) and Cainiao CW1-N into 5 groups (CW1~CW5). All datasets were split in chronological order to produce a training set (60%), a validation set (20%), and a test set (20%). Each group of both Galanz and Cainiao was separately trained and tested by our proposed model, UDM.

Table 2. Information of datasets used in this paper.

Dataset	Galanz	Cainiao
Warehouses	11	5
Product category quantity	38	27
Instances	583	200
Sample rate	1 day	1 day
Features	Product type Historical sales Amount of shop discount Perform discount amount Discount rate	Product type Historical sales User visits records Visits to cart Collections user visits

4.2. Metrics

To evaluate the accuracy of the models' performance on different datasets, we use four metrics, where N stands for the number of predictions, and y_i and $\hat{y}_i$ are the ground truth of the time series value and the prediction of the models, respectively. MAE stands for the average absolute error between the prediction and the ground truth, MAPE calculates the percentage difference between the prediction and the ground truth, RMSE represents the expectation of the squared error between the prediction and the ground truth, and CORR(Empirical correlation coefficient) represents the correlation between the two sequences. For the first three metrics, a lower value is better, while for CORR, a higher value is better. In terms of sales prediction tasks, MAE reflects the deviation between the actual sales and model predictions.

- Mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} |\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}|. \tag{17}$$

- Mean Absolute percentage error (MAPE):

$$\mathrm{MAPE} = \frac{100\%}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left| \frac{\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}}{\mathrm{y_i}} \right|. \tag{18}$$

- Root mean squared error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2}. \tag{19}$$

- Empirical correlation coefficient (CORR):

$$\mathrm{CORR} = \frac{1}{\mathrm{N}} \times \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} (\mathrm{y_i} - \mathrm{mean(y)})(\hat{\mathrm{y}}_\mathrm{i} - \mathrm{mean}(\hat{\mathrm{y}}))}{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{N}} (\mathrm{y_i} - \mathrm{mean(y)})^2 (\hat{\mathrm{y}}_\mathrm{i} - \mathrm{mean}(\hat{\mathrm{y}}))^2}}. \tag{20}$$

4.3. Baselines

We compare UDM with three categories of methods:

- Traditional TS modeling methods, including FBProphet [44], exponential smoothing [2], and ARIMA [45]. FBProphet is proposed by Facebook to forecast time series data based on an additive model, where non-linear trends are fit with yearly, weekly, and daily seasonalities. Exponential smoothing is one of the moving average methods, which is carried out according to the stability and regularity of the time series to reasonably extend the existing observation series and generate the prediction series. ARIMA stands for the autoregressive integrated moving average. It considers the previous values of the data, the degree of differencing required to achieve stationarity, and the moving average errors to make predictions for future values.
- Informer [10]: A model based on the transformer can effectively capture the dependencies in long sequences. It increases the capacities of long-time series predictions, and effectively controls the time and space complexities of model training.
- MLCNN [1]: This is a deep learning framework composed of a convolution neural network and recurrent neural network; it improves the predictive performance by fusing forecasting information of different future times.

The comparison is used mainly for the final multi-step commodity sales prediction performance using four metrics mentioned in Section 4.2, including *MAE*, *MAPE*, *RMSE*, and *CORR*.

4.4. Training Details

We conducted a grid search for the proposed UDM and all baselines, except ARIMA (we utilized auto-ARIMA with an automatic parameter adjustment function) to find the best hyperparameter settings. To begin with, for the task of multi-step sales sequence forecasts, we set the output length to 7, which means that the models should predict the sales sequence for 7 days. For all models, we set the maximum training iterations to 20 for the Galanz dataset and 10 for the Cainiao dataset. For UDM, we used a batch size of 16 and a learning rate of 0.0001. In the convolutional component, we set the CNN layers to 2, the dropout rate to 0.2, and the output size to 128. In the universality-extracting module, the CNN module was configured with 7 layers, the number of layers was set to 4, and the dropout rate to 0.2 for the shared GRU. Next, in the distinction-capture module, the head of the prob-sparse attention layer was 8, the sampling factor was $k = 7$, and the activation function was $gelu$. In the Pin-DTW loss function, we set the weight to $\alpha = 1/2$ and the target quantile to $\tau = 0.6$.

For the informer model, we set $d\_model = 512$, $n\_heads = 8$, $num\_workers = 2$, $e\_layers = 2$, $d\_layers = 2$, $batch\_size = 16$, $learning\_rate = 0.0001$, and $drop\_out = 0.05$. We used MSE as the loss function and chose a prob-sparse attention mechanism in the encoder. For MLCNN, we used the continuous mode based on the data type. We applied a collaborative span of three and a collaborative stride of one. We set $learning\_rate = 0.0001$, $n\_CNN = 7$, $drop\_out = 0.2$, $hidCNN = 10$, $hidRNN = 25$, $highway\_window = 3$, and we tuned $kernel\_size \in [3, 5]$. For both traditional time series prediction methods, ES and FBProphet, we set alpha as 0.5 and beta as 0.9.

4.5. Main Results

We will now compare the performances of UDM and other baselines on Galanz (GW1~GW11) and Cainiao datasets (CW1~CW5), as shown in Tables 3–5. The best results are highlighted in bold, and the second-best results are underlined for each metric. Compared with traditional time series-predicting methods and advanced deep learning models, our proposed model (UDM) outperforms the other models on both Galanz and Cainiao datasets. Compared with the best baseline, the improvements are 57.27%, 50.68%, and 35.26% on the Galanz dataset and 16.58%, 6.07%, and 5.27% on the Cainiao dataset in terms of MAE, MAPE, and RMSE. For 11 Galanz warehouses datasets, UDM achieved the five best results and the five second-best results on MAE, as well as the eight best results on both MAPE and RMSE. In the Cainiao dataset, UDM demonstrated the best results in terms of MAE, MAPE, and RMSE. Among the five baselines, the informer model achieved the second-best results on the Galanz datasets. However, its prediction ability was not stable, as shown by the large MAE, MAPE, and RMSE results on several datasets, such as GW1 and GW8. Although the informer model is a very competitive baseline model, from the overall effect evaluation, UDM is significantly better than the informer model. The average improvements compared to the informer model are 20.09%, 35.26%, and 78.58% on 11 Galanz warehouses and 27.95%, 12.57%, and 31.32% on 5 Cainiao warehouses in terms of MAE, RMSE, and MAPE. Compared to all the baseline models, the informer model can achieve the 7 best MAE results and 2 best MAPE and RMSE results on 11 Galanz warehouses. However, our model achieved the best results on the whole Galanz dataset (GW1-N) and the whole Cainiao dataset (CW1-CWN) in terms of MAE, RMSE, and MAPE. For 11 Galanz warehouses and GW1-N, UDM achieved the 5 best MAE results, 10 best RMSE results, and 8 best MAPE results. For 5 Cainiao warehouses and CW1-N, UDM achieves the best results in terms of MAE, RMSE, and MAPE. In inventory management, the predicted sales component of our model serves as a reference for marketing personnel when stocking the minimum inventory level. The predicted sales component is an important indicator when arranging the inventory plan for up to two weeks in the future, and the inventory plan is based on a strict calculation process. Based on actual testing in the enterprise, our method saved approximately 20% of costs compared to traditional methods and avoids the risk of out-of-stock methods, proving its effectiveness.

Table 3. Evaluation of all baselines on 6 Galanz warehouse datasets (GW1-GW6). The best results are highlighted in bold, and the second-best results are underlined for each metric.

Method	Metrics	GW1	GW2	GW3	GW4	GW5	GW6
FBProphet	MAE	15.2847	30.3044	31.8755	34.4644	16.2163	16.7275
	MAPE	**57.9891**	57.1674	55.8968	62.3766	55.8108	59.6338
	RMSE	52.4318	276.4578	277.0762	303.1195	61.2255	63.1584
	CORR	0.2373	**0.1636**	0.1613	0.2153	**0.2248**	0.2172
Informer	MAE	17.6240	**9.1548**	**4.1450**	**29.5429**	**5.1940**	15.0258
	MAPE	394.9109	22.5000	19.1176	80.0000	11.9055	457.2982
	RMSE	35.0193	38.1540	13.0211	**41.3107**	22.6037	17.4231
	CORR	0.3087	0.0000	0.0000	0.0000	0.0566	**0.2400**
MLCNN	MAE	18.3824	33.6053	36.7638	37.6184	20.1129	20.7452
	MAPE	371.8239	313.5209	312.2668	290.0065	354.7266	294.2883
	RMSE	60.9568	281.9341	286.2610	308.5424	72.1018	77.0757
	CORR	0.2163	0.1597	0.1681	0.1808	0.2084	0.1929
ES	MAE	21.2214	34.1332	37.0971	40.3494	22.1497	23.0650
	MAPE	267.1900	239.7035	254.3169	271.8555	258.0128	271.5501
	RMSE	63.8428	292.0610	294.4738	320.9468	72.0085	75.0383
	CORR	0.1986	0.1589	0.1545	0.1985	0.1934	0.1985
ARIMA	MAE	14.2295	28.4680	30.0798	32.6718	15.3041	15.8134
	MAPE	75.1275	67.6109	66.1797	76.7000	72.4418	75.8649
	RMSE	52.2138	282.8495	283.7393	309.1383	65.1573	66.4925
	CORR	0.1181	0.0961	0.0984	0.1154	0.1157	0.1146
UDM	MAE	**10.4198**	9.7407	5.2871	41.7966	7.3015	**2.8003**
	MAPE	82.7683	**16.1626**	**16.4438**	**52.9563**	**10.2945**	**16.9803**
	RMSE	**34.2683**	**37.6247**	**12.4147**	51.3284	**21.7823**	**7.4525**
	CORR	**0.3373**	0.0577	**0.2435**	**0.5420**	0.0703	0.0059

4.5.1. The Advantage of the Informer

The informer is a representative transformer-based model, and is a competitive baseline model in the experiment, especially in terms of its MAE performance. Although UDM is significantly superior to the informer model on the entire Galanz datasets in terms of all metrics, the informer model outperforms UDM on 6 Galanz warehouses, in terms of the MAE metric. However, the performances of the two models on MAE are very similar; our proposed UDM model achieved comparable results to the informer model on several datasets where the informer model had previously shown superior performance, with only minor deviations. The reason for this may be that, during the independent modeling of universality and distinct aspects in UDM, information loss issues may have occurred, causing a slight decrease in UDM's performance on the MAE metric.

4.5.2. The Advantage of UDM

The informer model's performance in terms of RMSE and MAPE was not as good as its performance in MAE. The MAPE of the informer model exhibited large deviations on Galanz warehouses GW1, GW6, and GW8, the MAPEs of which exceeded 300 (the MAPEs of UDM on the same warehouses were 82, 16, and 11, respectively). The reason for this is that the informer model is weaker than UDM at capturing fluctuation patterns. A large MAPE often indicates that the true number of sales is small, but the number of predicted sales is large, which means that the informer model often misjudges the sudden peak values or the fluctuation trend. UDM proposes a novel distinct mechanism, which can specifically model the unique characteristics of each time series, and learn fluctuation patterns separately. The mechanism can better solve this problem, which the informer model struggles to handle properly, and significantly improve the performance in terms of MAPE. RMSE is another metric that can evaluate the stability of the model's performance. UDM outperformed the informer model on 8 Galanz warehouses and all

Cainiao warehouses, which indicates that the performance of UDM is more stable than that of the informer model. The informer model exhibited very large variations in the testing datasets of certain Galanz warehouses. For example, the RMSE of the informer model on the dataset of Galanz warehouse GW8 was 139, while the value of UDM on the same dataset was only 31.

Table 4. Evaluation of all baselines over 5 Galanz warehouse datasets (GW7-GW11) and the whole Galanz dataset (GW1-N). The best results are highlighted in bold, and the second-best results are underlined for each metric.

Method	Metrics	GW7	GW8	GW9	GW10	GW11	GW1-N
FBProphet	MAE	9.0854	17.9634	28.0403	27.2752	11.2483	21.6805
	MAPE	70.7815	59.7207	58.4812	51.8571	63.3920	59.3717
	RMSE	25.1791	74.1755	265.7824	266.1724	40.2530	155.0029
	CORR	**0.2347**	**0.2153**	0.1605	**0.1716**	**0.2265**	**0.2025**
Informer	MAE	**0.0089**	135.4354	**3.8177**	5.6080	**0.0470**	20.5094
	MAPE	4.2647	399.4997	**25.7501**	87.0297	**1.2546**	136.6846
	RMSE	0.0945	139.5018	**8.4105**	5.9640	**0.1817**	29.2440
	CORR	0.0000	0.1250	0.1975	0.0986	0.0097	0.0942
MLCNN	MAE	10.5993	21.9535	31.4661	30.6287	13.3700	25.0223
	MAPE	395.6474	296.0490	348.4923	282.7980	377.6082	330.6571
	RMSE	26.1790	84.4675	273.0667	269.1123	43.8731	162.1428
	CORR	0.2212	0.2081	0.1561	0.1635	0.2249	0.1909
ES	MAE	12.4604	24.1590	31.9851	31.6545	14.2210	26.5905
	MAPE	201.3380	271.0901	241.4655	210.1611	189.3783	243.2783
	RMSE	31.4932	84.5216	282.4325	282.8159	43.0267	167.5147
	CORR	0.2028	0.1968	0.1601	0.1616	0.1976	0.1837
ARIMA	MAE	7.6219	16.8480	26.4030	25.6837	9.7364	20.2600
	MAPE	74.3844	75.8856	68.5079	62.2803	70.6236	71.4188
	RMSE	23.0878	77.0473	270.6166	271.3406	38.8939	158.2343
	CORR	0.1303	0.1180	0.0938	0.0988	0.1200	0.1108
UDM	MAE	**0.0089**	**9.4581**	7.7964	**0.5114**	0.1156	**8.6579**
	MAPE	**2.3156**	**11.5199**	95.4989	**7.0528**	10.0944	**29.2807**
	RMSE	**0.0943**	**31.7277**	9.6884	**1.6839**	0.1855	**18.9319**
	CORR	0.0000	0.1028	**0.2198**	0.0846	0.0089	0.1521

4.6. Ablation Study

To demonstrate the effectiveness of every UDM component, we compare UDM with five variants, as follows:

- w/U: The universality-extracting component is removed from UDM.
- w/D: The distinction-capture component is removed from UDM.
- w/Pin-DTW: Pin-DTW loss is replaced by the MAE as the loss function.
- w/DTW: DTW is removed from the Pin-DTW loss function.
- w/Pin: Pinball loss is removed from the Pin-DTW loss function.

We kept all variant parameters the same as the completed UDM model to eliminate the influence of model complexity. Figure 3a,b present the results of the Galanz and Cainiao datasets, respectively, in detail. The important observations from these results are listed as follows:

- Removing the distinction module causes great performance drops in terms of MAE metrics on the Galanz and Cainiao datasets; this proves that extracting the distinction module helps to achieve more accurate multi-step predictions.
- According to Figure 3a, the significant decline in MAE appears when the Pin-DTW loss is replaced by the MAE loss function. The metrics also clearly decrease when the pinball loss is removed from the Pin-DTW loss function, which illustrates the

significant contributions of the joint Pin-DTW loss function, especially the pinball loss function, to the Galanz dataset. However, Figure 3b shows that the DTW component in the Pin-DTW loss function contributes more to the Cainiao dataset.

- Removing the universality module results in a more obvious decrease in MAE in the Galanz dataset than the Cainiao dataset, which indicates that capturing the common features of products from the same warehouse is effective in the Galanz dataset, and great differences exist between the different warehouses.

Table 5. Evaluation of all baselines over 5 Cainiao warehouse datasets (CW1-CW5) and the whole Cainiao dataset (CW1-N). The best results are highlighted in bold, and the second-best results are underlined for each metric.

Method	Metrics	CW1	CW2	CW3	CW4	CW5	CW1-N
FBProphet	MAE	1.5912	1.3346	1.8584	2.3292	1.7956	1.7818
	MAPE	62.1643	50.6469	67.7299	66.1445	61.5287	61.6429
	RMSE	7.5401	6.5327	7.9424	12.1592	8.3002	8.4949
	CORR	0.2532	0.2264	0.2422	0.2462	**0.2480**	0.2432
Informer	MAE	1.8133	1.5807	2.1339	2.7799	2.0071	2.0630
	MAPE	87.9340	78.3794	61.2950	66.2499	93.0463	77.3809
	RMSE	8.2744	7.2051	8.7268	13.8907	8.7783	9.3751
	CORR	0.2476	**0.2369**	0.2628	**0.2601**	0.2405	**0.2496**
MLCNN	MAE	1.7142	1.4752	2.1640	2.6487	1.8236	1.9651
	MAPE	63.3250	60.5524	94.3953	63.0208	88.9358	74.0459
	RMSE	7.8343	6.9332	9.4610	13.4777	8.8779	9.3168
	CORR	0.2353	0.2021	0.2581	0.2338	0.2190	0.2296
ES	MAE	2.0410	1.9012	2.6317	3.2389	2.3219	2.4269
	MAPE	70.3472	65.9903	86.5732	86.4917	81.6856	78.2176
	RMSE	8.3194	7.3440	9.4947	13.5777	9.5196	9.6511
	CORR	0.2449	0.2059	0.2686	0.2394	0.2080	0.2333
ARIMA	MAE	1.6612	1.4361	1.8497	2.5363	1.6870	1.8340
	MAPE	59.6706	49.3459	59.6440	63.6416	50.5761	56.5757
	RMSE	7.8752	7.0230	7.9706	12.6939	8.4314	8.7988
	CORR	0.1462	0.1392	0.1483	0.1527	0.1483	0.1460
UDM	MAE	**1.3642**	**1.1498**	**1.5688**	**2.0236**	**1.3251**	**1.4863**
	MAPE	**55.6866**	**47.1187**	**55.7236**	**58.4229**	**48.7573**	**53.1418**
	RMSE	**7.2861**	**6.2559**	**7.3421**	**11.5921**	**7.7595**	**8.0472**
	CORR	**0.2680**	0.1985	**0.2697**	0.2065	0.1967	0.2279

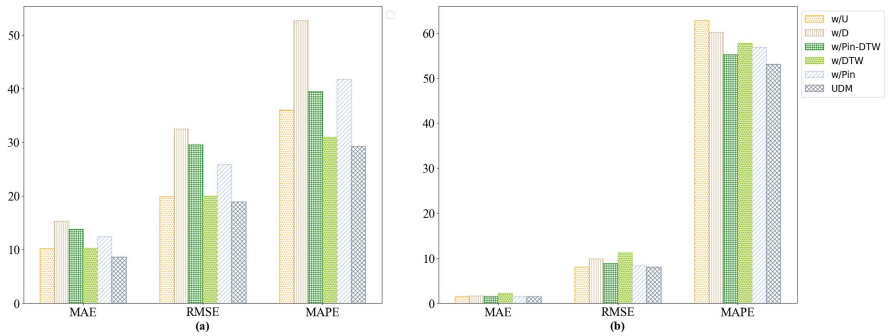

Figure 3. Ablation study: (**a**) Analysis of the Galanz dataset. (**b**) Analysis of the Cainiao dataset.

More importantly, as can be seen in Figure 3a, removing the distinction-capture component results in great drops in RMSE (41.77%) and MAPE (44.47%) in the Galanz dataset. According to Figure 3b, there are obvious decreases in performance (28.63% and 18.6%) in terms of RMSE in the Cainiao dataset when the DTW loss or the distinction-capture component is removed from UDM, and MAPE clearly goes down (15.40% and

11.83%) when the universality-extracting component or distinction-capture component is removed from UDM in the Cainiao dataset. The experimental results clearly exhibit that the distinction-capture component plays the most important role in the stability of UDM.

4.7. Further Analysis

4.7.1. Parameter Sensitivity Analysis

Fine-tuning experiments on the Galanz dataset were carried out for three parameters that can obviously affect the effectiveness of the UDM. These parameters are the hidden size, the weight α in the Pin-DTW loss function, and the sampling factor in the distinction module. As shown in Figure 4, we attained optimal results when the hidden size was set at 128, α was set at $1/2$, and k was set at 7.

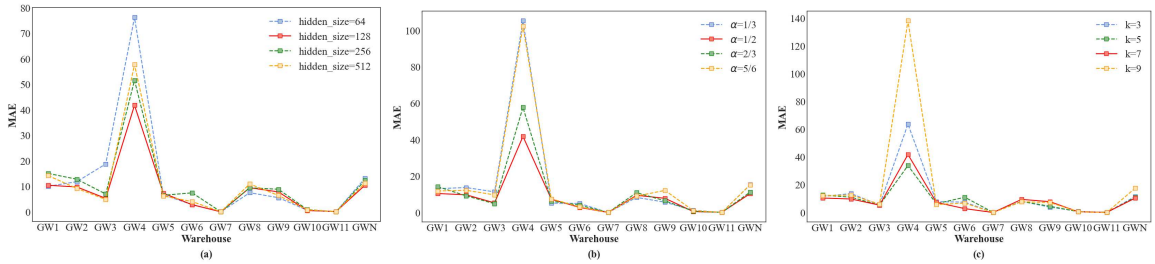

Figure 4. Results of the parameter sensitivity tests on the Galanz dataset. (**a**) Hidden size. (**b**) Weight of the loss function. (**c**) Sampling factor in the distinction module.

4.7.2. Comparative Analysis of Attention

We compared the canonical self-attention mechanism and the prob-sparse attention mechanism on Galanz and Cainiao datasets in terms of three metrics (running time, MAE, and MAPE), and the results can be seen in Figure 5a. From these two datasets, the prob-sparse attention mechanism performed better, with shorter running times on MAE and MAPE metrics when compared to canonical self-attention.

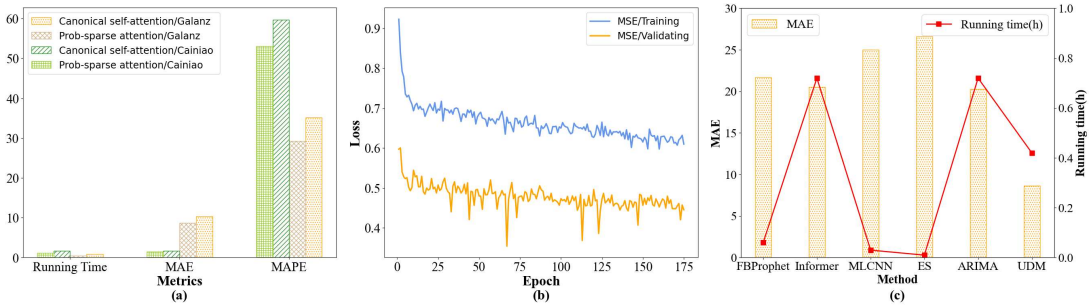

Figure 5. (**a**) Contrast between canonical self-attention and prob-sparse attention mechanisms. (**b**) Convergence analysis. (**c**) Training time analysis.

4.7.3. Convergence and Time Complexity Analysis

We analyzed the convergence of UDM by recording its MAE loss of training and validating on the Galanz dataset. As shown in Figure 5b, UDM can easily be trained with quick convergence. Figure 5c shows the training time and MAE comparison of FBProphet, informer, MLCNN, ES, ARIMA, and UDM on the Galanz dataset. UDM uses the attention mechanism in its distinction module, so it is reasonable that UDM takes more time for training than MLCNN, which is based on CNNs and traditional time series analysis methods (FBProphet and ES). In our study, we utilized auto-ARIMA, a model equipped with an automatic parameter tuning ability. As a result, the implementation of fitting and predicting with ARIMA requires comparatively more time. As we can see, the inference time of UDM is less than that of ARIMA and the informer model, and UDM performs better than other methods on the MAE metric.

4.8. Case Study

A case study comparing the stability in the multi-step forecast task of the proposed UDM model with five baselines can be seen in Figure 6. In this figure, the blue and green parts are used as input sequences for the prediction of item D63, where blue is a historical series and green is the source window. The yellow and red parts are found in the prediction window, in which yellow is the ground truth and red is the model prediction. In this case, the improvements in UDM for predicting the future seven-step sales are 78%, 76%, and 78%, respectively, when compared to the informer model, in terms of MAE, RMSE, and MAPE. The MAPE improvements are 74%, 73%, and 74%, respectively, compared to MLCNN. The absolute improvements in CORR are 22% and 20% compared to the informer model and MLCNN. Compared to the best performances from FBProphet, ES, and ARIMA, the improvements are 76%, 75%, and 78%, respectively, in terms of MAE, RMSE, and MAPE. Some valuable observations are as follows: (1) In our multi-step forecast tasks, the informer model provides a flat prediction and cannot simulate the real fluctuations, as shown in Figure 6a. Moreover, there is a large deviation between the predicted average and the real average. (2) The MLCNN and ARIMA models can capture fluctuations but often generate delayed predictions, resulting in an opposite change trend between actual and predicted sequences, as illustrated in Figure 6b,e. (3) The results of FBProphet are similar to those of the informer model; however, FBProphet presents a few waves that are contrary to the ground truth, which can be seen in Figure 6c. (4) The results of the exponential smoothing (ES) model are shown in Figure 6d; the errors between predictions and actual values gradually become more obvious with increasing time steps. (5) Figure 6f shows the results of the proposed UDM model, which demonstrates that not only can UDM make more accurate predictions but it can also more closely simulate the variation trend of the sales sequence compared to the baseline. Moreover, the UDM predictive accuracy remains stable in multi-step predictions. In other words, UDM achieves accurate predictions for each step, and the prediction error does not gradually increase over time steps, as seen in the ES results. The stability of our model is mainly attributed to the universality component in UDM. By extracting instinct features and common correlation patterns from multivariate time series with similar contexts, our model ensures that the multi-step predicted values remain within a reasonable range, thereby preventing error accumulation.

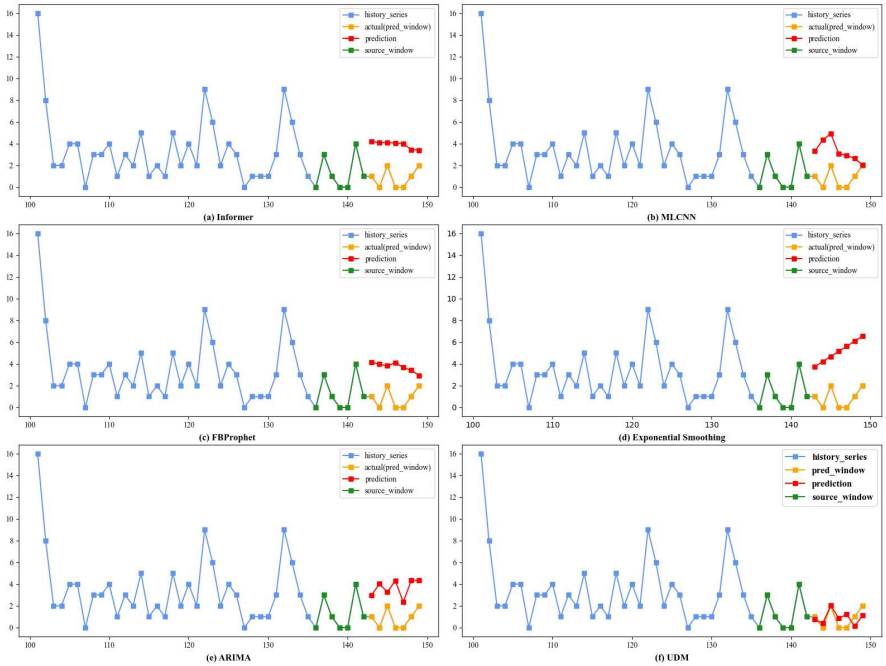

Figure 6. Case study: (**a**) informer, (**b**) MLCNN, (**c**) FBProphet, (**d**) exponential smoothing, (**e**) ARIMA (**f**) UDM.

5. Conclusions

In our paper, we propose a novel universality–distinction mechanism framework for the multiple-step sales prediction task. Firstly, the universality-extracting module generates construals for different predictive steps, which are integrated into the complete time window information. At the same time, this module can model the relationships among different future steps and learn their universality. An efficient self-attention mechanism was then employed to distinguish the information of multiple predictive steps, which could effectively capture future fluctuations. Finally, we developed a joint Pin-DTW loss function that addresses the optimizations of two aspects of the second challenge: (1) how to make better use of the extracted common and distinct representations from UDM to make more accurate future multi-step predictions; (2) how to ensure that predicted sales are bigger than the true sales as much as possible, on the premise of minimizing prediction deviations. Both the deformation error and time delay error are seen for the first aspect, which could be solved by a sequence-based DTW loss. For the second aspect, the main aim is to avoid the risk of shortages in inventory optimization, and a pinball loss is proposed to solve this problem. The combination of pinball and DTW, named Pin-DTW loss, can reduce the risk of inventory shortage while further improving prediction accuracy and stability. The experiments conducted on the Galanz and Cainiao datasets in the fourth section prove that our proposed UDM model ensures accurate and stable sales prediction capabilities while effectively reducing costs for enterprises.

However, some directions are worth exploring in the future. First, the outliers in the sales sequence can greatly affect the accuracy of the forecast; therefore, improving the model's handling of outliers or abnormal values is an important direction for future research. Second, we hope to further improve the forecasting efficiency and bring real value to the enterprise supply chain.

Author Contributions: Conceptualization, D.L. and X.L.; data curation, F.G. and Z.P.; formal analysis, X.L. and F.G.; funding acquisition, D.L.; investigation, X.L., F.G., and Z.P.; methodology, D.L. and X.L.; project administration, D.L. and D.C.; resources, D.L. and D.C.; software, D.L. and X.L.; supervision, D.L. and D.C.; validation, F.G. and Z.P.; writing—original draft, X.L.; writing—review and editing, D.L. and A.M. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: To verify the effectiveness of the proposed model, we utilized two datasets - the Galanz dataset and the Cainiao dataset. We would release the de-identified version of the Galanz dataset at https://github.com/lx237/2023UDM once we have obtained approval from the company. The Cainiao dataset is an official dataset provided by Aliyun for a specifically designed public algorithm competition: https://tianchi.aliyun.com/competition/entrance/231530/introduction.

Funding: This research was supported by the National Natural Science Foundation of China (NSFC) under grant no. 72074231 and the Soft Science Foundation of Guangdong Province in China under grant no. 2019A101002020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cheng, J.; Huang, K.; Zheng, Z. Towards Better Forecasting by Fusing Near and Distant Future Visions. *Natl. Conf. Artif. Intell.* **2020** , *34*, 3593–3600. [CrossRef]
2. Harrison, P.J. Exponential Smoothing and Short-Term Sales Forecasting. *Manag. Sci.* **1967**, *13*, 821–842. [CrossRef]
3. Slawek, S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *Int. J. Forecast.* **2020**, *36*, 75–85.
4. David, S.; Valentin, F.; Jan, G.;Tim J. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191.
5. Yudianto, M.R.A.; Agustin, T.; James, R.M.; Rahma, F.I.; Rahim, A.; Utami, E. Rainfall Forecasting to Recommend Crops Varieties Using Moving Average and Naive Bayes Methods. *Int. J. Mod. Educ. Comput. Sci.* **2021**, *13*, 23–33. [CrossRef]
6. Breiman, L. Random Forests. *Mach. Learn. Arch.* **2001**, *45*, 5–32. [CrossRef]
7. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *Knowl. Discov. Data Min.* **2016**, 785–794.

8. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In Proceedings of the International Acm Sigir Conference on Research and Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104.
9. Lim, B.; Arik, S.O.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
10. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Natl. Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
11. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G.W. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *arXiv* **2017**, arXiv:1704.02971.
12. Farnoosh, A.; Azari, B.; Ostadabbas, S. Deep Switching Auto-Regressive Factorization:Application to Time Series Forecasting. *Natl. Conf. Artif. Intell.* **2021**, *35*, 7394–7403.
13. Zhou, Q.; Han, R.; Li, T.; Xia, B. Joint prediction of time series data in inventory management. *Knowl. Inf. Syst.* **2019**, *61*, 905–929. [CrossRef]
14. Hmamouche, Y.; Lakhal, L.; Casali, A. A scalable framework for large time series prediction. *Knowl. Inf. Syst.* **2021**, *63*, 1093–1116. [CrossRef]
15. Tian, H.; Xu, Q. Time Series Prediction Method Based on E-CRBM. *Electronics* **2021**, *10*, 416. [CrossRef]
16. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. *arXiv* **2016**, arXiv:1607.00148.
17. Wu, Y.; Ni, J.; Cheng, W.; Zong, B.; Song, D.; Chen, Z.; Liu, Y.; Zhang, X.; Chen, H.; Davidson, S.B. Dynamic Gaussian Mixture based Deep Generative Model For Robust Forecasting on Sparse Multivariate Time Series. *Natl. Conf. Artif. Intell.* **2021**, *35*, 651–659. [CrossRef]
18. Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C.K.; Li, X.; Guan, C. Time-Series Representation Learning via Temporal and Contextual Contrasting. *arXiv* **2021**, arXiv:2106.14112.
19. Pan, Q.; Hu, W.; Chen, N. Two Birds with One Stone: Series Saliency for Accurate and Interpretable Multivariate Time Series Forecasting. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–27 August 2021; pp. 2884–2891.
20. Chen, T.; Yin, H.; Chen, H.; Wang, H.; Zhou, X.; Li, X. Online sales prediction via trend alignment-based multitask recurrent neural networks. *Knowl. Inf. Syst.* **2020**, *62*, 2139–2167. [CrossRef]
21. Hirche, M.; Farris, P.; Greenacre, L.; Quan, Y.; Wei, S. Predicting Under- and Overperforming SKUs within the Distribution–Market Share Relationship. *J. Retail.* **2021**, *97*, 697–714. [CrossRef]
22. Jiang, H.; Ruan, J.; Sun, J. Application of Machine Learning Model and Hybrid Model in Retail Sales Forecast. In Proceedings of the 2021 IEEE 6th International Conference on Big Data, Xiamen, China, 5–8 March 2021; IEEE: Toulouse, France, 2021; pp. 69–75.
23. Zhao, X.; Keikhosrokiani, P. Sales Prediction and Product Recommendation Model Through User Behavior Analytics. *Cmc-Comput. Mater. Contin.* **2022**, *70*, 3855–3874. [CrossRef]
24. Ntakolia, C.; Kokkotiis, C.; Moustakidis, S.; Papageorgiou, E. An explainable machine learning pipeline for backorder prediction in inventory management systems. In Proceedings of the 25th Pan-Hellenic Conference on Informatics, Volos, Greece, 26–28 November 2021; pp. 229–234.
25. Rohitash, C.; Shaurya, G.; Rishabh, G. Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction. *IEEE Access* **2021**, *9*, 83105–83123.
26. Matteo, S.; Fabio, D. Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. *Chaos Solitons Fractals* **2020**, *139*, 110045.
27. Ou-Yang, C.; Chou, S.C.; Juan, Y.C. Improving the Forecasting Performance of Taiwan Car Sales Movement Direction Using Online Sentiment Data and CNN-LSTM Model. *Appl. Sci.* **2022**, *12*, 1550. [CrossRef]
28. Paterakis, N.G.; Mocanu, E.; Gibescu, M.; Stappers, B.; van Alst, W. Deep learning versus traditional machine learning methods for aggregated energy demand prediction. In Proceedings of the IEEE Pes Innovative Smart Grid Technologies Conference, Turin, Italy, 26–29 September 2017; IEEE: Toulouse, France, 2017; pp. 1–6.
29. Loureiro, A.L.D.; Miguéis, V.L.; da Silva, L.F.M. Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decis. Support Syst.* **2018**, *114*, 81–93. [CrossRef]
30. Qi, Y.; Li, C.; Deng, H.; Cai, M.; Qi, Y.; Deng, Y. A Deep Neural Framework for Sales Forecasting in E-Commerce. In Proceedings of the Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 299–308.
31. Ma, S.; Fildes, R. Retail sales forecasting with meta-learning. *Eur. J. Oper. Res.* **2021**, *288*, 111–128. [CrossRef]
32. Yaşar, B. Comparison of Forecasting Performance of ARIMA LSTM and HYBRID Models for The Sales Volume Budget of a Manufacturing Enterprise. *Istanb. Bus. Res.* **2021**, *50*, 15–46.
33. Chen, I.F.; Lu, C.J. Demand Forecasting for Multichannel Fashion Retailers by Integrating Clustering and Machine Learning Algorithms. *Processes* **2021**, *9*, 1578. [CrossRef]
34. Azadi, Z.; Eksioglu, S.; Eksioglu, B.; Palak, G. Stochastic optimization models for joint pricing and inventory replenishment of perishable products. *Comput. Ind. Eng.* **2019**, *127*, 625–642. [CrossRef]
35. Perez, H.D.; Hubbs, C.D.; Li, C.; Grossmann, I.E. Algorithmic approaches to inventory management optimization. *Processes* **2021**, *9*, 102. [CrossRef]

36. Junyoung, C.; Caglar, G.; KyungHyun, C.; Yoshua, B. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
38. Shiyang, L.; Xiaoyong, J.; Yao, X.; Xiyou, Z.; Wenhu, C.; Yu-Xiang, W.; Xifeng, Y. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. *Adv. Neural Inf. Process. Syst.* **2019**, *471*, 5243–5253.
39. Nikita, K.; Lukasz, K.; Anselm, L. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451.
40. Fenglin, L.; Xuancheng, R.; Guangxiang, Z.; Chenyu, Y.; Xuewei, M.; Xian, W.; Xu, S. Rethinking and Improving Natural Language Generation with Layer-Wise Multi-View Decoding. *arXiv* **2020**, arXiv:2005.08081 .
41. Tanveer, M.; Gupta, T.; Shah, M. Pinball Loss Twin Support Vector Clustering. *Acm Trans. Multimed. Comput. Commun. Appl.* **2021**, *17*, 1–23. [CrossRef]
42. Guen, V.L.; Thome, N. Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models. *Neural Inf. Process. Syst.* **2019**, 377, 4189–4201.
43. Daifeng, L.; Kaixin, L.; Xuting, L.; Jianbin, L.; Ruo, D.; Dingquan, C.; Andrew, M. Improved sales time series predictions using deep neural networks with spatiotemporal dynamic pattern acquisition mechanism *Inf. Process. Manag.* **2022**, *59*, 102987.
44. Chikkakrishna, N.K.; Hardik, C.; Deepika, K.; Sparsha, N. Short-Term Traffic Prediction Using Sarima and FbPROPHET. In Proceedings of the IEEE India Conference, Rajkot, India, 13–15 December 2019; IEEE: Toulouse, France, 2019; pp. 1–4.
45. Gilbert, K. An ARIMA supply chain model. *Manag. Sci.* **2005**, *51*, 305–310. [CrossRef]