Online Purchase Prediction via Multi-Scale Modeling of Behavior Dynamics

Chao Huang[1] Xian Wu[1] Xuchao Zhang[3] Chuxu Zhang[1] Jiashu Zhao[2] Dawei Yin[2]
Nitesh V. Chawla[1]
University of Notre Dame[1] JD.com[2] Virginia Tech[3]
{chuang7, xwu9, czhang11, nchawla}@nd.edu xuczhang@vt.edu zhaojiashu1@jd.com yindawei@acm.org

ABSTRACT

Online purchase forecasting is of great importance in e-commerce, which is the basis of how to present personalized interesting product lists to individual customers. However, predicting online purchases is not trivial as it is influenced by many factors including: (i) the complex sequential transition regularities exhibited with time-dependent and high-order nature; (ii) the multi-scale temporal dynamics of user purchase behavior; (iii) arbitrary category dependencies. To address these factors, we develop a Graph Multi-Scale Pyramid Networks (GMP) framework to fully exploit users' latent behavioral patterns with both multi-scale temporal dynamics and arbitrary inter-dependencies among product categories. In GMP, we first design a multi-scale pyramid modulation network architecture which seamlessly preserves the underlying hierarchical temporal factors–governing users' purchase behaviors. Then, we develop a resolution-wise recalibration gating mechanism to automatically capture the importance of each scale-view representations. Finally, a context-graph neural network module is proposed to adaptively uncover complex dependencies among category-specific purchases. Extensive experiments on six real-world e-commerce datasets from two platforms demonstrate the superior performance of our method over state-of-the-art baselines across various settings.

ACM Reference Format:
Chao Huang[1] Xian Wu[1] Xuchao Zhang[3] Chuxu Zhang[1] Jiashu Zhao[2] Dawei Yin[2] Nitesh V. Chawla[1]. 2019. Online Purchase Prediction via Multi-Scale Modeling of Behavior Dynamics. In *Proceedings of ACM Conference (KDD'19)*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online purchase prediction is of great importance for a wide spectrum of user-centric applications in online retailing platforms, ranging from personalized recommender systems [34], user activity modeling [5] and resource management [10]. For instance, by knowing the categories of products, customers tend to purchase on a daily basis, online retailers can: (i) alleviate the information overload issue and help customers to meet a variety of their needs and tastes [37]; (ii) increase the profit for online retailers through increasing the traffic [32]. Hence, making predictions on customers' future online purchases is key to enhancing the experience and satisfaction of customers and online retailers. To facilitate this task, we seek to develop effective predictive models with the goal of forecasting purchase behavior of customers on each product category at the future time step given their historical purchase records.

KDD'19, Aug 2019, Anchorage,Alaska USA
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Intuitively, we can employ conventional time series forecasting techniques. However, the purchase sequences involve dynamic and non-linear temporal dependencies across time steps, which pose difficulties to many existing time series forecasting models–relying on the stationary and linear assumption of time series data, such as autoregressive integrated moving average (ARIMA) [20] and its variants (*e.g.*, seasonal-ARIMA [16]). To mitigate this issue, various types of non-linear deep neural network models (*e.g.*, LSTM [4] and GRU [1]) have been introduced to consider the time-varying sequential patterns. Nevertheless, a common drawback of the above approaches is that only one dimensional temporal dynamics is considered, which may not properly reflect the variation in many real world scenarios. In fact, real-world temporal patterns of customer purchase behavior are much more complicated, involving daily routines, weekly pattern, monthly periodicity [8], and even other personalized periodic transition regularities, which naturally form a type of multi-scale temporal dynamics. For example, in the scenarios that product lifespan usually vary among different categories (*e.g.*, daily necessities, electronic devices), purchase behavioral data is often exhibited with both time-dependent and multi-dimensional dependencies. This sheds light on the weakness of existing time series forecasting methods, and motivates us to capture multi-scale temporal dynamics in modeling customer purchase behavior that can result in more accurate prediction results.

In addition to the importance of considering multi-scale temporal patterns of online purchase behavior, another key dimension is to understand dependencies among product categories, to augment predictions with relevant context signals [19]. In real life, explicit and implicit dependencies among product categories are ubiquitous when users make online purchases [28]. For example, when a user in an online store is examining sports clothes for athletic activities, he/she might also be interested in buying fitness foods which are good for weight loss and building muscle. Shoppers may buy forks/knives/plates and beer together for a party. In such cases, purchases on different categories are no longer independent. If the context-aware relationships among categories are ignored and each individual category-specific purchases are treated as independent ones, it is likely that the modeling of users' purchase behavior is inaccurate, and thus the prediction performance is degraded.

We identify two key challenges of modeling online purchase behavior, which motivate the model design (as illustrated in Figure 1).

Pattern Fusion with Hierarchical Inter-Correlations. It is a significant challenge to learn a temporal representation which can comprehensively reflect time-ordered sequential patterns of users' purchase behavior from different scales (resolutions). Different scale

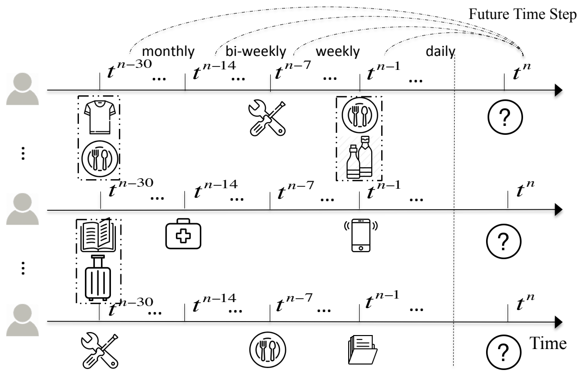


Figure 1: Illustration of the online purchase prediction problem with multi-scale temporal dynamics and arbitrary category dependencies.

views (*e.g.*, daily, weekly, bi-weekly, monthly) usually provide complementary information for model online purchase activities [37]. Additionally, each scale view may exhibit pairwise correlations or even higher-order cross-resolution correlations, and can be represented in a hierarchical way. If features are learned from different views separately and then loosely couple them together by directly concatenating the feature embeddings as the final representation, it cannot capture the cross-scale correlations and preserve the semantic multi-scale structural information of online purchase data. Hence, in order to learn meaningful purchase representations from long, broad and hierarchical temporal inputs, a robust across-scale temporal feature learning model with effective pattern fusion mechanism is needed in our studied problem.

Dynamic and Arbitrary Category Dependencies. Distinct to stationary inter-dependent relations, the process of inter-category influences on online purchase behavior is rather dynamic [31], since users' purchase preferences could change over time (*e.g.*, women may be no longer interested in pregnancy products after the childbirth). Prediction techniques for static scenarios can hardly be responsive to dynamic changes. Additionally, the dependencies across categories can be arbitrary since any pair of category-specific purchases could potentially be related in various online shopping scenarios [41]. For instance, a user can order snacks from a online store for a social party or outdoor activities. Users often make correlated purchases and exhibit different dependencies in choosing items of different categories due to his/her specialty. It is difficult to model such complex dependencies with predefined relation graphs. Therefore, to build effective online purchase predictive models with the context of category dependencies, it is crucial to generalize our framework to jointly capture dynamic and arbitrary category correlation structures in an adaptive way.

To address the aforementioned challenges, this work proposes a general and flexible framework–Graph Multi-Scale Pyramid Networks (GMP)–for online purchase prediction. Specifically, at the first stage, we design a multi-scale pyramid modulation network to model multi-resolution temporal factors that govern the sequential regularities of users' purchase behavior. The multi-scale pyramid architecture preserves the inter-dependent structures of evolving multi-dimensional temporal dependencies from low-level (locally) to high-level (globally). At the second stage, a resolution-wise recalibration gating mechanism is developed to promote the collaboration across different resolution views, and automatically capture the importance of contributed temporal patterns. Finally, a context-graph neural network module is proposed to handle dynamic dependencies among category-specific purchases. In this module, we leverage the graph convolution layers to explicitly model the arbitrary pairwise relationship among categories without predefined relation structures, which is able to adaptively aggregate global contextual information when modeling complementary purchase behavior.

In summary, we highlight our contributions as follows:

- We introduce a novel multi-scale pyramid modulation network architecture for predicting users' online purchases, which can explore the high-order and time-dependent structural correlations underlying multi-scale temporal dynamics of online purchase behavior in a hierarchical way.
- We develop a resolution-wise recalibration gating mechanism for fusing scale-specific pattern representations, and automatically capturing the importance of each scale-view in the predictive model. In addition, we propose a context-graph neural network module which is capable of adaptively uncover dynamic dependencies among categories.
- Extensive experiments on three real-world e-commerce datasets are performed along with comparisons to existing state-of-the-art predictive models, to demonstrate the advantages of our GMP model across various settings.

2 PROBLEM FORMULATION

In this section, we begin with some necessary notations and then formally present the online purchase forecasting problem in this paper. Suppose we have M users $U = \{u_1, ..., u_m, ..., u_M\}$ $(1 \leq m \leq M)$ and N project categories $C = \{c_1, ..., c_n, ..., c_N\}$ $(1 \leq n \leq N)$. We refer to an individual user as $u_m \in U$, an individual product category as $c_n \in C$, where m, n and t are defined as the index for the user, category and time step, respectively.

Definition 1. ***Purchase Matrix*** $\mathcal{X}_m$. *Given a window of T time steps (e.g., day), we define a matrix $\mathcal{X}_m \in \mathbb{R}^{N\times T}$ to represent the purchase records (i.e., category-time interactions) of each user u_m on each category $c_n \in C$ over time. The (n,t)-th entry of $\mathcal{X}^m$ is denoted as $x^m_{n,t}$. In particular, $x^t_{m,n} = 1$ if the m-th user have purchased items of n-th category at the t-th time step and $x^t_{m,n} = 0$ otherwise.*

Problem Statement. With the aforementioned definitions, online purchase prediction problem can be formulated as: given the purchase behavior data of each user (*i.e.*, $\mathcal{X}_m$) from previous T time steps, the objective is to learn a mapping function which predicts the unknown purchase behaviors of each user u_m on each product category c_n in h future time steps (*i.e.*, $x^{(T+p)}_{m,n}$).

$$x^{(T+h)}_{m,n} = F(\mathcal{X}_m \in \mathbb{R}^{N\times T}); \;\; (u_m \in U) \tag{1}$$

where $F(\cdot)$ is the mapping function we aim to learn.

3 METHODOLOGY

Generally speaking, our proposed GMP model consists of four components: 1) modeling temporal hierarchy with a multi-scale pyramid modulation network; 2) learning resolution-aware dynamic sequential patterns with a stacked convolutional recurrent

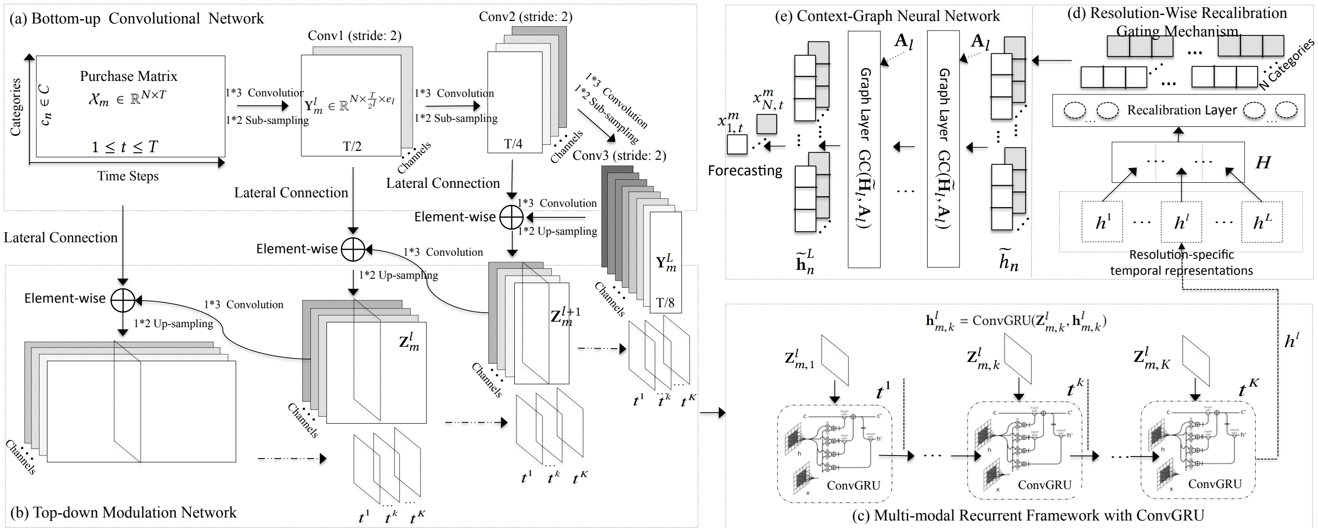


Figure 2: The Graph Multi-Scale Pyramid Networks (GMP) Framework. (a)-(b) The lateral connection leverages point-wise convolutional neural networks to transform feature representations (*i.e.*, $(\frac{T}{2^l} \times N \times e_l)$ to $(\frac{T}{2^l} \times N \times g_l)$) between bottom-up and top-down pathways. $\bigoplus$ represents the element-wise addition operation. (c) Each resolution-specific feature representation $\mathbf{Z}_m^l$ from l-th level of pyramid architecture will be fed into a ConvGRU encoder. We plot only one level of multi-modal recurrent framework, due to space limit. (d) $\mathbf{H} \in \mathbb{R}^{N \times (L \cdot d_c)}$ is a multi-resolution representation matrix which is concatenated from the encoded resolution-specific temporal representations $\mathbf{h}^l (1 \leq l \leq L)$.

encoder; 3) aggregating the multi-level temporal representations with a recalibration gating mechanism; 4) capturing implicit category dependencies with a context-graph neural network. Figure 4 presents the architecture of our GMP framework.

3.1 Multi-Scale Pyramid Modulation Network

We propose a multi-scale pyramid modulation network to capture the multi-grained temporal hierarchical structures of online purchase patterns. Following the pyramid architecture, we develop two multi-level convolutional networks, one (*i.e.*, bottom-up convolutional network) to encode the feature hierarchy of purchase patterns with respect to different time resolution levels, and another one (*i.e.*, top-down modulation network) to capture the semantic structures of feature maps across time resolutions.

3.1.1 **Bottom-Up Convolutional Network.** The bottom-up network is a hierarchical feedforwad convolutional framework that models temporal feature dynamics of user's online purchase patterns with multi-time granularity information. Each convolutional layer is able to generate high-level semantic representations of purchase patterns by increasing the receptive field of subsequent layers. In particular, given a user u_m, the bottom-up convolutional network takes the purchase matrix $\mathcal{X}^m \in \mathbb{R}^{N \times T}$ as input and perform convolutional operations on L levels (indexed by l) of time resolutions to generate feature representations $\mathbf{Y}_m^l \in \mathbb{R}^{N \times \frac{T}{2^l} \times e_l}$ in l-th level (e_l denotes the channel size of l-th layer), each of which operate at increasing temporal resolution. Formally, each convolutional layer can be represented as follows:

$$\mathbf{Y}_m^l = ReLU(\mathbf{W}_{bom}^l * \mathbf{Y}_m^{(l-1)} + \mathbf{b}_{bom}^l) \tag{2}$$

where , $*$ is the convolutional operation, $\mathbf{W}_{bom}^l$ and $\mathbf{b}_{bom}^l$ represents the transformation matrix and bias term in l-th layer of the bottom-up network, respectively. We output the feature representation on $\mathcal{X}^m$ of user u_m with highest level semantic signals of at the final layer, *i.e.*, $\mathbf{Y}_m^L$.

In the bottom-up network, we set the filter size of the convolutional operation as 3×1 at each layer with a stride of 2. The number of channels of in our bottom-up network is set as $(4, 16, 32, 64)$ for successive convolutional layers. After stacking L convolutional layers, the bottom-up network will generate intermediate latent representations (*i.e.*, $\mathbf{Y}_m^L$) of user u_m's purchase behavior on all categories from previous T time steps. In the process of feature map representation, to reduce the aliasing effect of sampling across dimensions with different semantics [30, 35], we perform convolutional operations on the time dimension and keep the input matrix with the same category dimension in each layer.

3.1.2 **Top-Down Modulation Network.** . In our multi-scale pyramid modulation architecture, the top-down modulation network starts from the last layer of the above bottom-up feedforward network. Then, we apply the stacked convolutional layers to transmit higher-level semantic signals into the feature representation learning process of purchase patterns with lower-level resolutions. More specifically, we also use convolutional neural network with upsampling operations to learn feature representations of each l-th layer $(1 \leq l \leq L)$ in the top-down architecture, and feed the $\mathbf{Y}_m^L$ as input $\mathbf{Z}_m^{(L+1)}$. The formulation of feature representation $\mathbf{Z}_m^l \in \mathbb{R}^{N \times \frac{T}{2^l} \times g_l}$ (g_l is the channel dimension in top-down network) for l-th layer can be given as:

$$\mathbf{Z}_m^l = ReLU(\mathbf{W}_{top}^l * \mathbf{Z}_m^{(l+1)} + \mathbf{b}_{top}^l) \tag{3}$$

where $\mathbf{W}_{top}^l$ and $\mathbf{b}_{top}^l$ are learned parameters. Furthermore, we enhance the feature learning of top-down framework with lateral connections between the bottom-up and top-down networks, which incorporates the resolution contextual signals to guide the feature learning of top-down network. Particularly, first, the lateral connection leverages point-wise convolutional neural networks to transform ($\frac{T}{2^l} \times N \times e_l$) from bottom-up architecture to ($\frac{T}{2^l} \times N \times g_l$), where g_l is the number of channels in top-down network. Then, the element-wise addition is applied to the same level of bottom-up and top-down layers. Therefore, The learned feature representations in which all levels (both low- and high-level resolutions) are semantically correlated with each other.

3.2 Multi-Modal Conv-Recurrent Encoder

Given the generated feature representations $\mathbf{Z}_m^l \in \mathbb{R}^{N \times T \times g_l}$ with tensor format from the above pyramid architecture, we develop a multi-modal convolutional recurrent encoder to model the resolution-aware temporal dynamics of purchase patterns across time steps. The ConvLSTM model has been introduced as a variant of LSTM for sequence modeling with both spatial-temporal information [33]. We consider GRU as the recurrent unit due to its computation efficiency without performance decay [2]. Our ConvGRU cell involves convolutional operations, which can be formally presented as:

$$\begin{aligned}
r_k &= \sigma(\mathbf{W}_{xr} * X_k + \mathbf{W}_{hr} * h_{k-1} + b_r) \\
z_k &= \sigma(\mathbf{W}_{xz} * X_k + \mathbf{W}_{hz} * h_{k-1} + b_z) \\
\mathbf{J}_k &= r_k \circ J_{k-1} + z_k \circ tanh(\mathbf{W}_{xc} * + \mathbf{W}_{hc} * h_{k-1} + b_c) \\
\mathbf{h}_k &= z_k \circ tanh(\mathbf{J}_k)
\end{aligned} \tag{4}$$

where r_k and z_k represents the outputs of reset gate and update gate at the k-th time step in recurrent neural network. $*$ is the convolutional operation. We denote the cell output at the k-th time step as J_k and the hidden state of a cell at the k-th time step as $\mathbf{h}_k$.

To model time-evolving temporal dependencies of multi-level feature representations from our pyramid modulation network, we develop a hierarchical convolutional recurrent encoder with stacked ConvGRU layers, to account for each resolution-specific feature representation $\mathbf{Z}_m^l$ of user u_m. In particular, with the learned feature representations $\mathbf{Z}_m^l$, we feed them into a stacked ConvGRU framework in which each ConvGRU layer encodes the resolution-specific sequential patterns corresponding to l-th layer in our pyramid modulation network. For simplicity, we denote the derivations of hidden vector representation $\mathbf{h}_k^l \in \mathbb{R}^{N \times d_c}$ for l-th pyramid scale as $\mathbf{h}_k^l = \text{ConvGRU}(\mathbf{Z}_m^l, \mathbf{h}_{k-1}^l)$. Formally, the multi-modal convolutional recurrent encoder can be given as: $\mathbf{h}_{m,k}^l = \text{ConvGRU}(\mathbf{Z}_{m,k}^l, \mathbf{h}_{k-1}^l)$.

The advantages of utilizing ConvGRU lie in: (i) the GRU part is able to capture temporal correlations across time steps; (ii) the convolution operator enables the consideration of topological information in $\mathbf{Z}_{m,k}^l$. An alternative way to consider temporal dependencies is to directly concatenate the vectors over category dimensions from representation matrix $\mathbf{Z}_{m,k}^l$. However, if we only considering single-dimensional data to encode non-linear temporal correlations, it is likely that a lot of contextual spatial signals would be lost and the learned temporal embeddings may not properly reflect the variation in purchases among categories.

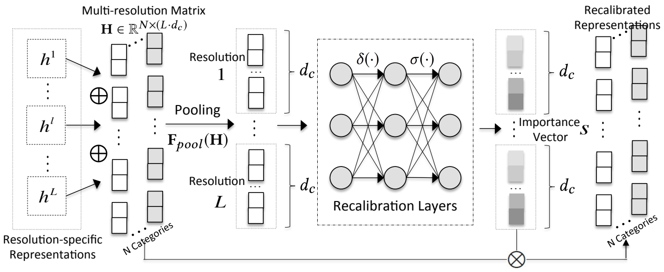


Figure 3: Illustration of the recalibration gating mechanism.

3.3 Resolution-Wise Gating Mechanism

In our GMP framework, the goal of the resolution-wise recalibration gating mechanism is to select the most informative components from the encoded resolution-specific temporal representations $\mathbf{h}^l (1 \leq l \leq L)$, and then aggregate the representation of informative resolution elements to characterize user's purchase patterns. In order to exploit the element dependencies over resolution dimension, we first concatenate $\mathbf{h}^l$ from each layer l to construct a multi-resolution representation matrix $\mathbf{H} \in \mathbb{R}^{N \times (L \cdot d_c)}$ as:

$$\mathbf{H} = \mathbf{h}^1 \oplus \ldots \oplus \mathbf{h}^l \oplus \ldots \oplus \mathbf{h}^L \tag{5}$$

where $\oplus$ represents the concatenation operation. Then, we apply global average pooling F_{pool} on $\mathbf{H}$ over N category dimensions to produce the summary of each element-wise ($[1, \ldots, p, \ldots, (L \cdot d_c)]$ indexed by p) representations as:

$$\varphi^{(p)} = \mathbf{F}_{pool}(\mathbf{H}) = \frac{1}{N} \sum_{n=1}^{N} H^{(p)} \tag{6}$$

where $\varphi^{(p)}$ denotes the p-th element in the pooling summarized vector $\varphi \in \mathbb{R}^{1 \times (L \cdot d_c)}$. We then propose a resolution-aware recalibration gating mechanism to recalibrate the information distribution among all fine-grained elements across resolutions, *i.e.*, $1 \leq q \leq (L \cdot d_c)$. We define s as the importance score vector to indicate the importance of all elements in $\mathbf{H}$. Formally, our gating mechanism can be represented as follows:

$$s = \sigma(\mathbf{W}_2 \cdot \delta(\mathbf{W}_1 \cdot \varphi)) \tag{7}$$

Here, $\mathbf{W}_1$ and $\mathbf{W}_2$ is the corresponding transformation matrix of two fully connected neural network layers. σ and δ dentes the Sigmoid and ReLU activation function, respectively. Finally, the aggregation process is given as follows:

$$\widetilde{h}_n = h_{n,1} \circ s_1 \uplus \ldots \uplus h_{n,l} \circ s_l \uplus \ldots \uplus h_{n,L} \circ s_L; l \in [1, \ldots, L] \tag{8}$$

where $\widetilde{h}_n$ denotes the summarized multi-resolution representation on category c_n and s_l is the sub-vector of s corresponding to the l-th resolution. $\uplus$ and $\circ$ is defined as the element-wise summation and multiplication, respectively. $s^{(p)}$ is the p-th entry of vector s.

Another intuitive way to fuse the resolution-specific patters is to combine resolution-specific patterns with attention mechanism. However, combining different feature representation $\mathbf{h}^l$ through a weighted summation fails to fully capture the inherent element-wise dependencies in a hierarchical manner [27]. To mitigate this problem, our lightweight recalibration gating mechanism is able

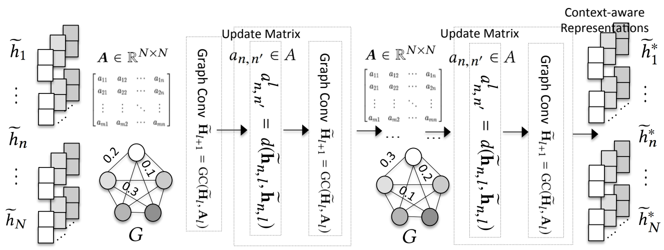


Figure 4: Illustration of the context-graph neural network.

to: 1) model the non-linear interactions between element channels; 2) enhance the representation ability of cross-resolution inter-dependencies with the constrain of mutually-exclusive relations among all element channels.

3.4 Context-Graph Neural Network

In this subsection, we show how to encode the contextual signals in our predictive solution GMP, by modeling the inter-dependencies of purchase behavior between different categories with graph convolution network. We first define the following input.

Definition 2. ***Category Graph*** *G. Our context-graph neural network is defined over the category graph $G = (V, E)$, where V and E is the set of all vertices and edges, respectively. Each vertex corresponds to a specific category c_n and each edge indicates the relationship between two categories.*

Definition 3. ***Adjacent Matrix*** *A. $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents the adjacency matrix whose entries (i.e., $a_{n,n'}$) is the correlations between categories (i.e., c_n and $c_{n'}$). In particular, $a_{n,n'}$ is estimated similarity between the learned summarized representation vector $\widetilde{h}_n$ and $\widetilde{h}_{n'}$ (encoded from the recalibration gating mechanism), i.e., $a^l_{n,n'} = exp^{-d(\widetilde{h}_{n,l}, \widetilde{h}_{n',l})}$, where $d(\cdot)$ represents the Euclidean distance [18] between $\widetilde{\boldsymbol{h}}_{n,l}$ and $\widetilde{\boldsymbol{h}}_{n',l}$.*

A graph convolution layer GC$(\cdot)$ receives an input $\widetilde{\mathbf{H}}_l \in \mathbb{R}^{N\times d_l}$ from l-th layer and produces $\widetilde{\mathbf{H}}_l \in \mathbb{R}^{N\times d_{l+1}}$ as:

$$\widetilde{\mathbf{H}}^*_{l+1} = \mathrm{GC}(\widetilde{\mathbf{H}}_l, \mathbf{A}_l) = \delta(\mathbf{A}_l \widetilde{\mathbf{H}}_l \mathbf{W}_g) \tag{9}$$

where $\mathbf{W}_g \in \mathbb{R}^{d_l \times d_{l+1}}$ denotes the feature transformation matrix. $\delta(\cdot)$ is a point-wise non-linearity ReLU activation function. In each l-th graph convolution layer, the adjacency matrix $\mathbf{A}_l$ is updated based on the new estimated vertex embedding vectors $\widetilde{h}_n$ ($n \in [1, ..., N]$). We further normalize the trainable adjacency matrix $\mathbf{A}$ with a stochastic kernel using a softmax along each row.

In our framework, we set the depth of our context-graph convolution network as the order of the graph diameter, so that all vertex contextual information from the the entire graph G could be incorporated in the learned vertex representation vectors. Instead of handcrafed category graph construction, our context-graph convolutional network enables the automatic discovery of meaningful and useful dependency features from category representations.

3.5 Model Inference

In general, our online purchase prediction can be regarded as a classification problem. We utilize cross entropy as the metric in our loss function which is defined as follows:

$$\mathscr{L}(\Theta) = -\sum_{(m,n,t)\in D} x^m_{n,t} \log \hat{x}^m_{n,t} + (1 - x^m_{n,t}) \log(1 - \hat{x}^m_{n,t}) \tag{10}$$

where Θ represents all learnable parameters in GMP. $\hat{x}^m_{n,t}$ denotes the estimated probability of user u_m purchase items of n-th category at t-th time step. Here, D denotes the set of observed purchase interactions in the training process. During the training phase, we use a Adam optimizer to infer model parameters by minimizing the loss function. In addition, we apply *Batch Normalization* [7] to reduce the internal covariance shift by transforming the input to zero mean/unit variance distributions in each mini-batch training.

Table 1: Statistics of the experimented datasets.

Dataset	# of Users	# of Categories	# of Purchases
Platform 1-City 1	3,362	12	470,634
Platform 1-City 2	11,593	12	2,127,351
Platform 1-City 3	59,476	12	3,377,133
Platform 2-Time 1	10482	24	135,815
Platform 2-Time 2	11970	24	143,547
Platform 2-Time 3	8276	24	89,334

4 EVALUATION

To comprehensively evaluate our proposed model, we perform experiments to answer the following questions:

- **Q1**: How is the overall prediction performance of our *GMP* as compared with various types of state-of-the-art methods across different categories?
- **Q2**: How does *GMP* perform compared with competitive approaches in forecasting individual category-specific purchases?
- **Q3**: How is the ranking-based performance of *GMP* in forecasting users' category-specific purchases?
- **Q4**: How do the different components (*e.g.*, resolution-wise recalibration gating mechanism and context-graph neural network) of *GMP* contribute to model performance?
- **Q5**: How does our *GMP* model work with different evaluated time resolution (*e.g.*, 1 day, 3days, and etc)?
- **Q6**: How do different hyper-parameter settings affect the performance of *GMP* (Refer to Appendix Section 7 for details)?
- **Q7**: How is the interpretation of our *GMP* framework in capturing dynamic category dependencies?

4.1 Experimental Settings

4.1.1 **Datasets.** We performed experiments with six purchase behavior datasets collected two real-world e-commerce platforms.

Platform 1: We experimented with three real-world online purchase datasets collected from the first online retailing platform in City 1, City 2 and City 3 within the time period from 01/01/2015 to 06/30/2018. Each online purchase record is in the format of (product category, user id, timestamp).

Platform 2: The other three datasets are collected from the second e-commerce site with three different time periods–09/01/2012 to 12/30/2013, 01/01/2013 to 03/03/2014, and 03/01/2013 to 06/30/2014. The format of each record is similar as that from platform 1.

These datasets were collected with different data scales (*i.e.*, the number of users and purchases), reflective of different degrees of online purchase activities. Table 1 summarizes the statistics about our experimented datasets. The details of data partition for training, validation, and testing are presented in Appendix Section 7.

4.1.2 **Methods for Comparison.** We compared it with the following state-of-the-art methods from various research lines (detailed baseline settings are presented in Appendix):

Recurrent Neural Network-based Methods: Since RNNs have shown their superiority in handling time-ordered sequential data as compared to conventional time series analysis methods (*e.g.*, ARIMA), we considered two developed variants of RNNs in the performance evaluation.

- **Sequential Prediction with Recurrent Neural Network (SP-RNN)** [40]: a deep learning approach which models the dependency on user's online behavior prediction via the recurrent neural network structure.
- **Stacked Long Short-Term Memory Model (ST-LSTM)** [36]: ST-LSTM is a mixture deep recurrent architecture on dynamic time series data by unifying stacked LSTM networks.

Attentive Recurrent Models: we further compare with another line of methods that models time-stamped data with the integration of attention mechanism and recurrent neural networks.

- **Dual-Stage Attentive Recurrent Networks (DA-RNN)** [21]: DA-RNN is a dual stage attentive time series prediction method which consists of an encoder with an input attention mechanism and a decoder with a temporal attention mechanism.
- **Attentive Bidirectional Recurrent Model (Dipole)** [17]: It models the temporal and high dimensional time series data by employing bidirectional recurrent neural networks and further interpreting learned representations with attention mechanisms.

Recommendations Techniques with Temporal Dynamics: In the performance comparison, we also include recommendation models which care about temporal drift effect.

- **Long- and Short-term Time-Series Network (LSTNet)** [12]: LSTNet combines the convolution neural network and the recurrent neural network to extract short-term local dependency patterns and to discover long-term patterns for time series trends.
- **Conv-Sequence Recommendation Model (Caser)** [26]: Caser first embeds a sequence of recent items into a tensor and then learns sequential patterns using convolutional filters.

Hybrid Multi-level Model: Finally, we compare our *GMP* with a hybrid time series analysis approach with multi-level wavelet decomposition networks.

- **Multilevel Wavelet Decomposition Network (mWDN)** [29]: mWDN is a wavelet-based neural network architecture which integrates the multilevel discrete wavelet decomposition and the frequency-aware LSTM for time series forecasting.

4.1.3 **Evaluation Protocols.** To fully measure the effectiveness of our *GMP* framework for online purchase prediction, we adopt three types of evaluation metrics:

- **Prediction across Categories**. We use *Marco-F1* and *Micro-F1* to evaluate the prediction accuracy across different product categories [14]. They indicate the overall performance across different classes (categories).
- **Prediction on Individual Categories**. We use *F1-score* (harmonic mean to balance precision and recall) and *Area Under Curve (AUC)* [3] as evaluation metrics for the accuracy of predicting purchases of each user on each individual category.

Table 2: Prediction results across different categories in terms of *Macro-F1* (Mac-F1) and *Micro-F1* (Mic-F1).

Data	Platform 1					
City	City 1		City 2		City 3	
Metrics	Mac-F1	Mic-F1	Mac-F1	Mic-F1	Mac-F1	Mic-F1
Caser	0.1994	0.1303	0.1969	0.1232	0.1423	0.0903
Dipole	0.2237	0.1490	0.1720	0.0626	0.1290	0.0634
LSTNet	0.2638	0.0784	0.3332	0.1325	0.0829	0.0393
SP-RNN	0.2100	0.1382	0.3050	0.1921	0.1467	0.0951
DARNN	0.2948	0.1537	0.1720	0.0626	0.0829	0.0393
ST-LSTM	0.2736	0.1823	0.208	0.1262	0.1514	0.0970
mWDN	0.3426	0.2270	0.2685	0.1717	0.1709	0.1638
GMP	**0.3799**	**0.2444**	**0.3531**	**0.2232**	**0.3118**	**0.1919**
Data	Platform 2					
Time	Time 1		Time 2		Time	
Metrics	Mac-F1	Mic-F1	Mac-F1	Mic-F1	Mac-F1	Mic-F1
Caser	0.0543	0.0253	0.0998	0.0675	0.1286	0.1011
Dipole	0.0639	0.0337	0.1391	0.0858	0.1285	0.1221
LSTNet	0.2770	0.0247	0.2226	0.1832	0.1460	0.1275
SP-RNN	0.3107	0.1826	0.3882	0.3362	0.3197	0.2346
DARNN	0.2979	0.1930	0.3873	0.3632	0.2277	0.2130
ST-LSTM	0.2997	0.1973	0.4234	0.3959	0.2342	0.2062
mWDN	0.3140	0.1798	0.4211	0.3932	0.3311	0.2958
GMP	**0.3408**	**0.2326**	**0.4373**	**0.4234**	**0.3830**	**0.3499**

- **Ranking-based Performance**. To assess the ranked list with the ground-truth user who has actual category-specific purchases, we adopt *Mean Average Precision (MAP)@k* and *Normalized Discounted Cumulative Gain (NDCG)@k* [9], where *MAP@k* computes the average precision of top-k ranked users and *NDCG@k* accounts for the position of hit for specific n-th category.

Note that all metrics are the higher the better.

4.1.4 **Reproducibility.** The parameter settings and implementation details are presented in Appendix (Section 7).

4.2 Overall Performance Comparison (Q1)

Table 2 shows both the forecasting accuracy across categories of different methods on six datasets in terms of Macro-F1 and Micro-F1. We summary key observations as follows:

(1) *GMP* achieves the best performance and obtains high improvements over different types of state-of-the-art methods in all cases. This sheds lights on the benefit of our model which jointly captures multi-scale temporal dynamics and arbitrary category dependencies. The performance is followed by mWDN which decomposes an online purchase series into a group of sub-series to capture multi-dimensional frequency factors. This further verifies the utility of considering multi-dimensional temporal information in predicting online purchases. However, mWDN fails to consider high-order inter-dependencies across time resolutions. In contrast, *GMP* dynamically learning cross-level semantics from data, which shows remarkably flexibility and superiority.

(2) With the increase of the number of users and purchases, the performance improvement of *GMP* compared with other baselines also increases. The reason may be that the learned feature representations are more informative for a relatively larger scale of

Table 3: Purchase prediction results on individual categories in terms of *F1-score* and *AUC* on Platform1-City 2 dataset.

Category	Density	Metrics	Caser	Dipole	LSTNet	SP-RNN	DARNN	ST-LSTM	mWDN	*GMP*
Beauty & Care	sp = 2.9%	F1-score	0.0440	0.0704	0.0584	0.0590	0.0295	0.1173	0.1899	**0.2439**
		AUC	0.7311	0.7424	0.7574	0.7284	0.7433	0.7555	0.7646	**0.7832**
Clothing & Shoes	sp = 3.83%	F1-score	0.2396	0.2592	0.2607	0.2291	0.2736	0.1505	0.2201	**0.2942**
		AUC	0.7049	0.7357	0.7308	0.7181	0.7227	0.7490	0.7490	**0.8212**
Computers & Office	sp = 1.25%	F1-score	0.1820	0.1935	0.1874	0.1790	0.1548	0.2283	0.2050	**0.2807**
		AUC	0.6888	0.7007	0.7052	0.6904	0.6729	0.7049	0.6984	**0.7420**
Electronics	sp = 3.16%	F1-score	0.3189	0.3418	0.3461	0.3277	0.3279	0.3755	0.1302	**0.4307**
		AUC	0.7562	0.7745	0.7464	0.7603	0.7551	0.7855	0.7901	**0.8233**
Food & Grocery	sp = 3.65%	F1-score	0.3511	0.3891	0.5146	0.3684	0.4959	0.4512	0.5743	**0.5884**
		AUC	0.6949	0.7008	0.7227	0.6959	0.7101	0.7022	0.6968	**0.7370**
Fresh Food	sp = 1.85%	F1-score	0.0997	0.1274	0.0896	0.1131	0.0689	0.1587	0.1361	**0.1516**
		AUC	0.6305	0.6484	0.6603	0.6352	0.6365	0.6607	0.6591	**0.6832**
Health & Medicine	sp = 1.61%	F1-score	0.1010	0.1088	0.1804	0.1077	0.1652	0.1227	0.1191	**0.1270**
		AUC	0.6894	0.7151	0.7328	0.6983	0.7267	0.7307	0.7383	**0.7788**
Home & Furniture	sp = 1.97%	F1-score	0.1847	0.2321	0.3442	0.2037	0.3804	0.3267	0.2301	**0.4045**
		AUC	0.7380	0.7563	0.7712	0.7418	0.7610	0.7626	0.7672	**0.7956**
Luggage & Gift	sp = 3.29%	F1-score	0.1896	0.2102	0.2546	0.1998	0.3204	0.2268	0.1645	**0.3440**
		AUC	0.6619	0.6812	0.6864	0.6695	0.6874	0.6893	0.6832	**0.7249**
Mother & Baby	sp = 2.11%	F1-score	0.1430	0.1447	0.1002	0.1361	0.1100	0.1452	0.1159	**0.1515**
		AUC	0.6108	0.6431	0.6520	0.6241	0.6469	0.6553	0.6542	**0.7190**
Travel & Outdoors	sp = 1.38%	F1-score	0.0436	0.0617	0.1382	0.0566	0.1562	0.0892	0.1957	**0.2611**
		AUC	0.7197	0.7412	0.7603	0.7207	0.7571	0.7548	0.7682	**0.8016**
Toys & Instruments	sp = 0.96%	F1-score	0.1179	0.1684	0.2426	0.1422	0.1274	0.2062	0.2790	**0.3172**
		AUC	0.6487	0.6613	0.6975	0.6545	0.6771	0.6750	0.6680	**0.7266**

purchase activities, in which the multi-scale dynamics of online purchase behavior is more obvious. In addition, there is no obvious winner among attentive recurrent models and temporal recommendation techniques. This again confirms that only considering data dependencies and interactions from singular temporal dimension is insufficient to model complex sequential purchase transition regularities exhibited with multi-scale behavior dynamics.

(3) For all recurrent neural network-based and hybrid deep learning methods, we also conduct experiments with more recurrent neural network layers to have more parameters in the time series data fitting scenario, but have not observed clear improvement. The potential reason is that the input sequence is of low-dimension (single or limited predefined dimensions) and larger/deeper recurrent networks tend to overfit [39].

4.3 Category-Specific Prediction Accuracy (Q2)

We investigate the effectiveness of *GMP* in predicting users' purchases on each individual category and report the evaluation results in Table 3. We can observe that our *GMP* achieves the best performance in all forecasting cases. Additionally, another interesting observation is that obvious improvements can also be obtained by *GMP* in predicting purchases of sparser categories. This observation indicates that *GMP* is capable of effectively modeling latent correlations from different contextual modalities (time and category), to alleviate the data sparsity issue in forecasting sparse online purchases. Due to space limit, we only report the results on Platform 1-City 2 data. Similar results can be observed on other datasets.

4.4 Ranking Performance Comparison (Q3)

Furthermore, we also measure the ranking quality of top-k predicted users for future category-specific purchase of our *GMP* method with varying k from 200 to 1000. The evaluation results (measured by *MAP@k* and *NDCG@k*) of *GMP* and the best performed baseline (mWDN) on Platform 1-City 2 data traces are presented in Figure 6 and Figure 5. We can observe that *GMP* achieves the best performance under different values of k, which suggests that our *GMP* model assigns higher score to the true users in the top-k ranked list and hit the ground truth at top positions.

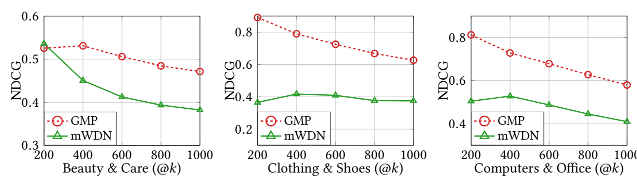


Figure 5: Purchase prediction results on individual categories in terms of *NDCG@k*.

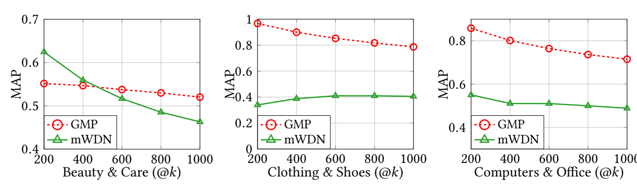
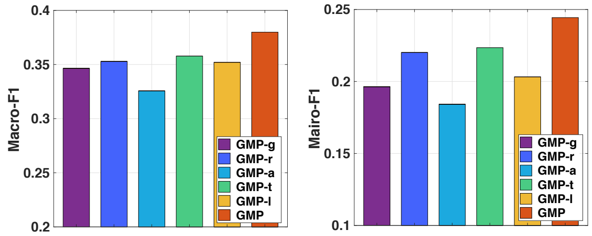


Figure 6: Purchase prediction results on individual categories in terms of *MAP@k*.

4.5 Component-Wise Evaluation of *GMP* (Q4)

In our evaluation, we consider five variants of the proposed method corresponding to different analytical aspects:



(a) Macro-F1 (b) Micro-F1

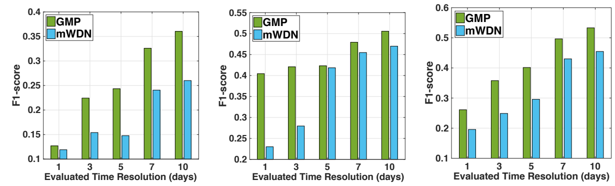
Figure 7: Evaluation on GMP variants.

- **Effect of Category Dependencies** *GMP*-g: A simplified version of *GMP* which does not include context-graph neural module to consider dependencies among categories.
- **Effect of Recalibration Gating Mechanism**. *GMP*-r: A simplified version of *GMP* which does not include the recalibration gating mechanism to model element-wise correlations from encoded resolution-aware representations.
- **Aliasing Effect in Pyramid Modulation Network**. *GMP*-a: During the process of feature map generation in the bottom-up convolutional networks, we perform the convolutional operation with 3×3 filter size to directly consider category correlations using the local structure information.
- **Impact of Modeling Cross-level Semantics**. *GMP*-t: We only reply on the bottom-up convolutional network to encode the underlying multi-dimensional structures of evolving temporal dependencies, *i.e.*, without the top-down modulation network in the multi-scale pyramid architecture.
- **Effectiveness of Encoded Multi-Scale Temporal Dynamics**. *GMP*-l: To further evaluate the quality of the latent representations learned from our designed multi-scale pyramid networks, we first perform flatten operation on the learned resolution-specific temporal representations and then feed them into an integrative framework of LSTM and MLP for making predictions.

We report the evaluation results in Figure 7. We can notice that the full version of our developed framework *GMP* achieves the best performance in all cases, which suggests: (i) the effectiveness of *GMP* in capturing the category-aware inter-relations between users' online purchases in a dynamic environment; (ii) the efficacy of the designed recalibration gating mechanism in handling hierarchical structural relations among resolution-aware purchase patterns; (iii) the rationality of *GMP* to address the aliasing effect in the feature representation process from different modalities; (iv) the effectiveness of the top-down modulation networks for helping *GMP* transmit high-level temporal semantics back to low-level latent representations; (v) *GMP* could generate good embeddings to capture and incorporate the underlying structural and semantic relationships between various types of sequential transition regularities. As such, it is necessary to build a joint framework to capture multi-scale temporal dynamics and category dependencies for predicting purchase behavior of users.

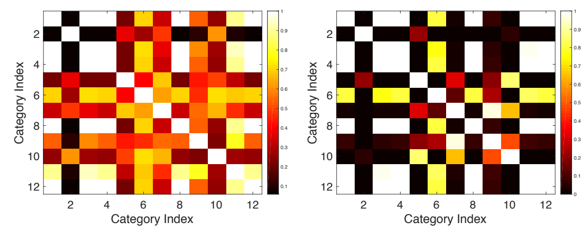
4.6 Effect of Evaluated Time Resolution (Q5)

To further investigate the robustness of *GMP*, we evaluate the model performance with different resolutions of evaluated time step (*i.e.*, from the finest resolution–1 day to the coarsest resolution–10 days) as shown in Figure 8. We can observe that *GMP* consistently outperforms the best performed baseline mWDN with respect to different



(a) Health & Medicine (b) Home & Furniture (c) Travel & Outdoors

Figure 8: Prediction results on individual categories v.s. time resolution of target period.



(a) Adjacent Matrix A from 1^{st} Layer (b) Adjacent Matrix A from 2^{nd} Layer

Figure 9: The adjacent matrix A across different categories updated from the first and second layer in our *GMP*. Category index is consistent with the order in Table 3.

evaluated resolutions for all category cases, which demonstrates the robustness of our *GMP* framework in various purchase prediction scenarios with different granularity of target time period.

4.7 Case Study (Q7)

Apart from the superior forecasting performance, another key advantage of *GMP* is its ability in interpreting the dependencies across different categories in predicting online purchases. To demonstrate this, we perform case studies to show the explainability of our framework by visualizing the dynamic adjacent matrix A (dependency weights between categories) as shown in Figure 9. *GMP* will update the adjacent matrix A through each layer. We could observe that *GMP* enables the dynamic modeling of dynamic and arbitrary correlations between different categories.

5 RELATED WORK

Neural Next-Item Recommendation. Many deep neural network-based frameworks for next-item recommendation seek to model item-item transition among successive items [11, 13, 38]. For instance, Yuan *et al.* [38] studied the next-item recommendation problem by considering both short- and long-range item dependencies. Kang *et al.* [11] proposed a self-attention based sequential model to make next item predictions based on relatively few items. To model user's sequential patterns. complementary to next-item recommendation applications, accurate and reliable prediction of online purchase activities can help sequential recommender systems to generate more relevant items for ranking and offer more effective personalized recommendations.

Time Series Data Forecasting. Most of conventional time series forecasting techniques are based on the predefined kernel functions or prior distributions [25], such as the Auto-regressive Integrated Moving Average (ARIMA) [20] and Support Vector Regression (SVR) [24]. Although these approaches work well for time series analysis in a static scenario, they are less applicable to capture the dynamic patterns among time series data. To address this problem, RNNs-based methods (*e.g.*, LSTM [31] and GRU [2]) were proposed and have shown in their success in modeling sequential data. Additionally, further attempt–the integration of CNN-RNN network structure and attention mechanisms–was made to adaptively identify values relevant from relevant time steps for making forecasting [21, 29]. However, most of the existing deep neural network methods failed to pay attention to multi-levels of temporal dynamics in the time series of user purchases, which is the key concern of this work.

Time-stamped Behavior Modeling. There exists a good amount of research work in modeling various behavioral time-stamped data [1, 15, 22]. For example, Lian *et al.* [15] aimed to explore user web activities with the consideration of high-order feature interactions. Qiu *et al.* [22] investigated user social activities using their local network information. Online purchases are different from the above time-stamped human behavior. The sequential transition regularities of purchase patterns is often exhibited with both high-ordered time-dependent and temporal hierarchy nature, which pose difficulties to comprehensively explore costumer purchases with a multi-dimensional structure.

6 CONCLUSION

This work contributes a new framework, named GMP for online purchase prediction via modeling behavior dynamics from both multi-scale temporal patterns and arbitrary category dependencies. Particularly, we develop a multi-scale pyramid neural network architecture to explore the high-order correlations underlying multi-resolution online purchase patterns. In addition, we propose a recalibration gating mechanism that is tailored to cooperate with a hierarchical recurrent framework for multi-resolution pattern fusion. With the help of context-graph neural network module, we further consider contextual signals in encoding complex relations between category-specific purchases. Finally, we perform extensive experiments on three real-world datasets and evaluation results shown that the proposed model significantly outperforms the state-of-the-art methods. There are two future directions for our work. *First*, it is interesting to investigate other auxiliary cross-domain information (*e.g.*, user demographics) in modeling users' preferences. *Second*, the framework GMP is general, and it is feasible to apply it on more various time-stamped data.

REFERENCES

[1] Bokai Cao, Lei Zheng, Chenwei Zhang, Philip S Yu, Andrea Piscitello, John Zulueta, Olu Ajilore, Kelly Ryan, and Alex D Leow. 2017. DeepMood: modeling mobile phone typing dynamics for mood detection. In *KDD*. ACM, 747–755.

[2] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *ICML*. 2067–2075.

[3] Yuxiao Dong, Jing Zhang, Jie Tang, Nitesh V Chawla, and Bai Wang. 2015. Coupledlp: Link prediction in coupled networks. In *KDD*. ACM, 199–208.

[4] Cong Du, Peng Shu, and Yong Li. 2018. CA-LSTM: Search Task Identification with Context Attention based LSTM. In *SIGIR*. ACM, 1101–1104.

[5] Jie Feng, Yong Li, Chao Zhang, and etc. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *WWW*. ACM, 1459–1468.

[6] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[7] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. ACM, 355–364.

[8] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *SIGIR*. ACM, 355–364.

[9] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *KDD*. ACM, 1531–1540.

[10] Diane J Hu, Rob Jall, and Josh Attenberg. 2014. Style in the long tail: discovering unique interests with latent variable models in large scale social e-commerce. In *KDD*. ACM, 1640–1649.

[11] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE, 197–206.

[12] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and etc. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*. ACM, 95–104.

[13] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In *KDD*. ACM, 1734–1743.

[14] Defu Lian, Kai Zheng, Vincent W Zheng, Yong Ge, Longbing Cao, Ivor W Tsang, and Xing Xie. 2018. High-order Proximity Preserving Information Network Hashing. In *KDD*. ACM, 1744–1753.

[15] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *KDD*.

[16] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 2013. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *TITS* 14, 2 (2013), 871–882.

[17] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *KDD*. ACM, 1903–1911.

[18] Calvin R Maurer, Rensheng Qi, and Vijay Raghavan. 2003. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *TPAMI* 25, 2 (2003), 265–270.

[19] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *KDD*. ACM, 785–794.

[20] Bei Pan, Ugur Demiryurek, and Cyrus Shahabi. 2012. Utilizing real-world transportation data for accurate traffic prediction. In *ICDM*. IEEE, 595–604.

[21] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *IJCAI*.

[22] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, and etc. 2018. DeepInf: Social Influence Prediction with Deep Learning. In *KDD*. ACM, 2110–2119.

[23] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. 2014. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *JMLR* 15, 1 (2014), 335–366.

[24] Goce Ristanoski, Wei Liu, and James Bailey. 2013. Time series forecasting using distribution enhanced linear regression. In *PAKDD*. Springer, 484–495.

[25] Nicholas I Sapankevych and Ravi Sankar. 2009. Time series prediction using support vector machines: a survey. *CIM* 4, 2 (2009).

[26] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. ACM, 565–573.

[27] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *ICML*. 3156–3164.

[28] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *CIKM*. ACM, 1133–1142.

[29] Jingyuan Wang, Ze Wang, and etc. 2018. Multilevel wavelet decomposition network for interpretable time series analysis. In *KDD*. ACM, 2437–2446.

[30] Maurice Weiler, Mario Geiger, and etc. 2018. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *NIPS*. 10402–10413.

[31] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. ACM, 495–503.

[32] Liang Wu, Diane Hu, Liangjie Hong, and etc. 2018. Turning Clicks into Purchases: Revenue Optimization for Product Search in E-Commerce. In *SIGIR*. ACM.

[33] Shi Xingjian, Zhourong Chen, and etc. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*. 802–810.

[34] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommener systems. In *IJCAI*. 3203–3209.

[35] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition.. In *IJCAI*, Vol. 15. 3995–4001.

[36] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep learning: A generic approach for extreme condition traffic forecasting. In *SDM*. SIAM, 777–785.

[37] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *WWW*. ACM, 649–658.

[38] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *WSDM*. ACM.

[39] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014).

[40] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks.. In *AAAI*, Vol. 14. 1369–1375.

[41] Dawei Zhou, Jingrui He, and etc. 2018. Sparc: Self-paced network representation for few-shot rare category characterization. In *KDD*. ACM, 2807–2816.

7 APPENDIX

7.1 Training/Testing Data Split

7.1.1 Platform 1. For training/test data split, we use the data from 01/01/2015 to 01/01/2018 for training, data from 01/02/2018 to 01/15/2018 as validation, and the data from 01/16/2018 to 06/30/2018 is used for testing.

7.1.2 Platform 2. For the three datasets from platform 2, we summarize the training/test data split in Table 4.

Table 4: Training/test data split for the datasets from Platform 2.

Data	Training
Platform 2-Time 1	09/01/2012 to 09/01/2013
Platform 2-Time 2	01/01/2013 to 01/01/2014
Platform 2-Time 3	03/01/2012 to 03/01/2014
Data	Validation
Platform 2-Time 1	09/02/2013 to 09/15/2013
Platform 2-Time 2	01/02/2014 to 01/15/2014
Platform 2-Time 3	03/02/2014 to 03/14/2014
Data	Test
Platform 2-Time 1	09/16/2013 to 12/30/2013
Platform 2-Time 2	01/16/2014 to 03/30/2014
Platform 2-Time 3	03/16/2014 to 06/30/2014

Table 5: Parameter Settings

Parameter	Value
Input Series length:	128
# of Time Steps for Conv_LSTM:	5
Hidden State Dimensionality:	32
Embedding Dimension:	32
Batch Size:	64
Learning Rate:	1e-3

7.2 Implementation Details

We implemented all the deep learning baselines and the proposed *GMP* framework with Tensorflow [1]. For training models, we divided the datasets into training, validation and testing set in chronological order. The validation set is used to select the best parameter values. The partition details of training/testing time windows for each dataset have been clarified in the dataset description. For the sake of fair comparison, all prediction experiments are conducted across the consecutive time steps (*i.e.*, day) in the test data (refer to data descriptions for details) and the average performance is reported. Furthermore, we repeated all the approaches 10 times and reported the average performance.

7.3 Parameter Settings

In our experiments, we set the dimension of hidden representation and sequence length in convolutional recurrent encoder as 32 and 5, respectively. We have the channel size of $[2^2, 2^4, 2^5, 2^6]$ which corresponds to the process of high-level feature map generation (with a larger latent representation space). The number of channels in the top-down modulation network is set as 32. For hyperparameter initialization, we respectively utilized the Glorot initialization [6]. For each method, the grid search is applied to find the optimal settings of hyperparameters using the validation set. The early stopping [23] is adopted to terminate the training process based on the validation performance. After the parameter tuning on all baselines, we reported their best performance in the evaluation results. During the model learning process, we used the Adam optimizer for gradient-based model optimization, where the batch size and learning rate were set as 64 and 0.001, respectively.

[1] The code is available at https://github.com/graphmp.

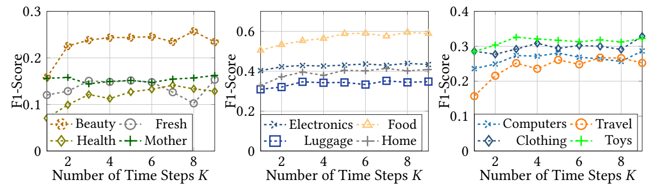


Figure 10: Hyperparameter Studies *w.r.t* the number of time steps K. Categories with similar performance range are presented in the same figure.

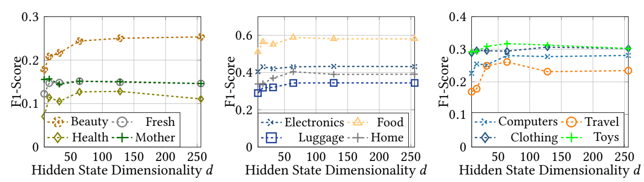


Figure 11: Hyperparameter Studies *w.r.t* hidden state dimensionality d. Categories with similar performance range are presented in the same figure.

7.4 Hyperparameters Studies (Q6)

To investigate the robustness of *GMP* framework, we examine how the different choices of parameters affect the performance of *GMP* in the prediction performance. Except for the parameter being tested, we set other parameters at the default values (see Table 5). Figure 10 and Figure 11 show the evaluation results (measured by Macro-F1 and Micro-F1) as a function of one selected hyperparameter when fixing others. Overall, we observe that *GMP* is not strictly sensitive to these parameters and is able to reach high performance under a cost-effective parameter choice, which demonstrates the robustness of our *GMP* framework. More specifically, there are two interesting insights from the result:

- As depicted in Figure 10, the performance of *GMP* becomes better as the number of encoded time steps K increases, which is not surprising since we consider a larger sequence length to capture the evovling temporal patterns. Furthermore, as K increases, the prediction performance remains stable when $K = 5$. One potential reason is that when considering larger sequence length, more parameters need to be learned in the recurrent network architecture. As a result, the training of *GMP* becomes harder. In our experiments, we set $K = 5$.

- Furthermore, from Figure 11, we can observe that the increase of prediction performance saturates as the representation dimensionality reaches around 64. This is because: at the beginning, a larger value of hidden state dimensionality d brings a stronger representation power for our representation learning framework, but the further increase of dimension size of latent representations to encode purchase patterns from multi-scale behavior dynamics might lead to the overfitting issue. In our experiments, we set the dimension size as 64, due to the consideration of the trade-off between the effectiveness and computational cost.