BASM: A Bottom-up Adaptive Spatiotemporal Model for Online Food Ordering Service

Boya Du, Shaochuan Lin*, Jiong Gao, Xiyu Ji, Mengya Wang, Taotao Zhou, Hengxu He*, Jia Jia*, Ning Hu

Alibaba Group
Hangzhou&Shanghai, China
{Boya.dby, lin.lsc, jionggao.gg, jixiyu.jxy, mia.wmy, taotao.zhou, hengxu.hhx, jj229618, huning.hu}@alibaba-inc.com

Abstract—Online Food Ordering Service (OFOS) is a popular location-based service that helps people order what they want. Compared with traditional e-commerce recommendation systems, users’ interests may be diverse under different spatiotemporal contexts, leading to various spatiotemporal data distributions, which increases the difficulty of model learning. However, numerous current works simply mix all samples to train a set of model parameters, which makes it challenging to capture the diversity in different spatiotemporal contexts. Therefore, we address this challenge by proposing a Bottom-up Adaptive Spatiotemporal Model (BASM) to adaptively fit the spatiotemporal data distribution, further improving the fitting capability of the model. Specifically, a spatiotemporal-aware embedding layer performs weight adaptation on field granularity in feature embedding to achieve the purpose of dynamically perceiving spatiotemporal contexts. Meanwhile, we propose a spatiotemporal semantic transformation layer to explicitly convert the concatenated input of the raw semantic to the spatiotemporal semantic, which can further enhance the semantic representation under different spatiotemporal contexts. Furthermore, we introduce a novel spatiotemporal adaptive bias tower to capture diverse spatiotemporal bias, reducing the difficulty of modeling spatiotemporal distinction. To further verify the effectiveness of BASM, we propose two new metrics, Time-period-wise AUC (TAUC) and City-wise AUC (CAUC). Extensive offline evaluations on public and industrial datasets are conducted to demonstrate the effectiveness of our proposed model. The online A/B experiment also further illustrates the practicability of the model online service. This proposed method has now been implemented on Ele.me, a major online food ordering platform in China, serving more than 100 million online users.

Index Terms—Online Food Ordering Services, Spatiotemporal Recommendation System, Click-Through Rate Prediction, Metric

I. INTRODUCTION

Online Food Ordering Service (OFOS), a convenient location-based service for ordering, delivering, or picking up food via a smartphone or website, has become popular in recent years. Nowadays, online food ordering platforms, such as Grubhub, DoorDash, Meituan, and Ele.me, serve millions of users every day. For instance, in the U.S., users place more than 400 million online orders on the DoorDash food ordering platform in Q2 of 20221.

* Corresponding author.

1https://ir.doordash.com/financials/quarterly-results/default.aspx

Fig. 1. The overview of one typical online food ordering service implementation process.

An example of one typical online food ordering service implementation process can be seen in Fig. 1. Users’ interests may vary in various spatiotemporal contexts, such as cities and hours, leading to various spatiotemporal data distributions, which makes learning a model more challenging. Therefore, how to adaptively fit this complex spatiotemporal data distribution is a challenging problem. From the perspective of model parameters, the existing work on spatiotemporal data modeling can be divided into static parameter-based methods [1]–[9] and dynamic parameter-based methods [10]–[20]. The static parameter-based methods adopt a set of model parameters to fit the entire data, while dynamic parameter-based methods exploit a set of adaptive model parameters or maintain multiple sets of model parameters simultaneously to realize the data’s dynamic modeling. In this paper, adaptive is synonymous with dynamic. Adaptive emphasizes the model’s capability to fit the data, while dynamic emphasizes the change of model parameters.

For static parameter-based methods, existing works mainly focus on the extraction of spatiotemporal interest and enhancement of spatiotemporal information. Among them, the spatiotemporal periodic user behavior modeling methods [3], [5] divide and combine user behavior sequences through spatiotemporal information and realize the extraction of users’ periodic spatiotemporal interests. TRISAN [2] enhances the representation of spatiotemporal information by constructing a triangular relationship between user geographic location, item geographic location, and user behavior click time. Long-
sequence user behavior modeling methods [6]–[9] extract users’ long-term interest preferences by extending the length of user behavior sequences. AutoInt [1] enriches the representation of input information by automatically interacting from multiple subspaces.

For dynamic parameter-based methods, the adaptive model parameter methods [16]–[20] achieve adaptive learning of the model by maintaining a set of dynamic parameters. M2M [16] constructs a meta-unit to generate tower weights in different scenarios to realize dynamic modeling of multi-scenario data. APG [20] realizes self-wise adaptive modeling by building a parameter generation network and parameter adaptor. The multi-parameter learning methods [10]–[15] achieve the fitting of different data distributions by maintaining multiple sets of model parameters simultaneously. STAR [11] realizes adaptive modeling of multi-scene data distribution by constructing shared and domain-specific parameters. ADI [13] further optimizes the fusion mechanism between shared and domain-specific parameters to improve the feature-level domain adaptation.

However, in online food ordering services, users have different preferences in different spatiotemporal contexts, resulting in significant differences in spatiotemporal data distribution. Taking the Ele.me recommendation platform as an example, as shown in Fig. 2, the data distribution (exposure and Click-Through-Rate (CTR)) would be varied at different times (e.g., 6 a.m vs. 6 p.m) and different locations (e.g., City 1 vs. City 2). Static parameter-based methods simply mix all samples to train a set of model parameters, making it challenging to capture diversities in different spatiotemporal contexts. Meanwhile, dynamic parameter-based methods ignore spatiotemporal factors’ effects or rely heavily on predefined parameter spaces, such as scenarios and crowds. In OFOS, spatiotemporal factors, as one of the dominant factors in user decision-making, play an essential role in spatiotemporal data modeling. Moreover, spatiotemporal scenarios are difficult to enumerate because the Cartesian product from temporal (i.e., hour, time-period, weekday) and spatial (i.e., geohash, aoi, city) granularity is enormous. Thus, it is difficult to predefine different spatiotemporal parameter spaces.

To echo the above challenge, which we refer to as spatiotemporal data distribution, we propose Bottom-up Adaptive Spatiotemporal Model, BASM for short, adaptive modeling of spatiotemporal data from the bottom embedding layer, middle semantic layer, and top classification bias tower. Specifically, at the bottom of the information input, we propose the Spatiotemporal-Aware Embedding Layer, which performs weight adaptation from the field granularity to dynamically perceive spatiotemporal contexts. In this case, the spatiotemporal weights of features are dynamically scaled as the spatiotemporal context changes. Concatenating the raw semantic inputs obtained from the above aware features, we further propose a Spatiotemporal Semantic Transformation Layer. The raw semantic is explicitly converted to spatiotemporal semantic to enhance the semantic representation under different spatiotemporal contexts. Meanwhile, we novelly propose a Spatiotemporal Adaptive Bias Tower, which modulates the parameters of the raw fully-connected layer and batch normalization layer through spatiotemporal information to realize the adaptive modeling of the spatiotemporal bias, thereby reducing the difficulty of spatiotemporal distinction modeling. To further verify the effectiveness of BASM, we also propose two new metrics, Time-period-wise Area Under the Curve (TAUC) and City-wise Area Under the Curve (CAUC), which can be seen in Section. III. Comprehensive experiments on public and industrial datasets demonstrate the superiority of our proposed method compared to state-of-the-art methods, and the online A/B test illustrates the effectiveness and efficiency of BASM.

In summary, the main contributions of this work can be listed as follow:

- We novelly propose a Bottom-up Adaptive Spatiotemporal Model (BASM) to characterize various spatiotemporal distributions, further improving the fitting capability of the model.
- We realize adaptive modeling of spatiotemporal data from the bottom embedding layer, middle semantic
layer, and top classification bias tower by proposing the Spatiotemporal-Aware Embedding Layer, the Spatiotemporal Semantic Transformation Layer, and the Spatiotemporal Adaptive Bias Tower.

- To further verify the effectiveness, we propose two new metrics, time-period-wise AUC (TAUC) and city-wise AUC (CAUC), which are used to measure the ranking capability when the time-period and city factors are omitted.

- Extensive experiments on an industrial dataset with more than 80 million users and a large-scale public dataset demonstrate the effectiveness of BASM, further validated by online A/B tests on industrial spatiotemporal scenarios. BASM has now been successfully deployed on Ele.me, one of the major online food ordering platforms in China, achieving a CTR improvement of 6.51%.

II. PROPOSED METHOD

In this section, we introduce the problem of spatiotemporal data distribution and detail the Bottom-up Adaptive Spatiotemporal Model (BASM), with its overall architecture illustrated in Fig. 3. BASM proposes the Spatiotemporal-Aware Embedding Layer, the Spatiotemporal Semantic Transformation Layer, and the Spatiotemporal Adaptive Bias Tower, which comprehensively solves the above problems from the bottom embedding layer, the middle semantic layer, and the top classification tower, respectively.

| Field                      | Detailed Features                                      |
|----------------------------|--------------------------------------------------------|
| User Feature               | User ID, Basic Profiles, User’s Exposures/Clicks/Orders/CTR/CVR for Item/Category in the last 1/3/7/30/90 days |
| User Behavior Sequence     | Item ID, Brand, General Category, Price, Time-Period, Hour, City ID, Weekday. |
| Candidate Item Feature     | Item ID, General Category, Brand, Location, Exposures/Clicks/Orders/CTR/CVR for Candidate Item in the last 1/3/7/30/90 days. |
| Spatiotemporal Context Feature | Time-period, Hour, Weekday, Geohash, City ID, District ID, AOI ID. |
| Combined Feature           | Some Hand-selected Cross-features between Users and Items (e.g., gender, category). |

A. Problem Formulation

In this paper, to better maintain and manage the features in our application, we classify the entire input features and declare each class as a field. As shown in TABLE I, let $\mathcal{X}$ represent the input data, which consists of the spatiotemporal context feature field $c$, the user feature field, the user behavior sequence field, the candidate item feature field, and the combined feature field. Note that the recommendation system collectively refers to several features with the same attributes as field [21]. For example, features on the item side are collectively referred to as the item field. To better observe the effects of spatiotemporal features, in the following, we refer to other features except spatiotemporal context features as $o$, which is $x_i = (o_i, c_i)$, where $x_i \in \mathcal{X}$ and $i$ represent the instance. $\mathcal{Y}$ denotes the click label. Formally, the spatiotemporal data distribution problem can be formulated by,

$$y_i = \mathcal{F}_\theta(x_i)$$

$$= \mathcal{F}_\theta(o_i, c_i)$$

where $y_i \in \mathcal{Y}$, $\mathcal{F}$ is the deep network and $\theta$ is the parameters of the deep network. Different from the e-commerce scenario, spatiotemporal context features $c$, as one of the main features of online order recommendation systems, play an important role in spatiotemporal data modeling. Moreover, spatiotemporal scenarios are difficult to enumerate, and it isn’t easy to predefine different spatiotemporal parameter spaces. Therefore, we propose to solve the spatiotemporal data distribution problem by dynamically adapting the parameter $\theta$ under various spatiotemporal factors $c$. Below we will introduce our model module by module.

B. Spatiotemporal-Aware Embedding Layer (StAEL)

As with other deep models, we first encode each discrete class feature into a high-dimensional one-hot vector. For the $i$th feature, its one-hot encoding is denoted as:

$$v_i = \text{onehot}(i)$$

where $v_i \in \mathbb{R}^N$ is a vector with one at the $i$th entry and zero elsewhere, and $N$ is the number of all unique features. We then embed the sparse and high-dimensional one-hot encoding vectors into dense and low-dimensional vectors, more suitable for neural network inputs. In particular, we define a learnable embedding matrix $E \in \mathbb{R}^{D \times N}$ and project the $i$th feature to its corresponding embedding vector:

$$e_i = Ev_i$$

| Symbol | Description |
|--------|-------------|
| $N$    | The number of all unique features. |
| $n$    | The number of feature fields. |
| $o$    | The spatiotemporal context feature. |
| $r$    | All other features except spatiotemporal context features. |
| $e_i$  | Embedding vector of the $i$th feature. |
| $h_j$  | Hidden representation of the $j$th feature field. |
| $h_c$  | Hidden representation of the spatiotemporal context feature field. |
| $\alpha_j$ | Spatiotemporal weight of the $j$th feature field learning by StAEL. |
| $W_{\text{stl}}, b_{\text{stl}}$ | Dynamic parameter weights and bias learned by StTIL. |
| $W_{\text{bias}}^{(m-1)}, b_{\text{bias}}^{(m-1)}$ | Dynamic bias of Fusion FCNs learning by Spatiotemporal Adaptive Bias Tower. |
| $W_{\text{bias}}^{(m)}, b_{\text{bias}}^{(m)}$ | Dynamic bias of Fusion BNs learning by Spatiotemporal Adaptive Bias Tower. |
| $y_i, y$ | The click label and prediction of instance $i$. |
| $\sigma()$ | A non-linear activation function. |
| $[;]$ | Concatenation operator. |
Fig. 3. Bottom-up Adaptive Spatiotemporal Model (BASM) architecture overview. It consists of three modules: the Spatiotemporal-Aware Embedding Layer (described in Section II-B), the Spatiotemporal Semantic Transformation Layer (described in Section II-C), and the Spatiotemporal Adaptive Bias Tower (described in Section II-D).

Fig. 4. The architecture of Spatiotemporal-Aware Embedding Layer. It applies a gate attention mechanism to perform weight adaptation from the field granularity.

where $e_i \in \mathbb{R}^D$, $D$ is the dimension, much smaller than $N$.

Most present approaches overlook the variations in the distribution of spatiotemporal data, provide the same feature relevance in various spatiotemporal scenarios, and are unable to perform feature weight adaptation, which restricts the model’s perception of varied spatiotemporal contexts. For example, during mealtimes, users are more concerned about the price of food, so that the price feature will be more important. In contrast, during the time-period of afternoon tea, users are more inclined to browse around and compare multiple shops, so the category feature of the shop will be more concerned. Uniform feature weights cannot dynamically perceive such spatiotemporal changes. What’s more, it is impossible to maintain a set of embedding for each spatiotemporal scenario. Because this method not only needs to maintain a large number of parameters but also is difficult to fit the few-shot distribution. Typically, due to the large number $N$ of unique features in industrial recommendation systems, more than hundreds, it is prohibitively expensive to learn different importance weights for each feature. This not only brings a huge amount of computation but also reduces the fitting effect of the model due to too much noise.
Therefore, we dynamically characterize each field-wise feature embedding according to the spatiotemporal context feature \( c \), and each field representation is defined as:

\[
h_j = \alpha_j x_j
\]

(5)

where \( x_j = [e_{1j}; \ldots; e_{kj}] \) denotes the \( j^{th} \) field embedding, which is the concatenation of all feature embeddings in it. \( k \) is the total number of features in the \( j^{th} \) field. \( \alpha_j \) is a spatiotemporal weight assigned to \( x_j \), indicating its importance in current spatiotemporal context. \( \{ j \in \mathbb{Z} : 0 \leq j < n \} \), where \( n \) denotes the number of feature fields. \( h_j \) denotes the adaptive representation of the \( j^{th} \) field of features.

Our Spatiotemporal-Aware Embedding Layer aims to calculate the appropriate spatiotemporal weight \( \alpha_j \). A naïve way is \( \alpha_j = 1/n \). That is, each feature field shares the same weight. However, this is obviously not a wise choice because the importance of features cannot be adapted to various spatiotemporal contexts. Illustrated in Fig. 4, we apply a gate attention mechanism to extract the spatiotemporal weight over different spatiotemporal contexts, which can be seen as follows:

\[
\alpha_j = 2 + \sigma(W_p x_j + b_p)
\]

(6)

where \( \sigma \) denotes the activation function, which is the Sigmoid function in our application. \( x_j \) and \( x_c \) represent the embeddings of the other feature field and spatiotemporal context features, respectively. [;] is the concatenation operator, and \( W_p \) and \( b_p \) are attention parameters. The spatiotemporal weight is activated by the Sigmoid function and then multiplied by 2 to scale the weight between 0 and 2. The purpose of this is mainly to ensure that different features are strengthened (i.e., > 1) or weakened (i.e., < 1) under different spatiotemporal contexts. Zero-value initialization is used to guarantee initial weights of 1. By adopting such a weight adaptation method, we can achieve the purpose of dynamically perceiving spatiotemporal contexts at the bottom layer.

C. Spatiotemporal Semantic Transformation Layer (StSTL)

After obtaining the dynamic spatiotemporal embedding by the above aware layer, we can obtain the raw semantics by a simple concatenation \( \hat{h} = [h_0; \ldots; h_{n-1}] \). The model’s capacity to fit spatiotemporal data can be improved further if spatiotemporal contextual differences are added to the original semantics. Hence, we propose to explicitly transform the raw semantic into the spatiotemporal semantic by bringing dynamic network parameters, illustrated in Fig. 5. The spatiotemporal semantic here denotes the feature representation in vector space that can better distinguish differences in different spatiotemporal contexts.

Inspired by [16], [18], a meta network is utilized to achieve semantic transformation purposes. The raw semantic is relevant to all behaviors of users, ignoring the contextual information between different behaviors. Therefore, we filter user behaviors by the current spatiotemporal context and introduce a method of dynamic network parameters to characterize user preferences in the current spatiotemporal context. Specially, we exploit the time-period and geohash to filter the user’s historical behavior and obtain the current spatiotemporal filtering behavior \( u_i \). According to our data analysis, the coverage and average length of the filtered sequence can reach 92.94% and 58.77, respectively. In industrial recommendation systems, sequence modeling solutions (e.g., DIN [22], DIEN [23]) usually deal with behavioral sequence data of length ~50. Therefore, the sparsity problem of the filtered sequence has little effect. It is worth mentioning that after filtering, \( u_i \) can better represent the interests of users in different spatiotemporal contexts. Then, we concatenate the spatiotemporal context embedding \( h_c \) and spatiotemporal filtering behavior embedding \( h_{stl} \) to obtain dynamic parameter weights \( W_{stl} \) and dynamic parameters bias \( b_{stl} \) through a meta network,

\[
W_{stl} = \text{Reshape}(W_w[h_c; u_i] + b_w)
\]

(7)

\[
b_{stl} = \text{Reshape}(W_b[h_c; u_i] + b_b)
\]

(8)

where Reshape is used to convert the output embedding of the meta network into a dynamic weight matrix. \( W_w \) and \( b_w \) are the weight and bias of meta network, so as \( W_b \) and \( b_b \). \( W_{stl} \) and \( b_{stl} \) are the dynamic weight and dynamic bias from meta network output.

After generating the dynamic weight matrix and bias vector, the raw input semantic \( \hat{h} \) is transformed into spatiotemporal semantic output,

\[
h_{stl}^* = W_{stl} \hat{h} + b_{stl}
\]

(9)

where \( h_{stl}^* \) is the spatiotemporal semantic output. Unlike traditional linear mapping methods lacking spatiotemporal
adaptation, we implement different mapping functions under different spatiotemporal contexts through a meta-learning network. In this way we achieve an explicit transformation of raw semantic to spatiotemporal semantic, enhancing the semantic representation of different spatiotemporal contexts.

D. Spatiotemporal Adaptive Bias Tower (StABT)

In OFOS, there are natural differences in CTR under different times and locations, and we call this phenomenon spatiotemporal bias. The spatiotemporal bias can be seen in Fig. 6. We can find that the user’s click tendency varies at different times and locations, which cannot be ignored in the spatiotemporal data modeling. As shown in Fig. 7, in order to capture diverse spatiotemporal bias given different spatiotemporal scenarios, we novelly present a Spatiotemporal Adaptive Bias Tower, which consists of Fusion Fully Connected layers (Fusion FCs) and Fusion Batch Normalization layers (Fusion BNs).

Fusion FCs: Static parameters-based FCs are easily dominated by the strong distribution and will cause the weak distribution to be submerged. Therefore, we exploit spatiotemporal context features to construct spatiotemporal specific parameters to modulate the weights of FCs to better characterize spatiotemporal bias given different spatiotemporal scenarios. The spatiotemporal dynamic parameters of FCs modulation can be obtained as follows:

\[
W_{bias}^{(m)} = \sigma \left( W_{dfc,1}^{(m)} h_c + b_{dfe,1}^{(m)} \right) \tag{10}
\]

\[
b_{bias}^{(m)} = \sigma \left( W_{dfe,2}^{(m)} h_c + b_{dfe,2}^{(m)} \right) \tag{11}
\]

where \( W_{bias}^{(m)} \) and \( b_{bias}^{(m)} \) are the spatiotemporal modulation parameters for FC. \( W_{dfe,1}^{(m)}, b_{dfe,1}^{(m)}, W_{dfe,2}^{(m)} \) and \( b_{dfe,2}^{(m)} \) are the network weights that output this modulation parameters. \( \{m \in \mathbb{Z} : 1 \leq j \leq L\} \), and \( L \) is the layers of the tower. Our dynamic bias modulation for FCs is obtained by:

\[
h_{fusion}^{(0)} = h_{stl}^{*} \tag{12}
\]

where \( \circ \) is Hadamard product, \( W_t^{(m-1)} \) is the layers of the tower. Our and \( \gamma \) are learnable parameters, \( \beta_t^{(m)} \) and \( \beta_{bias}^{(m)} \) are also the four and \( \beta^{(m)} \) denote the variance. The raw BN can be shown as follows:

\[
h_{fc}^{(m)} = \sigma \left( W_{bias}^{(m)} \circ W_{t}^{(m)} h_{bias}^{(m-1)} + \left( b_{bias}^{(m)} + \beta_t^{(m)} \right) \right), \forall m \in 1, 2, \ldots, L \tag{13}
\]

where \( \circ \) is Hadamard product, \( W_t^{(m)} \) and \( \beta_t^{(m)} \) are the MLP weights to be modulated. By this means, we implement Fusion FCs, which enable dynamic modeling under spatiotemporal bias by modulating the weights of FCs.

Fusion BNs: Batch normalization (BN) has been proven effective and widely applied in deep learning. Let \( \mu \) denote the mean value of input \( X \), while \( \sigma^2 \) represents the variance. The raw BN can be shown as follows:

\[
\hat{X} = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{14}
\]

where \( \gamma \) and \( \beta \) are learnable parameters, \( \epsilon \) is a small number to avoid the denominator being 0. Raw BN requires the input \( \hat{X} \) to satisfy the assumption of independent and identical distribution (i.i.d), which works well in a single scenario. However, the data distribution corresponding to each spatiotemporal context is diverse in our spatiotemporal scenario. The raw BN confuses the distributional differences across different spatiotemporal contexts from all samples. Therefore, we need to utilize the spatiotemporal features to modulate the raw BN,

\[
\gamma_{bias}^{(m)} = \sigma \left( W_{dbn,1}^{(m)} h_{fc} + b_{dbn,1}^{(m)} \right) \tag{15}
\]

\[
\beta_{bias}^{(m)} = \sigma \left( W_{dbn,2}^{(m)} h_{fc} + b_{dbn,2}^{(m)} \right) \tag{16}
\]

\[
h_{fusion}^{(m)} = \gamma_{bias}^{(m)} h_{fc}^{(m)} - \mu, \beta_t^{(m)} + \beta_{bias}^{(m)} \tag{17}
\]

where \( W_{dbn,1}^{(m)}, b_{dbn,1}^{(m)}, W_{dbn,2}^{(m)} \) and \( b_{dbn,2}^{(m)} \) are also the four network weights. Eq. 10, Eq. 11, Eq. 15 and Eq. 16 are collectively named FCN bias in Fig. 7. \( \gamma_{bias}^{(m)} \) and \( \beta_{bias}^{(m)} \) represent modulated parameters. \( \gamma^{(m)} \) and \( \beta^{(m)} \) denote the origin learnable BN parameters. By the above approach, we achieve fusion BNs by modulating the parameters of BNs and realizing
the dynamic normalization in different spatiotemporal data distributions. Experiments show that this method is more beneficial for spatiotemporal data modeling.

After Spatiotemporal Adaptive Bias Tower, our final predicted click probability can be presented as:

\[ \hat{y}_i = \sigma(W_o \hat{x}_i + b_o) \]  

where \( \hat{x}_i \) represents the final output of Spatiotemporal Adaptive Bias Tower of instance \( i \). Then we update the entire network by minimizing the binary cross-entropy loss with the prediction \( \hat{y}_i \) and the label \( y_i \):

\[ \min_\theta \sum_i -y_i \log \hat{y}_i - (1 - y_i) \log (1 - \hat{y}_i) \]  

(19)

III. EXPERIMENT

In this section, we evaluate both the offline model and the whole online system. Extensive comparisons and ablation studies are conducted to answer the following questions:

- **RQ1.** How does our proposed BASM performs compared to state-of-the-art methods?
- **RQ2.** Where does the performance gain come from?
- **RQ3.** How efficient are BASM and comparison methods?
- **RQ4.** How does BASM perform on the online platform?
- **RQ5.** Can BASM cope with the before-mentioned challenges in the spatiotemporal recommendation scenario?

A. Experimental Settings

1) **Datasets:** To validate the performance of BASM, a takeaway industrial dataset Ele.me and a public spatiotemporal recommendation dataset Spatiotemporal Public Data\(^2\) are selected. (1) **Ele.me:** Ele.me is collected from an industrial recommendation platform, Ele.me. It has more than 80 million users and 2 billion samples. 45-day samples are used for training, and the samples of the following day are used for testing. (2) **Spatiotemporal Public Data:** It contains a total of over 170 million data, including 7 days of training data and 1 day of testing. The statistics of the datasets are shown in TABLE III.

2) **Comparison Methods:** We select three representative static parameter-based methods and three dynamic parameter-based methods for comparison to verify the effectiveness of our method on spatiotemporal data. (1) **Wide&Deep** [24]: Wide&Deep is an classic recommendation model for feature interactions. It jointly trains a wide linear model for memorization and a deep neural network for generalization. (2) **DIN** [22]: DIN concentrates on the diversity and local activation of user interests. A local activation unit extracts user interests from their extensive history behaviors. (3) **AutoInt** [1]: AutoInt utilizes multiple layers of the multi-head self-attention neural networks to simulate various orders of feature combinations, which can automatically learn the high-order feature interactions. (4) **STAR** [25]: STAR uses a shared factorized network and a domain-specific factorized network in each domain to predict CTR for enumerated domains. In this paper, we use the time-period as the scenario division indicator, divided into breakfast, lunch, afternoon tea, dinner, and night, with 5 scenarios in total. (5) **M2M** [26]: M2M designs multiple meta units to learn the correlation between scenes and the characteristics of specific scenes to accomplish multi-task and multi-scenario modeling. In our experiments, we use spatiotemporal information as the input of the meta unit to achieve dynamic modeling of spatiotemporal data. (6) **APG** [20]: APG realizes self-wise adaptive modeling by building a parameter generation network and parameter adaptor to improve the performance of CTR prediction.

3) **Evaluation Metrics:** In our experiment, we use Area Under the Curve (AUC), Normalized Discounted Cumulative Gain(NDCG3, NDCG10), and Logloss, widely used in recommendation systems. In addition, in order to further evaluate the fitting capability of the model on the spatiotemporal data, we proposed TAUC(Time-period-wise AUC) and CAUC(City-wise AUC) as our additional metrics. The computations of TAUC and CAUC are defined as follows:

- **TAUC:** TAUC is the weighted average value of AUC for each time-period, it measures the quality of ranking in each time-period:

\[ \text{TAUC} = \frac{\sum_{t=1}^{T} \text{impression}_t \times \text{AUC}_t}{\sum_{t=1}^{T} \text{impression}_t} \]  

(20)

where \( \text{impression}_t \) is the number of exposures at \( t^{th} \) time-period and \( \text{AUC}_t \) is the AUC of \( t^{th} \) time-period.

- **CAUC:** CAUC is the weighted average value of AUC for each city, it measures the quality of ranking in each city:

\[ \text{CAUC} = \frac{\sum_{c=1}^{C} \text{impression}_c \times \text{AUC}_c}{\sum_{c=1}^{C} \text{impression}_c} \]  

(21)

where \( \text{impression}_c \) is the number of exposures at \( c^{th} \) city and \( \text{AUC}_c \) is the AUC of \( c^{th} \) city.

TAUC is to calculate the AUC of each time-period, and then get the weighted average. It can reduce the impact of meaningless comparisons of sorting results in different time-periods, because the distribution of data in different time-periods varies greatly, as shown in Fig. 2. For example, the CTR is naturally high during the high-activity period, and naturally low during the low-activity period. When evaluating AUC, the CTR of different time-periods will be mixed, making it impossible to evaluate the ranking effect of the same time-period accurately. The same is true for CAUC, which is evaluated from the city dimension. We evaluate from temporal and spatial dimensions, which can accurately measure the ranking ability of the model in different spatiotemporal dimensions.

4) **Experimental environment and parameter settings:** Models in this paper are implemented with Tensorflow 1.12 in Python 2.7 environment. Every comparison model in this paper is implemented using the Alibaba Onlien-learning Platform (AOP), a framework developed by Alibaba Group. AOP supports distributed training tasks written in Tensorflow style. We conduct distributed training with 40 parameter servers and

\(^2\)https://tianchi.aliyun.com/dataset/dataDetail?dataId=131047
TABLE III
THE BASIC STATISTICS OF DATASETS. # DENOTES THE NUMBER, AND ML INDICATES THE AVERAGE LENGTH OF USER BEHAVIOR SEQUENCES.

| Datasets                  | Total Size | #Feature | #Users | #Items | #Clicks | ML of User Behaviors |
|---------------------------|------------|----------|--------|--------|---------|----------------------|
| Ele.me                    | 2380427866 | 417      | 81086293 | 547354 | 86735276 | 58.77                |
| Spatiotemporal Public Data| 177114244  | 38       | 14427689 | 7446116 | 3140831  | 41.19                |

TABLE IV
OFFLINE PERFORMANCE COMPARISON WITH STATIC AND DYNAMIC PARAMETER-BASED STATE-OF-THE-ARTS ON PUBLIC AND INDUSTRIAL DATASETS. THE BEST RESULTS ARE IN BOLD.

| Methods              | AUC | TAUC | CAUC | NDCG3 | NDCG10 | Logloss |
|----------------------|-----|------|------|-------|--------|---------|
| Ele.me               |     |      |      |       |        |         |
| Wide&Deep            | 0.7037 | 0.7022 | 0.7016 | 0.8219 | 0.8410 | 0.1376 |
| DIN                  | 0.7327 | 0.7313 | 0.7302 | 0.8309 | 0.8482 | 0.1376 |
| AutoInt              | 0.7308 | 0.7295 | 0.7285 | 0.8272 | 0.8502 | 0.1346 |
| STAR                 | 0.7331 | 0.7321 | 0.7265 | 0.8300 | 0.8523 | 0.1343 |
| M2M                  | 0.7333 | 0.7319 | 0.7308 | 0.8296 | 0.8514 | 0.1345 |
| APG                  | 0.7339 | 0.7326 | 0.7315 | 0.8301 | 0.8519 | 0.1344 |
| BASM                 | 0.7373 | 0.7360 | 0.7348 | 0.8323 | 0.8535 | 0.1339 |

400 workers without relying on GPU. Each parameter server has 16 CPU cores with 64 GB RAM, and every worker has 16 CPU cores and 64 GB memory. AdagradDecay [27] is chosen as our optimizer for model training. We train all models using a warm-up [28] strategy to speed up convergence and improve model performance. Especially, the learning rate gradually increases from 0.001 to 0.012 during 0 and 1M steps. The activation function of the neural network is set to LeakyReLU, and the batch size is set to 1024. In addition, we averaged the metrics of all studies after five repetitions.

B. RQ1: Offline Performance Comparison

TABLE IV indicates that BASM outperforms all State-of-The-Art (SOTA) methods in all metrics on both industrial and public datasets. In particular, among our two proposed metrics (TAUC and CAUC), our method achieves state-of-the-art results in comparison with other methods, proving that BASM has a better modeling effect on spatiotemporal data. Specifically, on Spatiotemporal Public Data, BASM achieves a further significant improvement\(^3\) compared to the best comparison method M2M, i.e., AUC +0.46%, TAUC +0.39%, and CAUC +0.25%.

Compared with static parameter-based methods (Wide&Deep, DIN, and AutoInt), dynamic parameter-based methods (STAR, M2M, APG, and ours) significantly improve, because static parameter-based methods simply mix all samples to train a set of model parameters, which makes it difficult to capture diversities in different spatiotemporal contexts.

However, current dynamic parameter-based methods either ignore the effects of spatiotemporal factors or rely heavily on predefined parameter spaces, limiting the ability of models to fit spatiotemporal data. For instance, our model has gained 0.42% and 0.61% AUC improvement compared to STAR on Ele.me and Spatiotemporal Public Data, respectively. STAR

\(^3\)Note that the 0.1% absolute AUC gain is already considerable in practical application [1], [20], [22], [24].

Fig. 8. Statistics and importance heatmap from StAEL over different time-periods. (a) Statistics on user clicks, orders over time-periods, i.e., showing user activity. (b) The heatmap depicts the spatiotemporal weight \(\alpha_j\) of each feature field over different time-periods. (Best viewed in color)
mainly models an exclusive network for each domain, but it isn’t easy to enumerate all domains for spatiotemporal scenarios. The predefined scenario division limits the performance of STAR in spatiotemporal data. Compared to M2M and APG in Spatiotemporal Public Data, our model improves by 0.46% and 0.98% in AUC and decreases by 0.002 and 0.0006 in Logloss, respectively. On the one hand, this is due to the fact that the dynamic parameters do not thoroughly investigate the influence of spatiotemporal factors, and their direct application will result in poor interpretability. On the other hand, M2M and APG do not consider explicit debiasing solutions for distributions in various spatiotemporal scenarios.

It demonstrates that 1) for characterizing differences in spatiotemporal distributions, it is crucial to strengthen spatiotemporal factors as dynamic parameters; 2) the spatiotemporal scenarios are difficult to enumerate, and the method of dynamic modeling with multiple sets of parameters is challenging; 3) in OFOS, there are natural differences in CTR under different times and locations (spatiotemporal bias), which cannot be ignored in spatiotemporal data modeling.

C. RQ2: Ablation Study and Visualization

To demonstrate the effectiveness of each BASM module, we have conducted ablation studies on Ele.me. The result is illustrated in TABLE V.

1) Spatiotemporal-Aware Embedding Layer: This module has outstanding performance in TAUC, CAUC, and AUC, as shown in Table V, the Spatiotemporal-Aware Embedding Layer improves TAUC by +0.10%, CAUC by +0.08%, AUC +0.10%. It shows that the dynamic perception of different spatiotemporal contexts in the feature embedding layer of the model, which provides significant help for the subsequent task learning of the network.

2) Spatiotemporal Semantic Transformation Layer: We remove the Spatiotemporal Semantic Transformation Layer (w/o StSTL) from BASM to evaluate the performance. We observe that when it is removed, the AUC fall by 0.18% and the Logloss increase by 0.0208. This shows that adding this module can further enhance the semantic representation of different spatiotemporal contexts. Nonetheless, we observe that removing this module has a much larger drop in effect than the other two modules, suggesting the importance of semantic transformation for modeling spatiotemporal data.

3) Spatiotemporal Adaptive Bias Tower: In order to capture diverse spatiotemporal bias, we introduce Spatiotemporal Adaptive Bias Tower to reduce the difficulty of spatiotemporal distinction modeling. According to TABLE V, the removal of this module (w/o StABT) results in 0.24% reduction in AUC, 0.26% in TAUC, and 0.25% in CAUC, validating the effectiveness of the Spatiotemporal Adaptive Bias Tower.

4) Visualization about the spatiotemporal weight $\alpha_j$: In addition to proving the effectiveness of Spatiotemporal-Aware Embedding Layer (StAEL), we verified whether our gated attention could learn reasonable adaptive weights to perceive spatiotemporal context. We recorded the spatiotemporal weights $\alpha_j$ learned by StAEL in different spatiotemporal scenarios and visualized the heatmap of $\alpha_j$ in different time-periods and cities, shown in Fig. 8 and 9, respectively.

From Fig. 8(a), we can see that the averages of users’ clicks and orders are higher for lunch and dinner compared to others, indicating that users are more active at lunch and dinner periods.
Fig. 10. The t-SNE visualization of instance features generated by the Base model (variation of DIN) and BASM over different time-periods. (Best viewed in color)

dinner. Fig. 8(b) shows that, at lunch and dinner, our StAEL learns higher spatiotemporal weights $\alpha_j$ for the user field, user behavior sequence field, and combined field. This confirms that StAEL assigns higher spatiotemporal weights to user-side features during active periods, which is intuitive and consistent with data performance. The left image shows that users have a weaker click and purchase intention at breakfast and night, while the heatmap indicates that StAEL increases the spatiotemporal weights $\alpha_j$ of the item feature field and context feature field, and the two trends are consistent. Moreover, we select five typical cities where the number of users decreases from City 1 to City 5, in order to investigate the influence of different locations. Observed from Fig. 9(a), it is obvious to find that the average value of orders within 90 days and the average value of clicks within 1 day from City 1 to City 5 decrease, which can be regarded as the difference in the above-mentioned city-wise user activity. Correspondingly, as shown in Fig. 9(b), we can see that our method detects changes in the spatiotemporal weight $\alpha_j$ of feature field, i.e., with the increase of user activity, weights of the feature fields of user profile and user behavior sequences increase. In contrast, the weight of the item field decreases. Based on the above analysis, our method is able to capture the significance of feature fields in different spatiotemporal scenarios, which also demonstrates its superior ability in spatiotemporal context awareness.

5) Visualization about the model embedding: The t-SNE visualization of final embeddings between the Base model (variation of DIN, detail can be seen in Section III-E) and BASM over different time-periods and different cities can be seen in Fig. 10 and Fig. 11, respectively. We can find that our BASM makes instances in the same time-period or city more convergent, while instances in different time-periods or cities more dispersed.

Especially, compared with Fig. 10(a), instances between dinner and lunch would be segmented by a clear manifold structure, as shown in Fig. 10(b). Those samples with similar semantics are closely in the hyperspace, i.e., samples of night and samples of afternoon tea. What’s more, it is observed from Fig. 11(b) that clusters of instances in different cities are more convergent, while scattered in Fig. 11(a). Those prove that our BASM can better adapt to different spatiotemporal distributions.

D. RQ3: System Efficiency

In this section, we evaluate the time and memory efficiency of both BASM and its comparison methods by using the Ele.me dataset. Experimental results are displayed in TABLE VI, and we observe that static parameter-based methods (i.e., Wide&Deep, DIN, and AutoInt) consistently get shorter training time and memory usage compared with dynamic parameter-based methods, since these static parameter-based methods simply mix all samples to train a set of model parameters, which is at the cost of reducing the performance of the model. However, in our proposed method, we reduce the overall parameter size of BASM through the rational model structure design and matrix decomposition method. Compared with other dynamic parameter-based methods (i.e., STAR, M2M, and APG), BASM achieves the best model results with the lowest training time and memory consumption on the Ele.me dataset.

E. RQ4: Online A/B Test

In August 2022, we conducted an online experiment by deploying BASM to the recommendation scenario on the homepage of Ele.me for one week. We also deployed the Base model, the variation of DIN, mainly consisting of three Multi-head Target Attention modules on the user’s long/short/realt ime historical behavior sequence. CTR was used to evaluate the performance of the online experiment,
TABLE VI

| Methods | Ele.me | Time / Epoch (min) | Memory (G) |
|---------|--------|--------------------|------------|
| Wide&Deep | 168 | 12.3 |
| DIN | 183 | 14.2 |
| AutoInt | 189 | 13.1 |
| STAR | 301 | 35.3 |
| M2M | 299 | 34.3 |
| APG | 567 | 48.2 |
| BASM | 256 | 32.1 |

which was defined as the number of clicks over the number of item impressions. Strictly online A/B experiments are shown in Table VII. We can see that the proposed BASM consistently outperforms the Base model. On average, our model improves CTR by 6.51% compared to the Base model, which demonstrates the effectiveness of BASM in industrial spatiotemporal scenarios. BASM has been deployed on Ele.me, Alibaba’s online food delivery platform, and currently serves more than 100 million users.

F. RQ5: Online Spatiotemporal Result Analysis

We analyze the performance of the Base model and BASM in different time-periods (breakfast, lunch, afternoon tea, dinner, and night) and cities. As illustrated in Fig. 12, We found that: 1) BASM achieves consistent improvement in all time-periods and cities; 2) the CTR improvement is more significant in time-periods and cities with smaller exposure ratios. These results demonstrate the effectiveness of the Bottom-up Adaptive Spatiotemporal Model in fitting the variance of spatiotemporal data distribution and capturing the diversity of user interest under different locations and times.

IV. SYSTEM IMPLEMENTATION AND DEPLOYMENT

This section introduces the system implementation and deployment of BASM on Ele.me, one of the major online food recommendation platforms of Alibaba. The system implementation and deployment mainly include offline training and online serving, as shown in Fig. 13.
## TABLE VII

**ONLINE A/B PERFORMANCES FOR CONSECUTIVE 7 DAYS IN AUG 2022.**

| Day | Base model | BASM | Relative Improvement |
|-----|------------|------|----------------------|
| 1   | 4.26       | 4.63 | 8.69%                |
| 2   | 4.36       | 4.70 | 7.80%                |
| 3   | 4.35       | 4.56 | 4.83%                |
| 4   | 4.57       | 4.86 | 6.35%                |
| 5   | 4.98       | 5.24 | 5.22%                |
| 6   | 4.69       | 5.03 | 7.25%                |
| 7   | 5.09       | 5.35 | 5.11%                |
| Avg | 4.61       | 4.91 | 6.51%                |

### A. Offline Training

We first store the collected online behavior of users on the self-developed MaxCompute platform (MCP) in log format. By processing the data, train and test datasets can be generated. Through distributed training on Alibaba Online Learning Platform (AOP) and performance evaluation, we finally deploy BASM on a Real-Time Prediction (RTP) platform for online serving.

### B. Online Serving

As shown in Fig. 13, the online serving of BASM is implemented on the Personalization Platform (TPP). When a user logs in to the Ele.me APP, TPP obtains user-side features, including user basic features and behavior sequences, by calling Alibaba Basic Feature Server (ABFS). The candidate items are recalled based on Location-based Service, then user features, context features, and candidate items are fed to RTP for scoring prediction. Finally, according to the RTP score, the top k items are returned for exposure.

### V. RELATED WORKS

#### A. Spatiotemporal Modeling

Most current spatiotemporal modeling methods are based on static parameter-based approaches. STAN [4] proposes a spatiotemporal dual-attention model to learn the regularity between non-adjacent locations and non-sequential visits. ST-PIL [3] proposes two levels of attention to fully learn spatiotemporal periodic interests. SiEN [5] proposed three modules to extract spatiotemporal preference, where the Spatiotemporal-aware Target Attention mechanism employed different spatiotemporal information to generate different parameters and feed them into target attention to improve the personalized spatiotemporal awareness of the model. And TRISAN [2] achieves CTR prediction by capturing geographic and temporal information through an attention-based fusion mechanism.

#### B. Dynamic Network

LHUC [17] scales the activation of the hidden layer by treating each speaker as an element-wise multiplier so that each speaker learns a specific hidden unit contribution to improve speech recognition performance. CAN [18] achieves explicitly learning co-action representation by extending the user feature representation with a multi-layer perceptron, in which parameters are dynamically generated from items. APG [20] designs a new learning paradigm to capture custom and common patterns by dynamically generating parameters across different instances.

#### C. Debias

Bias is common in recommendation system, e.g., selection bias [29], [30], position bias [31], [32], exposure bias [33], [34], and popularity bias [35], [36], etc. PAL [37] proposes a Position-bias Aware Learning framework for CTR prediction in a live recommendation system. DICE [38] proposes a general framework to disentangle interest and conformity for recommendation with causal embedding. FPC [39] proposes a dynamic debiasing strategy by utilizing false positive signals.

### VI. CONCLUSION

In this paper, we propose a Bottom-up Adaptive Spatiotemporal Model (BASM) to adaptively fit the spatiotemporal data distribution, further improving the fitting ability of the model. To address the challenge of spatiotemporal data distribution, we realize adaptive modeling of spatiotemporal data from the bottom embedding layer, middle semantic layer, and top classification bias tower. Specifically, a spatiotemporal-aware embedding layer performs weight adaptation on field granularity in feature embedding to achieve the purpose of dynamically perceiving spatiotemporal contexts. Besides, we propose a spatiotemporal semantic transformation layer to explicitly convert the concatenated input of the raw semantic to spatiotemporal semantic, which can further enhance the semantic representation under different spatiotemporal contexts. What’s more, we introduce a novel spatiotemporal adaptive bias tower to capture diverse spatiotemporal bias, reducing the difficulty of modeling spatiotemporal distinction. Extensive offline experiments and online A/B demonstrate the effectiveness and efficiency of BASM.

Although BASM is mainly modeled for spatiotemporal scenarios, we believe that it can be generalized to other scenarios with multiple data distributions in the future.
[32] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen, “A novel click model and its applications to online advertising,” in Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010, B. D. Davison, T. Suel, N. Craswell, and B. Liu, Eds. ACM, 2010, pp. 321–330.

[33] J. Chen, C. Wang, S. Zhou, Q. Shi, J. Chen, Y. Feng, and C. Chen, “Fast adaptively weighted matrix factorization for recommendation with implicit feedback,” in The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020, pp. 3470–3477.

[34] J. Chen, Y. Feng, M. Ester, S. Zhou, C. Chen, and C. Wang, “Modeling users’ exposure with social knowledge influence and consumption influence for recommendation,” in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018. H. Liu, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, and H. Wang, Eds. ACM, 2018, pp. 953–962.

[35] A. Krishnan, A. Sharma, A. Sankar, and H. Sundaram, “An adversarial approach to improve long-tail performance in neural collaborative filtering,” in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, A. Cuzzocrea, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, and H. Wang, Eds. ACM, 2018, pp. 1491–1494.

[36] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, “Causal intervention for leveraging popularity bias in recommendation,” in SIGIR ’21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021. F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai, Eds. ACM, 2021, pp. 11–20.

[37] H. Guo, J. Yu, Q. Liu, R. Tang, and Y. Zhang, “Pal: a position-bias aware learning framework for ctr prediction in live recommender systems,” in Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 452–456.

[38] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, “Disentangling user interest and conformity for recommendation with causal embedding,” in Proceedings of the Web Conference 2021, 2021, pp. 2980–2991.

[39] H. Zhuang, Z. Qin, X. Wang, M. Bendersky, X. Qian, P. Hu, and D. C. Chen, “Cross-positional attention for debiasing clicks,” in WWW ’21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds. ACM / IW3C2, 2021, pp. 788–797.