

# DIGMN: Dynamic Intent Guided Meta Network for Differentiated User Engagement Forecasting in Online Professional Social Platforms

Feifan Li  
feili@linkedin.com  
LinkedIn Corporation

Lun Du\*  
lun.du@microsoft.com  
Microsoft Research

Qiang Fu  
qifu@microsoft.com  
Microsoft Research

Shi Han  
shihan@microsoft.com  
Microsoft Research

Yushu Du  
yusdu@linkedin.com  
LinkedIn Corporation

Guangming Lu  
glu@linkedin.com  
LinkedIn Corporation

Zi Li  
zili@linkedin.com  
LinkedIn Corporation

## ABSTRACT

User engagement prediction plays a critical role for designing interaction strategies to grow user engagement and increase revenue in online social platforms. Through the in-depth analysis of the real-world data from the world's largest professional social platforms, i.e., LinkedIn, we find that users expose diverse engagement patterns, and a major reason for the differences in user engagement patterns is that users have different intents. That is, people have different intents when using LinkedIn, e.g., applying for jobs, building connections, or checking notifications, which shows quite different engagement patterns. Meanwhile, user intents and the corresponding engagement patterns may change over time. Although such pattern differences and dynamics are essential for user engagement prediction, differentiating user engagement patterns based on user dynamic intents for better user engagement forecasting has not received enough attention in previous works. In this paper, we proposed a **Dynamic Intent Guided Meta Network (DIGMN)**, which can explicitly model user intent varying with time and perform differentiated user engagement forecasting. Specifically, we derive some interpretable basic user intents as prior knowledge from data mining and introduce prior intents in explicitly modeling dynamic user intent. Furthermore, based on the dynamic user intent representations, we propose a meta predictor to perform differentiated user engagement forecasting. Through a comprehensive evaluation on LinkedIn anonymous user data, our method outperforms state-of-the-art baselines significantly, i.e., 2.96% and 3.48% absolute error reduction, on coarse-grained and fine-grained user engagement

prediction tasks, respectively, demonstrating the effectiveness of our method.

## CCS CONCEPTS

• **Information systems** → **Enterprise applications**; • **Computing methodologies** → *Neural networks*;

## KEYWORDS

User Intent, User Engagement Forecasting, Meta Learning

## ACM Reference Format:

Feifan Li, Lun Du, Qiang Fu, Shi Han, Yushu Du, Guangming Lu, and Zi Li. 2023. DIGMN: Dynamic Intent Guided Meta Network for Differentiated User Engagement Forecasting in Online Professional Social Platforms. In *Proceedings of the 16th ACM International Conference on Web-inspired Research Involving Search and Data Mining (WSDM '23)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXXX.XXXXXXX>

## 1 INTRODUCTION

Online professional social platforms like LinkedIn have become a significant part of today's lives. People use these platforms to socialize, apply for jobs, read industry news, etc. Maintaining a high-level user engagement is vital for these platforms, which can lead to more revenue (e.g., more ad exposure). For the purpose of increasing user engagement in the future, these platforms need to formulate appropriate user interaction strategies, such as delivering content that satisfies user intents or interests. Accurate user engagement prediction is one of the core technologies for formulating these strategies, which can help the platform conduct user modeling, understand user needs, and provide personalized services.

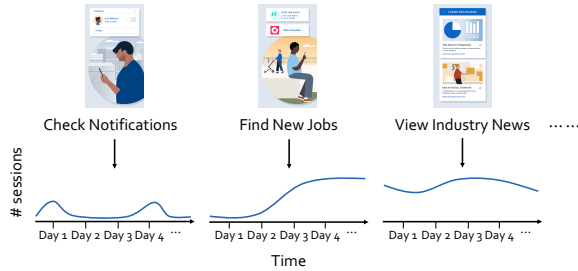
Real-world users often exhibit different behaviors in online social platforms, which leads to their diverse engagement patterns. Through data mining and analysis in the scenario of LinkedIn, we found that the diversity of user engagement patterns is related to the multiple intents of users using LinkedIn, as shown in Figure 1. For instance, some users have the intent of looking for jobs recently, and they will frequently visit LinkedIn to seek and apply for jobs, increasing their engagement level rapidly in a period. As another example, some users use LinkedIn to view industry news, and their

\*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

WSDM '23, February 27 to March 3, 2023, Singapore

© 2023 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/XXXXXXXX.XXXXXXX>



**Figure 1: Example of different user engagement patterns with different user intents.**

engagement pattern is usually maintained at a relatively high level because they regularly check the industry news on LinkedIn. These insights suggest that user intents can be signals to differentiate user engagement patterns.

However, user intents are usually not directly observable because they exist implicitly in human consciousness. How to extract implicit user intents is a challenging problem. At the same time, the user’s intents may change over time. For example, some users initially use LinkedIn to look for jobs, and when they finish looking for jobs, they may use LinkedIn to socialize (e.g., make new connections at a new company). Explicitly modeling dynamic user intent is vital. It can help the platform understand users’ recent intent (or interests) and provide users with content that matches their intent to increase user engagement.

Recently, there have been many works on user engagement forecasting in social network platforms. [1] firstly groups new users into some clusters and then uses a LSTM based model to predict user churn rate. [2] constructs user action graph to characterize and forecast new user engagement. [3] considers user interaction actions and builds a user graph that evolves over time to predict user engagement. Although engagement patterns vary between different users, these works use a model with static parameters for all users to predict their future engagement, which cannot sufficiently model diverse user engagement patterns and perform differentiated user engagement forecasting. Meanwhile, some works show that user intent can impact user engagement. [4] shows that user intent can influence user engagement (e.g., usage time and return time) in Pinterest. [5] demonstrates that user primary intents are associated with how likely the user is to re-engage in activity tracking applications. On the one hand, these works have limitations in extracting user intent. In [4], the user’s intent is obtained through a survey when the app is just opened, which may affect the user’s subsequent behaviors [6]. [5] adopts the activity that most commonly used by the user as a proxy for the user’s intent. However, the user’s intent may be diverse when using applications [7]. On the other hand, these works do not explicitly model changes in user intent over time.

To address the above challenges, we first use Latent Dirichlet Allocation (LDA) [8] to perform user intent mining on large-scale session data and identify basic user intents. Then, we propose a Dynamic Intent Guided Meta Network (DIGMN), which can capture the user’s dynamic intent and perform differentiated user engagement prediction. Specifically, DIGMN infers multiple user intents during each session based on similarity computation with the basic

user intents and captures the variation of user intents over time based on the sequence model. Besides, DIGMN contains a prediction network based on meta-learning, which adopts a dynamic intent guided attention mechanism to adjust network parameters by performing a linear combination of basic parameters shared by all users for differentiated user engagement forecasting. Extensive experiments conducted on coarse-grained and fine-grained user engagement forecasting tasks verify the effectiveness of our DIGMN method.

The major contributions of this paper can be summarized as:

- We find that user intent can be very useful for differentiated user engagement forecasting in online professional social platforms.
- We develop a **Dynamic Intent Guided Meta Network** (DIGMN) which explicitly model user intent’s evolution over time and leverage intent guided attention mechanism to adjust model parameters for differentiated user engagement modeling and forecasting.
- Through evaluation experiments on anonymous data from LinkedIn, our proposed model DIGMN has improvements of 2.96% (Macro F1-score) and 3.48% (AUROC) on coarse-grained and fine-grained user engagement prediction tasks when compared to the state-of-the-art model, showing the effectiveness of our proposed method.

## 2 RELATED WORK

In this section, we present related work on user engagement forecasting, user intent modeling, and meta learning for dynamic network parameters.

**User engagement forecasting.** Recently, there have been many works on user engagement predicting from different perspectives on the social platform. Such as: user behaviors and social attributes [1], user action graphs [2], interaction actions between users [3], and periodicity of user behaviors [9]. These works learn a model with static parameters for all users to make predictions, which cannot sufficiently model differentiated user engagement patterns. [10] leverages a decision tree model to divide users into disjoint groups and then learns a separate Logistic regression model for each group of users to predict user churn. However, such separate modeling compromises the model’s ability to capture similarities between users. [11] adopts the matrix factorization to predict the personalized user’s participation in mobile video. However, there is a large amount of user behavior sequence data in our scenario. This method can not effectively deal with the user’s behavior sequence and its change over time.

**User intent modeling.** Some previous works exploit LDA [2, 12, 13], n-gram [2, 14] and deep learning model [15, 16] to mine user intent from user behavior. At the same time, modeling user intents can help us understand user needs better and is significant in many scenarios; for instance, web searching [17, 18], e-commerce application [4], image sharing social platform [19], activity tracking application [5] and recommender systems [20–24]. However, to our best knowledge, no related work has explicitly modeled user intent and its variation for differentiated user engagement forecasting in online social platforms.

**Meta learning for dynamic network parameters.** Meta-learning (also known as learning to learn) can be used to learn dynamic model parameters, which is widely used in scenarios and tasks

**Table 1: Types of user events we collect on the platform.**

Event ID	Event Type	Explanation
1	Feed	view and react to updates
2	Search	search for members, jobs, or other
3	View Profile	view profiles of members or companies
4	Jobs	view or apply for jobs
5	PYMK	invite members to build connections
6	Notification	check notifications
7	Message	check or send messages
8	Edit Profile	edit personal profile
9	Share Content	share content
10	Follow	follow members, companies, or other

with diverse data distributions. There are usually two ways to learn dynamic model parameters [25]: dynamically generate model parameters or dynamically adjust model parameters. For dynamically generate model parameters: [26] uses a filter-generating network to dynamically generate convolutional filters according to the inputs. [27] adopts a leranet to generate the parameters of a pupil network from a single exemplar. [28] uses a hypernetworks to generate weights of another network. [29] leverages a shared meta-network to generate the parameters of the task-specific semantic composition models. [30] utilizes a meta-knowledge learner to generate model parameters for modeling diversity spatial-temporal correlations in urban traffic prediction. [31] adopts a meta network to learn a personalized mapping function for each user in cross-domain recommendation. [32], [33] and [34] adopt meta networks to generate dynamic parameters of scenario-specific models for CTR prediction. For dynamically adjust model parameters: [35] incorporates meta-information that learned from a supplementary neural network into a fixed base-level neural network to realize the generalization for each type of input images. [36], [37], and [38], in the light of different input images, use a separate network to learn different weights for adjusting the parameters of convolution kernels, which can obtain dynamic convolution kernels for feature extraction. However, no existing works utilize meta learning for predicting user engagement with diverse patterns in online social platforms.

### 3 PRELIMINARY

In this section, we first introduce the relevant definitions, and then formulate the user engagement forecasting tasks that we use in this work.

#### 3.1 Basic Definitions

We define  $U = \{u_1, u_2, \dots, u_N\}$  as a set of different users and  $I = \{i_1, i_2, \dots, i_M\}$  as a set of different event types, where  $N, M$  is the number of users and event types respectively.

**User event.** The user’s behaviors on the platform can be abstracted as events for collection. Formally, a user event  $e$  can be denoted by  $e = (u, i, t)$ , where  $u \in U$  is the user,  $i \in I$  is the event type and  $t$  is the time when the event happens. We collect ten major event types on the platform, as shown in Table 1. These 10 events can completely cover users’ activities on the platform.

**User session.** Generally, a session is a set of continuous browsing activities with gaps between full-page views not more than

a threshold. Formally, a session can be denoted as  $S = (u, c, \mathcal{E})$ , where  $u \in U$  is the user who starts the session,  $c$  is the session context information (e.g., the session start time, the session duration time and the software client) and  $\mathcal{E}$  is the sequence of events that take place during the session.

**User engagement.** User engagement measures how often and how long users interact with the website or application, which can reflect how much value users get from the website or app [39]. Different platforms use different metrics to track user engagement. In this paper, we adopt two metrics to measure the engagement of each user in LinkedIn: average active days (coarse-grained, which is the same as the definition of active rate in [2]) and average session numbers (fine-grained). The average session numbers  $\bar{s} = \frac{\# \text{ valid sessions}}{\# \text{ total days}}$ , where a valid session is a session that contains at least one event in Table 1. The average active days  $\bar{d} = \frac{\# \text{ active days}}{\# \text{ total days}}$ , where an active day is defined as a day that the user has at least one valid session.

**Platform actions.** Previous studies have demonstrated that user engagement on the platform can be increased through notifications [40, 41], incentives including badges [42, 43], etc. Therefore, we need to consider the impact of the interaction actions between the platform and the user when forecasting user engagement. For simplicity, here we only consider the platform delivery messages. The platform will deliver messages to users according to their statuses, such as updates on the people or companies they follow, reminders of unread messages, and job recommendations. Platform delivery messages with appropriate content at a suitable time can provide users with a better experience in the platform, thereby enhancing user engagement in the future. On the contrary, too many irrelevant platform delivery messages may damage the user experience and lead to user churn. Formally, a platform delivery message can be denoted as  $\mathcal{D} = (u, w, r, t)$ , where  $u \in U$  is the user being delivered,  $w$  is the way the message is delivered (i.e., email, SMS and in-app push),  $r$  is the content of delivery message (i.e., Feed relevant, Jobs relevant, PYMK relevant, Message relevant and others),  $t$  is the delivery time.

#### 3.2 Task Formulation

In this work, we predict the trend of user engagement instead of directly predicting the future engagement of users. This is because we pay more attention to the change in user engagement in actual business scenarios. For example, when designing a platform message delivery strategy in the scenario of user retention, the change value ( $\Delta \bar{s}$  or  $\Delta \bar{d}$ ) of the user engagement caused by the delivered message is an essential indicator of business decisions. Specifically, we define two user engagement trend forecasting tasks as follows:

- **Day-level Task (coarse-grained):** We predict the trend of the average active days of the user, which can be viewed as a 3-class (i.e., increase, decrease, or stay the same) classification task.
- **Session-level Task (fine-grained):** We predict the trend of the average session numbers of the user, which can be viewed as a 2-class (i.e., decrease or not decrease) classification task. The reason for setting it as a binary classification task is that the number of user sessions has a broader distribution than the number of

active days, and users with a large number of sessions are less likely to have the same number of sessions over a while.

Based on the above definitions, for any user  $u \in U$ , given the user's macroscopic features  $M_u$  (e.g., the number of connections, the average number of sessions per day in the past period), session sequence  $S_u = \langle S_1, S_2, \dots, S_T \rangle$ , and the latest platform delivery message  $D_u$  in the past period, forecasting user engagement trend  $y_u$  in the next period can be formulated as follows:

$$\Pr(y_u | M_u, S_u, D_u; \Theta) \quad (1)$$

where  $T$  is the number of sessions,  $y_u$  is the user engagement trend label, and  $\Theta$  is the parameters we need to learn.

## 4 INTENT MINING

In this section, we introduce how to perform user intent mining and show the basic user intents we obtain from data mining.

We assume that every user has at least one intent within each session [44]. The user's intents are usually not directly observed, and they exist implicitly in the user's consciousness. It is difficult for us to collect a large number of user behavior samples with user intent labels, so using unsupervised methods for user intent mining is a more feasible option. Intuitively, the user's intents influence the user's behaviors, and the user's behaviors constitute the set of events in the session. This process is similar to generating documents with an unsupervised topic model LDA. Like previous work [2], we treat each session as a document and each event type as a word, then apply LDA to mine basic user intents.

We use Spark MLlib<sup>1</sup> to perform LDA on approximately 6 million anonymous user session data for intent mining. To determine the optimal number of intents with semantic meaning, we adopt perplexity as the evaluation metric to search for the optimal intent number in  $\{2, 3, 4, \dots, 10\}$ . When the number of intents is equal to 7, making the LDA model has the least perplexity.

These 7 topics can be regarded as 7 basic user intents (denoted as  $t_1, t_2, \dots, t_7 \in \mathbb{R}^{10}$ ), each of which is composed of events with different weights (also can be view as probability) as shown in Figure 2. The meaning of each intent can be explained by the top weight events that make up them. For example, we can find that the two events with top weight making up intent 1 are PYMK and Profile View, this intent indicates that the users want to expand their connections in LinkedIn.

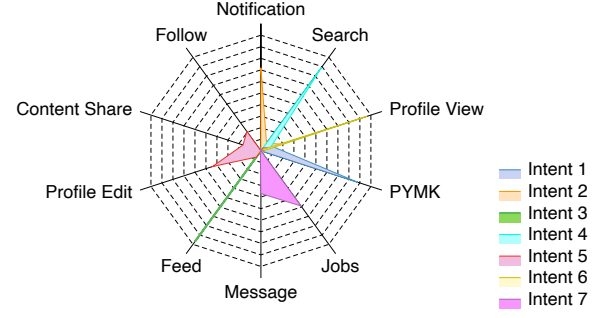
## 5 DYNAMIC INTENT GUIDED META NETWORK

In this section, we introduce our proposed model DIGMN. The framework of our proposed model is illustrated in Figure 3. It consists of three main components: behavior evolution layer, intent evolution layer and a meta predictor.

### 5.1 Behavior Evolution Layer

Users' behaviors change over time, and the length of the behavior sequence of different users may be different. LSTM[?] has been widely used for modeling variable-length user behavioral sequence. In this paper, we utilize a 1-layer LSTM to model user behavior evolution.

<sup>1</sup><https://spark.apache.org/mllib/>



**Figure 2: Basic user intents. There are 7 basic intents obtained by LDA, and we show the event weights that compose them.**

For each session  $S$ , we represent it by concatenating the session context features  $c$  and session event frequency  $\mathbf{v} = (v_1, v_2, \dots, v_{10}) \in \mathbb{R}^{10}$ , where  $v_i$  is the frequency of Event  $i$  (e.g., If there are 100 events happen in a session, and the Feed event occurs 10 times, then  $v_1 = 10/100 = 0.1$ ). We take  $s = (\mathbf{v}, c)$  as the input of the session LSTM unit at each timestep. Since the user's behavior sequences are of different lengths, we take the last timestep output of LSTM as the representation of the user's behavior features in the past period.

### 5.2 Intent Evolution Layer

A user may have multiple intents during one session. For example, the user can use LinkedIn to view the updates on the industry news (i.e., intent 3) and share content related to himself/herself (i.e., intent 5) at the same time. We can infer the user's possible intents based on the events happening during this session.

In this paper, we exploit a simple and effective way to infer user intent in each session. Given a session's event frequency  $\mathbf{v}$  and the basic user intents  $t_1, t_2, \dots, t_7$  obtained by LDA in the intent mining stage, we can infer the user intents  $\mathbf{i} = (i_1, i_2, \dots, i_7)$  during this session. To be specific,  $i_k$  is the cosine similarity between the session event frequency  $\mathbf{v}$  and the  $k$ -th basic user intent  $t_k$ , which measures the degree to which the current session contains the  $k$ -th intent and can be calculated as follows:

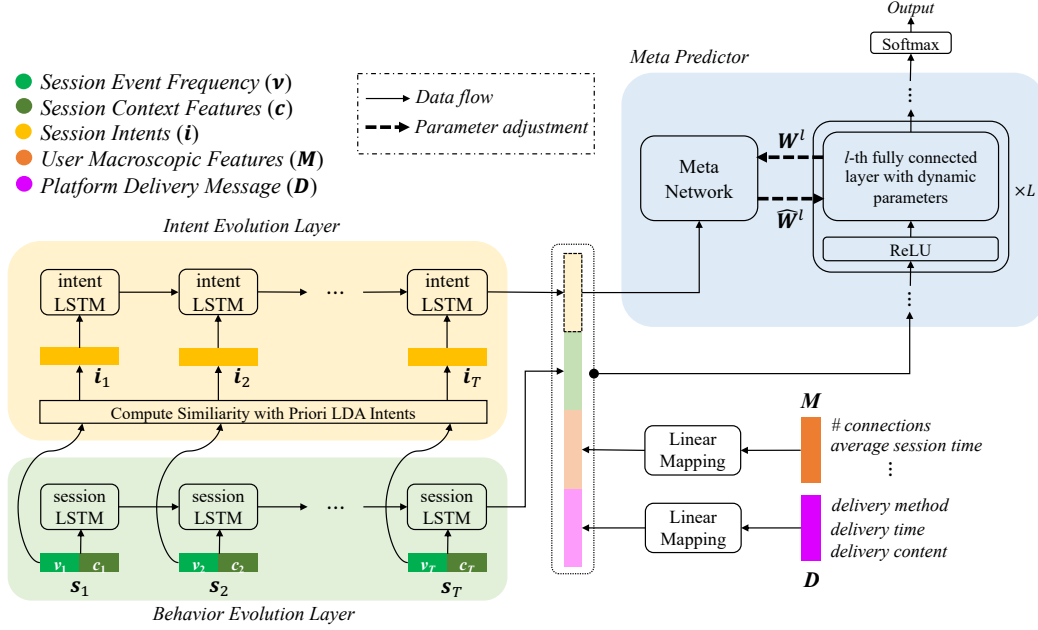
$$i_k = \frac{\mathbf{v} \cdot t_k}{\|\mathbf{v}\| \times \|t_k\|} \quad (2)$$

By performing the above intent extraction operation on each user session, we can obtain an intent sequence for each user with the same length as the user session sequence. Another 1-layer LSTM is adopted to model the variation of user intent over time. We take the intent representation  $\mathbf{i}$  of each session as the input of the intent LSTM unit at each timestep. We also take the LSTM's last timestep output to represent the user's dynamic intent over the past period.

### 5.3 Meta Predictor

Different users have different engagement patterns, and the user engagement pattern may change in different periods. Previous works [1–3] use a model with the static parameters  $\Theta$  for different users to predict their future engagement in social platform, which can be formulated as:

$$\mathbf{y} = F(\mathbf{x}, \Theta) \quad (3)$$



**Figure 3: The framework of our proposed model. It consists of three main components: behavior evolution layer, intent evolution layer and a meta predictor.**

where  $\mathbf{x}$  is the model input (i.e., user features) and  $\mathbf{y}$  is the model output. In order to make a good prediction, the model tends to capture the common patterns of all users, which is not optimal for modeling and predicting diverse user engagement patterns in real scenarios. The most straightforward idea is to learn a unique predictor for each user or a group of similar users. However, there is a certain similarity between users, while separate modeling may impair the model's ability to capture the similarity between users. Therefore, we need a predictor that can better model the diversity in user engagement patterns while can also capture the similarities between users.

Dynamic parameters neural networks have shown promising results in various real-world tasks and can dynamically adjust or generate model parameters  $\Theta^*$  according to different inputs  $\mathbf{x}$ , which have a more powerful representation ability. Compared with formula (3), it can be formulated as:

$$\mathbf{y} = F(\mathbf{x}, \Theta^*) = F(\mathbf{x}, \phi(\mathbf{x}, \Theta)) \quad (4)$$

where  $\phi(\cdot, \Theta)$  is the operation that adjusts or generates model parameters according to input  $\mathbf{x}$ . A common technique in dynamic parameters neural networks is attention on parameters [36, 37]. It assumes that there are some basic learnable parameters with the same shape in the dynamic parameter layer. Given different inputs, the meta network can generate different attention weights to combine these basic learnable parameters in dynamic parameter layer to obtain dynamic parameters and make transformations. In our scenario, the meta network can adjust the model parameters according to different user feature inputs for differentiated modeling. At the same time, all users share the basic learnable parameters of the dynamic parameter layer, which enables the model to capture the similarities between users.

According to the previous analysis, we can use the user dynamic intent representation obtained from the intent evolution layer as the input signal to parameter adjust operation  $\phi(\cdot, \Theta)$  in the meta network to perform differentiated user engagement modeling and forecasting. Specifically, we design a meta predictor which contains a meta network and multiple stacked fully connected layers with dynamic parameters (FC-D layer), as illustrated in Figure 3.

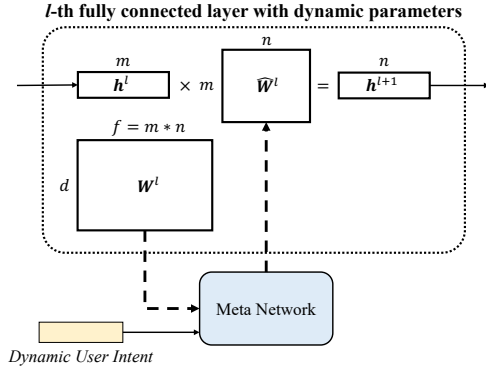
**5.3.1 Fully Connected Layers with Dynamic Parameters.** As illustrated in Figure 4, assuming that the  $l$ -th FC-D layer transforms the input feature  $\mathbf{h}^l \in \mathbb{R}^m$  into  $\mathbf{h}^{l+1} \in \mathbb{R}^n$ , and correspondingly assuming there are  $d$  basic learnable parameters  $\mathbf{W}_1^l, \mathbf{W}_2^l, \dots, \mathbf{W}_d^l \in \mathbb{R}^{(m \times n)}$  in it, denoted as  $\mathbf{W}^l = [\mathbf{W}_1^l, \mathbf{W}_2^l, \dots, \mathbf{W}_d^l]^T \in \mathbb{R}^{d \times (m \times n)}$ . The  $l$ -th FC-D layer computes the following transformation:

$$\mathbf{h}^{l+1} = \widehat{\mathbf{W}}^l \cdot \mathbf{h}^l + \hat{\mathbf{b}}^l \quad (5)$$

where  $\widehat{\mathbf{W}}^l \in \mathbb{R}^{m \times n}$ ,  $\hat{\mathbf{b}}^l \in \mathbb{R}^n$  are the weights and bias of the  $l$ -th FC-D layer adjusted by the meta network.

**5.3.2 Meta Network.** The adjustment process of  $\widehat{\mathbf{W}}^l$  in the meta network is shown in Figure 5. The combination weights  $\mathbf{a} = [a_1, \dots, a_d]^T \in \mathbb{R}^d$  (which can be seen as meta-knowledge) of basic learnable parameters  $\mathbf{W}^l$  is derived by applying a non-linear transformation and a Softmax operation to user dynamic intent (denoted as  $\tilde{\mathbf{i}}$ ). In this paper, we utilize two fully connected layers with ReLU activation function to perform non-transformation, which can be formulated as follows:

$$\begin{aligned} \mathbf{a} &= \text{Softmax}(\mathbf{W}_2(\text{ReLU}(\mathbf{W}_1 \tilde{\mathbf{i}} + \mathbf{b}_1)) + \mathbf{b}_2) \\ \text{s.t. } \sum_{i=1}^d a_i &= 1, 0 < a_i < 1 \end{aligned} \quad (6)$$



**Figure 4: The  $l$ -th fully connected layers with dynamic parameters.**

where  $W_1, W_2, b_1, b_2$  are learnable parameters. Then we can combine the basic learnable parameters  $W^l = [W_1^l, W_2^l, \dots, W_d^l]^T$  through  $\alpha$ , and obtain the adjusted transformation matrix  $\widehat{W}^l$  of  $l$ -th FC-D layer through the reshape operation:

$$\widehat{W}^l = \text{Reshape}(\alpha^T W^l) = \text{Reshape}\left(\sum_{i=1}^d a_i W_i^l\right) \quad (7)$$

$\widehat{b}^l$  can be calculated in the similar way as  $\widehat{W}^l$ .

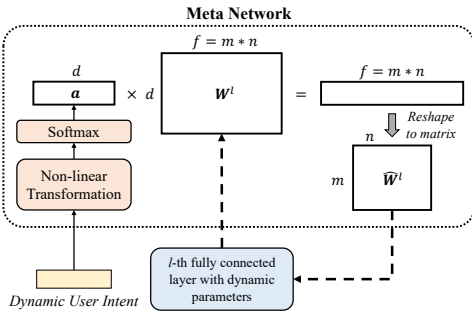
Each row of  $W^l$  (i.e.,  $W_i^l, i = 1, 2, \dots, d$ ) can be regarded as independent basic learnable parameters.  $\widehat{W}^l$  is a linear combination of  $W_1^l, W_2^l, \dots, W_d^l$ , and we hope that  $W_1^l, W_2^l, \dots, W_d^l$  are as orthogonal as possible to reduce redundant information. [45] summarizes some orthogonal regularization methods for parameters, including "selective" Soft Orthogonality Regularization, which can be written as follows:

$$\lambda \cdot \|(W^l (W^l)^T - I)\|_F^2 \quad (8)$$

where  $\lambda$  is the regularization coefficient,  $I \in \mathbb{R}^{d \times d}$  is the identity matrix. Here we do not limit the norm of  $W_i^l$  to 1, and adopt the following orthogonal regularization term:

$$\mathcal{L}_R = \sum_{l=1}^L \|(W^l (W^l)^T - I) \odot (1 - I)\|_F^2 \quad (9)$$

where  $\mathbf{1} \in \mathbb{R}^{d \times d}$  is a matrix whose elements are all 1,  $\odot$  is Hadamard product,  $L$  is the number of FC-D layers.



**Figure 5: Meta Network.**

## 5.4 End-to-end model training

We adopt end-to-end learning strategy and use error back propagation to train our proposed model. We choose cross-entropy as classification loss function, which can be formulated as:

$$\mathcal{L}_C = -\frac{1}{n} \sum_i \sum_c y_{ic} \log(\hat{y}_{ic}) \quad (10)$$

where  $n$  is the number of samples,  $i$  represents the  $i$ -th sample,  $y_{ic}$  equals to 1 if  $i$  belongs to class  $c$  otherwise 0,  $\hat{y}_{ic}$  is the predicted probability that sample  $i$  belongs to class  $c$ . After adding the regularization term for the parameters of meta predictor, the loss function can be written as:

$$\mathcal{L} = \mathcal{L}_C + \beta \cdot \mathcal{L}_R \quad (11)$$

where  $\beta$  is the hyperparameter that trades off classification loss  $\mathcal{L}_C$  and regularization term  $\mathcal{L}_R$ .

## 6 EVALUATION

To evaluate the performance of our proposed model, we conduct experiments on real-world anonymous users' data from LinkedIn. In this section, we first describe the dataset used in this work and introduce the detailed experimental settings. Then, we show that DIGMN outperforms current state-of-the-art models on the user engagement prediction task, demonstrating the effectiveness of DIGMN. To be more specific, we aim to answer the following research questions:

- **RQ1:** Can DIGMN outperform state-of-the-art baselines in user engagement forecasting tasks at different granularities?
- **RQ2:** Does DIGMN perform better than a network with static parameters with a similar number of parameters?
- **RQ3:** How does each part in DIGMN affect the performance?
- **RQ4:** Is dynamic user intent a good signal to differentiate diverse user engagement patterns?
- **RQ5:** How do hyperparameters affect the performance?
- **RQ6:** Is it effective to introduce prior intent obtained by LDA in DIGMN?
- **RQ7:** Can DIGMN learn interpretable user dynamic intent representations?

## 6.1 Experimental Setup

**6.1.1 Datasets.** In order to protect user privacy, we collect coarse-grained behavioral data (only the types of events, as shown in Table 1, not detailed user behaviors) of anonymous users during four weeks. To make the extracted dynamic intent information more meaningful, we filter out users with less than 7 (median) sessions during the first two weeks. As shown in Table 2, the user engagement trend label  $y$  is obtained by comparing the user engagement in the first two weeks (i.e.,  $\bar{d}_h, \bar{s}_h$ ) and the following two weeks (i.e.,  $\bar{d}_f, \bar{s}_f$ ).

**Table 2: Labels of user engagement forecasting tasks.**

Day-level Task	Session-level Task
$y = \begin{cases} -1, & \bar{d}_h > \bar{d}_f \\ 0, & \bar{d}_h = \bar{d}_f \\ 1, & \bar{d}_h < \bar{d}_f \end{cases}$	$y = \begin{cases} 0, & \bar{s}_h > \bar{s}_f \\ 1, & \bar{s}_h \leq \bar{s}_f \end{cases}$

The label distribution of day-level task is approximately  $y = -1 : y = 0 : y = 1 \approx 2 : 2 : 1$ , and the label distribution of session-level task is approximately  $y = 0 : y = 1 \approx 3 : 2$ . We randomly sample 200K users for each experiment and split them into three parts: 80% of the samples (i.e., users) for training, 10% for validation, and the rest 10% for testing.

**6.1.2 Evaluation Metrics.** For day-level task (3-class classification task), we adopt Macro F1-score as our metric following Liu et al.[2]’s work. Macro F1-score is the average F1-score for each class and can evaluate the classifier in the case of class imbalance. Furthermore, for session-level task (2-class classification task), AU-ROC is adopted as our metric, which is a popular metric for the binary-classification task.

**6.1.3 Comparison Methods.** We do not collect the interaction actions between users in this work. Considering comparability, we did not compare with the FATE [3], which uses interaction behaviors between users to predict user engagement. For brevity, we use  $\{h_1, h_2, \dots\}$  to denote the hidden state dimension of fully connected layer, we compare our proposed model against the following baselines:

- **Logistic Regression (LR):** We use user macroscopic features and the latest platform delivery message to make predictions.
- **XGBoost [46]:** Use the same input features as Logistic Regression to classify.
- **Multilayer Perceptron (MLP):** We implement a MLP with two hidden layers {64, 32} using the same features as Logistic Regression and XGBoost as input.
- **Activity LSTM [1]:** We count the number of events in Table 1 that occurred each day for the user in the past two weeks, and get a time series  $\mathbf{A} \in \mathbb{R}^{10 \times 14}$  as the input of the activity LSTM. We implement a two-layer LSTM following setups in [1], and fed the final output of the activity LSTM concatenating with the latest platform delivery message to a MLP with two hidden layers {64, 32} for prediction.
- **Temporal GCN-LSTM [2]:** We treat events as actions and construct user’s action graph according to [2]. Temporal GCN-LSTM first applies GCN on action graph to extract user fine-grained features, then fed it into LSTM to capture its temporal dynamics. We implement a MLP with two hidden layers {64, 32} for prediction based on the last final output of LSTM concatenating with the latest platform delivery message.
- **Deep Multi-channel [2]:** To achieve the best performance, it combines user macroscopic features, the latest platform delivery message, activity LSTM, and Temporal GCN-LSTM. It can be viewed as the state-of-the-art model for predicting user engagement that does not consider the interaction behaviors between users. For comparability, we implement a MLP with two hidden

layers {64, 32} to predict user engagement based on the output of the above all modules.

**6.1.4 Evaluation Settings.** The output dimension of the linear mapping layer is set to be half of the input dimension. The hidden unit of session LSTM and intent LSTM are empirically set to 32. We implement DIGMN with  $L$  FC-D layers, where  $L$  is searched from 1 to 5. The hidden units of the non-linear transformation layer in the meta network are set to 32 and the dimension of  $\mathbf{a}$  is searched from 2 to 10. The hyperparameter  $\beta$  is searched in  $[10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0]$ . We adopt Adam [47] as optimizer with initial learning rate  $10^{-3}$  and learning rate decay half every 20 epochs. According to [48], the authors argue against using  $\ell_2$  weight decay and the orthogonal regularization term together, so we only use  $\ell_2$  weight decay for learnable parameters other than  $\mathbf{W}^l (l = 1, 2, \dots, L)$  in FC-D layers with a decay rate  $10^{-5}$ . Early stopping strategy is also used on the validation set for avoiding overfitting. Our experiments are conducted on a single machine with a 2.4GHz 12-core CPU and 64GB of memory. Each experiment is repeated 5 times with different random seeds.

## 6.2 Performance Comparison (RQ1, RQ2)

Table 3 lists the experiment results of all compared methods. We observe that DIGMN performs best on both day-level and session-level user engagement prediction tasks, which shows the effectiveness of DIGMN. The possible reason for the poor performance of activity LSTM and Temporal GCN-LSTM on our dataset is that our user session data is sparse (each user averages around 2 session a day in our dataset, but 7 in Liu et al.[2]’s dataset), which results in sparse activity sequences and action graphs, reducing their performance.

Compared with the FC layer, the FC-D layer used in DIGMN increases the learnable parameters. Take one FC-D layer as an example, assuming that it maps the input  $\mathbf{h} \in \mathbb{R}^m$  to  $\mathbf{h}' \in \mathbb{R}^n$ , then the learnable parameters of FC-D layer is  $d * m * n$  (for brevity, ignoring bias), where  $d$  is the number of basic parameters used for linear combination. However, for the FC layer, its number of learnable parameters is  $m * n$ . For comparability, we replace FC-D layers (Dynamic) in DIGMN with FC layers (Static). As shown in Table 4, we observe DIGMN with dynamic parameters outperforms DIGMN with static parameters in two tasks, demonstrating the effectiveness of leveraging dynamic parameters network to achieve differentiated user engagement prediction.

## 6.3 Model Ablation Study (RQ3, RQ4)

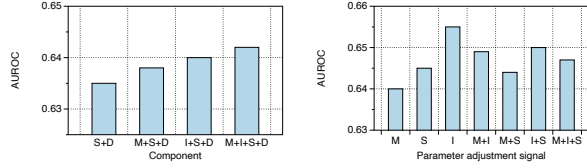
We conduct the following ablation study on the session-level user engagement prediction task.

We use a MLP with two hidden layers (# hidden units are 64 and 32) as the predictor to investigate the effectiveness of various components in our model. As shown in Figure 6(a), we find that simultaneously using the user’s macroscopic features, platform delivery message, session features, and intent features can achieve the best results.

We also study the influence of different adjust signal of DIGMN, as shown in Figure 6(b). We find that intent as a parameter adjust signal performs best, indicating that intent can distinguish different engagement levels well. Introducing too much information to adjust parameters may impair the model classification ability.

**Table 3: Performance comparison on the classification of user engagement trends at the day-level and the session-level.**

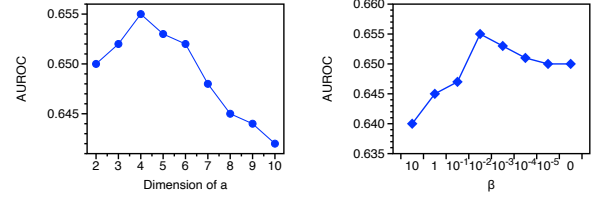
Model \ Task	Day-level	Session-level
	Macro F1-Score	AUROC
<i>Feature-based model</i>		
LR	0.446 ± 0.000	0.561 ± 0.000
XGBoost	0.460 ± 0.001	0.569 ± 0.001
<i>End-to-end neural network model</i>		
MLP	0.463 ± 0.002	0.571 ± 0.002
Activity LSTM	0.518 ± 0.001	0.605 ± 0.001
Temporal GCN-LSTM	0.522 ± 0.002	0.600 ± 0.003
Deep Multi-channel	0.575 ± 0.002	0.633 ± 0.002
<b>DIGMN (ours)</b>	<b>0.592 ± 0.001</b>	<b>0.655 ± 0.001</b>
Improvements	2.96%	3.48%

**Figure 6: Ablation study on different model components. (a) Ablation study on different model components. (b) Ablation study on different parameter adjustment signal of DIGMN.****Figure 6: Ablation study of components and different parameter adjustment signal.  $M, D, S, I$  represent macroscopic features, the latest platform delivery message, session features and intent features respectively.**

#### 6.4 Hyperparameter Sensitivity (RQ5)

We conduct the following hyperparameter sensitivity study on the session-level user engagement prediction task.

- **Hyperparameter sensitivity of  $\alpha$ :** As shown in Figure 7(a), we search for the optimal dimension of  $\alpha$  between 2 and 10, and find that when the dimension of  $\alpha$  is too small or too large, it will hurt the model’s performance. The possible reason is that when the dimension of  $\alpha$  is small, the expressive ability of the model is limited and cannot sufficiently capture the diverse engagement patterns of users. However, when the dimension of  $\alpha$  is large, the number of parameters of the model will rapidly increase, which may cause overfitting and weaken generalization ability. On our dataset, the model performs best when the dimension of  $\alpha$  is equal to 4.
- **Hyperparameter sensitivity of  $\beta$ :** As illustrated in Figure 7(b), we adopt grid search in  $[10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0]$ , and find that the model performs best when  $\beta = 10^{-2}$ . This indicates that too strong orthogonal constraints on FC-D layer will damage the classification ability of the model, and adding proper orthogonal constraints to the FC-D layer can improve the model’s performance.

**Figure 7: Hyperparameter sensitivity. (a) Prediction performance of different dimension of  $\alpha$ . (b) Prediction performance of different  $\beta$ .****Figure 7: Hyperparameter sensitivity.**

- **Hyperparameter sensitivity of  $L$ :** As illustrated in Table 5, we vary the number of FC-D layers (i.e.,  $L$ ) in the meta predictor from 1 to 5 and find that the model achieves the best performance when  $L = 3$ . The possible reason is that when the number of FC-D layers is small, the learning ability of the model is insufficient, and the model is prone to underfitting. In contrast, when the number of FC-D layers is large, the model is prone to overfitting, which damages the model’s generalization ability.

**Table 5: Prediction performance of different  $L$ .**

# FC-D Layers	# Hidden Units	AUROC
$L = 1$	-	0.645 ± 0.001
$L = 2$	{64}	0.651 ± 0.001
$L = 3$	{64,32}	0.655 ± 0.001
$L = 4$	{64,32,32}	0.652 ± 0.001
$L = 5$	{64,32,32,32}	0.646 ± 0.002

#### 6.5 Further Analysis (RQ6)

In order to verify the effectiveness of introducing the prior intent obtained by LDA in DIGMN, we replace the intent inference operation based on cosine similarity computation in the intent evolution layer with a linear mapping, which can be formulated as  $i = \tanh(Wv + b)$ , where  $W \in \mathbb{R}^{10 \times 7}$ ,  $b \in \mathbb{R}^7$  are learnable parameters,  $\tanh(\cdot)$  is hyperbolic tangent activation functions used to limit the output value between -1 and 1 (same range as cosine similarity). As shown in Table 6, we can observe that introducing prior user intents obtained by LDA in DIGMN performs better than directly learning the implicit intent representation. The possible reason is that the prior intents are learned on a larger amount of data, while the end-to-end learning user intent representation usually only use a part of the data, and the former contains more information.

**Table 6: Prediction performance of different methods to extract session intents.**

Methods to extract session intents	AUROC
End-to-end learning	0.644 ± 0.001
Prior intents	0.655 ± 0.001

#### 6.6 Visualization (RQ7)

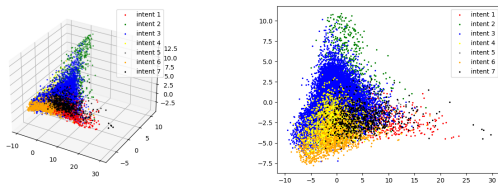
We use PCA to reduce the dimensions of dynamic user intent representation from 32 to 3 and 2 in the test dataset of the session-level user engagement forecasting task. The label of each sample is the user’s most frequent intent in the past two weeks. As shown in Figure 8(a) and 8(b), we can observe that the intent evolution layer



**Table 4: Performance comparison of DIGMN with dynamic predictor and DIGMN with static predictor.**

Model \ Task	# Hidden Units in Predictor	Day-level		Session-level	
		Macro F1-Score	# Parameters	AUROC	# Parameters
DIGMN (Static)	{64, 32}	0.580 ± 0.001	18.8K	0.642 ± 0.001	18.7K
DIGMN (Static)	{160, 96}	0.585 ± 0.001	38.6K	0.645 ± 0.002	38.5K
DIGMN (Dynamic)	{64, 32}	0.592 ± 0.001	40.8K	0.655 ± 0.001	40.7K

can learn meaningful and discriminative user dynamic intent representation. Subsequent meta predictors can leverage user dynamic intent representation to adjust network parameters and perform differentiated user engagement prediction.



(a) The distribution of user dynamic intent in to 3-dimensional space. (b) The distribution of user dynamic intent in to 2-dimensional space.

**Figure 8: The visualization of dynamic user intent representation. We apply zero-mean normalization to the input high-dimensional vectors before using PCA for dimension reduction.**

## 7 CONCLUSIONS

In this work, we use LDA on user session data to mine basic user intents. Meanwhile, we propose a dynamic intent guided meta network (DIGMN) to explicitly model user intent evolution over time and perform differentiated user engagement forecasting. Experiments on the real-world dataset from LinkedIn demonstrate the effectiveness of our method.

The future work will focus on the interpretability of the method and apply our model to more business scenarios, such as the platform message delivery system.

## 8 ACKNOWLEDGEMENT

Thanks to Xu Chen and Yanbin Kang for their insightful discussions. Thanks to Yihan Cao and Lu Chen for their help in data process and building dataset. Thanks to Yiping Yuan and Huichao Xue for their comprehensive review feedback.

## REFERENCES

- [1] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922, 2018.
- [2] Yozen Liu, Xiaolin Shi, Lucas Pierce, and Xiang Ren. Characterizing and forecasting user engagement with in-app action graph: A case study of snapchat. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2023–2031, 2019.
- [3] Xianfeng Tang, Yozen Liu, Neil Shah, Xiaolin Shi, Prasenjit Mitra, and Suhang Wang. Knowing your fate: Friendship, action and temporal explanations for user engagement prediction on social apps. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2269–2279, 2020.
- [4] Caroline Lo, Dan Frankowski, and Jure Leskovec. Understanding behaviors that lead to purchasing: A case study of pinterest. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 531–540, 2016.
- [5] Zhiyuan Lin, Tim Althoff, and Jure Leskovec. I'll be back: on the multiple lives of users of a mobile activity tracking application. In *Proceedings of the 2018 World Wide Web Conference*, pages 1501–1511, 2018.
- [6] Vicki G Morwitz, Eric Johnson, and David Schmittlein. Does measuring intent change behavior? *Journal of consumer research*, 20(1):46–61, 1993.
- [7] Martin Fishbein and Icek Ajzen. Belief, attitude, intention, and behavior: An introduction to theory and research. *Philosophy and Rhetoric*, 10(2), 1977.
- [8] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [9] Farhan Asif Chowdhury, Yozen Liu, Koustuv Saha, Nicholas Vincent, Leonardo Neves, Neil Shah, and Maarten W Bos. Ceam: The effectiveness of cyclic and ephemeral attention models of user behavior on social platforms. In *Proceedings of the 12th International AAAI Conference on Web and Social Media*, volume 15, pages 117–128, 2021.
- [10] Arno De Caigny, Kristof Coussement, and Koen W De Bock. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269(2):760–772, 2018.
- [11] Lin Yang, Mingxuan Yuan, Yanjiao Chen, Wei Wang, Qian Zhang, and Jia Zeng. Personalized user engagement modeling for mobile videos. *Computer Networks*, 126:256–267, 2017.
- [12] Guandong Xu, Yanchun Zhang, and Xun Yi. Modeling user behavior for web recommendation using lda model. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 529–532. IEEE, 2008.
- [13] Mark J Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1849–1852, 2010.
- [14] Jimmy Lin and W John Wilbur. Modeling actions of pubmed users with n-gram language models. *Information retrieval*, 12(4):487–503, 2009.
- [15] Pablo Loyola, Chen Liu, and Yu Hirate. Modeling user session and intent with an attention-based encoder-decoder architecture. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 147–151, 2017.
- [16] Rakshit Agrawal, Anwar Habeeb, and Chih-Hsin Hsueh. Learning user intent from action sequences on interactive systems. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [17] Qi Guo and Eugene Agichtein. Ready to buy or just browsing? detecting web searcher goals from interaction data. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 130–137, 2010.
- [18] Wenchao Gu, Yanlin Wang, Lun Du, Hongyu Zhang, Shi Han, Dongmei Zhang, and Michael R Lyu. Accelerating code search with deep hashing and code classification. *arXiv preprint arXiv:2203.15287*, 2022.
- [19] Justin Cheng, Caroline Lo, and Jure Leskovec. Predicting intent using activity logs: How goal specificity and temporal range affect user behavior. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 593–601, 2017.
- [20] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [21] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. *arXiv preprint arXiv:2202.02519*, 2022.
- [22] Nagaraj Kota, Venkatesh Duppada, Ashvini Jindal, and Mohit Wadhwa. Learnings from building the user intent embedding store towards job personalization at linkedin. *The 2nd International Workshop on Industrial Recommendation Systems*, 2021.
- [23] Yun Wang, Lun Du, Guojie Song, Xiaojun Ma, Lichen Jin, Wei Lin, and Fei Sun. Tag2gauss: Learning tag representations via gaussian distribution in tagged networks. In *IJCAI*, pages 3799–3805, 2019.
- [24] Junshan Wang, Zhicong Lu, Guojia Song, Yue Fan, Lun Du, and Wei Lin. Tag2vec: Learning tag representations in tag networks. In *The World Wide Web Conference*,

- pages 3314–3320, 2019.
- [25] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [26] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.
- [27] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. *Advances in neural information processing systems*, 29, 2016.
- [28] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.
- [29] Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang. Meta multi-task learning for sequence modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [30] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1720–1730, 2019.
- [31] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1507–1515, 2022.
- [32] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. Leaving no one behind: A multi-scenario multi-task meta learning approach for advertiser modeling. *arXiv preprint arXiv:2201.06814*, 2022.
- [33] Bencheng Yan, Pengjie Wang, Kai Zhang, Feng Li, Jian Xu, and Bo Zheng. Apg: Adaptive parameter generation network for click-through rate prediction. *arXiv preprint arXiv:2203.16218*, 2022.
- [34] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, et al. Can: Feature co-action network for click-through rate prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 57–65, 2022.
- [35] Feihu Zhang and Benjamin W Wah. Supplementary meta-learning: Towards a dynamic model for deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4344–4353, 2017.
- [36] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Conconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems*, 32, 2019.
- [37] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020.
- [38] Chao Li, Aojun Zhou, and Anbang Yao. Omni-dimensional dynamic convolution. In *International Conference on Learning Representations*, 2021.
- [39] CodeFuel Ltd. What is user engagement? – the ultimate guide, 2021.
- [40] Ghadah Alkhalidi, Fiona L Hamilton, Rosa Lau, Rosie Webster, Susan Michie, Elizabeth Murray, et al. The effectiveness of prompts to promote engagement with digital interventions: a systematic review. *Journal of medical Internet research*, 18(1):e4790, 2016.
- [41] Yiping Yuan, Jing Zhang, Shaunak Chatterjee, Shipeng Yu, and Romer Rosales. A state transition model for mobile notifications via survival analysis. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 123–131, 2019.
- [42] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd international conference on World Wide Web*, pages 95–106, 2013.
- [43] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*, pages 687–698, 2014.
- [44] Icek Ajzen. The theory of planned behavior. *Organizational behavior and human decision processes*, 50(2):179–211, 1991.
- [45] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.
- [46] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [47] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [48] Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6176–6185, 2017.