Tricolore: Multi-Behavior User Profiling for Enhanced Candidate Generation in Recommender Systems

Xiao Zhou, Zhongxiang Zhao, and Hanze Guo

***Abstract*—Online platforms aggregate extensive user feedback across diverse behaviors, providing a rich source for enhancing user engagement. Traditional recommender systems, however, typically optimize for a single target behavior and represent user preferences with a single vector, limiting their ability to handle multiple important behaviors or optimization objectives. This conventional approach also struggles to capture the full spectrum of user interests, resulting in a narrow item pool during candidate generation. To address these limitations, we present *Tricolore*, a versatile multi-vector learning framework that uncovers connections between different behavior types for more robust candidate generation. *Tricolore*'s adaptive multi-task structure is also customizable to specific platform needs. To manage the variability in sparsity across behavior types, we incorporate a behavior-wise multi-view fusion module that dynamically enhances learning. Moreover, a popularity-balanced strategy ensures the recommendation list balances accuracy with item popularity, fostering diversity and improving overall performance. Extensive experiments on public datasets demonstrate *Tricolore*'s effectiveness across various recommendation scenarios, from short video platforms to e-commerce. By leveraging a shared base embedding strategy, *Tricolore* also significantly improves the performance for cold-start users.**

***Index Terms*—Multi-behavior recommendation, candidate generation, user modeling, learning to rank, deep learning.**

I. Introduction

With the mission of delivering personalized recommendations to a massive user base and addressing information overload in the digital era, recommender systems have become integral to various platforms [1], [2], [3], [4], [5]. Despite notable progress, existing recommendation techniques predominantly focus on optimizing a single type of interaction, often tied directly to platform profitability, such as *purchase* in e-commerce, *rating* in movie recommendations, and *click* feedback for online news services [1], [6]. Unfortunately, these target behaviors, being high-cost and low-frequency digital traces, lead to cold-start and data sparsity issues, inaccurate user preference representations, and substantial performance degradation [6], [7]. To address this, some studies [6], [8], [9] have explored diverse user feedback types, giving rise to the emerging field of multi-behavior recommender systems (MBRS). These supplementary user behaviors, such as *browse, click, share* for news sites, *listen, favorite, add to playlist* on music platforms, and *view, click, add to cart* for online marketplaces, are often relatively richer and more readily available across domains. The core concept of MBRS is leveraging these multiple feedback types to learn more accurate user preferences for high-quality recommendations [7]. As a nascent field within recommender systems, there is ample room for enhancement.

One primary limitation of many existing MBRS is their tendency to manually ***predetermine one primary behavior*** and treat other feedback types as auxiliary data to optimize recommendations on the target behavior [1], [6]. While these approaches generally outperform their single-behavior counterparts, they rigidly prioritize one behavior over others, which may not always align with practical needs. For instance, in e-commerce, while purchasing is typically considered the target behavior, sharing an item with friends can sometimes indicate stronger user preference. Similarly, on platforms like WeChat Channels,[1] where optimizing both viewing time and interaction rates is crucial, fixed targeting of one behavior may be inadequate (see Fig. 1). This limitation can undermine the system's ability to model diverse user interests, particularly in early stages of the recommendation process, such as candidate generation. A narrow focus on a single target behavior risks filtering out potentially relevant items reflected in other types of user behavior, thereby complicating subsequent stages of the recommendation process.

Another common limitation in current MBRS is the ***insufficient detection of fine-grained interdependencies*** between different types of user behaviors. For instance, in e-commerce, behaviors like 'add-to-cart' and 'purchase' are typically positively correlated, while 'click' and 'hide' may exhibit negative correlations. Neglecting these associations in modeling can hinder the extraction of effective collaborative signals for user preference representation [10]. Many existing MBRS tend to model individual behavior types separately and then integrate them simplistically for predicting target behaviors, often overlooking these interdependencies. While recent studies like chainRec [11]

Received 6 April 2023; revised 5 July 2024; accepted 24 March 2025. Date of publication 7 April 2025; date of current version 28 May 2025. Recommended for acceptance by J. Tang. *(Corresponding author: Xiao Zhou.)*

Xiao Zhou and Hanze Guo are with the Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China (e-mail: xiaozhou@ruc.edu.cn; ghz@ruc.edu.cn).

Zhongxiang Zhao is with the WeChat Business Group, Tencent Inc., Beijing 100080, China (e-mail: abnerzxzhao@tencent.com).

The source code is publicly available at: https://github.com/abnering/Tricolore.

Digital Object Identifier 10.1109/TKDE.2025.3558503

[1] [Online]. Avaialble: https://www.wechat.com

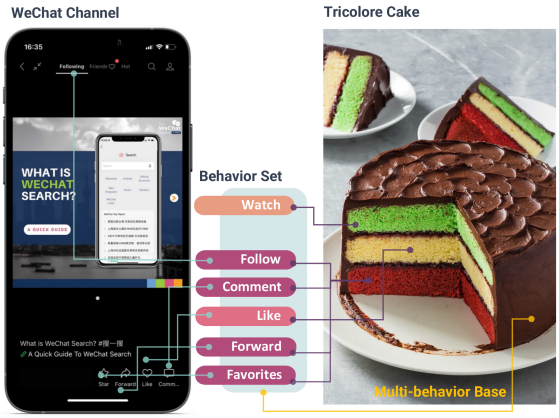

Fig. 1. An illustration of multiple feedback types in WeChat Channels and the design inspiration behind *Tricolore*.

and NMTR [9] have begun considering dependencies across feedback types, they often impose rigid sequential assumptions such as *click* $\rightarrow$ *add to cart* $\rightarrow$ *purchase* [12]. We argue that this assumption is overly simplified and too rigid to be universally applicable in practical settings. For instance, it is common for users to either add an item to their favorites or share its link with friends. However, the sequence in which these behaviors occur can vary among different users [12]. This issue is further complicated in emerging domains like short video platforms, where behavioral sequences are less defined. Overall, user behavioral patterns across different types intertwine in complex ways that defy rigid predefined sequences. While some sequential patterns exist in specific contexts, models should dynamically learn these patterns rather than imposing fixed sequences manually [2].

In addition to identifying commonalities and connections, it is crucial to address the sharp ***distinctions between individual behavior types*** in the development of MBRS. Each type of user behavior carries unique semantics that characterize diverse user preferences [2], [13]. Moreover, these behaviors exhibit highly unbalanced sparsity levels, a factor often overlooked in current approaches. Many existing multi-behavior recommendation algorithms rely on complex model structures with high computational demands, posing challenges for real-world industrial applications such as candidate generation tasks [14]. Hence, there is a pressing need for a multi-behavior recommendation framework capable of efficiently integrating multiple perspectives to capture both global commonalities and fine-grained distinctions among user behavior types simultaneously [6], [7].

To address the limitations of current MBRS techniques, we introduce *Tricolore*, a novel framework designed for comprehensive and hierarchical depiction of user preferences in multi-behavior recommendation scenarios. Inspired by the Tricolore cake (Fig. 1), our model adopts an elastic multi-bucket structure that simultaneously captures shared patterns within each behavior class while preserving the unique characteristics of individual classes. A foundational layer identifies associations across behavior types, enhancing the learning process with complementary information. We propose a lightweight and flexible multi-task architecture that allows platforms to tailor prediction objectives to specific recommendation needs. To address the varying levels of sparsity across behavior types, our framework incorporates a behavior-wise multi-view fusion module that adaptively integrates global and local features. Additionally, we introduce a popularity-balancing mechanism to mitigate the effects of popularity bias during negative sampling. Experimental evaluations on three public datasets demonstrate *Tricolore*'s effectiveness across diverse scenarios, including short video and e-commerce recommendations.

Our primary contributions can be summarized as follows:

- **We propose a novel multi-behavior recommendation framework**, which naturally enhances user preference modeling by leveraging multiple types of historical user feedback through advanced multi-view fusion techniques.
- **We design a multi-bucket structure with a custom gating mechanism**, built on rich global information, to enable behavior-wise representation learning. This allows *Tricolore* to effectively capture intricate associations between behavior types and deliver recommendations aligned with platform-specific needs.
- **We introduce a more effective negative sampling strategy** that mitigates popularity bias, balancing recommendation accuracy and item diversity during item representation learning.

II. Related Work

In this section, we provide a discussion about existing literature on multi-behavior recommender systems.

As a relatively new sub-domain in recommender systems, one of the earliest studies in MBRS field was conducted on LinkedIn products [1], where the authors presented a general discussion on three possible ways for incorporating multiple types of user feedback from an empirical standpoint. One basic idea is to train individual models for each feedback type in parallel, before combining them to get the final prediction. They also proposed a sequential training method that utilizes the previous feedback model as the prior to regularize the model for next behavior type. The third approach of joint training considers all types of feedback in a single optimization problem. Experimental results in [1] showed that the joint training method outperformed the others, especially when feedback types were correlated and some exhibited data sparsity. Even though elementary, this seminal work invoked the exploration of multi-behavior recommendations that subsequent studies were influenced by it more or less. It also pointed out the necessity of exploiting multiple user feedback types in industrial recommender systems.

To provide a clearer organization, MBRS studies can generally be categorized into three groups: ***sampling-based***, ***loss-based***, and ***model-based methods***. Within the first category that focused more on sampling refinements, Loni et al. [15] extended the vanilla Bayesian Personalized Ranking (BPR) [16] approach to distinguish the different strength levels of user preferences exhibited in various behavior types. Here, feedback types were ordered and endowed with weights according to their levels of importance, which would further influence how likely they were sampled in the training phase. Similarly, Ding

et al. [17] leveraged the information of behavior type of *view* in e-commerce and proposed a view-enhanced sampler technique for classical BPR. From a loss-based point of view, Ding et al. [18] also emphasized the importance of integrating *view* data to advance recommendation performance using the same datasets. Compared to their earlier work [17], the authors improved the model by adding pair-wise ranking relations between *purchase*, *view*, and *non-interacted actions* in the loss function instead of adopting point-wise matrix factorization methods.

Another line of MBRS research targeted on the model modification. Among them, Liu et al. [19] utilized both explicit feedback (e.g., *ratings*) and implicit feedback types (e.g., *view*, *click logs*) as input and employed explicit feedback to generate ordered partial pairs for training. Based on LSTM [20] networks, Li et al. [21] designed a framework that was capable of learning short-term intention and long-term preferences of users through different behavior combinations for the next purchase item recommendation. Another work that adopted an RNN-based model and exploited sequential user behaviors for recommendation in e-commerce scenario [22] utilized feedback types of *click, browsing, adding to cart, and dwell time*. Later, Zhou et al. [8] proposed the Multi-Relational Memory Network based on an investigation of the strength and diversity levels of behavior types and adopted the attention mechanism to capture fine-grained user preferences across multi-behavior space.

Some more recent investigations began to adopt graph neural network-based techniques for multi-behavior relational modeling. For instance, MBGCN [6] was proposed to utilize the power of graph convolutional network (GCN) in learning complicated user-item and item-item connections for the multi-behavior recommendation. Similarly, CRGCN [23] utilizes a cascading residual graph convolutional network structure to learn user preferences by refining embeddings across multiple types of behaviors. MB-CGCN [24] employs cascading graph convolution networks to learn sequential dependencies in behaviors. MBSSL [25] adopts a behavior-aware graph neural network with a self-attention mechanism to capture the multiplicity and dependencies of behaviors. Chen et al. [26] proposed the GHCF model that focused on the use of heterogeneous high-hop structural information of user-item interactions in multiple types. Some other recent attempts include FeedRec [27], in which authors employed an attention network to distinguish user engagement levels on different feedback types for news recommendations. Another framework named MMCLR [7] introduced contrastive learning (CL) in multi-behavior recommendations and designed three specific CL tasks to learn user representations from different views and behavior types.

In contrast to existing MBRS that either prioritize a single behavior or rely on rigid predefined sequences for behavior modeling, our proposed *Tricolore* model introduces a flexible and adaptive framework capable of simultaneously capturing both global commonalities and fine-grained distinctions across multiple feedback types. One of the key advantages of *Tricolore* is its ability to dynamically learn complex interdependencies between diverse behavior types without imposing overly simplistic or rigid assumptions, which is a limitation in many existing models. Furthermore, the inclusion of a popularity-balancing mechanism mitigates the impact of popularity bias, ensuring that the recommendation list not only prioritizes accuracy but also promotes diversity. These design choices enable *Tricolore* to outperform state-of-the-art MBRS models, particularly in candidate generation tasks. The ability to tailor the multi-task structure to specific platform needs also enhances its practical applicability, making *Tricolore* a versatile and robust solution for modern recommendation systems.

III. Preliminaries

In this section, we formulate our research problems and introduce the notations used throughout the paper.

Definition 1 (Candidate Generation). Candidate generation in recommendation systems is crucial for selecting a personalized subset of items from a large pool, aligning with user preferences by predicting relevant items based on user behavior and item characteristics.

Definition 2 (Multi-Behavior Recommendation). Let $\mathcal{U}$ denote the set of users and $\mathcal{V}$ denote the set of items. The multi-behavior interaction tensor $\mathcal{Y} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}|\times K}$ is defined to reflect multiple types of implicit user feedback, where each entry y_{uv}^k in $\mathcal{Y}$ records whether user $u \in \mathcal{U}$ has interacted with item $v \in \mathcal{V}$ under the behavior type k. $K(K \geq 2)$ is the number of user behavior types. Here given the behavior type k, y_{uv}^k is set to 1 if the interaction between user u and item v is observed. Otherwise, it is assigned the value of 0. Different from most existing studies on multi-behavior recommendations where a target behavior type is pre-selected manually for optimization, the multi-behavior recommender system in this work adopts a flexible go-setting strategy and allows researchers or engineers to pursue tailor-made designs for different recommendation scenarios and the shift of purpose. More specifically, taking advantage of the multi-task learning structure in *Tricolore*, one can either estimate the overall likelihood $\hat{y}_{uv}$ that user u would enjoy a non-interacted item v or her more refined preference on particular behaviors $\hat{y}_{uv}^k$ by fully utilizing the rich historical information across multiple types of user feedback and generate the Top-N item list for recommendation.

In addition, considering that online platforms usually allow users to interact with items in many ways, *Tricolore* encourages platforms to classify behavior types into several categories according to their needs with an aim to alleviate the data sparsity issue and streamline the model structure. For each item v, we compute key indicators (e.g., 10s completion rate, purchase rate, like rate) for K behaviors. Subsequently, across all items $\mathcal{V}$, we calculate the correlation between pairs of these indicators and organize the K behaviors into C categories based on the correlation results $(C \leq K)$. Behaviors with high positive correlations are grouped into the same category.

IV. Methodology

Next, we elaborate on the technical details of *Tricolore*.

Overview. Essentially, the *Tricolore* framework is composed of four key modules: i) *multi-behavior encoder* for initial user

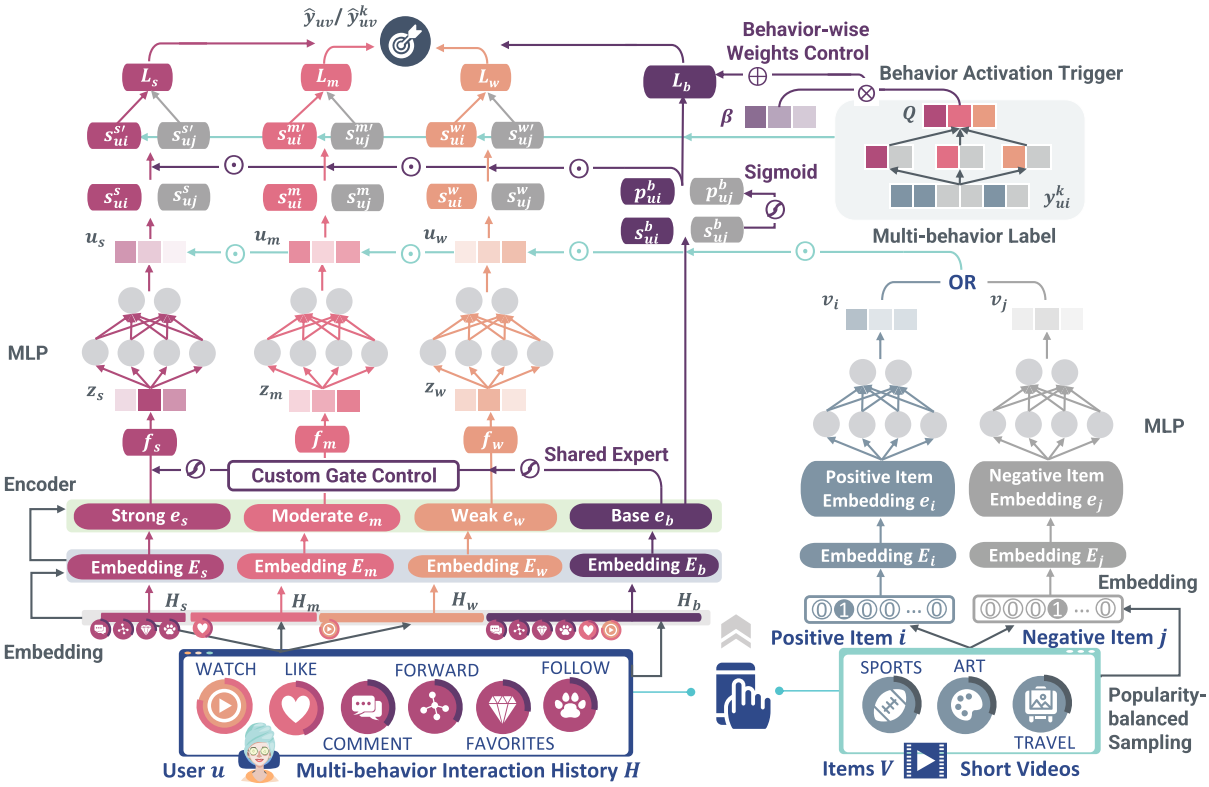

Fig. 2. The architecture of the proposed *Tricolore* framework.

embedding learning on various behaviors; ii) *behavior-wise multi-view fusion module* for global-information-enhanced user representation set formulation; iii) *popularity-balanced item representation learning* for item embedding; iv) *allied multi-task prediction module* for joint learning and the generation of multi-behavior recommendation lists. For a better illustration, we employ the short video recommendation scenario as an example to display the structural details of *Tricolore* in graphical representation (Fig. 2) and textual description below.

A. Multi-Behavior Encoder

Unlike most recommender systems that rely on a single type of behavior for user representation, or other multi-behavior approaches that merge feedback types into a single embedding, we introduce *Tricolore*, which hierarchically constructs multi-vector user embeddings. These embeddings capture a more comprehensive view of user preferences across different behaviors. We begin by categorizing K user feedback types into C groups based on similarities in user engagement and their impact on the platform. Using Pearson correlation analysis (as introduced in Section III), we classify behavior types. For instance, in the short video scenario, we select key indicators for each user behavior, such as 10-second completion rate for *watch*, *like rate* for *like*, *comment rate* for *comment*, *share rate* for *forward*, *save rate* for *favorite*, and *follow rate* for *follow*. In the e-commerce scenario, we adopt similar behavior-specific rates, including those for *purchase*, *cart*, *favorite*, and *click*.

We then compute Pearson correlation coefficients and group behaviors with high correlations. In the short video scenario, *watch* and *like* show low correlations with other behaviors (below 0.1), while *comment*, *forward*, *favorite*, and *follow* have higher correlations (above 0.3). As a result, we classify behaviors into three categories: "strong," "moderate," and "weak," reflecting the user's affinity for items. For example, *comment*, *forward*, *favorite*, and *follow* are categorized as "strong," *like* as "moderate," and *watch* as "weak." In e-commerce, *cart* and *favorite* are grouped as "moderate" (correlation 0.7630), *purchase* is "strong," and *click* is "weak."

Subsequently, we extract the interaction history of user u with the "strong" behavior set, denoting it as H_s. We then create a one-hot embedding matrix, $E_s \in \mathbb{R}^{K_s \times d}$, which is further encoded to generate the initial embedding for "strong" behavior, denoted as $e_s \in \mathbb{R}^d$. Here, K_s represents the number of behavior types in the "strong" category, and d denotes the dimension of user embeddings. Formally:

$$E_s = g_{emb}\left(H_s \mid H_s \in \mathcal{H}\right), \\ e_s = g_{enc}\left(E_s\right), \tag{1}$$

where $g_{enc}(\cdot)$ is the encoder function for initial user representations learning that one can choose to adopt the average pooling, self-attention, or other attention mechanisms according to need here. $g_{emb}(\cdot)$ is the embedding operation that maps interaction sequences to vectors. $\mathcal{H}$ is the complete user muti-behavior interaction history.

In a similar way, we can respectively obtain the embeddings of user behavior types that show moderate and weak liking, which are $e_m \in \mathbb{R}^d$ and $e_w \in \mathbb{R}^d$. Apart from these representations of user, we also learn a base embedding $e_b \in \mathbb{R}^d$ for the capture of the global preference of the user, which takes K types of user behaviors into consideration and serves as a significant supplement for individual behavior classes in the *behavior-wise multi-view fusion module* next.

B. Behavior-Wise Multi-View Fusion

The motivation behind *Tricolore*'s multi-bucket model and multi-view fusion strategy is to address sparsity issues in recommender systems while capturing both local preferences at each behavior level and global user preferences.

Most existing systems focus on a single primary behavior, like *click* in advertising or *purchase* in e-commerce. Although these behaviors are tightly linked to platform profit and user intent, they often suffer from data sparsity. While multi-behavior models help alleviate this, further improvements are needed to better model user preferences across behaviors. To this end, we propose the *behavior-wise multi-view fusion module*, which uses a custom gate control mechanism to adaptively refine user embeddings by allowing each behavior to leverage shared expert knowledge e_b. Specifically, we introduce learnable weight vectors $\mathcal{W} = \{W_s, W_m, W_w\}$, which enable each behavior to determine how much information to incorporate from other behavior types. For instance, the embedding for the *strong* behavior class is modeled as follows:

$$\begin{aligned} f_s &= \sigma\left(e_s W_s^T\right), \\ z_s &= f_s e_b + (1 - f_s)\, e_s, \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function. $W_s \in \mathbb{R}^d$ is the weight vector of *strong* behaviors. f_s is a learnable parameter that determines the extent to which information is incorporated from the *base* knowledge into the representation of *strong* behaviors. $z_s \in \mathbb{R}^d$ is the revised *strong* user embedding.

We then utilize z_s as the input of a multilayer perceptron (MLP) to obtain the final *strong* user embedding $u_s \in \mathbb{R}^d$ as:

$$u_s = MLP\left(z_s\right). \tag{3}$$

In a similar vein, *Tricolore* can generate the final user embeddings of u_m and u_w for *moderate* and *weak* behavior types, respectively.

This multi-view fusion approach enables us to learn user representations that maintain independent components at each behavior level, while sharing useful knowledge across behaviors. A key feature is the use of learnable parameters in the gate control for each behavior type.

For *strong* behaviors (e.g., *comment*, *forward*, *favorite*, *follow* in short video), the contribution from the base embedding e_b is expected to be larger due to sparsity. In contrast, *watch* behavior, though weaker, is still informative. Interestingly, stronger behaviors tend to correlate positively with *watch*, as users often engage in stronger interactions after watching for more than 10 seconds. Therefore, the multi-view fusion technique improves the quality of *weak* embeddings more by purifying the information rather than supplementing it. This is elaborated in Section V-D.

In special cases, such as cold-start users with limited interaction history across any behavior category, the shared base embedding e_b becomes even more crucial in shaping user representations. The varying parameters learned across different behavior types in this module, along with the performance of *Tricolore* in cold-start settings, are further discussed in Section V-G.

C. Popularity-Balanced Item Representation

For item embedding learning in *Tricolore*, the identities of a pair of positive and negative items, i and j, are initially represented as two binary sparse vectors E_i and E_j via one-hot encoding. Here positive item $i \in V_u^+$ is a video that user u has interacted with on more than one behavior type, while negative item $j \in V_u^-$ means that no interaction between user-item pair $\langle u, j \rangle$ is observed. Then they are projected to low-dimensional dense vectors to generate item embeddings, denoted as $e_i \in \mathbb{R}^d$ and $e_j \in \mathbb{R}^d$. After that, MLP is implemented to learn the final item embeddings v_i and v_j through:

$$v_i = MLP(e_i); \quad v_j = MLP\left(e_j\right). \tag{4}$$

The dimension of the generated vectors v_i and v_j is also d.

Considering the ubiquitous popularity bias problem that haunts many recommender systems [28], the proposed *Tricolore* framework adopts a popularity-balanced negative sampling technique inspired by that in [29] to penalize the sampling probability given to items according to their historical popularity. Mathematically, the probability for item v_j to be selected as a negative sample $P_n(v_j)$ can be computed by the formula:

$$P_n(v_j) = \frac{c(v_j)^\gamma}{\sum_j (c(v_j)^\tau)}, \forall j = 1, 2, \ldots, |V_u^-|, \tag{5}$$

where $c(v_j)$ is the frequency of item v_j that emerged in the historical interactions of all users $\mathcal{U}$, which reflects the overall popularity of the item. γ and τ denote smoothness powers.

The core idea behind this approach is to mitigate item popularity bias by increasing the negative sampling probability for more popular items. This encourages a more balanced distribution of negative samples and enhances the diversity of recommendation lists. The goal is to ensure that both popular and niche items are fairly represented during model training, preventing overfitting to frequently interacted items. This balanced treatment improves the model's generalization capability and promotes greater diversity in recommendations.

In recommender systems, particularly in two-tower models like *Tricolore*, this strategy helps address the challenge of imbalanced exposure between popular and less popular items. Our method specifically targets the imbalance issue in multi-behavior scenarios, where users engage with a wide range of behaviors. This is especially important, as popularity bias can blur behavior distinctions, and our approach ensures fairness in how items are treated across different behavior types.

The trade-off between recommendation accuracy and item popularity bias, as well as how to properly adjust the smoothness powers to strike the right balance, will be discussed in Section V-H.

D. Allied Multi-Task Prediction

Thus far, we have obtained multi-behavior user preference representations and embeddings for both positive and negative items. It is fairly straightforward to calculate the prediction scores by dot product for each user-item pair of embeddings. For instance, the scores for base user embedding and items can be achieved via:

$$s_{ui}^{b} = e_b \cdot v_i; \quad s_{uj}^{b} = e_b \cdot v_j. \tag{6}$$

We can also calculate the initial scores for *strong* user behaviors as:

$$s_{ui}^{s} = u_s \cdot v_i; \quad s_{uj}^{s} = u_s \cdot v_j. \tag{7}$$

Similarly, the relevant initial scores for moderate and weak user behaviors (s_{ui}^{m}, s_{uj}^{m}) and (s_{ui}^{w}, s_{uj}^{w}) can be also obtained, respectively.

However, considering the underlying sequential patterns may exist among user interactive behaviors, i.e., $click \rightarrow cart \rightarrow purchase$ or $watch \rightarrow like/forward/follow/comment$, we suggest taking one step further and propose an *allied multi-task learning* scheme. The key assumption here is that user behaviors do not exist independently but have mutual influences on each other. Only when a user shows basic interest in an item, he/she would create the following behaviors, such as following the author or forwarding the item to a friend. Instead of determining the sequential pattern of influence manually as most current multi-behavior studies do, we advocate a more elastic plan. Specifically, we convert the scores of base embedding into probabilities first by employing the *sigmoid* function as:

$$p_{ui}^{b} = \sigma\left(s_{ui}^{b}\right); \quad p_{uj}^{b} = \sigma\left(s_{uj}^{b}\right). \tag{8}$$

We then tune the initial prediction scores of *strong* behaviors to get its final version $(s_{ui}^{s\prime}, s_{uj}^{s\prime})$ by:

$$s_{ui}^{s\prime} = p_{ui}^{b} s_{ui}^{s}; \quad s_{uj}^{s\prime} = p_{uj}^{b} s_{uj}^{s}. \tag{9}$$

Similarly, we calculate the prediction scores for moderate and weak user feedback types, $(s_{ui}^{m\prime}, s_{uj}^{m\prime})$ and $(s_{ui}^{w\prime}, s_{uj}^{w\prime})$, which signal the likelihood that user u would be interested in an item on particular behavior classes. The higher the score, the more likely it would be.

E. Optimization

We leverage the pair-wise ranking loss for optimization in *Tricolore*. Instead of treating unobserved entries as negative feedback in point-wise loss [30], pair-wise learning [31] focuses on the relative positions of each observed-unobserved pair of items that observed entries are expected to rank higher than their counterparts. Mathematically, the objective function for each behavior level can be defined as:

$$\begin{aligned}
\mathcal{L}_s &= \sum_{(u,i,j)\in\mathcal{D}_s} \max\left(0, s_{uj}^{s\prime} - s_{ui}^{s\prime} + \lambda^{s}\right), \\
\mathcal{L}_m &= \sum_{(u,i,j)\in\mathcal{D}_m} \max\left(0, s_{uj}^{m\prime} - s_{ui}^{m\prime} + \lambda^{m}\right), \\
\mathcal{L}_w &= \sum_{(u,i,j)\in\mathcal{D}_w} \max\left(0, s_{uj}^{w\prime} - s_{ui}^{w\prime} + \lambda^{w}\right),
\end{aligned} \tag{10}$$

here $\mathcal{D}_s$, $\mathcal{D}_m$, and $\mathcal{D}_w$ denote the sub-datasets under *strong*, *moderate*, and *weak* behavior levels. λ^s, λ^m, and λ^w represent the safety margin sizes within the pairwise hinge loss function, which serve to separate negative items from positive items within each behavior class.

For the base loss function design in *Tricolore*, we consider both cross-class behavior dependencies and class-specific semantics by:

$$\begin{aligned}
\mathcal{L}_b = \sum_{(u,i,j)\in\mathcal{D}} &\max\left(0, s_{uj}^{b\prime} - s_{ui}^{b\prime} + \lambda^{b}\right) \\
&\left(\beta_s Q_s + \beta_m Q_m + \beta_w Q_w\right),
\end{aligned} \tag{11}$$

where $\mathcal{D}$ represents the whole dataset on various behavior types. λ^b is the margin separating positive and negative items in *base* view. β_s, β_m, and β_w are *behavior-wise control weights* to determine how much each behavior level contributes to the *base* loss $\mathcal{L}_b$. $\mathcal{Q} = \{Q_s, Q_m, Q_w\}$ denotes *behavior activation trigger* that indicates interactions are observed on which behavior types for each positive sample $\langle u, i\rangle$. Here each element in $\mathcal{Q}$ is a binary indicator vector that contains 1 for the observed behavior class and 0 otherwise. $\mathcal{Q}$ determines which branch(es) of behavior types would be activated for the individual level loss calculation and the *base* loss $\mathcal{L}_b$.

The overall objective function for *Tricolore* is a weighted sum of all the contributions of each related individual objective above as:

$$\mathcal{L} = \mathcal{L}_b + \alpha\mathcal{L}_s + \varepsilon\mathcal{L}_m + \zeta\mathcal{L}_w + \mu\left\|\theta\right\|_2^2, \tag{12}$$

here α, ε, and ζ are weighting parameters specified to influence to what degree each behavior level-specific effect should be taken into account during the optimization. θ is the model parameter set; and μ is a parameter that controls the importance of the last term, where we apply regularization to prevent overfitting by using the dropout strategy and adding L_2-norm terms.

It is also worth noting that *Tricolore* adopts a flexible multi-task framework, enabling platforms to choose whether to predict the overall likelihood that a user will enjoy an item $\hat{y}_{uv}$ or his/her interest in a specific behavior type $\hat{y}_{uv}^{k}$. Besides, by injecting discriminative signals λ, we can encode the *strength* deviation corresponding to different behavior levels and views. Here a larger setting of margin would push the positive and negative samples farther apart suggesting that this behavior class or view is more reliable in depicting user preferences.

TABLE I
STATISTICS OF THE DATASETS

Dataset	Behavior	#User	#Item	#Inter	Density
Channels	Watch*	218,919	44,846	2,157,358	0.022%
	Like	179,270	24,176	897,019	0.020%
	Comment	13,649	1,869	35,156	0.138%
	Forward	81,792	9,059	281,547	0.038%
	Favorite	25,507	3,953	90,989	0.090%
	Follow	26,473	3,312	55,513	0.063%
Tmall	Purchase*	4,760	3,037	6,237	0.043%
	Cart	1,958	2,353	3,588	0.078%
	Favorite	980	1,502	1,993	0.135%
	Click	4,675	66,927	76,536	0.245%
CIKM	Purchase*	13,673	10,629	17,051	0.012%
	Cart	3,208	3,544	4,260	0.037%
	Favorite	1,298	1,624	1,765	0.083%
	Click	13,325	26,216	59,659	0.017%

'*' denotes the primary behavior type in baseline models if applicable.

V. EXPERIMENTS

A. Experimental Settings

1) Datasets: The experiments are conducted using publicly available datasets from a short-video application and two e-commerce platforms, which are described below.

WeChat Channels: [2] This is a short-video dataset released by WeChat Big Data Challenge 2021. It contains six types of user-video interactions on the WeChat Channels application.

Tmall: [3] This is an open dataset from Tmall,[4] one of the largest e-commerce platforms in China. It contains four typical types of user behavior in e-commerce scenarios, including *click, favorite, add-to-cart*, and *purchase*.

CIKM: [5] It is offered by the CIKM 2019 EComm AI Challenge and includes the same user behaviors as the Tmall dataset.

We weed out users with less than ten interactions for the CIKM data and three interactions for WeChat Channels and Tmall datasets, respectively. To counter the imbalance between the rare and frequent user feedback types, the datasets are intentionally sampled so that each of them is well represented in the training phase. The statistics of the processed datasets are summarized in Table I. Here we color the cells for behavior types according to which bucket they belong to in *Tricolore* for the experiments. Although as a multi-task learning framework, *Tricolore* does not require us to specify a single target behavior to optimize, we use '$*$' to denote a primary behavior type generally employed by traditional single-behavior recommender systems in each scenario for further comparative analyses.

2) Baselines: To examine the effectiveness of the proposed *Tricolore*, a series of state-of-the-art baselines are employed for comparison. It should be mentioned that for multi-behavior recommendations, only models applicable to candidate generation tasks are selected. Besides, to guarantee the fairness in comparison and dispel doubts on the data size inequality caused by involving more feedback types in MBRS, we treat all behavior types as primary feedback in the single-behavior recommender systems. These baseline models include:

Classic Single-behavior Recommendation Algorithms.

- *MF-BPR [31]:* Bayesian Personalized Ranking is a classic method for item recommendation from implicit feedback, which is directly optimized for ranking.
- *DSSM [32]:* DSSM is an effective two-tower model for large-scale industrial recommender systems which makes predictions by matching the query and documents.
- *NCF [33]:* This is a typical recommendation algorithm that augments collaborative filtering with deep neural networks.

Multi-behavior Recommendation Algorithms.

- *MC-BPR [15]:* It is a sampling-based MBRS that samples positive and negative instants from multiple relations.
- *FeedRec [27]:* FeedRec is a recent MBRS model proposed for news recommendation based on an attention network.
- *MBGCN [6]:* It is a graph model with the ability to capture user-item and item-item level multi-behavior information.
- *MMCLR [7]:* MMCLR aims to predict the target behavior by the construction of contrastive learning tasks and the fusion strategy of the sequence model and graph model.
- *CRGCN [23]:* CRGCN utilizes a cascading residual graph convolutional network structure to learn user preferences by refining embeddings across multiple types of behaviors.
- *MB-CGCN [24]:* It employs cascading graph convolution networks to learn sequential dependencies in behaviors.
- *MBSSL [25]:* It adopts a behavior-aware graph neural network that incorporates a self-attention mechanism to capture the multiplicity and dependencies of behaviors.

Multi-task Learning Algorithms.

- *MMoE [34]:* It adapts MoE structure to multi-task learning, allowing expert submodels to be shared across tasks within a gating network optimized for each individual task.
- *ESMM [35]:* ESMM leverages sequential patterns in user actions and employs a transfer learning strategy to mitigate sample selection bias and address data sparsity issues.
- *PLE [36]:* It is a multi-task learning approach that employs a progressive routing mechanism to differentiate the semantic knowledge of shared and task-specific components.

3) Evaluation Metrics: The performance of the models is evaluated mainly by two widely-used ranking metrics, hit ratio (*HR@K*) and normalized discounted cumulative gain (*NDCG@K*), where K is set to $\{5, 10\}$. Besides, since the popularity-balanced strategy is adopted in the negative sampling phase, apart from metrics in accuracy, we also pay close attention to the model's performance in popularity bias. Here, we employ the average popularity of the top-10 item recommendation lists (ARP metric in [37]) for evaluation. The positive item is compared with 99 negative samples in the test stage to evaluate the ranking performance of the models.

4) Parameters Settings: We implement *Tricolore* using Tensorflow 2.0.0 and fine-tune the hyperparameters by grid-search according to its performance on the validation set. We search the sequence of behavior sample lengths in $\{1, 3, 5, 10\}$, embedding size in $\{16, 32, 64, 128\}$, and dropout rate [38] in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The model's embedding size is set

[2][Online]. Avaialble: https://algo.weixin.qq.com/2021/problem-description
[3][Online]. Avaialble: https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[4][Online]. Avaialble: https://www.tmall.com
[5][Online]. Avaialble: https://tianchi.aliyun.com/competition/entrance/231721/introduction

TABLE II
EXPERIMENTAL RESULTS ARE PRESENTED FOR THE THREE DATASETS, WITH THE BEST PERFORMANCE HIGHLIGHTED IN BOLDFACE AND THE BEST BASELINE MODEL MARKED WITH '$*$'

Methods	WeChat Channels				Tmall				CIKM			
	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10
MF-BPR	0.3009	0.4389	0.2027	0.2470	0.4498	0.4830	0.4069	0.4177	0.2425	0.3526	0.1428	0.1843
DSSM	0.2479	0.3834	0.1619	0.2057	0.3345	0.4045	0.2629	0.2854	0.2440	0.3520	0.1466	0.1815
NCF	0.2594	0.4044	0.1691	0.2156	0.3663	0.4186	0.3061	0.3231	0.2637	0.3457	0.1867	0.2130
MC-BPR	0.3216	0.4618	0.2173	0.2624	0.3720	0.4306	0.3123	0.3310	0.2701	0.3479	0.1957	0.2205
FeedRec	0.2269	0.3510	0.1456	0.1855	0.4470	0.4886	0.4040	0.4182	0.2651	0.3620	0.1821	0.2133
MBGCN	0.2294	0.3726	0.1440	0.1898	0.2258	0.3512	0.1471	0.1871	0.1910	0.2438	0.1420	0.1591
MMCLR	0.2106	0.3264	0.1413	0.1782	0.3330	0.4214	0.2395	0.2679	0.2704	0.3475	0.1932	0.2179
CRGCN	0.3060	0.4502	0.2016	0.2479	0.5078*	0.5255*	0.4825*	0.4881*	0.2924	0.3638	0.2242*	0.2470*
MB-CGCN	0.2751	0.5424	0.1477	0.2340	0.4965	0.5120	0.4674	0.4723	0.2921	0.3673*	0.2146	0.2386
MBSSL	0.4097*	0.5680*	0.2277*	0.3062*	0.4986	0.5099	0.4634	0.4670	0.2883	0.3638	0.2142	0.2384
MMoE	0.3083	0.4530	0.2082	0.2549	0.4052	0.4589	0.3540	0.3719	0.2730	0.3411	0.1790	0.2009
ESMM	0.3187	0.4682	0.2084	0.2566	0.2765	0.3338	0.2232	0.2417	0.3035*	0.3608	0.2032	0.2217
PLE	0.3171	0.4571	0.2114	0.2547	0.4490	0.4873	0.4069	0.4191	0.2745	0.3373	0.1797	0.1999
Ours	**0.4564**	**0.6134**	**0.3262**	**0.3769**	**0.5677**	**0.5890**	**0.5108**	**0.5290**	**0.4288**	**0.4797**	**0.2943**	**0.3186**
Impr.	+11.40%	+7.99%	+43.26%	+23.09%	+11.80%	+12.08%	+5.87%	+8.38%	+41.29%	+30.60%	+31.27%	+28.99%

Reported mean and standard deviation values are based on the results of 5 random runs.

to 32; the dropout rate is 0.1. The margin of λ employed in pair-wise ranking is explored among $\{0.05, 0.1, 0.3\}$ and set to 0.1. The weighting parameters of α, ε, and ζ in the overall loss are set to (0.1,0.1,0.1). We optimize all the baseline models as well as *Tricolore* by the Adam optimizer [39] and search the learning rate in $\{0.1, 0.01, 0.005, 0.001, 0.0001\}$. All models' batch size is set to 256 and other parameters are followed by default settings according to the respective papers.

B. Performance Comparison

Table II displays the performance of *Tricolore* and the baseline models across datasets. To ensure reliability, we conduct five tests for each method and average the outcomes for the final results. The analysis yields the following observations:

1) Performance of Tricolore*:* As depicted in Table II, *Tricolore* demonstrates state-of-the-art performance across all three datasets, outperforming baseline models in four evaluation metrics. Particularly noteworthy is its superiority over the best baseline model by over 40% in *NDCG@5* on WeChat Channels and in *HR@5* on CIKM.

2) Single-Behavior versus Multi-Behavior Methods: When comparing the performance of MBRS baseline models with their single-behavior counterparts, it is evident that the MBRS class as a whole offers superior results. Specifically, MBSSL stands out as the best baseline model across all evaluation metrics on WeChat Channels, while CRGCN performs exceptionally well in e-commerce scenarios on Tmall and CIKM.

In a nutshell, the overall experimental results demonstrate the effectiveness and generalization capabilities of *Tricolore* in multi-behavior recommendation tasks across various scenarios. This can probably be attributed to our fine-grained multi-vector learning strategy adopted in user representation and behavior-wise multi-view fusion mechanism in *Tricolore* framework, which will be discussed more in the following ablation study.

TABLE III
ABLATION STUDY ON WECHAT CHANNELS DATASET

	HR@5	HR@10	NDCG@5	NDCG@10
w/o MVF	0.3087	0.4614	0.2037	0.2539
w/o MVL	0.3034	0.4610	0.1985	0.2493
w/o MTL	0.2479	0.3834	0.1619	0.2057
w/o AMT	0.3057	0.4427	0.2062	0.2501
w/o CAT	0.2882	0.4420	0.1918	0.2412
Tricolore	**0.3230**	**0.4737**	**0.2189**	**0.2674**

C. Ablation Study

We further conduct the ablation study over key components of *Tricolore* to better understand the effects of each individual module. More specifically, we introduce the following four variants of the model:

- *w/o MVF:* The behavior-wise multi-view fusion module is removed in the stage of user preference modeling.
- *w/o MVL:* It removes the multi-vector learning strategy, treats all user-item interactions as input regardless of behavior types, and learns a single vector for user representation.
- *w/o MTL:* This variant changes the multi-task learning framework of *Tricolore* to a single-task structure.
- *w/o AMT:* The allied learning scheme in the final prediction phase is removed that base probabilities are not utilized here.
- *w/o CAT:* This is a variant without categorization that employs individual embeddings per behavior for prediction.

To see the evaluation results in Table III, we can draw the conclusion that all the key modules contribute to the overall performance of *Tricolore*. Among them, the multi-task learning strategy plays the most significant role that the performance would drop by around 20% in all the metrics without it. In

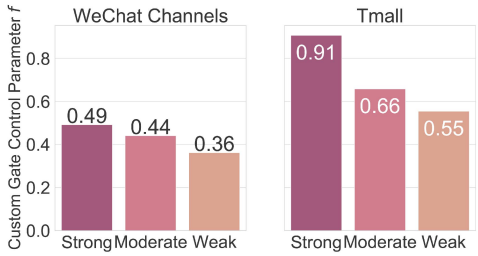

Fig. 3. Study of custom gate control parameters f on WeChat and Tmall.

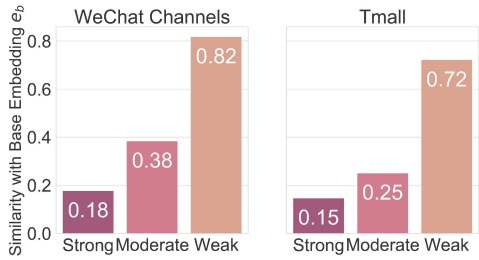

Fig. 4. Study of similarities between individual behavior class embeddings and the base embedding e_b.

addition, there are no evident differences among the contributions from the other four modules that a relative reduction of less than 10% is observed in each of the metrics.

D. Associations Between Behaviors

Next, we study the underlying associations between each individual behavior category and other behavior buckets as a whole, which correspond to local and global views, respectively. *Tricolore* enables us to do such analyses via introducing the custom gate control mechanism in the *Behavior-wise Multi-view Fusion* module.

1) Custom Gate Control Parameter f: As introduced in Section IV-B, the base embedding e_b that conveys information from all types of historical interactions of the user serves as a shared expert to assist in the representation learning on each behavior category according to their needs. The extent to which each behavior type in a branch relies on the base expert is controlled by a learnable gate control parameter f. Here a larger value indicates a greater reliance on the base expert (see (2)). In Fig. 3, we compare the custom gate control parameters of *strong*, *moderate*, and *weak* behaviors, $\{f_s, f_m, f_w\}$, learnt by *Tricolore* on the datasets of WeChat Channels and Tmall. As can be seen from the figure, behavior types falling into the *strong* group have the highest values of f on both datasets. This phenomenon is consistent with our expectations that strong user behaviors are generally high-cost and low-frequency digital traces, which suffer most from the sparsity issue and thus need more supplements from other behavior types. In addition, this result further illustrates the necessity of exploiting multiple behavior types in recommender systems since relying on single strong primary behavior may hard to represent user preferences well. Besides, this pattern is more evident in e-commerce scenarios where the f_s learned is up to 0.91, suggesting that the behavior types of *click, favorite*, and *add-to-cart* play significant roles in the prediction of *purchase* behavior that should not be neglected.

2) Embedding Similarities Between Behaviors: To further investigate the associations among various user feedback types, we also calculate the cosine similarities between the embedding of each behavior category in $\{e_s, e_m, e_w\}$ and the base embedding e_b on WeChat Channels and Tmall datasets. From the results in Fig. 4, we can observe that the user representation of *strong* behavior class presents the lowest similarities to the base embedding, followed by the *moderate* and *weak* classes. In addition, the order of behavior classes is consistent between the two datasets. This phenomenon may suggest that the user representations of behavior types with larger amounts of interactions (e.g., *watch* and *click*) tend to be more similar to the base user embeddings regardless of recommendation scenarios. When looking at the specific numeric values, we can find it varies across scenarios that WeChat Channels offers relatively higher values than Tmall. This observation may indicate that user behavior types in short-video applications seem to be more similar to each other, compared to e-commerce platforms.

TABLE IV
COMPARISON BETWEEN DIFFERENT NUMBER OF BUCKETS ON WECHAT

	HR@5	HR@10	NDCG@5	NDCG@10
Two Buckets	0.3109	0.4460	0.2077	0.2508
Three Buckets	0.3230	0.4737	0.2189	0.2674
Impr.	+3.89%	+6.21%	+5.41%	+6.62%

E. Optimum Number of Buckets

To ascertain the optimal number of buckets for behavior classification in *Tricolore*, we conducted comprehensive experiments under various settings and observed that employing three buckets generally yields superior results compared to other configurations. Using WeChat as an illustration, the enhancements achieved with three buckets, as opposed to two, are detailed in Table IV. In the two-bucket classification, *watch* is placed in one bucket, and all other behavior types in the other. Consistently, for the other two datasets employed, the optimal number of buckets is also determined to be three. However, it is crucial to acknowledge that the optimal number of buckets may vary when users apply *Tricolore* to their own data. It is important to clarify that the *Tricolore* framework proposed is a versatile one, and the recommendation of three buckets is not rigidly fixed.

F. Bucket-Wise Prediction Task

Having seen the overall effectiveness of *Tricolore*, we think it is also intriguing to assess its performance in each individual bucket recommendation task. Here we compare the bucket-wise results of *Tricolore* with the best baseline MC-BPR on WeChat data. It can be seen from Table V that our model offers better results in each individual task. In addition, stronger behaviors present relatively larger improvements. This pattern is also observed on Tmall that the improvements of *Tricolore* in strong

TABLE V
PREDICTION FOR INDIVIDUAL BUCKET ON WECHAT DATA

Bucket	Methods	HR@5	HR@10	NDCG@5	NDCG@10
Strong	MC-BPR	0.3567	0.5158	0.2428	0.2942
	Tricolore	0.4261	0.5352	0.2880	0.3238
	Impr.	+19.40%	+3.76%	+18.62%	+10.07%
Moderate	MC-BPR	0.3304	0.4780	0.2234	0.2713
	Tricolore	0.336	0.4815	0.2287	0.2755
	Impr.	+1.66%	+0.73%	+2.4%	+1.54%
Weak	MC-BPR	0.3023	0.4523	0.2062	0.2545
	Tricolore	0.3227	0.4549	0.2206	0.2631
	Impr.	+6.73%	+0.58%	+6.97%	+3.37%

TABLE VI
RESULTS FOR COLD-START USERS ON WECHAT DATASET

	HR@5	HR@10	NDCG@5	NDCG@10
MC-BPR	0.2653	0.4036	0.1764	0.2208
Δ	(+30.91%)	(+40.83%)	(+13.44%)	(+22.51%)
MBGCN	0.2125	0.3429	0.1334	0.1754
Δ	(+63.43%)	(+65.76%)	(+50.00%)	(+54.22%)
MMCLR	0.1769	0.2958	0.1177	0.1558
Δ	(+96.33%)	(+92.16%)	(+70.01%)	(+73.62%)
Ours	0.3473	0.5684	0.2001	0.2705

behaviors (*purchase*) is the largest, suggesting that the proposed model does not perform well at the cost of performance losses for stronger signals. Instead, for strong behaviors that many conventional single-task recommender systems have concerned most, *Tricolore* performs even better, demonstrating its business values for real-world platforms.

G. Experiments on Cold Start Users

With the implementation of the shared base embedding strategy in *Tricolore*, designed to assist users with limited interaction history in shaping user representations, we specifically targeted 20% of users with the fewest interactions from the WeChat dataset for cold-start experiments. The results of *Tricolore* in this cold-start setting are then compared with the outcomes among the MBRS methods, as detailed in Table VI. Encouragingly, our model demonstrates significant improvements in the cold-start recommendation setting compared to all MBRS baselines across various metrics, highlighting *Tricolore*'s superiority in addressing cold-start problems. In comparison to MMCLR, the improvements across metrics are substantial, exceeding 70% . Furthermore, concerning the two *HR* metrics, the enhancements of our approach compared to the baselines are more pronounced than the NDCG metrics, indicating the efficacy of our model for candidate generation tasks. These findings underscore that the advantages derived from the design of shared base embedding in MBRS are manifold, as *Tricolore* effectively mitigates both cold-start user issues and sparse behavior type issues simultaneously.

H. Trade-Off Between Popularity and Accuracy

As detailed in Section IV-C, the popularity bias can be naturally alleviated through the popularity-balance technique in

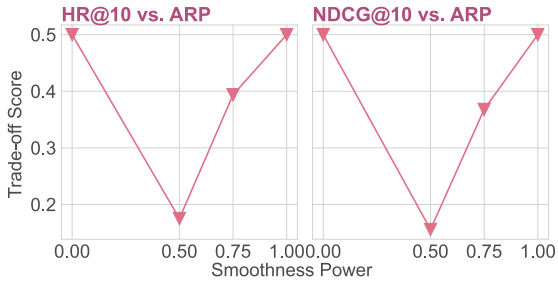

Fig. 5. Scoring the trade-off between accuracy and popularity metrics.

TABLE VII
POPULARITY BIAS FOR SMOOTHNESS POWERS ON WECHAT CHANNELS

	0.0 (w/o)	0.5	0.75	1.0
MC-BPR	1181.51	942.44	841.51	808.35
Ours	1073.29	1014.38	867.37	802.37

TABLE VIII
RESULTS FOR DIFFERENT SMOOTHNESS POWERS ON WECHAT CHANNELS

	HR@5	HR@10	NDCG@5	NDCG@10	ARP
0.0	0.4564	0.6134	0.3262	0.3769	1073.29
0.5	0.3407	0.4880	0.2318	0.2795	1014.38
0.75	0.3230	0.4737	0.2189	0.2674	867.37
1.0	0.3210	0.4608	0.2180	0.2637	802.37

negative sampling, penalizing the sampling probability assigned to items with higher popularity. To experimentally illustrate its effectiveness, we compare the models' performance in terms of the average popularity of the top-10 item recommendation lists (ARP metric in [37]). A higher ARP value indicates a more severe popularity bias. The results in Table VII demonstrate that, aided by the popularity-balance technique, *Tricolore* effectively reduces popularity bias. We apply this technique to MC-BPR as well and observe a significant reduction in popularity bias, highlighting the versatility of the strategy in more general MBRS.

Experiments conducted across various smoothness power settings reveal a discernible trade-off between item popularity and recommendation accuracy, as shown in Table VIII. To determine the optimal smoothness power value, we frame the task as a multi-objective optimization problem involving pairs of accuracy metrics (HR@5, HR@10, NDCG@5, or NDCG@10) and a popularity metric (ARP). We first standardize the values of each accuracy metric (A) and the popularity metric (R) according to:

$$A_{norm} = \frac{A - A_{min}}{A_{max} - A_{min}}, \tag{13}$$

$$R_{norm} = \frac{1/R - 1/R_{max}}{1/R_{min} - 1/R_{max}}. \tag{14}$$

We then compute the trade-off score based on the normalized values of pairs of metrics as follows:

$$t_s = \omega_1 A_{norm} + \omega_2 R_{norm}, \tag{15}$$

where ω_1 and ω_2 denote the weights assigned to the accuracy and popularity aspects, respectively. The choice of specific weight

values depends on the platform's prioritization of these two optimization objectives. In our case, we set $\omega_1 = \omega_2 = 0.5$ and present the corresponding trade-off scores for two accuracy-popularity pairs in the top 10 recommendations in Fig. 5. Notably, a smoothness power of 0.75 consistently yields higher scores, except in the extreme cases of 0 and 1. If smaller powers are employed, the performance of *Tricolore*, as shown in Table II, is expected to improve further.

I. Computational Complexity

Given the importance of computational complexity in candidate generation tasks for MBRS, we compare the time complexity of our model with that of several baselines: the graph-based MBGCN (the strongest baseline), MC-BPR, and the classical DSSM model.

Let N denote the number of samples and M denote the number of layers in the MLP. The computational complexity of *Tricolore* is $O(M \cdot N \cdot d^2)$. In comparison, the time complexity of MBGCN is $O(K \cdot E \cdot N \cdot \frac{d}{B})$, where E is the number of edges and B is the batch size. Our model is more computationally efficient because $\frac{E}{B} \gg d$ and $E \cdot N \cdot d/B \gg N \cdot d^2$. The time complexity of MC-BPR is $O(K \cdot M \cdot N \cdot d^2)$, which is on par with that of *Tricolore*. Compared to DSSM, a popular two-tower model in large-scale recommender systems, *Tricolore* also employs a two-tower structure without cross-interaction training before prediction. The only additional computational cost arises from handling multi-basket behaviors within a multi-task framework. As a result, the computational complexity of *Tricolore* is comparable to that of DSSM.

This analysis demonstrates that while *Tricolore* significantly outperforms existing MBRS models, its computational complexity is equal to or even lower than that of its counterparts.

VI. Discussion

We introduce a novel approach to mining nuanced preferences from ambiguous feedback. Unlike existing methods that impose rigid temporal or strength constraints on behaviors, *Tricolore* utilizes a hierarchical representation with base and fine-grained class embeddings. It employs sets of learnable parameters from the initial encoder to the final prediction, facilitating a thorough exploration of behavior associations, thereby making a unique contribution to MBRS research.

However, *Tricolore* has limitations. Currently, it does not address contextual recommendation scenarios like time-aware recommendations, which could enhance accuracy by considering temporal dynamics of user preferences. Moreover, due to the absence of direct negative user feedback in our datasets, modeling negative behavior types has not been emphasized. Yet, we propose refining *Tricolore* by incorporating subtle cues from weak feedback signals, such as short video watching duration or e-commerce click behavior without subsequent actions. Integrating such strategies into the negative sampling module could improve user representation learning. Furthermore, as a future research direction, we aim to integrate social information into the MBRS algorithms, an area largely unexplored by existing models. Many platforms, including the WeChat Channel analyzed in this study, offer social functionalities that could significantly enhance multi-behavior recommendations. For instance, user likes on short videos are visible to their WeChat friends, yet the platform does not share such recommendation reasons for videos watched beyond a certain duration. This design variation reflects different behavior preferences influenced by individual personality or social dynamics, prompting further exploration of multi-task learning frameworks within *Tricolore*.

VII. Conclusion

We present *Tricolore*, a versatile multi-behavior recommendation framework adept at adaptive learning across diverse behavior types. It hierarchically models comprehensive user preferences during candidate generation, tailored for various recommendation domains. Employing a multi-vector learning approach, it captures distinct behavior characteristics simultaneously. *Tricolore*'s flexible multi-task structure allows customization to specific recommendation needs, augmented by a popularity-balancing technique to mitigate bias. Extensive experiments across public datasets confirm its effectiveness in short videos and e-commerce, particularly excelling in cold-start scenarios. This framework promises to enhance user engagement and recommendation quality significantly.

References

[1] L. Tang, B. Long, B.-C. Chen, and D. Agarwal, "An empirical study on recommendation with multiple types of feedback," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 283–292.

[2] L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo, "Graph meta network for multi-behavior recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 757–766.

[3] C. Huang et al., "Online purchase prediction via multi-scale modeling of behavior dynamics," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2613–2622.

[4] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[5] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, Berlin, Germany: Springer, 2011, pp. 1–35.

[6] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 659–668.

[7] Y. Wu et al., "Multi-view multi-behavior contrastive learning in recommendation," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Springer, 2022, pp. 166–182.

[8] X. Zhou, D. Liu, J. Lian, and X. Xie, "Collaborative metric learning with memory network for multi-relational recommender systems," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4454–4460.

[9] C. Gao et al., "Neural multi-task recommendation from multi-behavior data," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1554–1557.

[10] L. Xia, C. Huang, Y. Xu, P. Dai, B. Zhang, and L. Bo, "Multiplex behavioral relation learning for recommendation via memory augmented transformer network," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 2397–2406.

[11] M. Wan and J. McAuley, "Item recommendation on monotonic behavior chains," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 86–94.

[12] W. Zhang, J. Mao, Y. Cao, and C. Xu, "Multiplex graph neural networks for multi-behavior recommendation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 2313–2316.

[13] L. Xia, C. Huang, Y. Xu, P. Dai, M. Lu, and L. Bo, "Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling," in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 1931–1936.

[14] M. Yan et al., "Cascading residual graph convolutional network for multi-behavior recommendation," 2022, *arXiv:2205.13128*.

[15] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 361–364.
[16] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 273–282.
[17] J. Ding, F. Feng, X. He, G. Yu, Y. Li, and D. Jin, "An improved sampler for Bayesian personalized ranking by leveraging view data," in *Proc. Companion Web Conf.*, 2018, pp. 13–14.
[18] J. Ding et al., "Improving implicit recommender systems with view data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3343–3349.
[19] J. Liu, C. Shi, B. Hu, S. Liu, and S. Y. Philip, "Personalized ranking recommendation via integrating multiple feedbacks," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, Springer, 2017, pp. 131–143.
[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[21] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen, "Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1734–1743.
[22] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 727–735.
[23] M. Yan et al., "Cascading residual graph convolutional network for multi-behavior recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 1, pp. 1–26, 2023.
[24] Z. Cheng, S. Han, F. Liu, L. Zhu, Z. Gao, and Y. Peng, "Multi-behavior recommendation with cascading graph convolution networks," in *Proc. ACM Web Conf.*, 2023, pp. 1181–1189.
[25] J. Xu et al., "Multi-behavior self-supervised learning for recommendation," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 496–505.
[26] C. Chen et al., "Graph heterogeneous multi-relational recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3958–3966.
[27] C. Wu et al., "FeedRec: News feed recommendation with various user feedbacks," in *Proc. ACM Web Conf.*, 2022, pp. 2088–2097.
[28] Y. Zhang et al., "Causal intervention for leveraging popularity bias in recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 11–20.
[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[30] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 263–272.
[31] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, AUAI Press, 2009, pp. 452–461.
[32] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2333–2338.
[33] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[34] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1930–1939.
[35] X. Ma et al., "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 1137–1140.
[36] H. Tang, J. Liu, M. Zhao, and X. Gong, "Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations," in *Proc. 14th ACM Conf. Recommender Syst.*, 2020, pp. 269–278.
[37] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing popularity bias in recommender systems with personalized re-ranking," in *Proc. 32nd Int. Flairs Conf.*, 2019, pp. 413–418.
[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Xiao Zhou received the PhD degree in computer science from the University of Cambridge, U.K. She joined the Gaoling School of Artificial Intelligence with Renmin University of China as a tenure-track assistant professor, in 2021. Prior to this, she served as a postdoctoral fellow with MIT. Her research interests include data mining, responsible recommender systems, urban computing, social computing, and large language models. She is particularly passionate about applying AI to social sciences.

Zhongxiang Zhao received the master's degree in computer science from the Harbin Institute of Technology, in 2016, and has since gained extensive experience at leading technology companies. He is currently a senior algorithm engineer with the WeChat Business Group, Tencent Corporation. His expertise focuses on developing recommendation systems for real-world applications, including image-text, short-video platforms, and e-commerce platforms.

Hanze Guo received the bachelor's degree in computer science from the University of Electronic Science and Technology of China, in 2022. Currently, he is working toward the PhD degree with the Gaoling School of Artificial Intelligence, Renmin University of China, under the supervision of Dr. Xiao Zhou. His research interests include deep learning, graph neural networks, and recommendation systems.