An E-Commerce Dataset Revealing Variations during Sales

Jianfu Zhang
c.sis@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Qingtao Yu
qingtao.yu@shopee.com
Shopee Pte. Ltd.
Singapore

Yizhou Chen
yizhou.chen@shopee.com
Shopee Pte. Ltd.
Singapore

Guoliang Zhou
guoliang.zhou@shopee.com
Shopee Pte. Ltd.
Singapore

Yawen Liu
yawen.liu@shopee.com
Shopee Pte. Ltd.
Singapore

Yawei Sun
yawei.sun@shopee.com
Shopee Pte. Ltd.
Singapore

Chen Liang
chen.liang@shopee.com
Shopee Pte. Ltd.
Singapore

Guangda Huzhang
guangda.huzhang@shopee.com
Shopee Pte. Ltd.
Singapore

Yabo Ni
yabo001@e.ntu.edu.sg
Nanyang Technological University
Singapore

Anxiang Zeng
zeng0118@ntu.edu.sg
Nanyang Technological University
Singapore

Han Yu*
han.yu@ntu.edu.sg
Nanyang Technological University
Singapore

ABSTRACT

Since the development of artificial intelligence technology, E-Commerce has gradually become one of the world's largest commercial markets. Within this domain, sales events, which are based on sociological mechanisms, play a significant role. E-Commerce platforms frequently offer sales and promotions to encourage users to purchase items, leading to significant changes in live environments. Learning-To-Rank (LTR) is a crucial component of E-Commerce search and recommendations, and substantial efforts have been devoted to this area. However, existing methods often assume an independent and identically distributed data setting, which does not account for the evolving distribution of online systems beyond online finetuning strategies. This limitation can lead to inaccurate predictions of user behaviors during sales events, resulting in significant loss of revenue. In addition, models must readjust themselves once sales have concluded in order to eliminate any effects caused by the sales events, leading to further regret. To address these limitations, we introduce a long-term E-Commerce search data set specifically designed to incubate LTR algorithms during such sales events, with the objective of advancing the capabilities of E-Commerce search engines. Our investigation focuses on typical industry practices and aims to identify potential solutions to address these challenges.

*Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA.

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657870

CCS CONCEPTS

• **Computing methodologies** → **Ranking**; *Neural networks*; • **Applied computing** → **Online shopping**.

KEYWORDS

E-Commerce, Learning-To-Rank, Sales Events

ACM Reference Format:
Jianfu Zhang, Qingtao Yu, Yizhou Chen, Guoliang Zhou, Yawen Liu, Yawei Sun, Chen Liang, Guangda Huzhang, Yabo Ni, Anxiang Zeng, and Han Yu. 2024. An E-Commerce Dataset Revealing Variations during Sales. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657870

1 INTRODUCTION

With the advent and evolution of artificial intelligence technology, E-Commerce has undergone exponential growth to become one of the world's largest commercial markets. Sales events (*i.e.*, promotional events) in the E-Commerce industry, driven by sociological mechanisms, play a crucial role in the success of online platforms. Sales events such as Black Friday, New Year's Day, and Double 11 Sales entice users to make purchases. For example, Alibaba reported that its Chinese market Gross Merchandise Volume (GMV) in 2021 (the most recent available report) exceeded 1,100 billion dollars, and Double 11 alone contributed 84.54 billion dollars, which is approximately 27.74 times the average daily GMV, and accounts for more than 7.6% of the one-year GMV, highlighting the importance of sales events. Alibaba's disclosure of its Double 11 GMV, showcasing an annual growth rate of approximately 30% until 2020, has been instrumental in promoting its business success. Additionally, smaller-scale sales events tied to national holidays and cultural

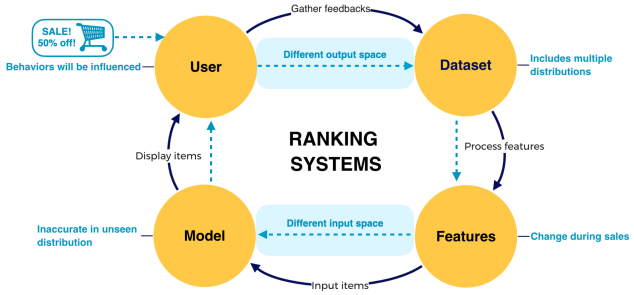

Figure 1: The illustration of a standard ranking system of an E-Commerce platform iteration cycle (black arrows): the cycle from user interactions to model predictions and back, including users providing feedback that shapes the dataset, which in turn influences the features processed by the model to display items. We also demonstrate the effects on predictive models when sales *spikes* occur (blue content): significant changes in the dataset output distribution caused by user behaviors, alter the feature input space, and impact the accuracy of the models in unseen data distribution.

observances in different countries also significantly impact user engagement.

Our study prioritizes evaluating Learning-to-Rank (LTR) models due to their crucial role in aligning user intent with item selection, significantly influencing GMV. Traditional LTR models and datasets (*e.g.* [14, 22]) typically operate under the assumption of Independent and Identically Distributed (IID) data, which is the foundation of model training. In contrast, sales events have a noticeable impact on user intent, leading to **spikes**, *i.e.*, significant changes in the data distribution under various metrics, which indicate distinctive challenges for predictive models based on IID assumption. Demonstrated by Figure 1, such events precipitate the variable nature of user behaviors [1] and the dynamic distributions during these periods, rendering a time-varying dataset with multiple distributions. When processing the dataset features, it is crucial to recognize that the distribution of these features may fluctuate across different periods of sales events. Conventional methods routinely overlook such intricate patterns, thus this inherent variability can introduce inaccuracies in LTR models tethered to the IID assumption, causing erroneously presented items that do not align well with users' preferences. As demonstrated in Figure 2, our comparative analysis underscores that LTR model performance can depreciate by approximately 10%, a deficit that could potentially lead to substantial GMV losses amounting to billions.

In this paper, we offer a first-of-its-kind E-Commerce dataset that explicitly overthrows the IID assumption. Our dataset encompasses data from sales events in the E-Commerce search scenario, a key component of online platforms, whose impact on GMV is particularly noteworthy. To facilitate our investigation, we curate a dataset specifically encompassing highly active users, items, and queries collected over an extended period, including multiple sales events, thereby constructing a time-varying dataset. Our main investigation is to examine the performance of LTR models across various days using this time-varying dataset. We emphasize the substantial variance observed in user behavior patterns across different days, especially during sales events. Traditional methodologies often fall short in adapting to such significant shifts in user behavior distribution. The cornerstone of our experimental analysis is to delineate the unique features of this dataset. Our approach is intentionally broad to extract more transparent insights into the dataset characteristics. Moreover, for methodologies that outperform on our dataset, live testing is employed to ascertain their effectiveness in real-world scenarios, underscoring the dataset's practical utility and impact in actual implementations.

Our contributions can be summarized as follows:

- **A time-varying dataset with spikes:** We introduce a time-varying dataset with explicit spikes from a large E-Commerce search engine, which is made publicly available for research purposes. This dataset is a critical resource for exploring time-variant phenomena, especially those associated with spikes, offering a unique perspective for investigation on the dynamics of E-Commerce activities.
- **Demonstration of benchmarks:** We conduct comprehensive evaluations of several benchmark methods and training strategies on our dataset, revealing their limitations in adapting to the evolving distribution of user purchasing data , particularly during sales events. We conduct live tests to verify the effectiveness of techniques for enhancements derived from our curated dataset, determining their utility in real-world live scenarios and, consequently, assuring the practical quality of our dataset. This effort highlights the dataset's practical relevance and its potential impact on advancing E-Commerce research and applications, providing valuable insights into its effectiveness.

Table 1: Datasets statistics and spike features. In contrast to other datasets, our dataset exhibits augmented daily density (barring Taobao UVA), which encompasses a truncated time frame insufficient for comprehending spike impacts. The "Implicty" column denotes whether user feedback is implicit (indicating behavioral presence or absence only), or explicit (encompassing scores or more concrete feedback by users).

	#Interactions	Time Range	E-Commerce	Implicty	Spikes
Amazon	230 millions	30 years	Yes	No	None
CIKMCUP	1 million	5 months	Yes	Yes	None
LETOR	< 1 million	Unknown	No	No	None
MovieLens 100K	0.1 millions	7 months	No	No	None
MovieLens 25M	25 millions	24 years	No	No	None
Taobao UVA	100 millions	9 days	Yes	Yes	None
Yelp	7 millions	8 years	Yes	Both	None
Ours	16 millions	80 days	Yes	Yes	Labeled

2 RELATED WORKS

Datasets for Evaluating LTR Models: Various public datasets exist for evaluating Learning-To-Rank models (*e.g.* [1, 3, 5, 9, 17, 19, 31]). Early works use small datasets with dense features [15, 16], or artificial data [2, 4, 19]. Recently there are many data resources contributed by the industry. For example, the Amazon dataset [22], which focuses on recommender systems in an E-Commerce context,

[1] For instance, a user u who clicks on an item i for many days without purchasing may eventually make a purchase during the next sales event.

Table 2: Statics of the proposed dataset.

Intervals	#Days	#Users	#Items	#Records	#Clicks	#Purchases	C/U	P/U
Total	80	31154 ± 3105	96201 ± 7231	202855 ± 29539	98839 ± 14522	4157 ± 1371	3.16 ± 0.18	0.131 ± 0.030
Regular	74	30769 ± 2816	95563 ± 7090	199710 ± 27844	97300 ± 13679	3858 ± 785	3.15 ± 0.18	0.125 ± 0.016
Sales	6	35895 ± 2539	104062 ± 3234	241643 ± 21118	117814 ± 10720	7846 ± 1667	3.28 ± 0.18	0.217 ± 0.030
Before Sales	12	30296 ± 1897	93653 ± 4262	192061 ± 16095	94557 ± 7863	3182 ± 318	3.12 ± 0.13	0.105 ± 0.009
After Sales	15	32268 ± 2544	99319 ± 6916	214744 ± 30625	104426 ± 14713	4375 ± 969	3.22 ± 0.21	0.134 ± 0.019

provides explicit review scores as feedback. The popular Movielens dataset [14], which is commonly used in recommender systems research, incorporates explicit user feedback in the form of ratings. Microsoft provides the LETOR dataset for LTR research [23]. Yahoo [7] held an LTR challenge and attracted many researchers. Taobao UVA [33] captures implicit feedback by observing user-item interactions without explicit scores. Yelp offers a diverse set of scenarios for examining models across various domains. Besides recommendation tasks, CIKMCUP dataset [27] has been specifically designed for the search domain. However, none of these datasets encompasses real-world out-of-distribution sales spikes commonly encountered in practice, which is the focus of our work. The mentioned datasets are summarized in Table 1.

Other Learning Paradigms: Our dataset can support the exploration of different learning paradigms including: 1) continual learning [10] which allows models to adapt to the latest distribution while preserving previous knowledge; 2) few-shot learning [11, 30] which enables models to learn the latest distribution swiftly; 3) sequence representation learning [33, 34] which enhances classic LTR models by behavior sequences of users and incorporating them into the ranking process; and 4) multi-task learning [6] can be examined as our dataset contains different types of user behaviors and we can also regard the multiple days as multiple tasks. We examine MMoE [20] in our work. Other common MTL options, including cross-stitch networks [21], sluice networks [25], and PLE [26], have similar properties and performances.

3 DATASET OVERVIEW

This section provides an overview of the dataset and the learning task. We begin by presenting the statistics of our time-varying dataset, highlighting the impact of *out-of-distributional sales spikes* (referred to as *spikes* hereafter for simplicity) on the data. Subsequently, we conduct a detailed analysis of the dataset and formally define the research objective the proposed dataset aims to support.

3.1 Preliminaries

In an E-Commerce search system, the i-th data sample is structured as $(x_i, u_i, q_i, d_i, y_i)$, where x_i represents the item, u_i denotes the user, q_i signifies the query[2], y_i represents the behavior, and d_i denotes the date. To represent items, users and queries, we employ the vanilla full embedding paradigm. Specifically, for each x_i, u_i, q_i, an integer ID is provided, and each ID corresponds to a unique trainable vector (*i.e.*, embedding) that serves as its vector representation. In a search service system, the core scoring model f takes samples as inputs and produces the corresponding score $\hat{y}$ for each sample, as follows:

$$f : x, u, q, d \rightarrow \hat{y}. \tag{1}$$

The final personalized suggestion of items is determined based on these scores. The objective is to assign higher scores to items that are more relevant to the matched users. To assess the accuracy of the models, various metrics, including AUC, NDCG, and MAP, are commonly adopted. These metrics exhibit high correlations in practice. Therefore, we adopt AUC as the evaluation metric.

3.2 Dataset Statistics

Our dataset was collected from a deployed E-Commerce platform over 80 days. It includes 109, 940 users, 217, 468 items and 19, 713 queries with desensitization. To ensure the dataset primarily focuses on meaningful interactions, we eliminated long-tail interactions, including both user-item interactions and queries. The user-item interactions in our dataset are divided into three categories: 1) **exposures** (no click or purchase), 2) **clicks** (but no purchase), and 3) **purchases**. Given the large number of exposures, we performed sub-sampling to balance the dataset, optimize storage, and maintain utility. Specifically, we sub-sampled the exposures to bring their quantity closer to that of the clicks, as exposure samples without clicks are less informative.

We analyze the dataset characteristics for different time intervals associated with sales events. As we have sub-sampled the exposures in our dataset, we focus our analysis on the statistics of clicks per user (C/U) and purchases per user (P/U), instead of the click-through rate (CTR) and the conversion rate (CR). This adjustment enables us to effectively capture the relative engagement and purchasing behaviors among users. The detailed statistics of the proposed dataset can be found in Table 2. We characterize the dataset distributions based on five types of time intervals, each is being associated with the sales events in a different manner:

(1) **Total**: provides the overall statistics.
(2) **Regular**: excludes the sales dates, exhibiting a lower P/U, while the C/U remains stable.
(3) **Sales**: represents the sales dates, characterized by spikes in P/U but no significant change in C/U.
(4) **Before Sales**: including data from the two days prior to the sales event. The P/U shows a downward trend, while the C/U shows an upward trend.
(5) **After Sales**: including data from three days following the sales event. Both C/U and P/U decrease.

Impact of Spikes: The sales strategy has a substantial impact on user behavior, resulting in distinct patterns on sales dates. To analyze this phenomenon, we present two statistics in Figure 2. From the sales dates, it is evident that the trend in C/U remains

[2]To facilitate the exploration and application of LTR algorithms, we simplify queries into query IDs though an industrial-level neural language understanding model.

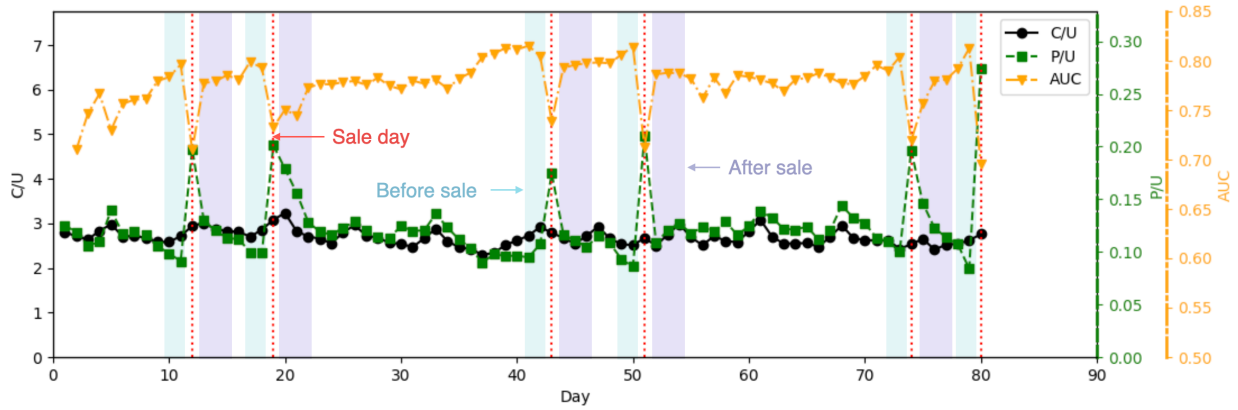

Figure 2: Visualization of different periods, C/U, P/U and AUC of our tested MLP model. The black line illustrates C/U, while the green line represents P/U. We also visualize three distinct periods: Sales (indicated by the red dotted line), Before Sales (indicated by the blue interval), and After Sales (indicated by the purple interval).

stable, whereas P/U exhibit spikes. These effects can be explained as follows. The daily user tendencies differ between **regular** days and **sales** days, resulting in different distributions. Specifically, sales days have a higher likelihood of selling items, leading to a higher P/U. Furthermore, user behaviors during the periods before and after the sales event also differ. **Before the sales event**, users may engage in more browsing and wait to make purchases on the sales, resulting in an increase in C/U. **After the sales event**, user activity typically decreases, leading to a decline in both C/U and P/U.

We further investigate the influence of spikes on model performance (Figure 2). We train a naive MLP (refer to Section 4.1) and evaluate it in terms of AUC. Note that the AUC values exhibit variations across the sales dates, with a notable decline observed. However, comparing AUC values across different days is not straightforward due to the presence of varying data distributions.

3.3 Dataset Settings

The main open research objective is to enhance the performance of LTR models, particularly on sales dates, while considering the time-varying characteristics of the dataset. We aim to establish a stable training paradigm that can be consistently applied across all types of days, enabling seamless and effective adaptation to the evolving nature of online systems. To effectively evaluate the performance of models on the time-varying dataset, it is crucial to utilize a suitable evaluation metric that can measure the overall loss for the next date in a sequence. To this end, the dataset is initially divided by dates. For a given test date d, all preceding data up to date $d-1$ are made available to the models. The test dataset on date d, denoted as D for convenience, is further partitioned into K batches by their occur order, represented as $D^1, D^2, \ldots, D^K$, with each batch corresponding to a set of samples. Prior to processing D^k, the previous batch D^{k-1} is used to finetune the models. Once all K batches have been processed, we compute the total loss as:

$$\sum_{k=1}^{K} L(D^k, f_{\theta_k}), \tag{2}$$

where f represents the LTR scoring model. The function L can be any loss function used for ranking. θ_k represents the model parameters after processing the $(k-1)$-th batch.

The above setting is referred to as the *online setting*, which is similar to online learning: the model f is capable of being updated after the data is revealed. An example of such an update is the application of a simple one-pass back-propagation update as:

$$\theta_k \leftarrow \theta_{k-1} + \alpha \partial L(D^{k-1}, f_{\theta_{k-1}})/\partial \theta_{k-1}, \tag{3}$$

where α is the learning rate.

In addition to the online setting, we introduce another setting, referred to as the *offline setting*. In the offline setting, models are not finetuned on new data collected on the test date. Consequently, θ_k remains fixed, representing the model obtained using the data preceding the test date. The distinctions between these two settings are illustrated in Figure 3.

The offline and online settings differ in their approaches to evaluating the performance of LTR algorithms. Under the offline setting, only data from the previous dates is utilized to predict the behavior of the next day. On the other hand, the online setting provides a more precise evaluation method where algorithm performance is assessed on a continuous stream of data, allowing for ongoing adaptation to the changing distribution of online systems.

4 EXPERIMENTAL STUDIES

The dataset consists of 80 files named from Day1 to Day80. Each file contains several lines, which are sorted by timestamp by default. Each line represents an interaction between a user and an item, given a query and a timestamp. The record format includes five integers in the following order: x_i, u_i, d_i, q_i, y_i. Details can be found in Appendix A.1. The primary objective of our experimental analysis is to investigate the distinct characteristics of the proposed dataset. It's pertinent to emphasize that our methodology adopts a macroscopic approach in this exploration. We deliberately steer clear of using overly intricate methods, aiming to gain clearer insights into the dataset's attributes. Additionally, for techniques that

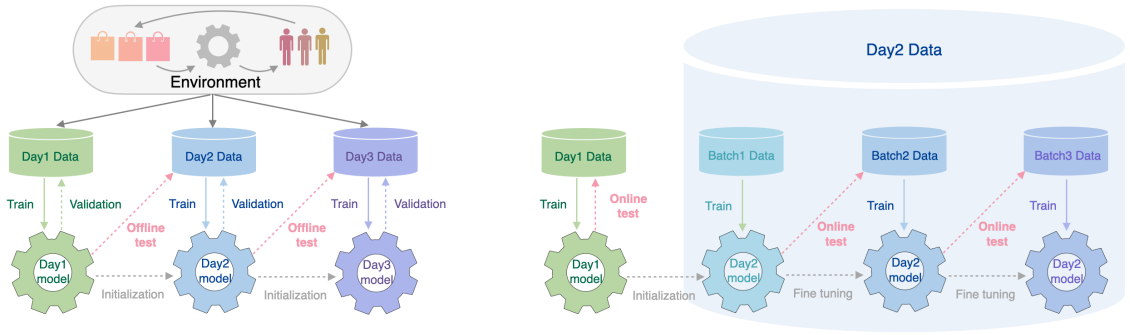

Figure 3: Difference between Offline and Online Test Settings. On the left: *Offline setting*, which uses previous data to predict the behavior of the next day, relies on historical data to make predictions. On the right: *Online setting* provides a more dynamic and ongoing evaluation as the model is updated with new data, which can be regarded as a more precise representation of real-time data. In the online test setting, metrics are accumulated as the model continues to train.

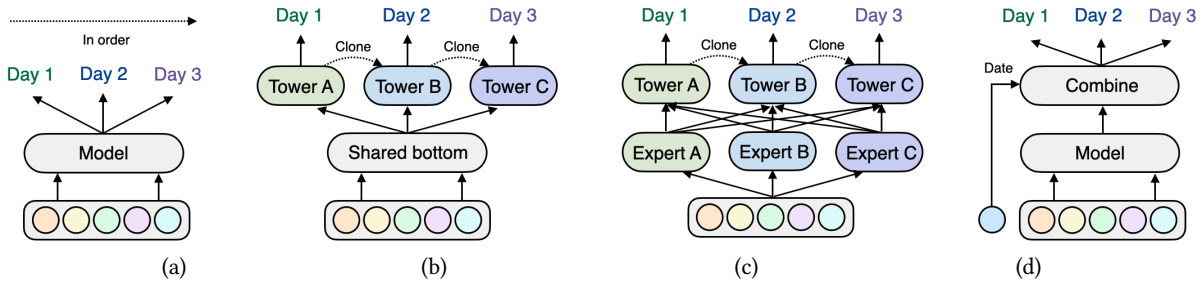

Figure 4: Illustration of standard LTR paradigms in our dataset, regarding each day as a single task. From left to right: (a) MLP. (b) Shared bottom, which is equivalent to an incrementally trained MLP. (c) MMoE. (d) WDL. The added date feature can absorb bias brought by different days.

demonstrate distinguishable performance on our dataset, we conduct live tests to verify their real-world efficacy. This step is critical to demonstrate the practical relevance and impact of our dataset in actual deployments.

We organize our proposed experiments around three key research questions: **RQ1**: Can the standard LTR paradigms overcome the challenges posed by the time-varying dataset? **RQ2**: Can more advanced approaches mitigate the challenges posed by the time-varying dataset? **RQ3**: Are there any potentially effective directions for tackling the challenges posed by time-varying data with spikes?

In the subsequent chapters, we begin by introducing the testing metrics, outlining the evaluation criteria for different methods on our dataset. This includes both the efficacy of the methods on our proposed dataset and their performance in real-world scenarios verified through live testing. Following this, we will validate baseline methodologies and then proceed to assess advanced methods. Finally, we explore different training strategies to address the identified challenges. More analysis can be found in Appendix A.4.

4.1 Live Test Setting

To demonstrate the practical utility of our dataset and its potential to aid research in LTR during sales events, we conduct several live experiments based on our industrial-level LTR model. Our experiments involve initially testing various techniques on our dataset for their AUC performance. For detailed information on the AUC metric, please refer to Appendix A.3. Methods demonstrating notably higher AUC are selected for further evaluation of their P/U performance in a live setting. These experiments aim to underscore the relevance and insights our dataset offers for practical online applications. We prioritize P/U as the main business metric, for its high relative to GMV and better stability than GMV (*i.e.*, GMV has higher variance in live testings). Given the scale of large E-Commerce platforms, even a 1% improvement in P/U is deemed substantial, as it can translate into millions in GMV on a daily basis.

4.2 Baselines Results and Analysis

In our study, we investigate the performance of multiple industry-standard baselines on our proposed dataset. These baselines include:

- Multi-Layer Perceptron (MLP): A vanilla MLP as shown in Figure 4(a). Note that incrementally trained MLP can be regarded as a shared bottom MTL when data is coming in the time order, as Figure 4(b) shows.
- Multi-gate Mixture-of-Experts (MMoE) [20]: a model that combines multiple implicit distributions from expert models, as Figure 4(c) shows. It divides the distribution of a day into several implicit distributions and effectively learns to ensemble them. Intuitively, the distribution of a new date can be more quickly adapted by ensembling some well-learned implicit distributions.
- Wide and Deep Learning (WDL) [8]: which incorporates sparse features in addition to the logit of MLP. The difference between our MLP and WDL lies in the inclusion of the date feature, as

Table 3: AUC achieved by the baselines on our proposed dataset for the last 7 days.

Settings	Labels	Baselines	Day 1 (Sales)	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7 (Sales)
Offline	Naive	MLP	71.89 ± 0.05	75.67 ± 0.04	77.90 ± 0.03	78.13 ± 0.06	79.11 ± 0.06	81.34 ± 0.05	69.22 ± 0.05
		MMoE	72.00 ± 0.04	75.62 ± 0.05	77.95 ± 0.07	78.04 ± 0.08	79.16 ± 0.07	79.16 ± 0.09	69.54 ± 0.03
		WDL	72.05 ± 0.06	75.78 ± 0.07	78.03 ± 0.05	78.18 ± 0.05	79.33 ± 0.06	81.61 ± 0.04	69.75 ± 0.07
	Cotrain	MLP	72.39 ± 0.07	76.19 ± 0.08	78.63 ± 0.05	78.35 ± 0.08	79.67 ± 0.08	82.13 ± 0.02	69.77 ± 0.15
		MMoE	72.61 ± 0.06	76.23 ± 0.04	78.64 ± 0.08	78.39 ± 0.06	79.75 ± 0.06	82.14 ± 0.04	70.12 ± 0.12
		WDL	72.61 ± 0.05	76.32 ± 0.03	78.74 ± 0.04	78.58 ± 0.06	79.94 ± 0.06	82.47 ± 0.03	70.34 ± 0.12
	Multiclass	MLP	73.80 ± 0.08	77.49 ± 0.06	79.68 ± 0.03	79.56 ± 0.03	80.59 ± 0.08	83.28 ± 0.07	70.83 ± 0.10
		MMoE	73.83 ± 0.11	77.61 ± 0.16	79.64 ± 0.11	79.56 ± 0.10	80.77 ± 0.06	83.53 ± 0.06	70.96 ± 0.12
		WDL	73.94 ± 0.09	77.59 ± 0.12	79.80 ± 0.05	79.63 ± 0.09	80.79 ± 0.09	83.36 ± 0.10	71.04 ± 0.14
Online	Naive	MLP	72.51 ± 0.04	75.78 ± 0.05	78.13 ± 0.06	78.45 ± 0.07	79.37 ± 0.03	81.78 ± 0.04	71.15 ± 0.09
		MMoE	72.61 ± 0.07	75.77 ± 0.06	78.17 ± 0.09	78.33 ± 0.06	79.43 ± 0.07	81.72 ± 0.05	71.07 ± 0.06
		WDL	72.58 ± 0.05	75.83 ± 0.07	78.17 ± 0.09	78.46 ± 0.08	79.39 ± 0.08	81.78 ± 0.05	71.28 ± 0.06
	Cotrain	MLP	73.22 ± 0.05	76.48 ± 0.08	78.80 ± 0.08	78.80 ± 0.08	79.89 ± 0.06	82.46 ± 0.09	72.11 ± 0.07
		MMoE	73.40 ± 0.06	76.57 ± 0.08	78.94 ± 0.11	78.80 ± 0.04	80.03 ± 0.06	82.50 ± 0.11	71.91 ± 0.09
		WDL	73.43 ± 0.09	76.63 ± 0.05	79.00 ± 0.05	78.86 ± 0.12	80.10 ± 0.03	82.59 ± 0.11	72.10 ± 0.09
	Multiclass	MLP	75.24 ± 0.05	78.15 ± 0.06	80.26 ± 0.06	80.23 ± 0.07	81.42 ± 0.08	83.72 ± 0.08	73.79 ± 0.06
		MMoE	75.40 ± 0.09	78.28 ± 0.07	80.27 ± 0.11	80.19 ± 0.12	81.39 ± 0.08	83.99 ± 0.11	73.81 ± 0.08
		WDL	75.24 ± 0.09	78.22 ± 0.05	80.22 ± 0.11	80.21 ± 0.11	81.40 ± 0.04	83.78 ± 0.07	73.79 ± 0.08

Figure 4(d) shows, aiming to address potential distribution variations across different dates and mitigate date bias. The wide part of WDL is designed to absorb the bias brought by contexts, such as the positional debiasing [13].

More details about implementations of baselines can be found in Appendix A.2. In our baselines, the input features are minimalistic, consisting of concatenated embeddings of item, user, and query. Some baselines additionally incorporate date embeddings.

We also explore three distinct multi-task learning settings to leverage the diverse behaviors represented by different labels. Firstly, we consider the **naive** setting, which involves direct binary classification of purchase prediction without considering the distinction between click and non-click behaviors. Then, by examining the impact of including additional side objective (click prediction) on the primary objective of purchase prediction, we can assess the effectiveness of different multi-task learning approaches. Thus, we introduce the **cotrain** setting. In this setting, we perform concurrent binary classification of clicks and purchases, employing two shared-bottom towers to predict both click and purchase behaviors. Furthermore, we explore the **multiclass** setting, where purchases, clicks, and exposures are treated as a three-class classification problem. By sharing the model parameters across the three labels, we can maximize parameter efficiency and capture the distinctive patterns and characteristics associated with each behavior.

Some insightful observations in Table 3 are discussed in the following paragraphs. In this table, Day 1-7 is the last 7 days (Days 74-80) of the dataset. Day 1 and Day 7 are Sales and Day 2-6 are Regular. They explain why some methods work better than others in our dataset. Additionally, we extend our observations with live test and show how they help our online system, which implies research in this dataset may benefit human society.

4.2.1 Learn from Multiple Behaviors. Upon analyzing the methods incorporating clicks data, namely the "cotrain" and "multiclass" rows in Table 3, it is evident that they outperform the approach in the "Naive" row. This observation suggests that incorporating click prediction objectives can enhance the learning of the purchase prediction task, as relying solely on the density of purchase behaviors might not effectively capture the latest distribution. Additionally, the "multiclass" method consistently outperforms "cotrain", which could be attributed to a higher utilization of parameters given both methods a similar number of parameters. This finding sheds light on **RQ3**, emphasizing that exploring learning from multiple behaviors is an important direction for training adaptive models.

Live test: Our live experiments unequivocally confirm the effectiveness of integrating click and add-to-cart targets during training. Moreover, finely tuning the weights of various target tasks can notably boost performance. In pursuit of this, we adjusted the weights of different label types during training, resulting in an enhanced model performance by appropriately increasing the weights of click and add-to-cart losses. Our online experiments demonstrated a significant +2.86% improvement in P/U with weights fine-tuned through cross validation.

4.2.2 Debiasing the date influence. In the offline setting, WDL showcases the ability to magnify the performance gap on sales dates, indicating its potential in mitigating the challenges caused by spikes. Notably, we observe a substantial improvement on sales dates, specifically Day 1 and Day 7, compared to regular dates. This suggests that debiasing features are particularly effective during sales dates. These findings partially address **RQ1** and **RQ3**, implying that debiasing approaches offer a promising solution for enhancing performance on sales dates within the offline setting.

Live test: When the debiasing feature is added, there is a noted improvement of +0.94% in P/U performance in the live environment. We also observe concatenating debiasing feature into the dense inputs (*i.e.*, not present as a bias in logit) will contrarily harm the performance, which indicates the benefit is from debiasing instead of information increase via inputting date feature.

4.2.3 Real-time updates. The models under the online setting consistently outperform the models under the offline setting, demonstrating significant performance gaps. This finding partially addresses **RQ3**, indicating that models capable of adapting more quickly to real-time distributions achieve higher accuracy. It emphasizes the importance of our second research objective. However, it is worth noting that in the online setting, the performance of all online models remains similar across different days, and neither MMoE nor WDL clearly outperforms MLP. This finding partially highlights **RQ1**, suggesting that a more sophisticated model design is required to better utilize the advantage of online updates.

Live test: In our system, shifting from daily to hourly updates led to significant improvements in live test. Specifically, P/U saw an increase of +1.31% on a sales day, with an overall daily P/U increase of +0.64%. These results highlight the importance of closely aligning with real-time data distributions. The transition to online updates, particularly during sales days, demonstrates more pronounced benefits than daily updates.

Table 4: AUC results achieved by advanced methodes on the proposed dataset for the last 7 days.

Baselines	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
MLP	0.7375	0.7752	0.7967	0.7958	0.8049	0.8337	0.7070
MMoE	0.7378	0.7751	0.7970	0.7968	0.8070	0.8348	0.7096
WDL	0.7409	0.7766	0.7973	0.7964	0.8078	0.8326	0.7110
DeepFM	0.7393	0.7754	0.7985	0.7962	0.8084	0.8326	0.7070
PNN	0.7382	0.7767	0.8010	0.7990	0.8084	0.8334	0.7098
DCN	0.7378	0.7753	0.7979	0.7948	0.8089	0.8333	0.7111
MAML	0.7284	0.7370	0.7510	0.7460	0.7541	0.7508	0.7290
DIN	0.7365	0.7788	0.8012	0.7968	0.8075	0.8337	0.7115
DIEN	0.7398	0.7795	0.8029	0.7992	0.8112	0.8353	0.7136

4.3 Analysis of Advanced Methods

The aforementioned results are derived from fundamental models. To more accurately assess the impact of sales events on sophisticated methodologies, we selected several state-of-the-art feature interaction models and advanced sequential modeling methods. For these advanced methods, specifically:

- Feature Interaction Models: We chose models from BarsCTR benchmark [35], where DeepFM [12], PNN [24], and DCN [29] are chosen for their superior performance reported in BarsCTR.
- Model-Agnostic Meta-Learning (MAML) [11] is a well-known few-shot learning strategy that can be utilized to adapt to the latest distribution by leveraging a few most recent interactions. Similar meta-learning strategies have been applied in industrial applications [18, 28].
- Deep Interest Network (DIN) [34] and Deep Interest Evolution Network (DIEN) [33] are benchmark sequence representation learning methods commonly employed in both LTR research and practical applications, allowing for capturing the latest distribution with greater ease.

Results of the last 7 days are in Table 4. Our analysis yields three findings: 1) A significant performance drop is observed in all selected feature interaction models on sale days (Day 1 and Day 7). It partially addresses **RQ1**: the classic LTR methods may not be entirely effective for sales without specific modifications. 2) MAML achieves the best performance on Day 7, while performing poorly on other days. This finding suggests that MAML exhibits some level of robustness to spikes, considering that the distribution on Day 7 is significantly different from the other days, as shown in Figure 2. However, this robustness comes at the expense of lower performance (underperforms other baselines) on regular days. 3) All advanced methods demonstrate only slightly better performance. This observation suggests that even models incorporating sequence information struggle to effectively adapt to the distribution shift that occurs on sales dates. Both facts partially answer **RQ2**: despite adding the ability to model sequential patterns, the tested models still face challenges in prediction accuracy during sales events.

4.4 Training Strategies

We first conduct an experiment where we stop the training of the model for the last 7 days and compare it to the model that continues training for the entire duration. The results are in Figure 5. Furthermore, we also conduct a similar comparison experiment using MLP, but stop training for the last 28 days, which will be discussed in Appendix A.4.1.

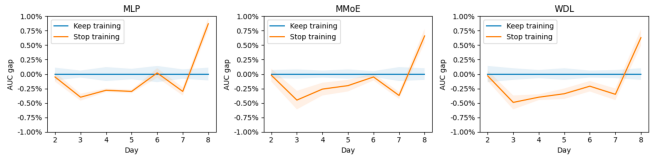

Figure 5: Comparison of performance between models with and without training for the last 7 days.

We observe that early stopping leads to a decline in regular performance. However, for the subsequent sale day, these models outperform the models that continue training. This observation indicates that while the distribution of regular days changes slowly and benefits incremental training, it can change rapidly during sales events, making incremental training less effective. This finding emphasizes the importance of utilizing samples from different days in a specially designed manner to mitigate the adverse effects of spikes. It provides valuable insights into addressing **RQ3** and optimizing incremental training in the presence of spikes.

Practically, a related strategy commonly employed in industrial E-Commerce models is to exclude data from sales dates during training, using it solely for testing purposes to maintain data consistency. We examine the effectiveness of this strategy using our dataset, as depicted in Figure 6.

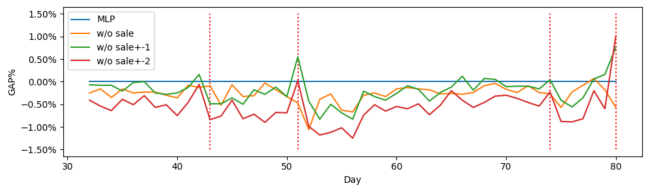

Figure 6: Performances of different data removal strategies among the last 49 days.

We observe that simply removing the data from sales dates does not lead to improvements in performance on both sales and regular

dates. However, when we extend the removal range to include the data from one day before or after the sale (*i.e.*, w/o sale+-1), the performance appears to improve on most sales dates, while the regular day performance remains similar to the scenario without sales dates. On the other hand, removing the data from two days before or after the sales event (*i.e.*, w/o sale+-2) shows inferior performance. This finding validates one of our proposed investigation direction for **RQ3**.

Live test: Inspired by the aforementioned findings, we experimented with omitting training data around sales dates. Identifying an optimal interval for skipping data proved crucial. Our best results were achieved by skipping data from 6 hours before the start of sales. This strategy is predicated on the notion that omitting excessive data can render model parameters outdated, while minimal omission might impair performance, as previous experiments have shown. Implementing this skip training approach yielded a +0.70% improvement in P/U.

5 CONCLUSIONS

In this paper, we present a novel problem in the field of LTR research, along with a benchmark dataset specifically designed to address this problem. The dataset exhibits several significant properties related to time-varying dynamics and spikes. By introducing this dataset and conducting our experiments, we aim to shed light on the challenges associated with adapting LTR algorithms to the evolving distribution of live environments, particularly during events such as sales in E-Commerce. The proposed dataset and the accompanying analysis can serve to inspire further investigation in this area, as it has implications for real-world applications that experience rapid and significant changes within short periods, such as social networks and E-Commerce platforms.

As a result of our analytical endeavors, several avenues for future research emerge, including the development of time-adaptive methodologies and refined techniques for data curation or removal. We will leave these avenues as future works.

ACKNOWLEDGMENTS

This research is supported, in part, by the National Natural Science Foundation of China (Grant No. 62302295, 62076162); the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (No: AISG2-RP-2020-019); and the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore.

A APPENDIX

A.1 Dataset Overview

We introduce a time-varying dataset from a large E-Commerce search engine, which is publicly available for research purposes. The presence of spikes in our proposed dataset is evident, making it a valuable resource for studying the impact of spikes.

Dataset Link: https://drive.google.com/drive/folders/1icPJtTjgPUW6BuOrajPEKP57vEZmuB2I?usp=sharing, password: 0569.

Industry Type: Academic - Tech.

Data Subject: Data about systems or products and their behaviors.

Content Description: The record format includes five integers in the following order: x_i, u_i, d_i, q_i, y_i.

Dataset Snapshot: See Tab 2. The dataset consists of 80 files named from Day1 to Day80.

Sensitivity of Data: The dataset comprises desensitized user activity data and business data.

Maintenance Status: The dataset is subject to limited maintenance. The data will not be updated, but any technical issues that arise will be addressed.

Example of Data Points: We present three examples as follows.

- "15614 94033 844852 11922 0". User 15614 views item 94033 at time 844852 given the query 11922 without click.
- "8319 4431 844852 3100 1". User 8319 views and clicks item 4431 at time 844852 given the query 3100.
- "47365 109929 844859 6924 2". User 47365 views and purchases item 109929 at time 844859 given query 6924.

Primary Data Modality: The dataset primarily consists of graph Data. The records can be regarded as edges from a *dynamic user-item interaction network*.

A.2 Implementation Details of Models

This subsection provides details on the implementation of the examined methods. Unless otherwise specified, all experiments employ the same set of hyperparameters and all trainable parameters are initialized via standard `tf.glorot_uniform_initializer`. The batch size is set to 64. Due to the observed one-epoch overfitting issue, we limited the epoch to 1. The embedding size for each ID is 64, resulting in an input length of 192. We use the Adam optimizer with a learning rate of 0.0001. Please note that the above hyperparameters are specific to TensorFlow models, whereas MAML is implemented in PyTorch. Further details regarding the MAML implementation can be found in the accompanying code.

MLP: All MLPs consist of layers with sizes of 128, 64, 16, and 8.

MMoE: The MMoE model utilizes 8 different experts, each with layer sizes of 16 and 8. The outputs of these experts, which form a vector of dimension 8, are aggregated through a trainable weighted sum gate and subsequently forwarded to the task-specific towers. The task-specific tower has two layers, with sizes of 16 and 8. The sizes are designed to be similar to the MLP model, ensuring a fair comparison without leveraging differences in parameter quantity.

WDL: WDL incorporates an additional date embedding in its wide component, which is processed by a linear layer. All date embeddings are initialized as all-0 vectors.

DIN and DIEN: We employ the implementations provided by the DIEN paper [33] [3] and integrate them into our framework. The behavior sequences are additionally maintained in memory.

MAML: In our 2-way classification task, we used a support set size of 400 and a query set size of 800. This means we utilized the latest 800 behaviors for finetuning to predict the subsequent 800 behaviors. The meta batch size was set to 1. Our implementation of MAML was based on the publicly available PyTorch implementation [4], which we adapted for our purposes.

[3] https://github.com/mouna99/dien

[4] https://github.com/dragen1860/MAML-Pytorch

A.3 Evaluation Criteria: AUC Metric

The Area Under the Curve (AUC) can be computed as:

$$\mathrm{AUC}_D(f) = \frac{\sum_{i \in D^-} \sum_{j \in D^+} \mathbf{1}[f(i) < f(j)]}{|D^-| \cdot |D^+|}, \tag{4}$$

where D represents the dataset, D^- denotes the negative samples (*i.e.*, unmatched items and users) in D, and D^+ denotes the positive samples (*i.e.*, matched items and users) in D. The function f represents the model.

AUC reflects the frequency with which samples with better labels receive higher scores compared to samples with lower labels. Given the specific nature of E-Commerce, we consider purchase behaviors as the primary positive samples in our evaluation.

A.4 More Experimental Results

A.4.1 Stop Training for Longer Term. To extend the above experiment, Figure 7 visually demonstrates the impact of stopping the training after Day 51. We observe that the model maintains competitive performance for the following three days, gradually declining thereafter. It demonstrates that the distribution is smoothly changing, highlighting the importance of continuously training models with the latest data to maintain a high level of accuracy. However, the model can achieve better performance on the last sales event, implying that the distribution can significantly differ between daily and sales events. This finding also suggests that the skip training strategy potentially addresses spikes and offers a solution to improve the AUC.

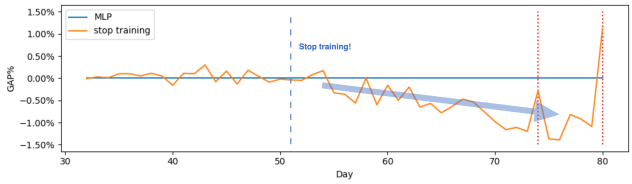

Figure 7: Performances of the model which stops training after Day 51.

A.4.2 Regularization Techniques for Overfitting Problem. We replicate the phenomenon known as "one-epoch overfitting" as previously reported in [32, 34, 36]. Figure 8 illustrates this issue. A decline in the performance of the examined models after 2 epochs can be clearly observed. We would like to emphasize that this overfitting phenomenon exhibits some distinctive characteristics compared to the common overfitting in machine learning research, such that standard regularization tricks solving overfitting may not be helpful [32]. Specifically:

- *Transfer-related.* In our dataset, the test data may exhibit variations in distribution compared to the training data, highlighting the importance of the model's generalization ability. The desired methods should demonstrate the capability to effectively adapt to diverse data distributions.
- *Induction-required.* Optimal performance is achieved when models encounter samples only once, which is different from the common machine learning overfitting that occurs after several epochs. This phenomenon suggests a strong tendency of models to prioritize memorization rather than induction, resulting in the ability to memorize data seen only once.To address this issue, desired methods should focus on enabling extrapolation and encouraging learning induction of behaviors, rather than relying on memorization.

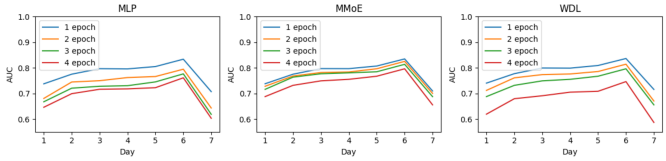

Figure 8: The reproduced one-epoch overfitting problem.

Additionally, we investigate the effectiveness of two regularization techniques to mitigate the overfitting problem: dropout and batch normalization. Results can be found in Table 5. Both techniques are applied after each fully connected layer, and dropout rate is set to 0.5.

Table 5: AUC of baselines with tricks solving overfitting.

Settings	Methods	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Offline	MLP	0.7375	0.7752	0.7967	0.7958	0.8049	0.8337	0.7070
	+dropout	0.6671	0.6979	0.7118	0.7188	0.7214	0.7570	0.6566
	+batchnorm	0.6010	0.6806	0.6967	0.6817	0.6723	0.7365	0.6064
Online	MLP	0.7528	0.7807	0.8021	0.8033	0.8146	0.8362	0.7376
	+dropout	0.6651	0.6925	0.7040	0.7162	0.7095	0.7570	0.6553
	+batchnorm	0.7447	0.7778	0.7958	0.7978	0.8106	0.8343	0.7313

The results reveal that both techniques consistently lead to a decrease in performance. Given that the model has been trained on 73 days' worth of data, it is less likely to suffer from underfitting. We hypothesize that dropout reduces the efficiency of the model's parameters. Furthermore, batch normalization demonstrates a significant decline in performance in the offline setting, while only exhibiting slightly worse performance in the online setting. We speculate that the optimal parameters for batch normalization are subject to change in a time-varying dataset. Therefore, if we update the parameters on a daily basis, the performance on the last day would already differ greatly from that on the subsequent day. However, if the model can access recent data for training, it may be able to properly learn the parameters required for batch normalization. Both results imply the one epoch overfitting problem is associated with certain properties inherent to the time-varying dataset.

REFERENCES

[1] Ai, Q., Wang, X., Bruch, S., Golbandi, N., Bendersky, M., and Najork, M. Learning groupwise multivariate scoring functions using deep neural networks. In *SIGIR* (2019), pp. 85–92.

[2] Burges, C., Ragno, R., and Le, Q. Learning to rank with nonsmooth cost functions. *NeurIPS* (2006).

[3] Burges, C. J. C. From ranknet to lambdarank to lambdamart: An overview. *Learning 11*, 23-581 (2010), 81.

[4] Burges, C. J. C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. N. Learning to rank using gradient descent. In *ICML* (2005), pp. 89–96.

[5] Cao, Z., Qin, T., Liu, T., Tsai, M., and Li, H. Learning to rank: from pairwise approach to listwise approach. In *ICML* (2007), pp. 129–136.

[6] Caruana, R. Multitask learning. *Machine Learning 28* (1997), 41–75.

[7] Chapelle, O., and Chang, Y. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge* (2011), pp. 1–24.

[8] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (2016), pp. 7–10.

[9] Cossock, D., and Zhang, T. Statistical analysis of bayes optimal subset ranking. *IEEE Trans. Information Theory 54*, 11 (2008), 5140–5154.

[10] De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 7 (2021), 3366–3385.

[11] Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML* (2017), pp. 1126–1135.

[12] Guo, H., Tang, R., Ye, Y., Li, Z., and He, X. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (2017), pp. 1725–1731.

[13] Guo, H., Yu, J., Liu, Q., Tang, R., and Zhang, Y. Pal: a position-bias aware learning framework for ctr prediction in live recommender systems. In *RecSys* (2019), pp. 452–456.

[14] Harper, F. M., and Konstan, J. A. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems 5*, 4 (2015), 1–19.

[15] Hawking, D., and Craswell, N. Overview of the trec-2001 web track. *Nist Special Publication Sp*, 250 (2002), 61–67.

[16] Hersh, W., Buckley, C., Leone, T., and Hickam, D. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *SIGIR* (1994), pp. 192–201.

[17] Huzhang, G., Pang, Z., Gao, Y., Liu, Y., Shen, W., Zhou, W.-J., Da, Q., Zeng, A., Yu, H., Yu, Y., et al. Aliexpress learning-to-rank: Maximizing online model performance without going online. *TKDE* (2021).

[18] Li, J., Jing, M., Lu, K., Zhu, L., Yang, Y., and Huang, Z. From zero-shot learning to cold-start recommendation. In *AAAI* (2019), pp. 4189–4196.

[19] Li, P., Burges, C. J. C., and Wu, Q. Mcrank: Learning to rank using multiple classification and gradient boosting. In *NeurIPS* (2007), pp. 897–904.

[20] Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., and Chi, E. H. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD* (2018), pp. 1930–1939.

[21] Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. Cross-stitch networks for multi-task learning. In *CVPR* (2016), pp. 3994–4003.

[22] Ni, J., Li, J., and McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP* (2019), pp. 188–197.

[23] Qin, T., and Liu, T. Introducing LETOR 4.0 datasets. *CoRR abs/1306.2597* (2013).

[24] Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., and Wang, J. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)* (2016), IEEE, pp. 1149–1154.

[25] Ruder12, S., Bingel, J., Augenstein, I., and Søgaard, A. Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142* (2017).

[26] Tang, H., Liu, J., Zhao, M., and Gong, X. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *RecSys* (2020), pp. 269–278.

[27] Tianjunzi, T. Cikm cup 2016 track 2: Personalized e-commerce search challenge, Sep 2022.

[28] Vartak, M., Thiagarajan, A., Miranda, C., Bratman, J., and Larochelle, H. A meta-learning perspective on cold-start recommendations for items. *NeurIPS* (2017).

[29] Wang, R., Fu, B., Fu, G., and Wang, M. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 2017, pp. 1–7.

[30] Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys 53*, 3 (2020), 1–34.

[31] Xia, F., Liu, T., Wang, J., Zhang, W., and Li, H. Listwise approach to learning to rank: theory and algorithm. In *ICML* (2008), pp. 1192–1199.

[32] Zhang, Z.-Y., Sheng, X.-R., Zhang, Y., Jiang, B., Han, S., Deng, H., and Zheng, B. Towards understanding the overfitting phenomenon of deep click-through rate prediction models. *arXiv preprint arXiv:2209.06053* (2022).

[33] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., and Gai, K. Deep interest evolution network for click-through rate prediction. In *AAAI* (2019), vol. 33, pp. 5941–5948.

[34] Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. Deep interest network for click-through rate prediction. In *KDD* (2018), pp. 1059–1068.

[35] Zhu, J., Dai, Q., Su, L., Ma, R., Liu, J., Cai, G., Xiao, X., and Zhang, R. BARS: towards open benchmarking for recommender systems. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022* (2022), E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, Eds., ACM, pp. 2912–2923.

[36] Zhu, J., Liu, J., Yang, S., Zhang, Q., and He, X. Open benchmarking for click-through rate prediction. In *CIKM* (2021), pp. 2759–2769.