



# Capturing Conversion Rate Fluctuation during Sales Promotions: A Novel Historical Data Reuse Approach

Zhangming Chan  
zhangming.czm@alibaba-inc.com  
Alibaba Group  
Beijing, People's Republic of China

Yu Zhang  
xieyuan.zy@alibaba-inc.com  
Alibaba Group  
Beijing, People's Republic of China

Shuguang Han\*  
shuguang.sh@alibaba-inc.com  
Alibaba Group  
Hangzhou, People's Republic of China

Yong Bai  
baiy@smail.nju.edu.cn  
Nanjing University  
Nanjing, People's Republic of China

Xiang-Rong Sheng  
xiangrong.sxr@alibaba-inc.com  
Alibaba Group  
Beijing, People's Republic of China

Siyuan Lou  
lousiyuan.lsy@alibaba-inc.com  
Alibaba Group  
Beijing, People's Republic of China

Jiacen Hu  
Baolin Liu†  
hujiacen@ustb.edu.cn  
liubaolin@ustb.edu.cn  
USTB  
Beijing, People's Republic of China

Yuning Jiang  
mengzhu.jyn@alibaba-inc.com  
Alibaba Group  
Beijing, People's Republic of China

Jian Xu  
Bo Zheng  
xiyu.xj@alibaba-inc.com  
bozheng@alibaba-inc.com  
Alibaba Group  
Beijing, People's Republic of China

## ABSTRACT

Conversion rate (CVR) prediction is one of the core components in online recommender systems, and various approaches have been proposed to obtain accurate and well-calibrated CVR estimation. However, we observe that a well-trained CVR prediction model often performs sub-optimally during sales promotions. This can be largely ascribed to the problem of the data distribution shift, in which the conventional methods no longer work. To this end, we seek to develop alternative modeling techniques for CVR prediction. Observing similar purchase patterns across different promotions, we propose reusing the historical promotion data to capture the promotional conversion patterns. Herein, we propose a novel **Historical Data Reuse (HDR)** approach that first retrieves historically similar promotion data and then fine-tunes the CVR prediction model with the acquired data for better adaptation to the promotion mode. HDR consists of three components: an automated data retrieval module that seeks similar data from historical promotions, a distribution shift correction module that re-weights the retrieved data for better aligning with the target promotion, and a TransBlock module that quickly fine-tunes the original model for better adaptation to the promotion mode. Experiments conducted with real-world data demonstrate the effectiveness of HDR, as it improves both ranking and calibration metrics to a large extent. HDR has also been

deployed on the display advertising system in Alibaba, bringing a lift of 9% RPM and 16% CVR during Double 11 Sales in 2022.

## CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking; Online advertising.**

## KEYWORDS

Conversion Rate Prediction, Recommender System, Label Shift, Online Advertising, Data-centric AI

## ACM Reference Format:

Zhangming Chan, Yu Zhang, Shuguang Han, Yong Bai, Xiang-Rong Sheng, Siyuan Lou, Jiacen Hu, Baolin Liu, Yuning Jiang, Jian Xu, Bo Zheng. 2023. Capturing Conversion Rate Fluctuation during Sales Promotions: A Novel Historical Data Reuse Approach. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, August 6–10, 2023, Long Beach, CA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3580305.3599788>

## 1 INTRODUCTION

As one of the largest e-commerce platforms in the world, Taobao provides an online shopping service for hundreds of millions of users on a daily basis. To facilitate an effective matching of user interest with millions of online products, we spend significant efforts developing deep machine learning models on Taobao. Among them, the click-through rate (CTR) prediction model [4, 24, 25, 32, 38] and the conversion rate (CVR) prediction model are the two most common ones. In this study, we are interested in the CVR prediction problem. Researchers and practitioners have proposed a variety of approaches for obtaining accurate and well-calibrated estimation of CVR [8, 10, 13, 20, 30, 31, 34, 36]. For instance, to tackle the problem of data sparsity, Wen et al. [30, 31] and Yang et al. [36] utilized a series of post-click behaviors for supplementing the rare conversion event, which resulted in an improved performance on CVR

\*Shuguang Han and Baolin Liu are the corresponding authors.

†USTB is the abbreviation of University of Science and Technology Beijing.

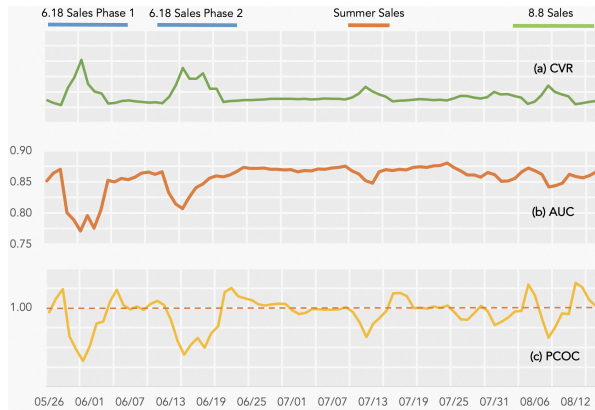
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components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

KDD '23, August 6–10, 2023, Long Beach, CA, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0103-0/23/08...\$15.00

<https://doi.org/10.1145/3580305.3599788>



**Figure 1: An illustration of the actual CVR and AUC/PCOC performance for the production CVR prediction model. To ensure data anonymity, we limit the number of labels on the Y-axis for CVR and PCOC to a minimum.**

prediction. Other researchers dealt with the selection bias problem in CVR prediction and proposed the entire space estimation method by jointly learning with multiple behavior prediction tasks [20, 34].

Despite the previous advancement, we still find that our production CVR prediction model performs sub-optimally during **sales promotions**. Sales promotion is a popular marketing strategy in e-commerce platforms that stimulate consumers' demand by offering a significant discount within limited time. The Black Friday Sales on Amazon and the Double 11 Sales on Taobao are the two well-known online promotion festivals worldwide. It is worth noting that sales promotion is frequently held on e-commerce platforms. For example, there are 33 promotions in Taobao 2022, taking about half of the time in a year. According to Figure 1(b) and Figure 1(c), there is a deteriorated performance of the production model during a sales promotion. Both the AUC and PCOC<sup>1</sup> metrics drop significantly. For a concrete example, during the 6.18 Sales, the online AUC falls as much as 10%, and the value of PCOC is only 1/3. This seriously affects the effectiveness of the production system.

Such a phenomenon can be primarily ascribed to the rapid change of data distribution, as we observe a substantial fluctuation of the actual CVR during sales promotion, as shown in Figure 1(a). Therefore, conventional CVR prediction models with i.i.d. (independent and identically distributed) assumption on the training and test data would certainly fail during the promotion. To resolve this problem, one may improve the model freshness so that the prediction model can be quickly adapted to the *promotion mode*. However, the delayed feedback nature of the conversion event further complicates the utilization of recent data.

Unlike click behaviors, feedback labels for the conversion event cannot be collected within a short time as conversion may happen in days or even weeks afterward. This introduces a dilemma – to employ recent data with incomplete labels or to wait for more accurate labels. By assuming a stable delayed conversion pattern (e.g., the delay time for the final conversion remains steady), a delayed feedback model resolves this problem by utilizing the relationship

<sup>1</sup>Predicted CVR Over the actual CVR,  $PCOC = \frac{\sum \text{predicted CVR}}{\sum \text{actual CVR}}$  is generally applied for evaluating the calibration ability, with 1.0 denoting a perfect calibration.

between the ultimate conversion and the conversion time to correct the incomplete real-time labels. However, such an assumption does not hold during the sales promotion since there is a rapid fluctuation of conversion behaviors at this time, which calls for the development of alternative modeling techniques.

In spite of the discrepancy between non-promotional and promotional conversions, the conversion patterns in different sales promotions remain to be similar (see more details in Figure 3 and Table 1). This motivates us to fine-tune our model with the historical data from similar promotions. Since the past promotions have already finished and the complete conversion labels are now easily accessible, the historical data no longer encounter the delayed feedback problem. To this end, we propose a novel **Historical Data Reuse (HDR)** approach with the first stage concentrating on retrieving historically similar promotional data, and the second stage focusing on model fine-tuning with the retrieved data.

Specifically, we first represent each day with a vector of designed features. Then, for a target promotion, we search for the most similar promotion in history with the nearest neighbor algorithm. After that, the retrieved data is applied for model fine-tuning so that the model can quickly adapt to the promotion mode. To further address the disparity of conversion behaviors between the retrieved promotion and the target promotion, we design a distribution shift correction method to further refine the fine-tuning mechanism. Concretely, we first estimate an importance weight to measure the discrepancy of different promotions. Then, we utilize the obtained importance weight to revise the fine-tuning loss following the Importance-Weighted Empirical Risk Minimization framework [26].

In practice, the model fine-tuning with the historical data often encounters the one-epoch problem [40], i.e., the model will be overfitted when the same data has been trained twice. For this reason, we design a TransBlock module (see Figure 5) to isolate the fine-tuning parameters from the main model and employ two separate training configurations. To capture different conversion patterns, TransBlock explicitly models the transition between promotional conversions and non-promotional conversions. In this way, the main model trained from the non-promotional conversion data will be less impacted, while the fine-tuning module helps the production model quickly adapt to the promotion mode.

The main contributions of our work are summarized as follows:

- With regard to the deteriorated performance of the production CVR prediction model during sales promotions, we propose a novel Historical Data Reuse (HDR) algorithm, aiming to learn common patterns from different historical promotions.
- To address the overfitting problem from a direct fine-tuning with the historical data, we design a TransBlock module that separates the fine-tuning parameters from the main model and explicitly models the transitions between the non-promotional conversion to the promotional conversion.
- HDR has been successfully deployed in the display advertising system of Alibaba, bringing a substantial lift of core production metrics (RPM+9% and CVR+16%) during Double 11 Sales in 2022.

## 2 RELATED WORK

CVR prediction during sales promotion is a complicated problem that suffers from the distribution shift and delayed conversions.

To provide a comprehensive overview of relevant techniques, we introduce a variety of related works, including conversion rate prediction models and distribution shift correction algorithms.

## 2.1 Conversion Rate Prediction

CVR prediction is essential for many industrial machine-learning systems, including search, recommendation, and online advertising. A variety of approaches have been proposed to deal with the accurate estimation of conversion rate [8, 10, 13, 20, 29–31, 34, 36]. In practice, due to the inherent similarity in training sample construction, CVR model often adopts a resembling modeling architecture as the CTR model. Previous studies have also developed a set of novel solutions in considering the unique characteristics of CVR prediction. For instance, to address the data sparsity issue, researchers have used a series of post-click behaviors to supplement the rare conversion event [30, 31, 36]. Other researchers focused on dealing with the problem of selection bias in CVR estimation and proposed the entire space estimation method by jointly learning with several other user behavior prediction tasks [20, 34].

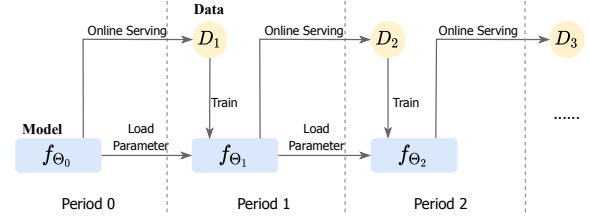
## 2.2 Delayed Feedback Modeling

Since the conversion events may happen long after clicks, the collection of actual conversion labels may delay by hours or even days. The importance of delayed feedback modeling is first emphasized in the pioneering work DFM by Chapelle [6]. DFM introduces an additional model that captures the expected conversion delay, assuming the delay distribution follows the exponential distribution. Ktena et al. [16] first studied the delayed feedback problem under the online training setting. They treated all data samples as negatives in the beginning. Once the positive engagement occurs, the sample will be duplicated with a positive label and trained for the second time. Importance sampling is adopted to re-weight training samples to further learn the model from the bias distribution. However, the resulting performance can be less satisfied due to the inaccuracy of weight computation. Therefore, Yang et al. [37] proposed to wait longer to obtain more conversions. In the meantime, the delayed positive samples were also duplicated in the training pipeline. Gu et al. [10] and Chen et al. [8] argued that the duplicated positive samples change the training distribution, which leads to sub-optimal performance. Hence, they proposed duplicating real negative samples and refining the importance weights to maintain consistency between training and testing distribution.

The above approaches often assume a stable delayed conversion pattern, e.g., the delay distribution for the conversion remains the same between training and test data. This presumption may hold in non-promotional data. However, as mentioned in Section 1, such a presumption does not hold during the sales promotion.

## 2.3 Distribution Shift

The fluctuation of actual CVR during sales promotion is mainly due to the problem of the rapid distribution shift. Distribution shift states that there is an inconsistency between the training distribution  $p(x, y)$  and the testing distribution  $q(x, y)$ . Previous research has studied this problem from two perspectives: covariate shift [5, 7, 9, 18, 21, 26, 28] and label shift [1–3, 17, 22, 33].



**Figure 2: An illustration of the online learning process for our production CVR model, which is updated on a daily basis.**

Covariate shift assumes that  $p(y|x)$  is constant and  $p(x)$  changes between training and testing. Under this presumption, Shimodaira [26] proposed an Importance-Weighted Empirical Risk Minimization (IW-ERM) framework that utilizes importance weights to adjust the training distribution. Similar approaches have also been adopted by other researchers [5, 11]. Different from the covariate shift, label shift states that  $p(x|y)$  is stable between training and testing distributions. Black Box Shift Estimation (BBSE) is such an algorithm to tackle the problem of label shift with a black-box predictor [17]. In addition, Schölkopf et al. [22] proposed to utilize causal inference to solve the label shift problem, and Azizzadenesheli et al. [2] developed a Regularized Learning under Label Shift (RLLS) approach that adopted regularization to de-bias the BBSE algorithm. Note that BBSE and RLLS are also the basis of our work.

## 3 METHODOLOGY

Before delving into the details of our proposed approach, we first provide a formal description of the problem. Let  $f_{\Theta}(\cdot)$  denote the CVR prediction model, and  $\Theta$  indicates the trainable parameters. Our production model optimizes  $\Theta$  with the training data  $\mathcal{A}(x, y)$  in a supervised learning way. After being deployed online, it will be used to serve an incoming new data distribution  $\mathcal{B}(x, y)$ . In practice, as shown in Figure 2, our production model is trained in an online learning manner [4, 19, 38] and is updated on a daily basis to make a trade-off between delayed conversion and data freshness.

The validity of the above training paradigm relies on the i.i.d. assumption between  $\mathcal{A}$  and  $\mathcal{B}$ . However, as illustrated by Figure 1, this hypothesis fails during sales promotions. To deal with the data distribution shift problem, a potential solution is to seek a historical distribution  $\mathcal{B}'$  that resembles the incoming promotion data  $\mathcal{B}$ , and further fine-tune the production model with  $\mathcal{B}'$ . This can be formalized using Formula 1, in which  $f'_{\Theta}$  denotes the fine-tuned model that is meant to adapt the promotion mode. Note that instead of directly training a model with  $\mathcal{B}'$  (while discarding  $\mathcal{A}$ ), we adopt the **training-and-finetuning** approach to guarantee the recent data remain to be included in the model.

$$f_{\Theta} : \text{Train}(\mathcal{A}) \implies f'_{\Theta} : \text{Train}(\mathcal{A}) + \text{Finetune}(\mathcal{B}') \quad (1)$$

The above formulation introduces several important challenges: (1) how to obtain the historical promotion data distribution  $\mathcal{B}'$  that is similar to  $\mathcal{B}$ , (2) how to deal with the discrepancy, if any, between  $\mathcal{B}$  and  $\mathcal{B}'$ , and (3) what are the effective architectures for model fine-tuning over  $\mathcal{B}'$ . The proposed HDR approach focuses on tackling these three obstacles, and the detailed implementations are provided in the below subsections.



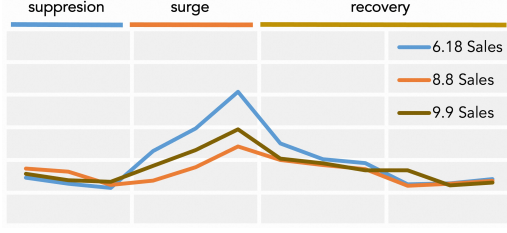


Figure 3: The change of conversion rate (Y-axis) over time in three different sales promotions. They all exhibit a similar pattern: pre-promotion suppression, on-promotion surging, and post-promotion recovery. For the purpose of data anonymity, we hide the labels on the Y-axis.

### 3.1 Obtaining $\mathcal{B}'$ with Historical Data Retrieval

For challenge (1), we design a simple yet effective vector-based approach to retrieve historically similar promotion data distribution  $\mathcal{B}'(x, y)$ . In our production system, the historical data is processed in the granularity of natural days, we thus regard each day as a candidate distribution for retrieval.

**3.1.1 Vector-Based Representation.** To facilitate automated data retrieval, we first represent each day with a vector of numerical features. As illustrated by Figure 3, the conversion pattern in different sales promotions looks quite similar – they exhibit a resembling pattern of pre-promotion suppression, on-promotion surging, and post-promotion recovery. By capturing such a trend, we can easily distinguish ordinary time from promotion activity. To this end, we adopt two kinds of **conversion-related features** from the most recent three days when computing the data similarity.

**CVRs from previous days.** Average CVR per day is obviously a good indicator of different conversion patterns. We first adopt a set of CVR metrics from the most recent three days for vector representation. Note that our production system utilizes three days as the attribution window; thereby, a non-conversion within one or two days (incomplete attribution) may end up with a conversion in three days (complete attribution). We thus consider both complete and incomplete attribution of conversions, including the conversion in one day, two days, and three days, for vector representation.

**Impression ratio of representative categories.** In practice, we find that the purchase of certain product categories such as cosmetics often surges during promotion, and this phenomenon is consistent across different sales promotions. As a consequence, we consider several representative categories of such and utilize the impression ratio of each category as a numerical feature. Here, the impression ratio of a category is computed with the number of impressions for the given category over the total number of impressions. Meanwhile, these categories are selected based on the discrepancy of impression ratio between the promotion and non-promotion periods of time, which include Toddler, House Cleaning, Personal Care, Cosmetics and etc.

Since the impression ratio does not rely on the collection of user feedback, it can be readily available. Therefore, we further compute the impression ratios with the data from the first ten hours of a target promotion day. These new features result in a performance

Table 1: An analysis of the conditional input distribution  $p(x|y)$  for 8.8 Sales and 9.9 Sales. We examine the conversion patterns for two groups of users ( $U_1, U_2$ ) and two product categories ( $C_1, C_2$ ). The value in each cell denotes the corresponding conditional probability.

User Group $\times$ Category	8.8 Sales		9.9 Sales	
	$y = 0$	$y = 1$	$y = 0$	$y = 1$
$U_1, C_1$	0.9597%	0.3831%	1.1013%	0.3595%
$U_1, C_2$	0.0064%	0.0472%	0.0065%	0.0575%
$U_2, C_1$	0.3125%	0.1738%	0.3441%	0.1597%
$U_2, C_2$	0.1769%	0.0258%	0.1738%	0.0288%

boost for data retrieval and lead to a further improvement in online performance despite a 10-hour delay for model serving.

**3.1.2 The Retrieval Process.** With the vector-based representation, we then utilize the nearest neighbor algorithm for retrieving the most similar historical data distribution. Here, cosine distance is adopted as the similarity measure while searching for the nearest neighbors. In our production system, the retrieved top-two days will be kept for later model fine-tuning.

### 3.2 Distribution Shift Correction

As mentioned in challenge (2), despite being similar, there is no guarantee of complete equivalence for the two data distributions  $\mathcal{B}(x, y)$  and  $\mathcal{B}'(x, y)$ . Figure 3 demonstrates that there are still discrepancies in the actual conversion rates for different sales promotions, particularly at the on-promotion surging phase. This calls for the fine-tuning module, as formulated by Equation 1, to come out with a resolution for the distribution shift problem.

As a consequence, we redesign the fine-tuning module under the Importance-Weighted Empirical Risk Minimization (IW-ERM) framework [27] by minimizing the below empirical risk:

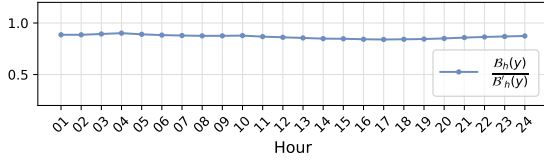
$$\mathcal{L} = \int \frac{\mathcal{B}(x, y)}{\mathcal{B}'(x, y)} \cdot \ell(x, y) d\mathcal{B}'(x, y) \quad (2)$$

where  $\ell(x, y)$  denotes the standard cross-entropy loss computed with the distribution  $\mathcal{B}'(x, y)$ .  $\frac{\mathcal{B}(x, y)}{\mathcal{B}'(x, y)}$  measures the importance weight that aims to correct the disparity between the two distributions. With Bayes' theorem, this can be further transformed to:

$$\mathcal{L} = \int \frac{\mathcal{B}(x|y)}{\mathcal{B}'(x|y)} \cdot \frac{\mathcal{B}(y)}{\mathcal{B}'(y)} \cdot \ell(x, y) d\mathcal{B}'(x, y) \quad (3)$$

in which  $\mathcal{B}(x|y)$  and  $\mathcal{B}'(x|y)$  represent the conditional input distributions,  $\mathcal{B}(y)$  and  $\mathcal{B}'(y)$  stand for the label distributions. For better understanding, we re-emphasize that  $\mathcal{B}(x, y)$  denotes the target promotion data distribution, and  $\mathcal{B}'(x, y)$  refers to the data distribution retrieved from historical promotion data. However, since  $\mathcal{B}(y)$ ,  $\mathcal{B}(x|y)$  and  $\mathcal{B}(x, y)$  are both unavailable at the serving time, we need to make a few presumptions.

**3.2.1 Presumption I:  $\mathcal{B}(x|y) = \mathcal{B}'(x|y)$ .** We first assume the equivalence of the two conditional input distributions  $\mathcal{B}(x|y)$  and  $\mathcal{B}'(x|y)$ , which means that among the purchased (non-purchased) items, the input distributions stay to be similar. It is worth noting that this actually shares the same presumption as the Black Box Shift Estimation (BBSE) algorithm [17], which is also applied for estimating the target label distribution  $\mathcal{B}(y)$  later on.



**Figure 4: An illustration of  $\mathcal{B}_h(y)/\mathcal{B}'_h(y)$  over different selections of the hour  $h$ . Here, the largest difference is less than 5% as the maximum and minimum values are 0.89 and 0.85, respectively. This clearly shows the stability of the ratio.**

To seek evidence from the actual data, we analyze the conditional input distribution with two sales promotions. More specifically, we select user group and product category as two representative input features and fill the conditional probabilities  $p(U_i, C_j|y=0)$  and  $p(U_i, C_j|y=1)$  in Table 1. It is apparent that  $p(x|y)$  holds steadily across different sales promotions, lending significant support for the above presumption. With such an assumption, the loss function can be further approximated by:

$$\mathcal{L} = \int \frac{\mathcal{B}(y)}{\mathcal{B}'(y)} \cdot \ell(x, y) d\mathcal{B}'(x, y) \quad (4)$$

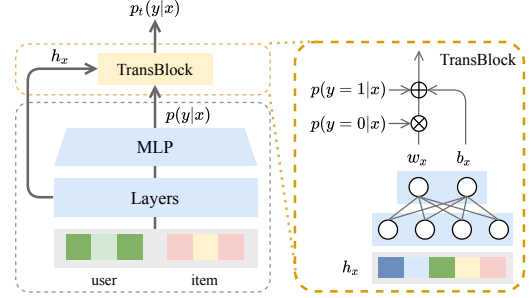
To be consistent with the model update frequency, the label distributions  $\mathcal{B}(y)$  and  $\mathcal{B}'(y)$  are estimated in the granularity of a natural day. Here,  $\mathcal{B}'(y)$  can be easily computed with the historical data while  $\mathcal{B}(y)$ , by the nature of delayed feedback, remains unavailable at the serving time. Inspired by Lipton et al. [17], we develop a distribution estimation approach for obtaining  $\mathcal{B}(y)$ . More specifically, we first introduce  $\hat{y}$  to indicate the model prediction of the input  $x$ , i.e.,  $f_{\Theta}(x)$ . Since  $\hat{y}$  only relies on the data input, with Presumption I, we can easily demonstrate the equivalence of  $\mathcal{B}(\hat{y}|y)$  and  $\mathcal{B}'(\hat{y}|y)$ . In this way, the distribution of model prediction  $\mathcal{B}(\hat{y})$  can be expressed with the below Equation 5.

$$\begin{aligned} \mathcal{B}(\hat{y}) &= \sum_{y \in \{0,1\}} \mathcal{B}(\hat{y}|y) \cdot \mathcal{B}(y) \quad (\text{Total Probability}) \\ &= \sum_{y \in \{0,1\}} \mathcal{B}'(\hat{y}|y) \cdot \mathcal{B}(y), \quad (\text{Presumption I}) \end{aligned} \quad (5)$$

Once  $\mathcal{B}(\hat{y})$  and  $\mathcal{B}'(\hat{y}|y)$  are both available, one can solve this Equation and obtain an analytic solution for  $\mathcal{B}(y)$ . However,  $\mathcal{B}(\hat{y})$  denotes the prediction distribution for the entire day, which is not fully accessible until the end of the day. We thus make the below presumption and try to use partial data of the given day for distribution estimation despite a certain delay in model serving.

**3.2.2 Presumption II:**  $\frac{\mathcal{B}(y)}{\mathcal{B}'(y)} = \frac{\mathcal{B}_h(y)}{\mathcal{B}'_h(y)}$ . We denote  $\mathcal{B}_h(y)$  as label distribution computed with the data from the midnight to the  $h$ -th hour of the day. This assumption states that, within one day, the disparity between two sales promotions remains steady no matter how many hours of data are employed<sup>2</sup>. Using the actual data, we study the stability of  $\mathcal{B}_h(y)/\mathcal{B}'_h(y)$  with different selections of the hour  $h$ . As shown in Figure 4, we do observe a relatively stable ratio as the largest difference within one day does not exceed 5%. In the production system, we set  $h=10$ ; therefore, the model serving is

<sup>2</sup>However,  $\mathcal{B}_h(y)$  itself is not stable and may change with  $h$ . For example, the CVR at night is larger than that in the morning as customers are often at work during the day time and shop at night. We put more discussion in Appendix E.



**Figure 5: The overall architecture of TransBlock. To overcome the one-epoch problem, the main model and TransBlock are updated with separate training configurations.**

delayed by 10 hours. However, we argue that it won't affect much as the conversion mostly happens during the other half of the day, and such a delay is also important for feature computation in historical data retrieval, as mentioned in Section 3.1.1.

Under this presumption, the loss in Equation 4 now turns into:

$$\mathcal{L} = \int \frac{\mathcal{B}_h(y)}{\mathcal{B}'_h(y)} \cdot \ell(x, y) d\mathcal{B}'(x, y) \quad (6)$$

Again,  $\mathcal{B}'_h(y)$  can be computed from the historical data. In terms of  $\mathcal{B}_h(y)$ , we repeat the same reasoning process as did in Section 3.2.1 and acquire the below Equation 7 for  $\mathcal{B}_h(\hat{y})$ . Now, we have both  $\mathcal{B}_h(\hat{y})$  and  $\mathcal{B}'_h(\hat{y}|y)$  available at the  $(h+1)$ -th hour. With  $\mathcal{B}_h(y)$  being the only unknown variable, we can solve the equation and obtain an analytic solution. More details about the final solution can be found in Appendix A.

$$\mathcal{B}_h(\hat{y}) = \sum_{y \in \{0,1\}} \mathcal{B}'_h(\hat{y}|y) \cdot \mathcal{B}_h(y). \quad (7)$$

### 3.3 Effective Fine-tuning with TransBlock

The above two sections haven't discussed the detailed modeling architecture for the fine-tuning module and how it interacts with the main model, which is the focus of this section. However, in practice, we encounter a serious overfitting problem during fine-tuning. This has been studied as the one-epoch issue in prior studies, that is, the model will be overfitted once the same data has been seen twice [40–42]. We thus develop a TransBlock module to resolve this challenge. It is worth emphasizing that TransBlock also captures the transition between non-promotional and promotional conversions, making the predictions easily interpretable.

As illustrated by Figure 5, TransBlock introduces a few extra parameters stacked on top of the main model. During fine-tuning, the parameters of TransBlock and the main model will be both updated through training the retrieved data  $\mathcal{B}'$ . To prevent the overfitting problem caused by re-training with the same data, we set an extensively smaller learning rate for the main model while keeping the regular learning rate for TransBlock. In this way, the main model trained from the non-promotional data will be less impacted, while the TransBlock module helps this main model quickly adapt to the promotion mode after fine-tuning.

To capture the difference between non-promotional conversion and promotional conversion, we explicitly model such a pattern

with the below Equation 8. More specifically, we compute the probability of promotional conversion with the following three components: (1) the probability of non-promotional conversion  $p(y = 1|x)$  obtained from the main model, (2) the transition from non-promotional non-conversion to promotional conversion  $v_x \cdot p(y = 0|x)$ , and (3) the transition from non-promotional conversion to promotional non-conversion  $u_x \cdot p(y = 1|x)$ , that is,

$$p_t(y = 1|x) = p(y = 1|x) + v_x \cdot p(y = 0|x) - u_x \cdot p(y = 1|x) \quad (8)$$

Here,  $u_x$  and  $v_x$  denote the instance-level transition probabilities with regard to the input feature  $x$ . They are both obtained from TransBlock. With a further expansion of the equation, we acquire:

$$\begin{aligned} p_t(y = 1|x) &= p(y = 1|x) + v_x \cdot p(y = 0|x) - u_x \cdot p(y = 1|x) \\ &= p(y = 1|x) + v_x \cdot p(y = 0|x) - u_x \cdot (1 - p(y = 0|x)) \\ &= p(y = 1|x) + (v_x + u_x) \cdot p(y = 0|x) - u_x \\ &= p(y = 1|x) + \underbrace{w_x \cdot p(y = 0|x) + b_x}_{w_x = v_x + u_x, b_x = -u_x} \end{aligned} \quad (9)$$

In practice, we utilize a simple Multi-layer Perceptron (MLP) that takes the transformed input feature  $h_x$  (after the embedding transformation of  $x$ ) and the predictions of the main model as input, and outputs  $w_x$  and  $b_x$  for computing the probability of promotional conversion. As illustrated by Equation 10,  $W_{\text{trans}}$  and  $b_{\text{trans}}$  are trainable parameters in TransBlock.

$$\begin{bmatrix} w_x \\ b_x \end{bmatrix} = W_{\text{trans}} \cdot h_x + b_{\text{trans}} \quad (10)$$

Since the output of TransBlock has now turned into the summation of three different components, there is no guarantee of the output range. Hence, we truncate the final prediction between  $[0, 1]$  with the below Equation 11,

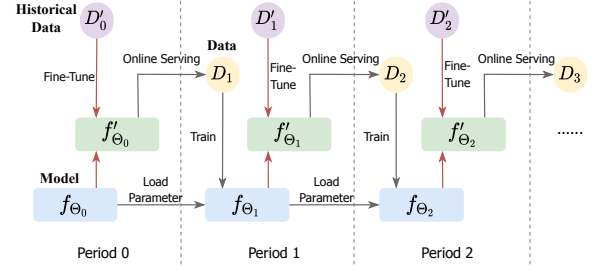
$$p'_t(y = 1|x) = \begin{cases} 0, & p_t(y = 1|x) < 0 \\ 1, & p_t(y = 1|x) > 1 \\ p_t(y = 1|x), & \text{otherwise.} \end{cases} \quad (11)$$

### 3.4 Optimization

With the above probability  $p'_t(y = 1|x)$ , we then compute the corresponding **cross-entropy loss**  $\ell(x, y)$ . Along with the importance weight  $\frac{\mathcal{B}_h(y)}{\mathcal{B}'_h(y)}$  (calculated in Section 3.2), we further place them back to Equation 6 for obtaining the final fine-tuning loss. Up to now, we have a complete picture of the loss function for HDR. During fine-tuning,  $W_{\text{trans}}$  and  $b_{\text{trans}}$  are updated with a regular learning rate  $\eta_1$ , whereas the original main model utilizes a much smaller learning rate  $\eta_2$ . Such an optimization trick is important and significantly improves model performance as shown in Section 5.2.2. It is worth noting that HDR only concentrates on the fine-tuning of the main model for better adaptation to the promotion mode, i.e., the  $\text{Finetune}(\mathcal{B}')$  term in Equation 1.

### 3.5 Overall Online Learning Process for HDR

HDR introduces a new training-and-finetuning paradigm that includes an automated historical data retrieval module and a model fine-tuning module. As a consequence, the original online learning process as illustrated by Figure 2 is no longer applicable. As a result, we redesign the existing online learning framework by injecting a



**Figure 6: An illustration of the training-and-finetuning online learning process for the proposed HDR approach.**

fine-tuning node. This new online learning process has been validated and successfully deployed in the display advertising platform of Alibaba, serving tens of millions of users on daily basis.

More specifically, as illustrated by Figure 6, the new online learning framework keeps the original process intact, that is the main model  $f_{\Theta}$  still updates on a daily basis. In addition, the fine-tuning node reloads the parameters from the main model and initializes the parameters for TransBlock. Once the retrieved data is ready, the fine-tuning process will be invoked automatically. After the fine-tuned model  $f'_{\Theta}$  is ready, it will be deployed to the production system for online serving. Note that, the fine-tuning node also runs on a daily basis to match the update frequency for the main model.

## 4 EXPERIMENT SETUP

To understand the effectiveness of the proposed HDR approach, we conduct an extensive number of offline and online experiments. This section starts with introducing the dataset for the offline experiments, which is collected from the display advertising system of Alibaba. Afterward, we introduce a set of evaluation metrics to measure the calibration ability and ranking performance of the proposed approach. For a complete understanding of HDR, we further compare its performance with a variety of baselines.

### 4.1 Dataset

Due to the lack of relevant public datasets, we first collect the user click log from the online display advertising system of Alibaba, which contains almost three-month data from 2022/05/22 to 2022/09/06. For each click, we store a binary 0/1 label indicating whether such a click leads to an ultimate purchase in three days. In addition, we also record the specific timestamp when the purchase occurred, and 'N/A' when there are no purchases. In this dataset, the average number of clicks (purchases) each day is around 40 million (600 thousand). For offline experiments, we treat the click and purchase information on 2022/09/06 (i.e., the promotion time for 9.9 Sales) as the testing data, and the rest as the training data.

### 4.2 Evaluation Metrics

In this paper, we evaluate both the ranking performance and calibration ability of the proposed HDR approach.

**4.2.1 Ranking Performance.** We first regard CVR prediction as a classification problem and thereby utilize AUC [23] to evaluate its ranking ability. AUC is commonly adopted to measure how well the

model differentiates between the distributions of different classes, and thus it is widely used in the two-class classification problem.

**4.2.2 Calibration Performance.** Calibration measures how well a prediction aligns with the actual conversion rate, which is of great importance for applications such as online advertising [24]. To measure the calibration performance, we employ three widely-adopted metrics including **LogLoss**, expected calibration error (**ECE**), and predicted CVR over the actual CVR (**PCOC**) [12, 35].

LogLoss measures the sample-level calibration error, while PCOC (or ECE) offers a macro view of the overall dataset-level (or subset-level) calibration performance. A smaller value of LogLoss or ECE usually implies a better calibration performance; and for PCOC, our goal is to obtain a value that is close to 1<sup>3</sup>.

**4.2.3 Online Performance.** With respect to the online A/B testing, we employ a set of top-line business metrics that include conversion rate (CVR), Revenue Per Mille (RPM), and Return Over Investment (ROI). Online CVR is a direct measure of prediction performance, and RPM and ROI are important metrics with regard to the industrial advertising system. RPM reflects the monetization ability of the production system, whereas ROI suggests the actual effect that our system can help improve the business of retailers.

### 4.3 Compared Methods

To understand the effectiveness of the proposed HDR approach, we introduce the below CVR prediction methods for comparison.

- **Base.** DEFER [10] is adopted as the base approach since it is our production CVR model and serves the main traffic of Alibaba display advertising system. Note that this method is one of the State-Of-The-Art delayed feedback models for CVR estimation.
- **ES-DFM.** ES-DFM [37] is another delayed feedback modeling algorithm that optimizes the expectation of true conversion distribution via importance sampling. It estimates the importance weight for each training sample and utilizes the estimated weight to revise the loss function for model training.
- **DEFUSE.** DEFUSE [8] is designed to tackle the delayed feedback problem in CVR prediction as well. Specifically, it divides the data samples into four groups — immediate positive samples, delayed positive samples, fake negative samples, and real negative samples, and further re-designs the importance sampling strategies for each of them, respectively.
- **Base w/o DFM.** The irregular conversion pattern during sales promotions violates the assumption of most existing delayed feedback models. For this reason, we experiment with a non-delayed-feedback modeling approach by discarding recent data without the complete conversion labels. Since the attribution window for conversion is three days in the production system, this method discards user behaviors of the recent two days.
- **Base w/ promotion indicator.** In addition to the fine-tuning of historical promotion data, another viable approach is to differentiate the promotion and non-promotion data with a binary indicator feature. In this way, the model may learn separate patterns for promotional and non-promotional conversions.
- **Base w/ direct retraining.** The simplest data reuse approach is to directly re-train the production model on top of the retrieved

historical data. In fact, such a method is equivalent to HDR with the removal of the distribution shift correction module and the TransBlock module. However, this may introduce the one-epoch problem as mentioned in Zhang et al. [40].

- **HDR.** To further analyze the usefulness of each module in HDR, we conduct a set of ablation studies on the distribution shift correction module and the TransBlock module, in which we take into account **HDR w/o DSC** (distribution shift correction) and **HDR w/o TransBlock**, respectively.

It is worth noting that **ES-DFM**, **DEFUSE**, and **Base w/o DFM** are introduced for the purpose of understanding the effectiveness of different delayed feedback models, particularly how well they perform during sales promotions. The introduction of **Base w/ promotion indicator** and **Base w/ direct retraining** is to examine the usefulness of data utilization methods other than HDR. Moreover, by examining a number of HDR variants, we can acquire a better understanding of the efficacy for each HDR component<sup>4</sup>.

## 5 EXPERIMENTAL RESULTS

This section starts with analyzing model performance for all of the compared methods in Section 4.3. After that, we conduct a number of ablation studies to understand the effectiveness of the distribution shift correction algorithm, the TransBlock module, and the automated historical data retrieval process. In the end, we evaluate the online performance of HDR after production deployment.

### 5.1 Overall Performance

Table 2 provides a comparison of the ranking ability and calibration performance for all of the methods listed in Section 4.3. We find that the **Base** model performs poorly on the promotion data, in which the AUC metric is 0.793 and PCOC is only 0.471. This is in line with the declined online performance as illustrated by Figure 1 and, again, demonstrates the necessity to develop a promotion-compatible CVR prediction model for production use.

We further investigate the effectiveness of different delayed feedback modeling techniques. According to Table 2, we identify that same as DEFER (i.e., the **Base** model), other delayed feedback models including **ES-DFM** and **DEFUSE** are still unable to alleviate the deteriorated model performance. On the contrary, after the removal of the delayed feedback module (i.e., **Base w/o DFM**), the **Base** model now outperforms **ES-DFM** and **DEFUSE**. Both the ranking and calibration metrics have increased by a large proportion (despite the PCOC metric remaining at a low level). This illustrates the failure of the underlying assumption for delayed feedback models during sales promotions and calls for the development of novel techniques to deal with the fluctuation of conversion behaviors.

In considering the uniqueness of the promotional conversion patterns, **HDR** takes a data reuse and fine-tuning approach. However, there might also be other simpler ways for promotional data utilization. Here, we evaluate the usefulness of two additional methods as mentioned in Section 4.3, which are **Base w/ promotion indicator** and **Base w/ direct retraining**. Again, from Table 2, we find that both of them produce on-par performances as the **Base** model. The introduction of a binary promotion indicator does not help much, which might be due to the forgetting of previous

<sup>3</sup>We put the computation of these three calibration metrics in Appendix B.

<sup>4</sup>We put the details of implementation in Appendix C.



**Table 2: A comparison of AUC, PCOC, Logloss, and ECE performance for different experiments.**

Methods	AUC	PCOC	Logloss	ECE
Base	0.793	0.471	0.143	0.0212
ES-DFM	0.798	0.441	0.149	0.0245
DEFUSE	0.806	0.452	0.145	0.0151
Base w/o DFM	0.826	0.539	0.103	0.0120
Base w/ promotion indicator	0.801	0.479	0.147	0.0191
Base w/ direct retraining	0.779	0.718	0.156	0.0241
<b>HDR</b>	<b>0.875</b>	<b>0.990</b>	<b>0.090</b>	<b>0.0010</b>
w/o DSC	0.874	0.803	0.091	0.0051
w/o TransBlock	0.771	0.861	0.167	0.0232

**Table 3: Model performance over different values of the learning rate  $\eta_2$ . Hereafter,  $\eta_2$  is set to  $1e^{-5}$  for production models.**

Setting	AUC	PCOC	Logloss	ECE
$\eta_2 = 0$	0.866	0.953	0.094	0.0015
$\eta_2 = 1e^{-5}$	0.875	0.990	0.090	0.0010
$\eta_2 = 1e^{-4}$	0.842	1.012	0.085	0.0023

promotional conversion patterns in continuous online learning. In the meantime, direct retraining of the historical data also fails to achieve a good performance. In fact, the AUC even drops by 2.8%. This is due to the one-epoch overfitting problem as discussed in Zhang et al. [40]. Overall, this again demonstrates the necessity to develop more proper historical data reuse mechanisms.

Our proposed **HDR** approach actually provides such a mechanism. As reported in Table 2, it outperforms all of the baseline methods in terms of both ranking ability and calibration performance. Specifically, HDR outperforms the Base method by 10.3% on AUC, and the PCOC is 0.99 which almost reaches the ideal value of 1.0. Meanwhile, it also achieves the best performance for Logloss and ECE. They all demonstrate the effectiveness of HDR.

## 5.2 Ablation Study

To further examine the usefulness of each module in HDR, we conduct the below ablation studies in this section.

**5.2.1 The Effect of Distribution Shift Correction (DSC).** To understand the influence of DSC, we experiment with the **HDR w/o DSC** approach by removing the DSC module. This is achieved by setting  $\mathcal{B}_h(y)/\mathcal{B}'_h(y)$  to 1.0 in Equation 6. As shown in Table 2, we acquire a roughly identical AUC score as the **HDR** model. However, this leads to a significant drop in calibration performance – there is a nearly 20% decline in PCOC, and the ECE also increases by a large extent (since ECE measures the error, a larger value indicates worse performance). This illustrates the importance of the DSC module in improving calibration performance.

**5.2.2 The Effect of TransBlock.** We also experiment with the **HDR w/o TransBlock** approach by removing the TransBlock module. More concretely, this method still utilizes the retrieved historical data for model fine-tuning, and re-weights the loss function through DSC, whereas the purposefully-designed TransBlock probability in Equation 9 and 11 is discarded for loss computation. As reported in

**Table 4: Case studies for the historical data retrieval results.**

	Date		CVR	Similarity
Target	2022/09/06	9.9 Sales	<b>x</b>	-
Top Two Dates	2022/08/08	8.8 Sales	0.82 <b>x</b>	0.9923
	2022/06/14	6.18 Sales	0.84 <b>x</b>	0.9919
Randomly-picked Date	2022/05/28	No Sales	0.58 <b>x</b>	0.9212
Target	2022/08/08	8.8 Sales	<b>y</b>	-
Top Two Dates	2022/07/12	Summer Sales	0.97 <b>y</b>	0.9937
	2022/07/31	Qixi Sales	0.83 <b>y</b>	0.9910
Randomly-picked Date	2022/05/28	No Sales	0.71 <b>y</b>	0.8902

\* For the purpose of data anonymity, we only report the relative CVR values.

Table 2, we find a significant drop in model performance compared to HDR, which illustrates the usefulness of TransBlock once more.

As mentioned in Section 3.4, during fine-tuning, we adopt two separate learning rates:  $\eta_1$  for TransBlock and  $\eta_2$  for the original main model. In this section, we set  $\eta_1$  to  $1e^{-3}$ , and examine the model performance over different values of  $\eta_2$ . In Table 3, we observe a sub-optimal model performance when  $\eta_2$  is not small enough. This is due to the one-epoch issue as mentioned in Zhang et al. [40]. However, a complete freeze of the original model parameters (i.e.,  $\eta_2 = 0$ ) is not the best choice. Thus, the selection of a proper learning rate is another important step for fine-tuning.

**5.2.3 The Effect of Historical Data Retrieval.** As shown in Table 4, we provide several case studies to better illustrate the accuracy of historical data retrieval as well. The first one is to find similar promotions for 9.9 Sales (on the date of 2022/09/06 and with a CVR of **x**). The top two dates we retrieved are 8.8 Sales on 2022/08/08 with a CVR of 0.82**x**, and 6.18 Sales on 2022/06/14 with a CVR of 0.84**x**. The second one is to seek similar promotions for 8.8 Sales (on the date of 2022/08/08 and with a CVR of **y**). The top two retrieved results are Summer Sales on 2022/07/12 with a CVR of 0.97**y**, and Qixi Sales on 2022/07/31 with a CVR of 0.83**y**. In addition to the top two results, we also randomly pick a date with a much lower similarity score. Clearly, the overall CVR of such a random date differs from the target promotion date by a very large extent. This proves the effectiveness of the data retrieval process in HDR.

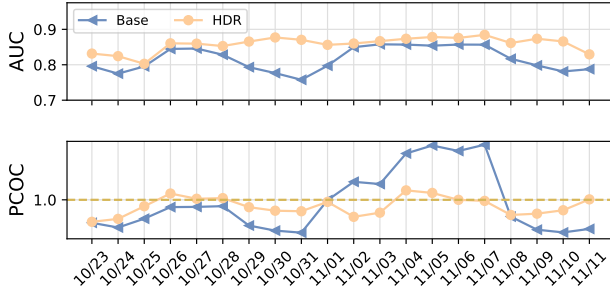
## 5.3 Production Deployment

HDR has been deployed on the display advertising system of Alibaba, serving the main traffic during the Double 11 Sales in 2022. Here, we provide the details of our production deployment.

**5.3.1 System Deployment.** In the historical data retrieval stage, we create a daily map-reduce task to build a vector representation for each day. After that, we compute the cosine similarity between the present day and each of the historical dates based on their vector representations and then select the top two dates for fine-tuning. Since we also adopt several features computed from user behaviors in the first 10 hours of the present day, this stage usually completes around 10:30 am each day.

During fine-tuning, we need to calculate the importance weight  $\frac{\mathcal{B}_h(y)}{\mathcal{B}'_h(y)}$  for the loss function in Equation 6. In practice,  $\mathcal{B}'_h(y)$  is estimated from the historical data, and  $\mathcal{B}_h(y)$  can be calculated with Equation 7. As for  $\mathcal{B}_h(y)$ , we need to access the data for both  $\mathcal{B}_h(\hat{y})$  and  $\mathcal{B}'_h(\hat{y}|y)$ . The former is obtained through parsing the





**Figure 7: A comparison of the model performance (AUC and PCOC) between Base and HDR with online A/B testing.**

real-time prediction log, and the latter is collected by running a batch prediction script to score the historical data with the same prediction model. Note that model fine-tuning is conducted after the data retrieval stage; thereby, the fine-tuned model is usually accessible for online serving at around 11:00 am each day.

**5.3.2 Ranking and Calibration Performance.** We first compare the model performance between HDR and Base through online A/B testing. Here, we adopt both the AUC and PCOC metrics to measure the ranking and calibration performance, respectively. As shown in Figure 7, throughout the whole sales promotion, HDR exhibits consistently superior performance over the Base model in terms of both AUC and PCOC. More importantly, it nearly resolves the problem of substantial performance drop around the promotion dates (e.g. 10/24, 10/31, and 11/10). With the proposed HDR approach, the PCOC and AUC curves have become much flatter.

To further understand the model performance in different product categories, we compute the corresponding PCOC metrics. According to Table 5, we observe severe underestimation of the Base model in the listed categories, in which the predicted CVR is only 1/7 of the actual CVR. HDR effectively tackles this problem by utilizing the historical promotion data, and PCOC has largely improved to 0.7. Note that the CVR prediction of sports shoes and sportswear, which are seriously underestimated by the Base model, now performs as well as other product categories. This proves that our method is capable of capturing the category difference.

**5.3.3 Production Metrics.** HDR has been deployed in the display advertising system of Alibaba. During the Double 11 Sales of 2022, we conduct a strict online A/B test to validate the effectiveness of HDR. **Compared to the production baseline, HDR achieves a 9% increase in RPM.** Such a significant improvement mainly comes from the optimization of the calibration ability, which allows the online advertising system to measure the value of production traffic in a more accurate way. **In the meantime, there is a 16% increase in CVR and an 11% increase in ROI,** which are largely ascribed to the improved ranking ability (i.e. AUC).

**5.3.4 Discussion.** One potential drawback for HDR is that the reuse of historical data may encounter the problem of data obsolescence. Specifically, online advertising systems are dynamic as active advertisers and advertising campaigns are constantly changing; thus, the active advertisements may differ significantly across different promotions [6]. Particularly, during promotion, the surge of marketing

**Table 5: An comparison of the PCOC metrics between Base and HDR over different product categories on 2022/10/31.**

Category	Base	HDR
women’s clothing	0.139	0.695
children’s clothing	0.135	0.721
sports shoes	0.054	0.735
women’s shoes	0.152	0.699
sportswear	0.071	0.743

demands also brings lots of new campaigns that are impossible to observe from history. Models fine-tuned on outdated historical data may be sub-optimal or even harmful to online performance.

However, in practice, we do not observe any sign of counterproductive performance. This might be attributed to the design of the fine-tuning mechanism, in which the original model (with a sparse ID denoting each specific ads campaign) is rarely updated because of the small learning rate, whereas TransBlock (only with a few parameters) learns coarse-grained knowledge, such as category-level difference, for data adaptation. Such a phenomenon is worth further exploration for a better understanding of the historical data reuse mechanism, and we will take it into account in the future.

## 6 CONCLUSION

In this work, we propose a novel data reuse mechanism, named HDR, to tackle the complicated CVR prediction problem during sales promotions. HDR consists of three components: an automated data retrieval module that seeks the most similar data from historical promotions, a distribution shift correction module that re-weights the retrieved data for better aligning with the target promotion, and a TransBlock module that quickly fine-tunes the original model for better adaptation to the promotion mode. Note that HDR only concentrates on fine-tuning the production model with historical promotion data whereas the original online learning process keeps intact. Experiments conducted on top of real-world datasets demonstrate the effectiveness of HDR, as it improves both ranking and calibration metrics to a large extent. HDR has also been deployed on the display advertising system in Alibaba, bringing a lift of 9% RPM and 16% CVR during Double 11 Sales in 2022.

In light of the great performance of HDR, we would like to explore other occasions that are also suitable for data reuse. For instance, we identify that conversion also differs between weekdays and weekends, middle-of-month and end-of-month, etc. Moreover, we believe that the data reuse mechanism is not limited to the historical data, we can also explore utilizing the data from other scenes. In the end, we would like to develop a comprehensive data-centric mechanism with a unified data utilization technique.

## ACKNOWLEDGEMENTS

This work was supported by Alibaba Group through Alibaba Research Intern Program, the National Natural Science Foundation of China (No.U2133218), the National Key Research and Development Program of China (No.2018YFB0204304) and the Fundamental Research Funds for the Central Universities of China (No.FRF-MP-19-007 and No. FRF-TP-20-065A1Z).

## REFERENCES

- [1] Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. 2020. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*. PMLR, 222–232.
- [2] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. 2019. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734* (2019).
- [3] Yong Bai, Yu-Jie Zhang, Peng Zhao, Masashi Sugiyama, and Zhi-Hua Zhou. 2022. Adapting to Online Label Shift with Provable Guarantees. *arXiv preprint arXiv:2207.02121* (2022).
- [4] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, et al. 2022. CAN: feature co-action network for click-through rate prediction. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 57–65.
- [5] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2009. Discriminative learning under covariate shift. *Journal of Machine Learning Research* 10, 9 (2009).
- [6] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1097–1105.
- [7] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. 2022. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 295–305.
- [8] Yu Chen, Jiaqi Jin, Hui Zhao, Pengjie Wang, Guojun Liu, Jian Xu, and Bo Zheng. 2022. Asymptotically Unbiased Estimation for Delayed Feedback Modeling via Label Correction. In *Proceedings of the ACM Web Conference 2022*. 369–379.
- [9] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungchul Yun. 2022. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 440–458.
- [10] Siyu Gu, Xiang-Rong Sheng, Ying Fan, Guorui Zhou, and Xiaoqiang Zhu. 2021. Real Negatives Matter: Continuous Training with Real Negatives for Delayed Feedback Modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2890–2898.
- [11] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. 2006. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems* 19 (2006).
- [12] Siguang Huang, Yunli Wang, Lili Mou, Huayue Zhang, Han Zhu, Chuan Yu, and Bo Zheng. 2022. MBCT: Tree-Based Feature-Aware Binning for Individual Uncertainty Calibration. In *Proceedings of the ACM Web Conference 2022*. 2236–2246.
- [13] Zhigang Huangfu, Gong-Duo Zhang, Zhengwei Wu, Qintong Wu, Zhiqiang Zhang, Lihong Gu, Jun Zhou, and Jinjie Gu. 2022. A Multi-Task Learning Approach for Delayed Feedback Modeling. In *Companion Proceedings of the Web Conference 2022*. 116–120.
- [14] Biye Jiang, Chao Deng, Huimin Yi, Zelin Hu, Guorui Zhou, Yang Zheng, Sui Huang, Xinyang Guo, Dongyue Wang, Yue Song, et al. 2019. Xdl: an industrial deep learning framework for high-dimensional sparse data. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–9.
- [15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [16] Sofia Ira Ktena, Alykhan Tejani, Lucas Theis, Pranay Kumar Myana, Deepak Dilipkumar, Ferenc Huszár, Steven Yoo, and Wenzhe Shi. 2019. Addressing delayed feedback for continuous training with neural networks in CTR prediction. In *Proceedings of the 13th ACM conference on recommender systems*. 187–195.
- [17] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. 2018. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*. PMLR, 3122–3130.
- [18] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. 2021. TTT++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems* 34 (2021), 21808–21820.
- [19] Qiang Luo, Xiang-Rong Sheng, Jun Jiang, Chi Ma, Shuguang Wang, Haiyang He, Pengtao Yi, Guowang Zhang, Yue Song, Guorui Zhou, et al. 2021. What Do We Need for Industrial Machine Learning Systems? Bernoulli, A Streaming System with Structured Designs. (2021).
- [20] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.
- [21] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofu Chen, Shijian Zheng, Peilin Zhao, and Minghui Tan. 2022. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*. PMLR, 16888–16905.
- [22] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. 2012. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471* (2012).
- [23] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.
- [24] Xiang-Rong Sheng, Jingyue Gao, Yueyao Cheng, Siran Yang, Shuguang Han, Hongbo Deng, Yuning Jiang, Jian Xu, and Bo Zheng. 2022. Joint Optimization of Ranking and Calibration with Contextualized Hybrid Model. *arXiv preprint arXiv:2208.06164* (2022).
- [25] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, and Xiaoqiang Zhu. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. *arXiv:2101.11427* [cs.LG]
- [26] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.
- [27] H. Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90, 2 (2000), 227–244.
- [28] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726* (2020).
- [29] Yanshi Wang, Jie Zhang, Qing Da, and Anxiang Zeng. 2020. Delayed feedback modeling for the entire space conversion rate prediction. *arXiv preprint arXiv:2011.11826* (2020).
- [30] Hong Wen, Jing Zhang, Fuyu Lv, Wentian Bao, Tianyi Wang, and Zulong Chen. 2021. Hierarchically modeling micro and macro behaviors via multi-task learning for conversion rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2187–2191.
- [31] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2377–2386.
- [32] Kailun Wu, Zhangming Chan, Weijie Bian, Lejian Ren, Shiming Xiang, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. Adversarial Gradient Driven Exploration for Deep Click-Through Rate Prediction. *arXiv:2112.11136* [cs.LG]
- [33] Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q Weinberger. 2021. Online adaptation to label distribution shift. *Advances in Neural Information Processing Systems* 34 (2021), 11340–11351.
- [34] Zixuan Xu, Penghui Wei, Weimin Zhang, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. UKD: Debiasing Conversion Rate Estimation via Uncertainty-regularized Knowledge Distillation. In *Proceedings of the ACM Web Conference 2022*. 2078–2087.
- [35] Le Yan, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Scale Calibration of Deep Ranking Models. In *KDD*. 4300–4309.
- [36] Jiaqi Yang and De-Chuan Zhan. 2022. Generalized Delayed Feedback Model with Post-Click Information in Recommender Systems. *Advances in Neural Information Processing Systems* 35 (2022), 26192–26203.
- [37] Jia-Qi Yang, Xiang Li, Shuguang Han, Tao Zhuang, De-Chuan Zhan, Xiaoyi Zeng, and Bin Tong. 2021. Capturing delayed feedback in conversion rate prediction via elapsed-time sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4582–4589.
- [38] Yujing Zhang, Zhangming Chan, Shuhao Xu, Weijie Bian, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. KEEP: An Industrial Pre-Training Framework for Online Recommendation via Knowledge Extraction and Plugging. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3684–3693.
- [39] Yuanxing Zhang, Langshi Chen, Siran Yang, Man Yuan, Huimin Yi, et al. 2022. PICASSO: Unleashing the Potential of GPU-centric Training for Wide-and-deep Recommender Systems. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE.
- [40] Zhao-Yu Zhang, Xiang-Rong Sheng, Yujing Zhang, Biye Jiang, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. Towards Understanding the Overfitting Phenomenon of Deep Click-Through Rate Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2671–2680.
- [41] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.
- [42] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2759–2769.

**Table 6: An analysis of the conditional input distribution  $p(x|y)$  for 9.9 Sales (in 2022/09) and 3.8 Sales (in 2023/03).**

User Group $\times$ Category	9.9 Sales		3.8 Sales		No Sales	
	$y = 0$	$y = 1$	$y = 0$	$y = 1$	$y = 0$	$y = 1$
$U_1, C_1$	1.1013%	0.3595%	1.1524%	0.3573%	0.0235%	0.0938%
$U_1, C_2$	0.0065%	0.0575%	0.0068%	0.0579%	0.0014%	0.1309%
$U_2, C_1$	0.3441%	0.1597%	0.3861%	0.1601%	0.0039%	0.1109%
$U_2, C_2$	0.1738%	0.0288%	0.1922%	0.0298%	0.0098%	0.1405%

## A SOLUTION FOR $\mathcal{B}_h(y)$

For a short representation, we rewrite Equation 7 in the matrix form as shown in Equation 12.

$$\underbrace{\mathcal{B}_h(\hat{y})}_{\mathbf{M}_{\hat{y}}} = \underbrace{[\mathcal{B}'_h(\hat{y}|y=1) \quad \mathcal{B}'_h(\hat{y}|y=0)]}_{\mathbf{M}'_{\hat{y}|y}} \cdot \underbrace{\begin{bmatrix} \mathcal{B}_h(y=1) \\ \mathcal{B}_h(y=0) \end{bmatrix}}_{\mathbf{M}_y} \quad (12)$$

Since  $\mathbf{M}_y$ , i.e.  $\mathcal{B}_h(y)$ , is the only unknown variable, we can reconstruct an optimization problem as illustrated by Equation 13 and seek the solution for  $\mathcal{B}_h(y)$ . Here, the first term (13.1) corresponds to the solution for Equation 12, and the second term (13.2) introduces a regularization constraint to keep the stability of result prediction. More specifically, we denote  $\mathcal{B}'_h(y)$  as  $\mathbf{M}'_{\hat{y}|y}$  and constrain the estimated  $\mathcal{B}_h(y)$  is close to historical  $\mathcal{B}'_h(y)$ , i.e.,  $\frac{\mathcal{B}_h(y)}{\mathcal{B}'_h(y)} \rightarrow 1$ . In this way, we could regulate the influence of importance weight on the fine-tuning process.

$$\arg \min_{\mathbf{M}_y} \underbrace{\|\mathbf{M}'_{\hat{y}|y} \cdot \mathbf{M}_y - \mathbf{M}_{\hat{y}}\|^2}_{(13.1)} + \lambda \cdot \underbrace{\|\mathbf{M}'_{\hat{y}|y} - \mathbf{M}_y\|^2}_{(13.2)} \quad (13)$$

The closed-form solution for  $\mathcal{B}_h(y)$  is provided in Equation 14. Placing it back to Equation 6, we acquire our final loss function.

$$\mathbf{M}_y = \left( \mathbf{M}'_{\hat{y}|y}{}^T \mathbf{M}'_{\hat{y}|y} + \lambda \mathbf{I} \right)^{-1} \left( \mathbf{M}'_{\hat{y}|y}{}^T \mathbf{M}_{\hat{y}} + \lambda \mathbf{M}'_{\hat{y}|y} \right) \quad (14)$$

## B COMPUTATION OF CALIBRATION METRICS

Let  $\hat{p}_i$  denote the predicted conversion rate for the  $i$ -th sample with a conversion label  $y_i$ , and  $N$  stands for the total number of data samples. With these notations, the computation of LogLoss and PCOC can be expressed with the below Equations. In terms of ECE, we first sort data samples by their predicted probabilities and divide them into  $K$  buckets, each containing approximately the same number of samples. Then, we compute the prediction error for each bucket and further aggregate the errors of all of the buckets. In the below Equation,  $\mathbb{1}(\hat{p}_i \in B_k)$  refers to a binary indicator with the value of 1 if the predicted probability locates in the  $k$ -th bucket  $B_k$ , and 0 otherwise.

$$\begin{aligned} \text{LogLoss} &= -\frac{1}{N} \sum_{i=1}^N (y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)); \\ \text{PCOC} &= \frac{\sum_{i=1}^N \hat{p}_i}{\sum_{i=1}^N y_i}; \quad \text{ECE} = \frac{1}{N} \sum_{k=1}^K \left| \sum_{i=1}^N (y_i - \hat{p}_i) \mathbb{1}(\hat{p}_i \in B_k) \right| \end{aligned} \quad (15)$$

## C IMPLEMENTATION DETAILS

In practice, we keep the original online learning process intact while only focusing on the model fine-tuning. More specifically, we adopt a single-layer fully-connected neural network for TransBlock<sup>5</sup>, and the number of hidden units is set to [100, 2]. During fine-tuning, we adopt the Adam optimizer [15] with a learning rate of 1e-3 and a batch size of 5,000 for TransBlock; parameters of the main model are also updated with the Adam optimizer but using a much smaller learning rate of 1e-5. In the meantime, the regularization strength  $\lambda$  in Equation 14 is set to 1.0. All of our experimental models are trained with the XDL platform [14, 39].

## D ILLUSTRATION FOR VECTOR-BASED REPRESENTATION IN RETRIEVAL

We give an example to illustrate our feature engineering effort. Suppose that a click happens on 2022/09/03, we can then compute the one-day conversion with the purchase data collected in [09/03], the two-day conversion with the data collected in [09/03, 09/04], and so forth. To retrieve similar historical promotion data for 2022/09/06, we utilize the below features in this paper:

(1). One-day CVR for clicks happen on 2022/09/05; (2). One-day CVR and Two-day CVR for clicks happen on 2022/09/04; (3). One-day CVR, Two-day CVR and Three-day CVR for clicks happen on 2022/09/03; (4). The impression ratios of representative categories (such as Toddler, Cosmetics and etc) from 2022/09/03 to 09/05, respectively; (5). The impression ratios of representative categories in the first 10 hours in 2022/09/06.

We utilize all of the above features to build a vector-based representation and retrieve historically similar promotion data for the target promotion date 2022/09/06.

## E MORE ANALYSIS ABOUT THE PRESUMPTIONS

It is notable that the two presumptions are derived from large-scale data analysis and we find that they hold in almost every promotion in our system. Here, We provide more analysis to improve the credibility of our presumptions. In Table 6, we provide more statistics from the most recent 3.8 Sales (in 2023/03) as well as the 9.9 Sales from the last year. We can see that Presumption I holds for both of them, e.g.  $p(U_1, C_1|y=0)$  in 9.9 Sales and 3.8 Sales are both around 1.1%. Moreover, on a normal day without any sales events (No Sales), such a probability has a large deviation from 1.1%.

As for Presumption II, we also conduct a similar large-scale analysis and do observe a common pattern as illustrated by Figure 4, in almost every sales promotion. The rhythm of sales promotions in our system is generally aligned, thereby, we propose Presumption II to simplify our problem. When it is not valid, we have the following alternatives. In practice, we observe that user behaviors are highly correlated to the time difference between the current time and the promotion start time. Therefore, we can calculate  $B_h(y)/B'_h(y)$  by aligning the time interval instead of aligning it to an absolute time point. Another alternative is to increase model update frequency. For example, we may calculate  $B_h(y)/B'_h(y)$  on an hourly basis, which allows us to capture changes more frequently and accurately.

<sup>5</sup>We provide a demo in <https://github.com/MingMTC/HDR>.