

Asymptotically Unbiased Estimation for Delayed Feedback Modeling via Label Correction

Yu Chen[†], Jiaqi Jin[†], Hui Zhao[‡], Pengjie Wang, Guojun Liu, Jian Xu and Bo Zheng*

Alibaba Group

{shuyuan.cy, jinjiaqi.jjq, shuqian.zh, pengjie.wpj, guojun.liugj, xiyu.xj, bozheng}@alibaba-inc.com

ABSTRACT

Alleviating the delayed feedback problem is of crucial importance for the conversion rate(CVR) prediction in online advertising. Previous delayed feedback modeling methods using an observation window to balance the trade-off between waiting for accurate labels and consuming fresh feedback. Moreover, to estimate CVR upon the freshly observed but biased distribution with fake negatives, the importance sampling is widely used to reduce the distribution bias. While effective, we argue that previous approaches falsely treat fake negative samples as real negative during the importance weighting and have not fully utilized the observed positive samples, leading to suboptimal performance.

In this work, we propose a new method, **DE**layed Feedback modeling with **Un**biaSed Estimation, (**DEFUSE**), which aim to respectively correct the importance weights of the immediate positive, the fake negative, the real negative, and the delay positive samples at finer granularity. Specifically, we propose a two-step optimization approach that first infers the probability of fake negatives among observed negatives before applying importance sampling. To fully exploit the ground-truth immediate positives from the observed distribution, we further develop a bi-distribution modeling framework to jointly model the unbiased immediate positives and the biased delay conversions. Experimental results on both public and our industrial datasets validate the superiority of DEFUSE. Codes are available at <https://github.com/ychen216/DEFUSE.git>.

CCS CONCEPTS

• Information systems → Computational advertising.

KEYWORDS

Delayed Feedback, Online Advertising, CVR prediction

ACM Reference Format:

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 (WWW '22)*, April 25–29, 2022, Virtual Event, Lyon, France. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3485447.3511965>

[†]Co-first authorship. [‡] This author is the one who gives a lot of guidance in the work
*Corresponding author.

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

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9096-5/22/04...\$15.00

<https://doi.org/10.1145/3485447.3511965>

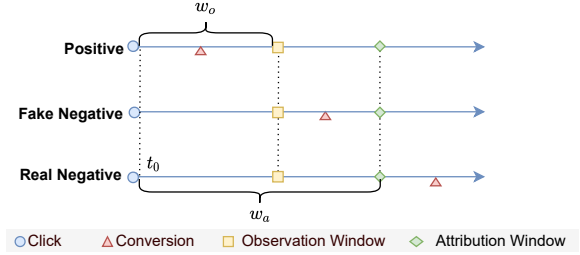


Figure 1: An illustration of different label types. The observation window w_o denotes the least time interval between the click time and the streaming training time; while the attribution window w_a determine the actual label.

1 INTRODUCTION

Online advertising has become the primary business model for intelligent e-commerce, which helps advertisers target potential customers [3, 4, 14]. Generally, cost per action (CPA) and cost per click (CPC) are two widely used payment options, which directly influence both revenue of the platform and the return-on-investment (ROI) of all advertisers. As a fundamental part of both kinds of price bidding, conversion rate(CVR) prediction, which focuses on ROI-oriented optimization, always keeps an irreplaceable component to ensure a healthy advertising platform [12].

As a widely used training framework, streaming learning, which continuously finetunes the model according to real-time feedback, has shown promising performance in click-through rate(CTR) prediction tasks [13, 18, 21, 22]. However, as shown in Table 1, it is non-trivial to achieve better results via streaming learning due to the pervasively delayed and long-tailed conversion feedback for CVR prediction. More specifically, as illustrated in Figure 1, a click that happened at time t_0 needs to wait for a sufficiently long attribution window w_a to determine its actual label — only samples convert before $t_0 + w_a$ are labeled as positive. Typically, the setting for w_a ranges from one day to several weeks for different business scenarios. The issue, even for an attribution window that as short as one day is still too long to ensure sample freshness, which remains a major obstacle for achieving effective streaming CVR prediction.

To solve this challenge, existing efforts focus on introducing a much shorter observation window w_o , e.g., 30min, allowing clicks with observed labels to be collected and distributed to the training pipeline right after $t_0 + w_o$. Optimizing w_o provides the ability to balance the trade-off between utilizing more fresh samples and accepting less accurate labels. This greatly improves sample freshness, with acceptable coverage of conversions within the observation window, at the cost of temporarily marking feedback with long

Table 1: The distribution for delay conversion on Taobao and Criteo datasets, where Inc and Acc respectively denote the incremental and accumulate proportion.

Datasets	Taobao Dataset		Criteo	
delay interval	Inc (%)	Acc (%)	Inc (%)	Acc (%)
<30min	61	61	42	42
30min - 12hour	13	74	14	56
12hour - 1day	4	78	5	61
1day - 3day	7	85	10	71
3day - 7day	6	91	10	81
7day - 30day	9	100	19	100

delay as *fake negative*. Thus, current works mainly focus on making CVR estimations upon the freshly observed but biased distribution with fake negatives.

Since it is hard to achieve unbiased estimation by using standard binary classification loss, e.g. cross-entropy, current efforts implement various auxiliary tasks to model conversion delay, so as to alleviate the bias caused by the fake negatives. Early methods [2, 29] attempts to address the delayed feedback problem by jointly optimizing CVR prediction with a delay model that predicts the delay time d from an assumed delay distribution. However, these approaches are directly trained on the biased observed distribution and have not fully utilized the rare and sparse delayed positive feedback. Having realized such drawbacks, recent studies mainly focus on reusing the delayed conversions as positive samples upon conversion. Various sample duplicating mechanisms have been designed to fully exploit each conversion. For instance, FNC/FNW [10] set $w_o = 0$ and re-sent all positive samples when conversion arrives. ES-DFM [27] only duplicate delay positive samples which previously have been incorrectly labeled as fake negatives; While DEFER [5] reuse all samples with actual label after completing the label attribution to maintain equal feature distribution and to utilize the real negative samples. Moreover, to bridge the distribution bias, importance sampling [1] is adopted to correct the disparity between the ground-truth and the observed but biased distribution.

Despite effectiveness, we argue that current methods still have some limitations. Firstly, they mainly focus on designing appropriate training pipelines to reduce the bias in feature space and only weight the loss of observed positives and negatives through importance sampling. The issue is, observed negatives may potentially be fake negatives, and these methods falsely treat them as real negatives, leading to sub-optimal performance. Second, observed positives can be further divided into *immediate positives*(IP) and *delay positives*(DP), which implies two potential improvements: (1) Intuitively, IPs and DPs contribute differently to the CVR model due to duplication. (2) By excluding DPs, an unbiased estimation of IP prediction can be established directly based on the observed dataset consistent with the actual distribution of IPs.

In this paper, We propose **DE**layed Feedback modeling with **U**nbiased Estimation (DEFUSE) for streaming CVR prediction, which investigates the influence of fake negatives and makes full use of DPs on importance sampling. Distinct from previous methods only

modelling observed positives and negatives, we formally recognize the samples into four types, namely immediate positives(IP), fake negatives(FN), real negatives(RN), and delay positives(DP). Since FNs are adopted in the observed negatives, we propose a two-step optimization, which firstly infers the probability of observed negatives being fake negatives before performing unbiased CVR prediction via importance sampling on each of the four types of samples. Moreover, we design a bi-distribution framework to make full use of the immediate positives. Comprehensive experiments show DEFUSE achieves better performance than the state-of-the-art methods on both public and industrial datasets.

Our main contributions can be summarized as follows:

- We emphasize the importance of dividing observed samples in a more granular manner, which is crucial for accurate importance sampling modeling.
- We proposed an unbiased importance sampling method, DEFUSE, with two-step optimization to address the delayed feedback issue. Moreover, we implement a bi-distribution modeling framework to fully exploit immediate positives during streaming learning.
- We conduct extensive experiments on both public and industrial datasets to demonstrate the state-of-the-art performance of our DEFUSE.

2 RELATED WORK

2.1 Delayed Feedback Models

Learning with delayed feedback has received considerable attention in the studies of predicting conversion rate (CVR). Chapelle [2] assumed that the delay distribution is exponential and proposed two generalized linear models for predicting the CVR and the delay time, respectively. However, such a strong hypothesis may be hard to model the delay distribution in practice. To address this issue, [29] proposed a non-parametric delayed feedback model for CVR prediction, which exploits the kernel density estimation and combines multiple Gaussian distributions to approximate the actual delay distribution. Moreover, several recent works [23, 25] discretize the delay time by day slot to achieve fine-grain survival analysis for delayed feedback problem. However, one significant drawback of the above methods was that all of them only attempted to optimize the observed conversion information rather than the actual delayed conversion, which cannot fully utilize the sparse positive feedback.

2.2 Unbiased CVR Estimation

Distinct from previous methods, current mainstream approaches employ the importance sampling method to estimate the real expectation *w.r.t* another observed distribution [5, 10, 27, 28]. Ktena et al. [10] assumes that all samples are initially labeled as negative, then duplicate samples with a positive label and ingest them to the training pipeline upon their conversion. To further model CVR prediction from the biased distribution, they propose two fake negative weighted(FNW) and fake negative calibration(FNC) utilizing importance sampling [1]. However, it only focuses on the time-liness of samples and neglects the accuracy of labels. To address this, ES-DFM [12] introduces an observation window to study the trade-off between waiting for more accurate labels in the window and exploiting fresher training data out of the window. Gu et al. [5]

Table 2: Main notations used in the paper.

Notation	Explanation
\mathbf{x}, y, d	the input features, label, and interval between click and pay time
$p(\mathbf{x}, y), q(\mathbf{x}, y)$	the ground-truth and observed distribution
w_o, w_a	the observation and attribution window
$w_i(\mathbf{x}, y), z(\mathbf{x})$	the importance weight and hidden variable of \mathbf{x}

further duplicate the real negative and sparse positive in the observation window to eliminate the feature distribution bias introduced by duplicating delayed positive samples.

2.3 Delay Bandits

The delayed feedback has attracted much attention in bandit methods [17, 20, 20]. Previous methods consider the delayed feedback modeling as a sequential decision-making problem and maximum the long-term rewards [6, 8, 24]. Joulani et al. [8] provided meta-algorithms that transform algorithms developed for the non-delayed, and analyzed the effect of delayed feedback in streaming learning problems. [24] provided a stochastic delayed bandit model and proof the algorithms in censored and uncensored settings under the hypothesis that the delay distribution is known. [6] tried to examine the bandit streaming learning in games with continuous action spaces and introduced a gradient-free learning policy with delayed rewards and bandit feedback.

3 PRELIMINARY

In this section, we first formulate the problem of streaming CVR prediction with delayed feedback. Then we give a brief introduction to the standard importance sampling algorithms used in the previous methods. The notations used throughout this paper are summarized in Table 2.

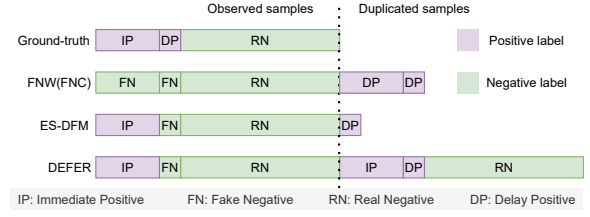
3.1 Problem Formulation

In a standard CVR prediction task, the input can be formally defined as $(\mathbf{x}, y) \sim p(\mathbf{x}, y)$, where \mathbf{x} denotes the features and $y \in \{0, 1\}$ is the conversion label. A generic CVR prediction model aims to learn the parameters θ of the binary classifier function f by optimizing following ideal loss [12, 14]:

$$\mathcal{L}_{ideal} = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} \ell(y, f_{\theta}(\mathbf{x})), \quad (1)$$

where (\mathbf{x}, y) is the training samples drawn from the ground-truth distribution $p(\mathbf{x}, y)$, and ℓ denotes the classification loss, e.g., the widely used cross-entropy loss. However, as mentioned above, due to the introduction of the observation window, the clicks with conversions that happened outside the observation window will firstly be treated as fake negatives. Thus, the observation distribution $q(\mathbf{x}, y)$ is always biased from the ground-truth distribution $p(\mathbf{x}, y)$. More specifically, as illustrated in Figure 1, there are four types of samples in the online advertising system:

- Immediate Positive(IP), e.g., $d < w_o$. The samples convert inside the observation window are labeled as immediate positive.

**Figure 2: Data Distribution of previous online delayed feedback methods.**

- Fake Negative(FN), e.g., $w_o < d < w_a$. Fake negative denotes samples that incorrectly labeled as negative at training time due to delay conversion.
- Real Negative(RN), e.g., $d > w_a/d = \infty$. The samples not convert after waiting a sufficient long attribution window w_a are labeled as real negative.
- Delay Positive(DP). These samples are duplicated and investigated into the training pipeline with a positive label upon conversion.

3.2 Importance Sampling

Importance sampling has been widely studied and applied in many recent tasks, e.g., counterfactual learning [26] and unbiased estimation [5, 27]. Typically, previous approaches use importance sampling to estimate the expectation of training loss from the observed distribution and rewrite the ideal CVR loss function as follows:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} \ell(y, f_{\theta}(\mathbf{x})) \quad (2)$$

$$= \mathbb{E}_{(\mathbf{x}, y) \sim q(\mathbf{x}, y)} w(\mathbf{x}, y) \ell(y, f_{\theta}(\mathbf{x})), \quad (3)$$

where f_{θ} is the desired CVR model pursuing unbiased CVR prediction, $p(\mathbf{x}, y)$ and $q(\mathbf{x}, y)$ respectively denote the joint density function of the *ground truth* and the *observed* and *duplicated* distribution, and $w(\mathbf{x}, y)$ is the likelihood ratio of the *ground truth* distribution with respect to the *observed* and *duplicated* distribution introduced by importance sampling, chasing for an unbiased $f_{\theta}^*(\mathbf{x})$.

Currently, by assuming or ensuring $p(\mathbf{x}) \approx q(\mathbf{x})$ and carefully designing the sample duplicating mechanisms, all published methods [5, 10, 27] apply the same derivation of the formulation of $w(\mathbf{x}, y)$ that firstly published at [10] as follows:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}, y) \sim q(\mathbf{x}, y)} w(\mathbf{x}, y) \ell(y, f_{\theta}(\mathbf{x})) \quad (4)$$

$$= \int q(\mathbf{x}) d\mathbf{x} \int q(y|\mathbf{x}) \frac{p(\mathbf{x}, y)}{q(\mathbf{x}, y)} \ell(\mathbf{x}, y; f_{\theta}(\mathbf{x})) dy \quad (5)$$

$$= \int q(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \int q(y|\mathbf{x}) \frac{p(y|\mathbf{x})}{q(y|\mathbf{x})} \ell(\mathbf{x}, y; f_{\theta}(\mathbf{x})) dy \quad (6)$$

$$\approx \int q(\mathbf{x}) d\mathbf{x} \int q(y|\mathbf{x}) \frac{p(y|\mathbf{x})}{q(y|\mathbf{x})} \ell(\mathbf{x}, y; f_{\theta}(\mathbf{x})) dy \quad (7)$$

$$\approx \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \left[y_i \frac{p(y_i=1|\mathbf{x}_i)}{q(y_i=1|\mathbf{x}_i)} \log f_{\theta}(\mathbf{x}_i) + (1-y_i) \frac{p(y_i=0|\mathbf{x}_i)}{q(y_i=0|\mathbf{x}_i)} \log(1-f_{\theta}(\mathbf{x}_i)) \right], \quad (8)$$

where \mathcal{D} is the observed dataset. The difference among these published methods mainly lies in:

Table 3: The label disparity of observed distribution and actual distribution after attribution in terms of four sample types. The observability denotes whether the model can distinguish the sample type during streaming training.

	IP	FN	RN	DP
observability	True	False	False	True
observed label	1	0	0	1
attributed label	1	1	0	1

- (1) Different designs of the training pipeline, e.g. choice of w_o and the definition of duplicated samples as illustrated in Figure 2, which eventually results in different formulations of $q(y|x)$
- (2) Different choices of modelling $p(d > w_o | x, y = 1)$ or $p(d > w_o | y = 1)p(y = 1|x)$, etc.

As demonstrated in Figure 2, FNW/FNC [10] firstly sets $w_o = 0$ and marks all clicks as negative samples at click time and all positive samples are collected and replayed as DP at conversion time; ES-DFM [27] and DEFER [5] keep a reasonable observation time w_o , hence clicks with conversions happened within $t_0 + w_o$ can be correctly labelled as IP. The only difference between ES-DFM and DEFER is that ES-DFM only replay delayed positives while DEFER duplicates all clicks(including IP and RN). Both ES-DFM and DEFER choose to model $f_{dp}(x) = p(d > w_o, y = 1 | x) = p(d > w_o | x, y = 1)p(y = 1 | x)$ as a whole. Such differences of these methods in sample duplicating mechanisms eventually result in their different formulation of $q(y|x)$ as shown in equation (9), (10) and (11), respectively.

$$q_{fnw}(y = 0 | x) = \frac{1}{1 + p(y = 1 | x)} \quad (9)$$

$$q_{esdfm}(y = 0 | x) = \frac{p(y = 0 | x) + f_{dp}(x)}{1 + f_{dp}(x)} \quad (10)$$

$$q_{defer}(y = 0 | x) = p(y = 0 | x) + \frac{1}{2}f_{dp}(x), \quad (11)$$

3.2.1 Limitations. Despite their success in bias reduction, we notice that these published methods still failed to achieve unbiased CVR prediction due to a hidden flaw introduced while deriving the formulation of $w(x, y)$. Normally, importance sampling assumes no value modification during the transition from $p(x, y)$ to $q(x, y)$, whereas in CVR prediction as mentioned in Section 3.1, even for the same clicks, observed label from $q(x, y)$ can be temporarily deviated to ground-truth label from $p(x, y)$. More specifically and rigorously, if we distinguish the observed label as v and re-express the biased distribution as $q(x, v)$, we have:

$$y = y(v, d) = \begin{cases} 1, & v = 1 \\ 0, & v = 0, d = +\infty \\ 1, & v = 0, d > w_o. \end{cases} \quad (12)$$

As a result, fake negative samples which should be denoted as $\frac{p(d > w_o, y=1|x)}{q(y=0|x)}$, are falsely treated as real negative in equation (8), leading to suboptimal performance and biased CVR prediction.

4 METHODOLOGY

In this section, we present our proposed method DElayed Feedback modeling with UnbiaSed Estimation (DEFUSE) in detail. We First introduce our correction of unbiased estimation, which respectively weight the importance of four types of samples. We then propose a two-step optimization for DEFUSE. Finally, to further reduce the influence caused by the observed bias distribution, we devise a bi-distribution modeling framework to sufficiently utilize the immediate conversion under the actual distribution. Note that our DEFUSE is applicable for different training pipelines, but for ease of description, we will introduce our approach on top of the design of the training pipeline in ES-DFM.

4.1 Unbiased Delayed Feedback Modeling

As we describe in Section 3.2.1, our goal is to achieve unbiased delayed feedback modeling by further optimizing the loss of the fake negative samples. From Equation (5,12), we can obtain the unbiased estimation as:

$$\mathcal{L}_{ub} = \int q(x) dx \int q(v|x) \frac{p(x)}{q(x)} \frac{p(y(v, d)|x)}{q(y|x)} \ell(x, y(v, d); f_\theta(x)) dv. \quad (13)$$

Where $\ell(x, y(v, d); f_\theta(x))$ is the loss function for observed samples with label $y(v, d)$. Typically, Previous approaches eliminate $\frac{p(x)}{q(x)}$ by assuming $p(x) \approx q(x)$ [10, 27] or design proper training pipeline to guarantee equal feature distribution [5].

4.1.1 Importance weighting of DEFUSE. In this work, different from previous works that focus on the duplicating mechanism, we aim for unbiased CVR estimation by properly evaluating the importance weight for $\ell(x, y(v, d); f_\theta(x))$. As shown in Table 3, the observed samples can be formally divided into four parts. Intuitively, if we have all labels of each part, the Equation (13) can be rewritten as:

$$\mathcal{L}_{ub} = \int q(x) \left[\sum_{v_i} q(v_i|x) w_i(x, y(v_i, d)) \ell(x, y(v_i, d); f_\theta(x)) \right] dx, \quad (14)$$

where $w_i = \frac{p(x, y(v_i, d))}{q(x, v_i)}$ and $i \in \{IP, FN, RN, DP\}$, subject to $\sum v_i = 1$ and $v_i \in \{0, 1\}$. Note that present works merely model the observed positives and negatives in Equation (8), which ignore the impact of fake negatives(FN) and lead to bias in label distribution. To solve equation (14), we first introduce a latent variable z , which is used to inference whether an observed negative is FN or not, then respectively modeling the importance weights w_i of these four types of observed samples. Thus, equation (14) is equivalent to:

$$\begin{aligned} & \min_{\theta} \mathcal{L}_{ub} \\ \Leftrightarrow & \min_{\theta} \int q(x) \left[v(w_{DP} \log f_\theta(x) + \mathbb{I}_{IP}(w_{IP} - w_{DP}) \log f_\theta(x)) \right. \\ & \left. + (1-v)(w_{FN} \log f_\theta(x)z + w_{RN} \log(1 - f_\theta(x))(1-z)) \right] dx \end{aligned} \quad (15)$$

s.t.

$$\begin{aligned} w_{IP}(x) &= w_{RN}(x) = 1 + f_{dp}(x) \\ w_{DP}(x) + w_{FN}(x) &= 1 + f_{dp}(x), \end{aligned}$$

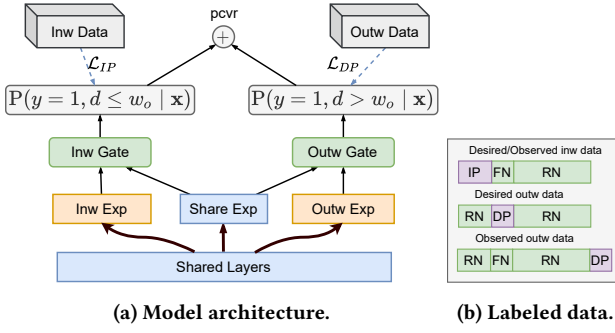


Figure 3: An illustration of Bi-Distribution model, where Exp, Share Exp, Outw Exp respectively denotes the single layer in_window, shared, and out_window expert network.

where $w_{IP}(\mathbf{x})$, $w_{DP}(\mathbf{x})$, $w_{FN}(\mathbf{x})$, $w_{RN}(\mathbf{x})$ denotes the importance weights; \mathbb{I}_{IP} is the indicator of observed immediate positives. Empirically, we set $w_{DP}(\mathbf{x}) = 1$ and $w_{FN}(\mathbf{x}) = f_{dp}(\mathbf{x})$ since DP can be observed. A detailed proof is given in the supplementary material.

Compared with the standard cross-entropy loss, we integrate an auxiliary task $f_{dp}(\mathbf{x})$ to model the importance weights w_i for each type of sample, rather than directly using the observed label.

4.1.2 Optimization. Hereafter, the remaining question is how to optimize the unbiased loss function. Since the z is inaccessible in Equation (15), we implement a two-step optimization by introducing another auxiliary model $z(\mathbf{x})$ predicting the hidden z to further decouple the observed negative samples into real negative and fake negative samples:

$$\mathcal{L}_{neg} = z(\mathbf{x})w_{FN} \log f_{\theta}(\mathbf{x}) + (1 - z(\mathbf{x}))w_{RN} \log(1 - f_{\theta}(\mathbf{x})) \quad (16)$$

where

$$z(\mathbf{x}) = \frac{p(y = 1, d > w_o | \mathbf{x})}{p(y = 0 | \mathbf{x}) + p(y = 1, d > w_o | \mathbf{x})}, \quad (17)$$

where $z(\mathbf{x})$ is the fake negative probability, denotes the probability that an observed negative is a ground truth positive.

In practice, we implement two ways to model the $z(\mathbf{x})$:

- $z_1(\mathbf{x}) = 1 - f_{rn}(\mathbf{x})$. This adopts a binary classification model $f_{rn}(\mathbf{x})$ to predict the probability of an observed negative being a real negative [27]. For the training of f_{rn} model, the observed positives are excluded, then the negatives are labeled as 1 and the delayed positives are labeled as 0.
- $z_2(\mathbf{x}) = \frac{f_{dp}(\mathbf{x})}{f_{dp}(\mathbf{x}) + 1 - f_{\theta}(\mathbf{x})}$. This adopts the CVR model $f_{\theta}(\mathbf{x})$ and delay model $f_{dp}(\mathbf{x})$ to indirectly model the fake negative probability. For the learning of $f_{dp}(\mathbf{x})$, the delayed positives are labeled as 1, the others are labeled as 0.

4.2 Bi-Distribution Modeling

Although theoretically unbiased, a potential drawback of our DEFUSE is that the estimation of importance weights w , hidden model $z(\mathbf{x})$ and especially the multiplicative term $z(\mathbf{x})w_{FN}$ and $(1 - z(\mathbf{x}))w_{RN}$ may cause high variance. This typically implies slow convergence and leads to sub-optimal performance, especially when the feedback

is relatively sparse. As such, we strive to build an alternative learning framework that can fully exploit samples from the observed distribution directly.

Recall that, distinct from previous methods merely using the observed positive and negative samples, we divide the samples into four types. The IP and DP denote the immediate conversion and the delay conversion, respectively.

We thus adopt an multi-task learning [11, 15, 19, 26] framework to jointly optimizing following subtasks: 1) In window(Inw) model: predicting the IP probability $\mathcal{F}_{IP}(\mathbf{x}) = p(y = 1, d \leq w_o | \mathbf{x})$ within the observation window w_o . 2) Out window(Outw) model: predicting the DP probability $\mathcal{F}_{DP}(\mathbf{x})$ out of w_o . Then the overall conversion probability can be formalized as:

$$p(y = 1 | \mathbf{x}) = \mathcal{F}_{IP}(\mathbf{x}) + \mathcal{F}_{DP}(\mathbf{x}). \quad (18)$$

It's worth mentioning that, as demonstrated in Figure 3b, for task 1), the samples used are nonduplicate and correctly labeled, so the $\mathcal{F}_{IP}(\mathbf{x})$ model can be directly trained upon the ground-truth distribution without bias. For task 2), the $\mathcal{F}_{DP}(\mathbf{x})$ model has to be trained on a biased observed distribution identical to that of FNW [10] with $w'_o = 0$. Thus, we implement our DEFUSE to the $\mathcal{F}_{DP}(\mathbf{x})$ model to achieve unbiased estimation with importance sampling. Similar to the derivation of equation (15), we have:

$$\mathcal{L}_{IP} = \int p(\mathbf{x}, y_{IP}) \left[y_{IP} \log f_{IP}(\mathbf{x}) + (1 - y_{IP}) \log(1 - f_{IP}(\mathbf{x})) \right] d\mathbf{x} \quad (19)$$

$$\begin{aligned} \mathcal{L}_{DP} = \int q(\mathbf{x}, v_{DP}) & \left[v_{DP} w'_{DP}(\mathbf{x}) \log f_{DP}(\mathbf{x}) \right. \\ & + (1 - v_{DP}) w'_{FN}(\mathbf{x}) z'(\mathbf{x}) \log f_{DP}(\mathbf{x}) \\ & \left. + (1 - v_{DP}) w'_{RN}(\mathbf{x}) (1 - z'(\mathbf{x})) \log(1 - f_{DP}(\mathbf{x})) \right] d\mathbf{x}, \end{aligned} \quad (20)$$

s.t.

$$w'_{DP}(\mathbf{x}) + w'_{FN}(\mathbf{x}) = 1 + f_{dp}(\mathbf{x}), \quad w'_{RN}(\mathbf{x}) = 1 + f_{dp}(\mathbf{x}),$$

where $p(\mathbf{x}, y_{IP})$, $q(\mathbf{x}, v_{DP})$ respectively denote distributions of the training datasets for the sub-tasks, $w'_{DP}(\mathbf{x})$, $w'_{FN}(\mathbf{x})$, and $w'_{RN}(\mathbf{x})$ are the importance weights, and $z'(\mathbf{x})$ as the hidden model to further infer fake negatives. Finally, we devise our multi-task learning architecture as illustrated in Figure 3a to learn the desired CVR model by jointly optimizing the union loss:

$$\mathcal{L} = \mathcal{L}_{IP} + \mathcal{L}_{DP}. \quad (21)$$

By doing so, we divide the delayed feedback modeling into an unbiased *in_window* prediction and an importance sampling-based *out_window* prediction task. Note that only the second part needs to be trained with importance weights and hidden variable z , which implies that the negative impact introduced by the high variance of inferring w and z can be effectively limited.

5 EXPERIMENTS

In this section, we first describe experimental settings and then conduct experiments on both public and industry advertising datasets to evaluate our proposed model by answering the following research questions:

- **RQ1** How does DEFUSE perform on streaming CVR prediction tasks as compared to other state-of-the-art methods?

Table 4: Statistic of Criteo and Taobao Dataset.

Dataset	#Users	#Items	#Features	#Conversions	#Samples	avg CVR	Duration
Criteo	-	5443	17	3619801	15898883	0.2269	60 days
Taobao Dataset	382 million	10.6 million	23	208 million	5.2 billion	0.04005	21 days

- **RQ2** How does DEFUSE perform under different duplicating mechanisms?
- **RQ3** How do different components (e.g., hidden variable estimation, observation window size) and hyper-parameter settings affect the results of DEFUSE?

5.1 Datasets

We evaluate our experiment on both public and industrial datasets. The statistics of the processed datasets are shown in Table 4.

- **Criteo**¹ is the well-researched public dataset for the delayed feedback modeling task [2, 5, 10]. It is collected from the Criteo live traffic data in a period of 60 days with 30 days attribution window. we use the click and pay(if exists) timestamps and all the hashed categorical features and continuous features for train and evaluation. In particular, since 30 days attribution period is unbearable for industrial online advertising, we further derive a one-day attribution version, namely Criteo-1d, which uses samples that convert within one day as positive.
- **Taobao Dataset** is collected from the daily click and conversion logs in Taobao systems. The industrial dataset contains about 5.2 billion interactions between nearly 400 million users and 10 million items. We set $w_d = 1$ day to wait for the actual label for each sample.

5.1.1 Data Stream. We divide each dataset into two parts to simulate the streaming training environment. Specifically, the first shuffled part is used for pre-training a well-initialized model. To prevent label leakage, we refer to the practice of [5] and set the labels as 0 if the conversion occurs in the second part of the data. For the second part, observed samples are sorted by click time except that the delayed and duplicated samples are ordered by conversion time. We then divide the data into pieces by hour. To simulate the online streaming, we train models on the t -th hour data and test them on the $t + 1$ -th hour. The reported metrics are the weighted average across different hours on streaming data.

5.2 Experimental Setting

5.2.1 Evaluation Metrics. We applied three widely used evaluation metrics to evaluate the streaming CVR prediction performance:

- **AUC** is the area under ROC curve which assesses the pairwise ranking performance of the classification results between the conversion and non-conversion samples.
- **PR-AUC** is the area under the precision-recall curve, which is more sensitive than AUC in skewed data for CVR prediction task.
- **NLL** is originally used in DFM [2], which is sensitive to the absolute value of the CVR prediction. In a CPA model, the predicted probabilities are important since they are directly used to compute the value of an impression.

¹<https://labs.criteo.com/2013/12/conversion-logs-dataset/>

To demonstrate the relative improvement over the pretrained model, follow previous works [5, 27], we also evaluate the RI-AUC as:

$$\text{RI-AUC}_{\text{DEFUSE}} = \frac{\text{AUC}_{\text{DEFUSE}} - \text{AUC}_{\text{Pre-trained}}}{\text{AUC}_{\text{Oracle}} - \text{AUC}_{\text{Pre-trained}}} \times 100\%.$$

This indicates the relative improvements for DEFUSE. Obviously, the closer the relative improvement to 100%, the better the method performs.

5.2.2 Baselines. We compared our DEFUSE with the following state-of-the-art methods:

- **Pre-trained:** This model is trained by the first part of data but without continuous training on the streaming data. The rest methods are all finetuned on top of this model during streaming simulation.
- **Oracle:** A model finetuned with the ground-truth label other than the observed label, which denotes the upper bound of the delayed feedback modeling.
- **Vanilla:** A model training with the waiting window but without any duplicate samples, using standard cross-entropy loss.
- **Vanilla-Win:** Vanilla-Win is trained on the streaming data with a waiting window. DP samples are duplicated with the actual label and re-sent to the training pipeline after conversion.
- **FNW** [10]: A model finetuned on top of the Pre-trained model using the fake negative weighted loss.
- **FNC** [10]: A model finetuned on top of the Pre-trained model using the fake negative calibration loss.
- **ES-DFM** [27]: It is trained on the same streaming data as Vanilla-Win but introduces auxiliary tasks and using the ES-DFM loss.
- **DEFER** [5]: This model is trained on the DEFER pipeline as illustrated in Figure 2 with the DEFER loss.

We also tried **DFM** [2] but found that the delayed feedback loss is difficult to converge on our sizeable industrial dataset due to the difficulty of estimating the delay time based on a strong distribution assumption. Hence, although it achieved promising performance in Criteo, we did not select it for comparison.

5.2.3 Parameter Settings. We implement the DEFUSE in Tensorflow. For a fair comparison, we tune the parameter settings of each model. The hidden units are fixed for all models with hidden size {256, 256, 128}. The Leaky ReLU [16] and BatchNorm layer [7] are attached to each hidden layer. All methods are trained with Adam [9] for optimization. For the basic parameter settings of all models, we apply the grid search strategy to tune the coefficient of L_2 normalization in {0.0001, 0.0005, 0.001, 0.01} and search the learning rate among {0.0001, 0.0005, 0.001}, or directly copy the best parameter settings reported in the original papers [10, 27]. Moreover, we tune the waiting window in {0.25, 0.5, 1} hour. The same pretrained model is used to initialize the online models.

Table 5: The overall performance comparison in terms of AUC, RI-AUC, PR-AUC, and NLL. The results in terms of RI-PR-AUC and RI-NLL are omitted due to space constraints and similar trends. Bold indicates the top-performing method.

Dataset	Criteo-30d				Criteo-1d				Taobao Dataset			
Metrics	AUC	RI-AUC	PR-AUC	NLL	AUC	RI-AUC	PR-AUC	NLL	AUC	RI-AUC	PR-AUC	NLL
Pre-trained	0.8307	0%	0.6251	0.4009	0.8285	0%	0.5218	0.3001	0.8015	0%	0.6074	0.1516
Vanilla	0.8098	-108.29%	0.5902	0.5453	0.8384	53.38%	0.5408	0.3088	0.8047	32.65%	0.6168	0.1504
Vanilla-Win	0.8375	35.23%	0.6288	0.4055	0.8385	52.91%	0.5408	0.3088	0.8059	44.90%	0.6169	0.1495
Oracle	0.8500	100%	0.6468	0.3869	0.8474	100%	0.5519	0.2874	0.8113	100%	0.6197	0.1470
FNC	0.8373	34.20%	0.6222	0.4688	0.8343	30.69%	0.4806	0.3145	0.8053	38.78%	0.6150	0.1495
FNW	0.8376	35.75%	0.6310	0.3971	0.8348	33.33%	0.4982	0.3367	0.8054	39.80%	0.6148	0.1497
ES-DFM	0.8396	46.11%	0.6384	0.3947	0.8459	92.06%	0.5492	0.2885	0.8066	52.04%	0.6155	0.1494
DEFER	0.8382	38.86%	0.6338	0.4800	0.8463	94.17%	0.5490	0.3098	0.8065	51.02%	0.6153	0.1529
DEFUSE	0.8408	52.33%	0.6400	0.3946	<u>0.8465</u>	<u>95.24%</u>	0.5490	<u>0.3086</u>	<u>0.8069</u>	<u>55.10%</u>	<u>0.6177</u>	0.1489
Bi-DEFUSE	0.8379	37.31%	0.6301	0.3963	0.8467	96.30%	0.5499	0.3092	0.8080	66.33%	0.6185	0.1512

5.3 Performance Comparison (RQ1)

To demonstrate the overall performance of DEFUSE, we conduct 5 random runs on the Criteo and Taobao Dataset, and report the average results of all methods in Table 5. The best-performing method is boldfaced. Analyzing such performance comparison, we have the following observations:

- Our approach consistently yields significant improvements on all the datasets. In particular, DEFUSE and Bi-DEFUSE improves over the strongest baselines *w.r.t.* RI-AUC by 6.22%, 2.13%, and 15.31% in Criteo-30d, Criteo-1d, and Taobao Dataset, respectively. Unlike previous approaches that only utilize the observed positive and negative samples during importance sampling, we divide the observed distribution into four types of feedbacks and introduce an auxiliary task to infer fake negatives from the observed negatives. Moreover, on Criteo-1d, our method can narrow the delayed feedback gap significantly compared to other methods by comparing the relative metrics, RI-AUC. Note that as reported in [30], a small improvement of offline AUC can lead to a significant increase in streaming CTR. In our scenario, even 0.1% of AUC improvement in CVR prediction is substantial and achieves significant online promotion.
- ES-DFM and DEFER generally achieve better performance than FNW and FNC. Such improvement can be attributed to the duplicating mechanism with a properly tuned w_o , which provides a good balance for the trade-off between label accuracy and sample freshness. Comparison between Vanilla-Win and Vanilla also indicates the importance of duplicating mechanisms.
- Compared with the pre-trained model, almost all the continuous learning methods demonstrate promising performance. This verifies the significant advantages of utilizing fresh samples for CVR prediction. Vanilla performs poorly on Criteo-30d but obtains a better result on Criteo-1d, possibly caused by the fact that the observed distribution deviated much more from the ground-truth distribution with longer w_a . This also indicates the importance of utilizing all conversions and performing unbiased estimation with delayed feedback modeling. Moreover, DEFER also demonstrates similar performance when compared with ES-DFM. Such

Table 6: Performance of DEFUSE under different duplicating mechanisms in Criteo.

Dataset	Criteo-30d		Criteo-1d	
Metrics	AUC	RI-AUC	AUC	RI-AUC
FNW	0.8376	35.75%	0.8348	33.33%
FNW+DEFUSE	0.8393	44.60%	0.8351	34.92%
ES-DFM	0.8396	46.11%	0.8459	92.06%
ES-DFM+DEFUSE	0.8408	52.33%	0.8465	95.24%
DEFER	0.8382	38.86%	0.8463	94.17%
DEFER+DEFUSE	0.8387	41.45%	0.8466	95.77%

a result can be attributed to the distribution bias caused by the long-term attribution window — it ingests real negative samples from 30 days before into the training pipeline.

5.4 Experiments under Different Duplicating Mechanisms (RQ2)

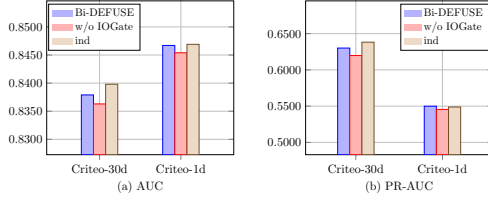
Recall that, our DEFUSE is unbiased and can be applied to different duplicating mechanisms. To further verify the performance, we have conducted experiments using different training pipelines on Criteo dataset applied by FNW, ES-DFM and DEFER respectively, and report the AUC and RI-AUC results in Table 6. In general, by label correction and weighting the importance of four types of observed samples, our unbiased estimation demonstrates consistent improvement on performance among the three duplicating pipelines.

5.5 Study of DEFUSE (RQ3)

Ablation studies on DEFUSE are also conducted to investigate the rationality and effectiveness of some designs — to be more specific, (1) how different estimations of hidden variable z may affect performance, (2) contributions of each components of Bi-DEFUSE, and (3) influence of different attribution window length w_a .

Table 7: Impact of the ways of hidden variable inference w.r.t. AUC and RI-AUC in Criteo.

Dataset	Criteo-30d		Criteo-1d	
Metrics	AUC	RI-AUC	AUC	RI-AUC
DEFUSE + z_1	0.8408	52.33%	0.8465	96.30%
DEFUSE + z_2	0.8405	50.78%	0.8464	94.71%
DEFUSE + z_{oracle}	0.8426	61.66%	0.8468	96.83%

**Figure 4: Ablation study for Bi-DEFUSE.**

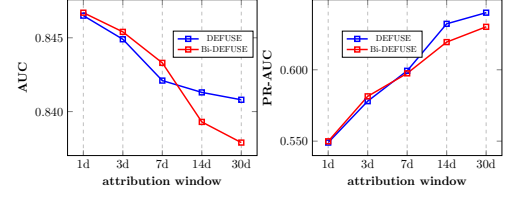
5.5.1 Impact of the estimation of $z(x)$. Since fake negatives are falsely labelled, a hidden variable z is further introduced to infer whether the observed negatives are fake negatives. We hence implement an additional auxiliary model – $z(x)$ that directly predicts FNs from observed negatives. As introduced in Section 4.1.2, different choices of modelling $z(x)$ is experimented. Besides, to further explore the upper bound of our two-step optimization, we additionally investigate the ideal performance of DEFUSE given $z_{oracle} \in \{0, 1\}$, which indicates the ground-truth label for z . As shown in Table 7, here are some observations:

- DEFUSE+ z_1 consistently outperforms DEFUSE+ z_2 . We credit such improvement to the reduction of high variance of $z(x)$. Prediction of $z_2(x)$ obviously involves divisions between two independent models, which may lead to unstable estimation and sub-optimal performance.
- DEFUSE+ z_{oracle} achieves best performance uniformly, this indicates the potential of optimizing our unbiased estimation by further improving the prediction of $z(x)$.
- We also notice that the gap between + z_1 and + z_{oracle} on Criteo-1d is smaller than that on Criteo-30d, indicating that lower proportion of fake negatives with relatively shorter attribution window makes it easier to estimate $z(x)$.

5.5.2 Contributions of different components in Bi-DEFUSE.

To have a better understanding of Bi-DEFUSE, we do ablation study by evaluating three formulations of Bi-DEFUSE: (1) Bi-DEFUSE as demonstrated in Figure 3a; (2) a simpler version replacing the gates, experts and heads by MLPs; marked as w/o IOGate; (3) completely disabling the shared network and estimating the in/out window CVR with two independent models, marked as *ind*. From Figure 4, we have:

- Removal of the MMoE gates and experts always degrades model performance since Bi-DEFUSE consistently outperforms w/o IOGate.

**Figure 5: The effect of different window length for w_a .**

- *ind* uniformly dominates, indicating that IPs and DPs can be greatly different, independent models effectively avoid the conflict caused by predicting both IPs and long DPs using shared models. Yet, *ind* is highly impractical in industrial scenarios, since it doubles the computational and storage consumption.
- Bi-DEFUSE still achieves comparable performance with *ind* on Criteo-1d, likely a result of the fact that smaller w_a not only greatly raises the ratio of IPs, making the unbiased estimation of IPs more decisive; but also effectively limits the difference between IPs and DPs, allowing the introduction of shared networks.

5.5.3 Performance of Bi-DEFUSE w.r.t. different w_a . As implied in Table 5, performance of Bi-DEFUSE may degrade with long w_a . Towards this end, we investigate the influence of different attribution windows in depth. Since DEFUSE consistently outperforms baseline methods, for clear comparison, we only evaluate the performance of DEFUSE and Bi-DEFUSE with $w_a = \{1, 3, 7, 14, 30\}$ days on the Criteo dataset. Results reported in Figure 5 suggest that Bi-DEFUSE achieves the better performance with smaller w_a , e.g., $w_a \leq 7$ in terms of AUC and PR-AUC, since smaller w_a not only makes the unbiased prediction of IPs more important, but also implies smaller challenges in estimating $z(x)$ and predicting DPs, which greatly amplifies the advantages of Bi-DEFUSE.

5.6 Online Evaluation

We conducted an A/B test in our online scenario to evaluate the proposed Bi-DEFUSE framework. We set 30min observation window and 1day attribution window and observed a steady performance improvement in terms of CVR(+2.28%). This align with our offline streaming results and demonstrates the effectiveness of our DEFUSE in industrial systems.

6 CONCLUSION

In this paper, we propose an asymptotically unbiased estimation method for the streaming CVR prediction, which addresses the delay feedback problem by weighting the importance of four types of observed samples. We carefully devise a new delayed feedback modeling approach, DEFUSE, with a two-step optimization mechanism. Specifically, we first infer the probability of fake negative among observed negatives, then respectively correct the importance weights of each type of observed samples. Furthermore, to fully exploit the ground-truth immediate positives, we propose a bi-distribution modeling framework. Comprehensive experiments show that the proposed DEFUSE and Bi-DEFUSE significantly outperform the competitive baselines.

REFERENCES

- [1] Christopher M Bishop. 2006. Pattern recognition. *Machine learning* 128, 9 (2006).
- [2] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani (Eds.). ACM, 1097–1105. <https://doi.org/10.1145/2623330.2623634>
- [3] David S Evans. 2009. The online advertising industry: Economics, evolution, and privacy. *Journal of economic perspectives* 23, 3 (2009), 37–60.
- [4] Avi Goldfarb and Catherine Tucker. 2011. Online display advertising: Targeting and obtrusiveness. *Marketing Science* 30, 3 (2011), 389–404.
- [5] 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 *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14–18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 2890–2898. <https://doi.org/10.1145/3447548.3467086>
- [6] Amélie Hélieu, Panayotis Mertikopoulos, and Zhengyuan Zhou. 2020. Gradient-free Online Learning in Games with Delayed Rewards. In *ICML '20: The 37th International Conference on Machine Learning*.
- [7] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, 448–456. <http://proceedings.mlr.press/v37/loffe15.html>
- [8] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. 2013. Online learning under delayed feedback. In *International Conference on Machine Learning*. PMLR, 1453–1461.
- [9] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- [10] 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, RecSys 2019, Copenhagen, Denmark, September 16–20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 187–195. <https://doi.org/10.1145/3298689.3347002>
- [11] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12–16, 2012*, Qiang Yang, Deepak Agarwal, and Jian Pei (Eds.). ACM, 768–776. <https://doi.org/10.1145/2339530.2339651>
- [12] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 768–776.
- [13] Xun Liu, Wei Xue, Lei Xiao, and Bo Zhang. 2017. PBODL: Parallel Bayesian Online Deep Learning for Click-Through Rate Prediction in Tencent Advertising System. *CoRR* abs/1707.00802 (2017). [arXiv:1707.00802](http://arxiv.org/abs/1707.00802) <http://arxiv.org/abs/1707.00802>
- [14] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the ADKDD'17*, 1–9.
- [15] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19–23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 1930–1939. <https://doi.org/10.1145/3219819.3220007>
- [16] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. Citeseer, 3.
- [17] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. 2015. The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- [18] Taesup Moon, Lihong Li, Wei Chu, Ciya Liao, Zhaohui Zheng, and Yi Chang. 2010. Online learning for recency search ranking using real-time user feedback. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26–30, 2010*, Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). ACM, 1501–1504. <https://doi.org/10.1145/1871437.1871657>
- [19] Junwei Pan, Yizhi Mao, Alfonso Lobos Ruiz, Yu Sun, and Aaron Flores. 2019. Predicting Different Types of Conversions with Multi-Task Learning in Online Advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4–8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Römer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 2689–2697. <https://doi.org/10.1145/3292500.3330783>
- [20] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvári, and Steffen Grunewalder. 2018. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*. PMLR, 4105–4113.
- [21] Doyen Sahoo, Quang Pham, Jing Lu, and Steven C. H. Hoi. 2018. Online Deep Learning: Learning Deep Neural Networks on the Fly. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 2660–2666. <https://doi.org/10.24963/ijcai.2018/369>
- [22] Linqi Song, Cem Tekin, and Mihaela van der Schaar. 2016. Online Learning in Large-Scale Contextual Recommender Systems. *IEEE Trans. Serv. Comput.* 9, 3 (2016), 433–445. <https://doi.org/10.1109/TSC.2014.2365795>
- [23] Yumin Su, Liang Zhang, Quanyu Dai, Bo Zhang, Jinyao Yan, Dan Wang, Yongjun Bao, Sulong Xu, Yang He, and Weipeng Yan. 2020. An Attention-based Model for Conversion Rate Prediction with Delayed Feedback via Post-click Calibration. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, Christian Bessiere (Ed.)*. ijcai.org, 3522–3528. <https://doi.org/10.24963/ijcai.2020/487>
- [24] Claire Vernade, Olivier Cappé, and Vianney Perchet. 2017. Stochastic bandit models for delayed conversions. *arXiv preprint arXiv:1706.09186* (2017).
- [25] Yanshi Wang, Jie Zhang, Qing Da, and Anxiang Zeng. 2020. Delayed Feedback Modeling for the Entire Space Conversion Rate Prediction. *CoRR* abs/2011.11826 (2020). [arXiv:2011.11826](https://arxiv.org/abs/2011.11826) <https://arxiv.org/abs/2011.11826>
- [26] Zhuojian Xiao, Yunjiang Jiang, Guoyu Tang, Lin Liu, Sulong Xu, Yun Xiao, and Weipeng Yan. 2021. Adversarial Mixture Of Experts with Category Hierarchy Soft Constraint. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19–22, 2021*. IEEE, 2453–2463. <https://doi.org/10.1109/ICDE51399.2021.00278>
- [27] 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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 4582–4589. <https://ojs.aaai.org/index.php/AAAI/article/view/16587>
- [28] Shota Yasui, Gota Morishita, Komei Fujita, and Masashi Shibata. 2020. A Feedback Shift Correction in Predicting Conversion Rates under Delayed Feedback. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 2740–2746. <https://doi.org/10.1145/3366423.3380032>
- [29] Yuya Yoshikawa and Yusaku Imai. 2018. A Nonparametric Delayed Feedback Model for Conversion Rate Prediction. *CoRR* abs/1802.00255 (2018). [arXiv:1802.00255](https://arxiv.org/abs/1802.00255) [http://arxiv.org/abs/1802.00255](https://arxiv.org/abs/1802.00255)
- [30] 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.

A APPENDIX

A.1 Detailed demonstration of Equation (15) and derivations of w

We first provide the demonstration of the equivalence of \mathcal{L}_{ub} , then show the detailed derivations for the importance weight $w_i(\mathbf{x}, y)$, where $i \in \{IP, FN, RN, DP\}$.

A.1.1 Demonstration of \mathcal{L}_{ub} . Firstly, let v denotes the observed label and v_{IP} , v_{DP} respectively denote the label for immediate positive and delay positive sample, we formulate the minimization for \mathcal{L}_{ub} as:

$$\begin{aligned} \min_{\theta} \mathcal{L}_{ub} = & \min_{\theta} \int q(\mathbf{x}) \left[v_{IP} w_{IP}(\mathbf{x}, y) \log f_{\theta}(\mathbf{x}) \right. \\ & + v_{IP} w_{IP}(\mathbf{x}, y) \log p(d \leq w_o) \\ & + (v_{DP} w_{DP}(\mathbf{x}, y) + z(1-v)w_{FN}(\mathbf{x}, y)) \log f_{\theta}(\mathbf{x}) \\ & + (v_{DP} w_{DP}(\mathbf{x}, y) + z(1-v)w_{FN}(\mathbf{x}, y)) \log p(d > w_o) \\ & \left. + (1-v)(1-z)w_{RN}(\mathbf{x}, y) \log(1-f_{\theta}(\mathbf{x})) \right] d\mathbf{x}, \end{aligned} \quad (22)$$

where z is the latent variable. Since the derivation of \mathcal{L}_{ub} w.r.t. f_{θ} is irrelevant with $\log p(d \leq w_o)$ and $\log p(d > w_o)$, it's trivial to obtain following equivalence formula:

$$\begin{aligned} \min_{\theta} \mathcal{L}_{ub} \\ \Leftrightarrow & \min_{\theta} \int q(\mathbf{x}) \left[v_{IP} w_{IP}(\mathbf{x}, y) \log f_{\theta}(\mathbf{x}) \right. \\ & + (v_{DP} w_{DP}(\mathbf{x}, y) + z(1-v)w_{FN}(\mathbf{x}, y)) \log f_{\theta}(\mathbf{x}) \\ & \left. + (1-v)(1-z)w_{RN}(\mathbf{x}, y) \log(1-f_{\theta}(\mathbf{x})) \right] d\mathbf{x} \\ = & \min_{\theta} \int q(\mathbf{x}) \left[v(w_{DP} \log f_{\theta}(\mathbf{x}) + \mathbb{I}_{IP}(w_{IP} - w_{DP}) \log f_{\theta}(\mathbf{x})) \right. \\ & + (1-v)(w_{FN} \log f_{\theta}(\mathbf{x}) + z \\ & \left. + w_{RN} \log(1-f_{\theta}(\mathbf{x}))(1-z)) \right] d\mathbf{x}, \end{aligned} \quad (23)$$

Intuitively, to ensure an unbiased estimation, we must have \mathcal{L}_{ub} from equation (14) equals to \mathcal{L}_{ideal} from equation (1). That is,

$$\begin{aligned} & \int q(\mathbf{x}) \left[\sum_{v_i} q(v_i | \mathbf{x}) w_i(\mathbf{x}, y(v_i, d)) \ell(\mathbf{x}, y(v_i, d); f_{\theta}(\mathbf{x})) \right] d\mathbf{x} \\ = & \int p(\mathbf{x}) \left[\sum_{y(v_i, d)} p(y(v_i, d) | \mathbf{x}) \ell(\mathbf{x}, y(v_i, d); f_{\theta}(\mathbf{x})) \right] d\mathbf{x}, \end{aligned} \quad (24)$$

Empirically, we can express the left part in the form of discrete summation as:

$$\begin{aligned} \mathcal{L}_{left} = & \sum \left[v_{IP} w_{IP}(\mathbf{x}, y) \log(f_{\theta}(\mathbf{x}) p(d \leq w_o | \mathbf{x}, y = 1)) \right. \\ & + [v_{DP} w_{DP}(\mathbf{x}, y) + z(\mathbf{x})(1-v)w_{FN}(\mathbf{x}, y)] \ell_{DP}(\mathbf{x}, y) \\ & \left. + (1-v)(1-z(\mathbf{x}))w_{RN}(\mathbf{x}, y) \log(1-f_{\theta}(\mathbf{x})) \right]. \end{aligned} \quad (25)$$

where $\ell_{DP}(\mathbf{x}, y) = \log(f_{\theta}(\mathbf{x}) p(d > w_o | \mathbf{x}, y = 1))$ and $z(\mathbf{x})$ denotes fake negative probability. Similarly, we can express the right part as:

$$\begin{aligned} \mathcal{L}_{right} = & \sum \left[y_{IP} \log(f_{\theta}(\mathbf{x}) p(d < w_o | \mathbf{x}, y = 1)) \right. \\ & + y_{DP} \log(f_{\theta}(\mathbf{x}) p(d > w_o | \mathbf{x}, y = 1)) \\ & \left. + (1-y) \log(1-f_{\theta}(\mathbf{x})) \right] \end{aligned} \quad (26)$$

Take the middle term of equation (25,26) as an example, unbiased estimation would require either

$$\begin{aligned} & \sum [v_{DP} w_{DP}(\mathbf{x}, y) + z(\mathbf{x})(1-v)w_{FN}(\mathbf{x}, y)] \ell_{DP}(\mathbf{x}, y) \\ = & \sum y_{DP} \log(f_{\theta}(\mathbf{x}) p(d > w_o | \mathbf{x}, y = 1)) \end{aligned}$$

or

$$v_{DP}(\mathbf{x}) w_{DP}(\mathbf{x}, y) + z(\mathbf{x})(1-v)w_{FN}(\mathbf{x}, y) = y_{DP}(\mathbf{x}), \forall \mathbf{x}. \quad (27)$$

Although directly solving w from above equations, we can rewrite equation (27) into following expectation form:

$$E_q[v_{DP} w_{DP}(\mathbf{x}, y) + z(1-v)w_{FN}(\mathbf{x}, y)] = E_p[y_{DP}]. \quad (28)$$

Empirically, equation always stands as well as the following unbiased estimations:

$$E_q[v_{DP}] (\mathbf{x}) = \frac{f_{dp}(\mathbf{x})}{1 + f_{dp}(\mathbf{x})} \quad (29)$$

$$E_q[z] (\mathbf{x}) = \frac{f_{dp}(\mathbf{x})}{1 - p_{win}(\mathbf{x})} \quad (30)$$

$$E_p[y_{DP}] (\mathbf{x}) = f_{dp}(\mathbf{x}) \quad (31)$$

$$E_q[1-v] (\mathbf{x}) = E_q[1 - (v_{IP} + v_{DP})] (\mathbf{x}) = \frac{1 - p_{win}(\mathbf{x})}{1 + f_{dp}(\mathbf{x})}, \quad (32)$$

where $p_{win}(\mathbf{x}) = p(y = 1 | \mathbf{x}) p(d < w_o | \mathbf{x}, y = 1)$. With sufficient data, which is usually abundant in digital advertising, we may reasonably assume following asymptotically unbiased estimation:

$$\frac{f_{dp}(\mathbf{x})}{1 + f_{dp}(\mathbf{x})} w_{DP}(\mathbf{x}, y) + \frac{f_{dp}}{1 - p_{win}} \frac{1 - p_{win}}{1 + f_{dp}(\mathbf{x})} w_{FN}(\mathbf{x}, y) \rightarrow f_{dp}(\mathbf{x}). \quad (33)$$

This leads to $w_{DP}(\mathbf{x}, y) + w_{FN}(\mathbf{x}, y) = 1 + f_{dp}(\mathbf{x})$. Likewise, we may derive that $w_{IP}(\mathbf{x}, y) = w_{RN}(\mathbf{x}, y) = 1 + f_{dp}(\mathbf{x})$.

A.2 Unbiased Estimation for other duplicating mechanisms

Next, we state the unbiased estimation for other duplicating mechanisms(FNW+DEFUSE, DEFER+DEFUSE). Since the derivation process is similar, we directly show the loss form.

A.2.1 \mathcal{L}_{ub} for dup-mechanism in FNW. For unbiased estimation for the dup-mechanism in FNW, since the size of w_o is 0, we divide the observed samples into DP , FN , and RN . We have following unbiased loss form:

$$\begin{aligned} \min_{\theta} \mathcal{L}_{ub} \\ \Leftrightarrow & \min_{\theta} \int q(\mathbf{x}) \left[v w_{DP}(\mathbf{x}, y) \log f_{\theta}(\mathbf{x}) \right. \\ & + z(1-v)w_{FN}(\mathbf{x}, y) \log f_{\theta}(\mathbf{x}) \\ & \left. + (1-v)(1-z)w_{RN}(\mathbf{x}, y) \log(1-f_{\theta}(\mathbf{x})) \right] d\mathbf{x}, \end{aligned} \quad (34)$$

s.t.

$$\begin{aligned} w_{RN}(\mathbf{x}, y) &= 1 + f_{dp-fnw}(\mathbf{x}) \\ w_{DP}(\mathbf{x}, y) + w_{FN}(\mathbf{x}, y) &= 1 + f_{dp-fnw}(\mathbf{x}), \end{aligned}$$

where $f_{dp-fnw} = f_\theta$ denotes the probability of observed negative samples to be fake negative. Emprically, we set $w_{DP}(\mathbf{x}, y) = 1$ since DP can be observed, and reduce the importance weight of fake negative($w_{FN}(\mathbf{x}, y)$) to f_θ .

A.2.2 \mathcal{L}_{ub} for dup-mechanism in DEFER. The loss form of unbiased DEFER loss is same to ES-DFM+DEFUSE, as they both have observation window w_o and attribution window w_a . The main difference lies in the way of duplication for real negative samples,

which will lead to different estimation forms of importance weights.

$$\begin{aligned} & \min_{\theta} \mathcal{L}_{ub} \\ \Leftrightarrow & \min_{\theta} \int q(\mathbf{x}) \left[v_{IP} w_{IP}(\mathbf{x}, y) \log f_\theta(\mathbf{x}) \right. \\ & + (v_{DP} w_{DP}(\mathbf{x}, y) + z(1-v) w_{FN}(\mathbf{x}, y)) \log f_\theta(\mathbf{x}) \\ & \left. + (1-v)(1-z) w_{RN}(\mathbf{x}, y) \log(1 - f_\theta(\mathbf{x})) \right] dx, \end{aligned} \quad (35)$$

s.t.

$$w_{IP}(\mathbf{x}, y) = w_{RN}(\mathbf{x}) = 2$$

$$w_{DP}(\mathbf{x}, y) + w_{FN}(\mathbf{x}) = 2.$$

The weights of both IPs and RNs are duplicated. Moreover, since DP samples has same feature distribution with FN samples, we emprically set $w_{DP}(\mathbf{x}, y) = w_{FN}(\mathbf{x}, y) = 1$.