Dually Enhanced Delayed Feedback Modeling for Streaming Conversion Rate Prediction

Sunhao Dai
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
sunhaodai@ruc.edu.cn

Yuqi Zhou
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
yuqizhou@ruc.edu.cn

Jun Xu*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

ABSTRACT

In online industrial advertising systems, conversion actions (e.g., purchases or downloads) often occur significantly delayed, even up to several days or weeks after the user clicks. This phenomenon leads to the crucial challenge called *delayed feedback* problem in streaming CVR prediction, that is, the online systems cannot receive the true label of conversions immediately for continuous training. To mitigate the delayed feedback problem, recent state-of-the-art methods often apply sample duplicate mechanisms to introduce early certain conversion information. Nevertheless, these works have overlooked a crucial issue of rapid shifts in data distribution and considered both the newly observed data and duplicated early data together, resulting in biases in both distributions. In this work, we propose a **D**ually enhanced **D**elayed **F**eedback **M**odel (DDFM), which tackles the above issues by treating the newly observed data and duplicated early data separately. DDFM consists of dual unbiased CVR estimators that share the same form but utilize different latent variables as weights: one for the newly observed data and the other for the duplicated early data. To avoid high variance, we adopt an addition-only formula for these latent variables, eliminating multiplication or division operations. Furthermore, we design a shared-bottom network that efficiently and jointly estimates the latent variables in DDFM. Theoretical analysis demonstrates the unbiasedness and convergence properties of DDFM. Extensive experiments on both public and industrial large-scale real-world datasets exhibit that our proposed DDFM consistently outperforms existing state-of-the-art methods.

*Jun Xu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614856

CCS CONCEPTS

• **Information systems** → **Computational advertising**; **Online advertising**.

KEYWORDS

Delayed Feedback; Conversion Prediction; Online Advertising

ACM Reference Format:
Sunhao Dai, Yuqi Zhou, Jun Xu, and Ji-Rong Wen. 2023. Dually Enhanced Delayed Feedback Modeling for Streaming Conversion Rate Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614856

1 INTRODUCTION

In online industrial advertising and recommender systems, conversion rate (CVR) prediction has become a core task as it directly relates to the profit of the platform in the widely-used cost-per-conversion (CPA) advertising payment mechanism [14–16, 41]. To capture the distribution shifts and maintain the model freshness, continuous training is commonly employed in practice for streaming CVR prediction task [5, 10, 12].

However, conversion actions (e.g., purchases and downloads) often occur up to even several days or weeks after the user clicks on real-world scenarios. Conversions that occur delayed are immediately falsely treated as negatives (i.e., *fake negatives*) in the current training pipeline, posing a significant challenge known as the *delayed feedback* problem in streaming CVR prediction. Delayed feedback leads to a dilemma for continuous training: a longer time window retains more certain label information but sacrifices model freshness, while a shorter window results in more fake negatives due to more delayed conversions.

To mitigate the delayed feedback problem, recent state-of-the-art methods have explored various sample duplicate mechanisms to leverage early available certain conversion information. For instance, FNW and FNC [12] immediately ingest all samples with negative labels into the training pipeline and re-ingest the duplicated delayed positives when conversions occur. ES-DFM [34] introduces a short waiting window to balance label correctness and model freshness. Conversions that do not happen within the window are

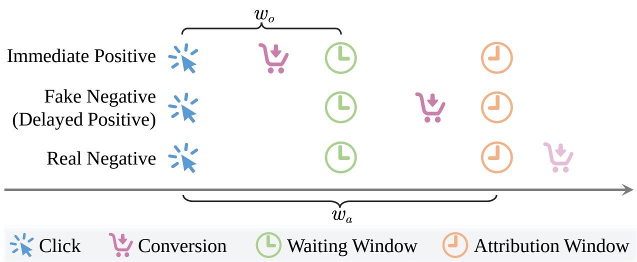


Figure 1: An illustration of delayed feedback problem in streaming conversion rate prediction. The attribution window w_a indicates a conversion is attributed to a click only if it occurred within this time window [4, 27], thereby a real negative is a sample that conversion does not occur before the attribution window. The waiting window w_o represents the time interval between the click time and the model updating time. An immediate positive is a sample that converts immediately within the waiting window. A fake negative is a sample whose conversion doesn't occur within the waiting window but will finally convert before the attribution window, i.e., a delayed positive.

first labeled as negatives and will be duplicated and ingested into the training pipeline with positive labels upon conversion. DEFER [4] further duplicates both delayed positives and real negatives to introduce more certain label information. Moreover, these methods adopt importance weighting methods [23] to re-weight the loss for unbiased delayed feedback modeling with fake negative samples.

Although existing methods have shown effectiveness to some extent, we argue that there are still some limitations. One significant limitation is that these methods assume the observed and duplicated feature distributions are identical, aiming to achieve an overall unbiased estimation for both types of data. However, in online systems, the data distribution rapidly shifts over time, such as the introduction of new campaigns to the system [2, 36]. As a result, the observed streaming samples collected from the latest short waiting window and the duplicated samples corresponding to clicks that occurred a considerable time ago inherently differ in their distributions. Treating these two types of data equally can introduce biases in both distributions and lead to sub-optimal performance. Moreover, most existing methods assign the same importance weight to the loss of both the newly observed and the duplicated early samples. This lack of differentiation makes it challenging for the CVR model to balance label correctness (i.e., correcting delayed conversions) and model freshness (i.e., adapting to recent streaming data).

In this work, we propose a novel **D**ually enhanced **D**elayed **F**eedback **M**odel (DDFM) for streaming CVR prediction. DDFM addresses the above issues of existing methods by treating the newly observed streaming data and the duplicated early data separately, enabling a more accurate and fine-grained delayed feedback modeling. Specifically, DDFM introduces dual unbiased CVR estimators through re-weighting, one for the observed data and the other for the duplicated data. These estimators follow the same form but utilize different latent variables as weight. Furthermore, we analyze the convergence properties of these unbiased estimators, demonstrating their ability to optimize in the correct gradient directions. To overcome the high variance issue that previous importance weighting methods suffer from, we only employ addition operations in the formula instead of multiplication or division operations. We then propose a shared-bottom network for efficiently and jointly estimating the latent variables. To evaluate the effectiveness of DDFM, we conduct extensive experiments on two large-scale datasets from real-world online advertising systems.

The major contributions of this paper are summarized as follows:

(1) We propose a novel dually enhanced model DDFM for delayed feedback modeling that incorporates two unbiased estimators with the same form, one for the newly observed streaming data and the other for the duplicated early data.

(2) We adopt an addition-only way for estimating weights in DDFM, which can avoid high variance issues. Moreover, we utilize a shared-bottom network to efficiently and jointly learn the latent variables in DDFM.

(3) We theoretically analyze the learning objectives of DDFM from both unbiasedness and convergence perspectives.

(4) We conduct extensive experiments on both large-scale public and industrial datasets, and the results show the superior performance of our proposed DDFM against state-of-the-art methods.

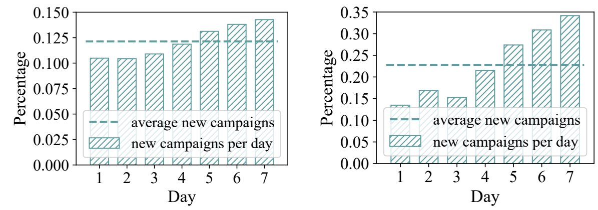
2 RELATED WORK

Deleayed Feedback Modeling is a crucial challenge in online conversion rate (CVR) prediction [6, 15, 22, 33, 35]. Chapelle [2] was the first study to emphasize the delayed feedback problem and assumed an exponential distribution for the delay time. Yoshikawa and Imai [37] relaxed this assumption and proposed a non-parametric delayed feedback model with kernel density estimation. Saito et al. [21] addressed both the positive-unlabeled and missing-not-at-random problems of delayed conversions by applying inverse propensity weighting [8]. Moreover, several studies have explored the use of importance weighting method [23] to correct the observed biased data distribution [7, 12, 36]. Following this series of works, recent state-of-the-art methods have designed various sample duplicate mechanisms to further introduce early certain conversion information due to the sparsity and rarity of conversions [3, 4, 34]. However, these methods overlook the inherent differences between the newly observed data distribution and the duplicated early data distribution, and conflate them up when modeling, which results in biases in both data distributions.

Bandits with Delayed Feedback has also garnered much research attention in recent years [18, 20, 25, 29]. The goal of bandit algorithms is to maximize the cumulative reward over a period of time [13, 24, 38]. Joulani et al. [9] analyzed the impact of delayed feedback in online learning and proposed to transform non-delayed bandit algorithms for delayed settings. Pike-Burke et al. [20] studied bandit algorithms with delayed, aggregated, and anonymous feedback, while Vernade et al. [25] explored the stochastic delayed bandit algorithms and provided a comprehensive analysis assuming known delay distributions. Mertikopoulos et al. [19] proposed a gradient-free learning policy for bandit streaming learning in games with continuous action spaces, which integrates delayed rewards and bandit feedback. Zhang et al. [39] proposed a counterfactual

Table 1: The data types in delayed feedback modeling.

Data source	Data type	Training label	True label
Newly observed data	IP	1	1
	FN	**0**	**1**
	RN	0	0
Duplicated early data	DP	1	1
	DN	0	0



(a) Criteo (b) Industrial

Figure 2: Percentages of new campaigns in the following week after a reference day on Criteo and Industrial datasets.

bandit algorithm with reward modification under the delayed bandit feedback. Different from the above studies, our work focuses on the real-world online streaming CVR prediction task with delayed feedback and aims to achieve an unbiased CVR estimation.

3 PRELIMINARY

3.1 Problem Formulation

In this work, we focus on the streaming CVR prediction task, where the CVR model is continuously updated with newly arrived data to keep the model fresh. Formally, the CVR model is trained on samples $(\mathbf{x}, y) \sim p(\mathbf{x}, y)$ drawn from the *ground-truth* data distribution $p(\mathbf{x}, y)$, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ denotes the features. The conversion label $y \in \mathcal{Y} = \{0, 1\}$, where $y = 1$ means the conversion, otherwise $y = 0$. Usually, the goal of CVR prediction is to learn a binary classifier $f_\theta : \mathcal{X} \to [0, 1]$ parameterized with θ by optimizing the following ideal learning objective [14, 15, 34]:

$$\mathcal{L}_{ideal} = -\mathbb{E}_{(\mathrm{x},y)\sim p(\mathrm{x},y)} \left[y \log(f_\theta(\mathbf{x}) + (1-y) \log(1 - f_\theta(\mathbf{x})) \right], \quad (1)$$

where the widely-used binary cross-entropy loss is adopted.

However, as CVR models are updated immediately after the waiting window in continuous training, samples where clicks occurred within the waiting window but conversions occurred outside the window are first ingested into the training pipeline as fake negatives. As a result, the observed data distribution $q^{obs}(\mathbf{x}, y)$ is often biased from the ground-truth data distribution $p(\mathbf{x}, y)$ and it is infeasible to directly optimize the ideal learning objective in Eq. (1).

To illustrate the delayed feedback problem in streaming CVR prediction more clearly, let w_o and w_a be the length of *waiting window* and *attribution window* ($w_a > w_o$), respectively. The time interval between click and conversion is denoted as *delayed time* d. As shown in Figure 1, there are three types of newly observed training samples in the delayed feedback modeling:

- **Immediate Positives** (IP, $d < w_o$): Immediate positives denote samples that convert within the waiting window and are labeled as positive.
- **Fake Negatives** (FN, $w_o < d < w_a$): Fake negatives denote samples that the conversion does not occur before the waiting window and are falsely labeled as negatives, which is also referred to as delayed positives.
- **Real Negatives** (RN, $d > w_a$): Real negatives (RN) denote samples that the conversion does not occur or convert after the attribution window and are labeled as negative.

To mitigate the delayed feedback problem, recent state-of-the-art methods such as ES-DFM [34] and DEFER [4] have proposed sample replay mechanisms to introduce early certain conversion information into the training pipeline. The difference between ES-DFM and DEFER is that ES-DFM only duplicates fake negatives with the true labels when conversions occur, while DEFER also duplicates the real negatives to bring more certain label information. Consequently, two additional types of training samples are introduced following the duplication mechanisms:

- **Duplicated Positive** (DP, $w_o < d < w_a$): Duplicated positive samples are the duplication of *fake negatives* but with true labels and will be ingested into the training pipeline with positive labels upon conversion.
- **Duplicated Negative** (DN, $d > w_a$): Duplicated negative samples are the duplication of *real negatives* and will be ingested into the training pipeline with negative labels when conversions do not occur finally.

Overall, as summarized in Table 1, there are five kinds of data samples in the whole training pipeline: the newly observed samples $(\mathrm{x}, y) \sim q^{obs}(\mathbf{x}, y)$ with partially true labels (i.e., IP, FN, and RN), and the duplicated samples $(\mathrm{x}, y) \sim q^{dup}(\mathbf{x}, y)$ with true labels (i.e., DP and DN). Note that the newly observed data is fresh but may contain fake negatives, whereas the labels of the duplicated early data are all true but stale.

3.2 Distribution Shift in Streaming Data

In this subsection, we conduct further empirical analysis of the data stream to shed light on the limitations of existing methods and outline our motivation. In real-world online display advertising scenarios, the data distribution undergoes constant shifts due to various factors, including introducing new campaigns, advertisers, and users [2, 4, 36]. To illustrate this point, we select a reference day and examine the statistics of new campaigns in the following week using two real-world datasets. Figure 2 clearly demonstrates a significant percentage of new campaigns per day. Typically, the percentage reaches as high as 23% in the Industrial dataset. These findings confirm that the data distribution rapidly evolves over time, necessitating continuous training to adapt to these shifts and maintain model freshness. Furthermore, our analysis reveals that the distributions of newly observed and duplicate data naturally differ. It is important to note that duplicate samples consist of clicks that occurred in the early period, potentially ranging up to 30 days ago in the case of the Criteo dataset [2]. On the other hand, newly observed samples are collected from the most recent short waiting window. This distinction further emphasizes the inherent dissimilarity between these two types of data.

4 OUR APPROACH: DDFM

In this section, we propose a Dually enhanced Delayed Feedback Model (DDFM) to overcome the limitations described in the previous sections. Our motivation is to model the newly observed and duplicated data from a dual perspective, recognizing their inherent differences rather than treating them as a single entity. To achieve this, we introduce two unbiased estimators in DDFM that account for the unique characteristics of both the newly observed data and the duplicated data separately. In this way, we aim to obtain a more comprehensive and accurate delayed feedback modeling.

4.1 Dually Enhanced Delayed Feedback Model

4.1.1 Unbiased Learning Objective Under Observed Data. For observed data distribution $q^{obs}(\mathrm{x}, y)$, they have the following obvious relations with true data distribution $p(\mathrm{x}, y)$:

$$q^{obs}(y=1|\mathbf{x}) = p(y=1, d \leq w_o|\mathbf{x}), \tag{2}$$

$$q^{obs}(y=0|\mathbf{x}) = p(y=0|\mathbf{x}) + p(y=1, d > w_o|\mathbf{x}). \tag{3}$$

Based on the above two equations, we design the following objective for learning an unbiased model under the observed data:

$$\begin{aligned}\mathcal{L}_{DDFM}^{obs} &= -\mathbb{E}_{q^{obs}(\mathrm{x},y)}\big[y\log(f_\theta(\mathbf{x})) \\ &\quad + (1-y)\big((1-z_1)\log(f_\theta(\mathbf{x})) + z_1\log(1-f_\theta(\mathbf{x}))\big)\big],\end{aligned} \tag{4}$$

where latent variable z_1 is the real negative probability, denotes the probability that an observed negative is a real negative:

$$z_1 = \frac{p(y=0|\mathbf{x}))}{q^{obs}(y=0|\mathbf{x}))} = \frac{p(y=0|\mathbf{x}))}{p(y=0|\mathbf{x})) + p(y=1, d > w_o|\mathbf{x}))}. \tag{5}$$

Intuitively, since the observed negatives can be real negatives and fake negatives (i.e., delayed positives), z_1 is a soft label that can indicate whether an observed negative sample is a real negative or a fake negative. Moreover, this learning objective in Eq. (4) is fully based on the observed data distribution and thus can be directly optimized in practice. The unbiasedness of this loss is shown in the following theorem and the proof can be found in Appendix A.1.

THEOREM 4.1 (UNBIASED ESTIMATOR UNDER OBSERVED DATA). *The learning objective defined in Eq. (4) is unbiased in terms of the ideal learning objective in Eq. (1):* $\mathbb{E}_{q^{obs}(\mathrm{x},y)}\left[\mathcal{L}_{DDFM}^{obs}\right] = \mathcal{L}_{ideal}$.

We further analyze the convergence of this unbiased estimator by calculating the gradient of $\mathcal{L}_{DDFM}^{obs}$ with respect to f_θ as:

$$\begin{aligned}\frac{\partial\mathcal{L}_{DDFM}^{obs}}{\partial f_\theta} &= \frac{q^{obs}(y=0|\mathbf{x})z_1}{1-f_\theta(\mathbf{x})} - \frac{q^{obs}(y=1|\mathbf{x}) + q^{obs}(y=0|\mathbf{x})(1-z_1)}{f_\theta(\mathbf{x})} \\ &= \frac{f_\theta(\mathbf{x}) - p(y=1|\mathbf{x})}{f_\theta(\mathbf{x})(1-f_\theta(\mathbf{x}))}.\end{aligned}$$

We can see that $\partial\mathcal{L}_{DDFM}^{obs}/\partial f_\theta > 0$ when $f_\theta(\mathbf{x}) > p(y=1|\mathbf{x})$, and $\partial\mathcal{L}_{DDFM}^{obs}/\partial f_\theta < 0$ when $f_\theta(\mathbf{x}) < p(y=1|\mathbf{x})$. This observation demonstrates that our proposed unbiased estimator can consistently optimize in the correct gradient directions, resulting in the convergence of $f_\theta(\mathbf{x})$ towards $p(y=1|\mathbf{x})$ [1, 12].

4.1.2 Unbiased Learning Objective Under Duplicated Data. Similarly, we have the following relations between duplicated data distribution $q^{dup}(\mathrm{x}, y)$ and the true data distribution $p(\mathrm{x}, y)$ as:

$$q^{dup}(y=1|\mathbf{x}) = p(y=1, d > w_o|\mathbf{x}),$$
$$q^{dup}(y=0|\mathbf{x}) = p(y=0|\mathbf{x}).$$

Since only fake positives and real negatives will be duplicated, then $q^{dup}(y|\mathbf{x})$ should be re-normalized by dividing $p(y=0|\mathbf{x}) + p(y=1, d > w_o|\mathbf{x})$ as follows:

$$q^{dup}(y=1|\mathbf{x}) = \frac{p(y=1, d > w_o|\mathbf{x})}{p(y=0|\mathbf{x}) + p(y=1, d > w_o|\mathbf{x})}, \tag{6}$$

$$q^{dup}(y=0|\mathbf{x}) = \frac{p(y=0|\mathbf{x})}{p(y=0|\mathbf{x}) + p(y=1, d > w_o|\mathbf{x})}. \tag{7}$$

Again, relying on the above equations, we design the following objective for learning an unbiased model under the duplicated data:

$$\begin{aligned}\mathcal{L}_{DDFM}^{dup} &= -\mathbb{E}_{q^{dup}(\mathrm{x},y)}\big[y\log(f_\theta(\mathbf{x})), \\ &\quad + (1-y)\big((1-z_2)\log(f_\theta(\mathbf{x})) + z_2\log(1-f_\theta(\mathbf{x}))\big)\big],\end{aligned} \tag{8}$$

where latent variable z_2 is the observed negative probability:

$$z_2 = p(y=0|\mathbf{x})) + p(y=1, d > w_o|\mathbf{x})). \tag{9}$$

Note that $\mathcal{L}_{DDFM}^{dup}$ has the same form as $\mathcal{L}_{DDFM}^{obs}$ but with a different latent variable as they are designed under different types of data. It can be proved that Eq. (8) is also an unbiased estimator:

THEOREM 4.2 (UNBIASED ESTIMATOR UNDER DUPLICATED DATA). *The learning objective defined in Eq. (8) is unbiased in terms of the ideal learning objective in Eq. (1):* $\mathbb{E}_{q^{dup}(\mathrm{x},y)}\left[\mathcal{L}_{DDFM}^{dup}\right] = \mathcal{L}_{ideal}^{dup}$.

The proof of Theorem 4.2 is similar to that of Theorem 4.1 and can be found in Appendix A.2.

We also analyze the convergence of $\mathcal{L}_{DDFM}^{dup}$ in the same way as for $\mathcal{L}_{DDFM}^{obs}$:

$$\begin{aligned}\frac{\partial\mathcal{L}_{DDFM}^{dup}}{\partial f_\theta} &= \frac{q^{dup}(y=0|\mathbf{x})z_2}{1-f_\theta(\mathbf{x})} - \frac{q^{dup}(y=1|\mathbf{x}) + q^{dup}(y=0|\mathbf{x})(1-z_2)}{f_\theta(\mathbf{x})} \\ &= \frac{f_\theta(\mathbf{x}) - p(y=1|\mathbf{x})}{f_\theta(\mathbf{x})(1-f_\theta(\mathbf{x}))}.\end{aligned}$$

The conclusion is the same as above, that is, our proposed estimator $\mathcal{L}_{DDFM}^{dup}$ can consistently optimize in the correct gradient directions, resulting in the convergence of $f_\theta(\mathbf{x})$ towards $p(y=1|\mathbf{x})$.

4.1.3 Final Unbiased Learning Objective. Finally, we give the final unbiased learning objective of DDFM as follows:

$$\mathcal{L}_{DDFM} = \alpha\mathcal{L}_{DDFM}^{obs} + (1-\alpha)\mathcal{L}_{DDFM}^{dup}, \tag{10}$$

where $\alpha \in [0,1]$ is the co-efficient that can balance the two unbiased estimators (trade-off between model freshness and label correctness). Given the $\mathcal{L}_{DDFM}^{obs}$ and $\mathcal{L}_{DDFM}^{dup}$ are two unbiased losses against the ideal loss, it is obvious that the averaged loss $\mathcal{L}_{DDFM}$ is also an unbiased loss, i.e., $\mathbb{E}\left[\mathcal{L}_{DDFM}\right] = \mathcal{L}_{ideal}$.

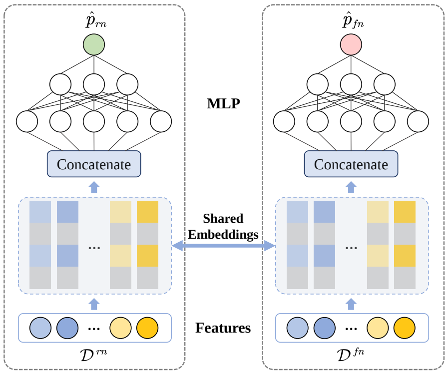


Figure 3: Overview of our shared-bottom architecture for estimating latent variables in DDFM.

4.2 Estimating Latent Variables and Learning

4.2.1 Estimating Latent Variables in DDFM. Given the two unbiased estimators, the remaining challenge is how to estimate the latent variables z_1 and z_2 in Eq. (4) and Eq. (8). We first denote $p_{rn}(\mathbf{x}) := \frac{p(y=0|\mathbf{x})}{p(y=0|\mathbf{x})+p(y=1,d>w_o|\mathbf{x})}$ and $p_{fn}(\mathbf{x}) := p(y = 1, d > w_o|\mathbf{x})$, where $p_{rn}(\mathbf{x})$ is the probability of an observed negative is a real negative and $p_{fn}(\mathbf{x})$ is the probability of a sample is a fake negative (i.e., delayed positive). Then we can rewrite the z_1 and z_2 according to Eq. (5) and Eq. (9) as:

$$z_1 = p_{rn}(\mathbf{x}), \tag{11}$$

$$z_2 = p(y = 0|\mathbf{x}) + p_{fn}(\mathbf{x}). \tag{12}$$

Thus, based on the above two equations, we can get the estimation of z_1 and z_2 as:

$$\hat{z}_1 = \hat{p}_{rn}(\mathbf{x}), \tag{13}$$

$$\hat{z}_2 = [1 - f_\theta(\mathbf{x})] + \hat{p}_{fn}(\mathbf{x}), \tag{14}$$

where $[\cdot]$ means the stop gradient calculation and the estimation $\hat{p}_{rn}(\mathbf{x})$ and $\hat{p}_{fn}(\mathbf{x})$ can be directly obtained from two classifiers, respectively. In Eq. (13) and Eq. (14), we only use addition operations for estimating the weights. By avoiding multiplication or division operations, we effectively mitigate the issue of high variance. We will further show the advantages of our estimation method by comparing it with existing approaches in Section 4.3.2.

Subsequently, we adopt an Embedding&MLP network architecture for modeling these two classifiers. As shown in Figure 3, these two classifiers employ a shared embedding module to better represent all input features [16, 30]. For training these two classifiers, we can construct two corresponding training datasets by adopting a mask mechanism as follows:

- $\mathcal{D}^{rn}$: The observed immediate positives are excluded, then the real negatives are labeled as 1 and the fake negatives (i.e., delayed positives) are labeled as 0.
- $\mathcal{D}^{fn}$: The fake negatives (i.e., delayed positives) are labeled as 1 and the other samples are labeled as 0.

4.2.2 Model Learning. Based on the final unbiased objective defined in Eq. (10) and the estimated latent variables $\hat{z}_1$ and $\hat{z}_2$, the final loss for deriving the model parameters θ in our proposed DDFM is formally defined as:

$$\begin{aligned}\widehat{\mathcal{L}}_{DDFM} = -\sum_{\mathcal{D}} \Big\{ &\alpha o \big[y\ell^+ + (1-y)\big((1-\hat{z}_1)\ell^+ + \hat{z}_1\ell^-\big)\big] \\ &+ (1-\alpha)(1-o)\big[y\ell^+ + (1-y)\big((1-\hat{z}_2)\ell^+ + \hat{z}_2\ell^-\big)\big]\Big\},\end{aligned} \tag{15}$$

where $o \in \{0, 1\}$ is a binary indicator, $o = 1$ indicates a training sample is from newly observed data, and $o = 0$ signifies that the training sample is from duplicated data. And for simplicity, we denote $\ell^+ := \log(f_\theta(\mathbf{x}))$ and $\ell^- := \log(1 - f_\theta(\mathbf{x}))$.

4.3 Discussion

4.3.1 Bias Analysis Under the Estimated Latent Variables. Though Theorem 4.1 and Theorem 4.2 demonstrate the unbiasedness of our proposed two unbiased estimators, the estimated latent variables can still be inaccurate due to the optimization process and potentially introduce bias to the final CVR prediction. The following theorem analyzes the biases of two estimators in DDFM with the estimated latent variables $\hat{z}_1$ and $\hat{z}_2$ (see Appendix A.3 for proofs).

THEOREM 4.3 (BIAS OF DDFM WITH ESTIMATED LATENT VARIABLES). *Given the estimated latent variables $\hat{z}_1$ and $\hat{z}_2$, the bias of two estimators in DDFM are:*

$$\text{Bias}\left[\mathcal{L}^{obs}_{DDFM} \mid \hat{z}_1\right] = \left|\left(\frac{\hat{z}_1}{z_1} - 1\right) p(y = 0|\mathbf{x})(\ell^+ - \ell^-)\right|,$$

$$\text{Bias}\left[\mathcal{L}^{dup}_{DDFM} \mid \hat{z}_2\right] = \left|\left(\frac{\hat{z}_2}{z_2} - 1\right) p(y = 0|\mathbf{x})(\ell^+ - \ell^-)\right|,$$

where $\ell^+ := \log(f_\theta(\mathbf{x})$ and $\ell^- := \log(1 - f_\theta(\mathbf{x}))$.

Theorem 4.3 indicates that better estimation of $\hat{z}_1$ and $\hat{z}_2$ (more closely align with $\hat{z}_1$ and $\hat{z}_2$) can effectively reduce bias. Thus, we will further analyze different ways of estimating $\hat{z}_1$ and $\hat{z}_2$ in the next section. Additionally, it is observed that DDFM demonstrates a reduced bias as the true conditional CVR (i.e., $1 - p(y = 0|\mathbf{x})$) increases. This implies that a higher true CVR leads to the incorporation of more certain positive samples, thereby alleviating the challenges posed by delayed feedback.

4.3.2 Different Estimating Ways of Latent Variables. Indeed, we have observed a relationship between the latent variables z_1 and z_2: $z_1 z_2 = p(y = 0|\mathbf{x})$. As a result, there are two alternative approaches for estimating the latent variables using only one classifier. The first approach involves solely estimating $\hat{p}_{rn}$ as:

$$\hat{z}_1 = \hat{p}_{rn}(\mathbf{x}), \tag{16}$$

$$\hat{z}_2 = \frac{[1 - f_\theta(\mathbf{x})]}{\hat{p}_{rn}(\mathbf{x})}. \tag{17}$$

The second approach only depends on the estimation of $\hat{p}_{fn}$, which is also employed in a similar manner in previous works [3, 4]:

$$\hat{z}_1 = \frac{[1 - f_\theta(\mathbf{x})]}{[1 - f_\theta(\mathbf{x})] + \hat{p}_{fn}(\mathbf{x})}, \tag{18}$$

$$\hat{z}_2 = [1 - f_\theta(\mathbf{x})] + \hat{p}_{fn}(\mathbf{x}). \tag{19}$$

However, both of these methods have complex fractions such that a small deviation will result in serious bias. Thus, as we mentioned in Section 4.2.1, we adopt Eq. (16) and Eq. (19) to directly estimate the latent variables with two classifiers, which can reduce the error accumulation and high variance (see comparisons in Section 5.3.2).

Table 2: Statistics of Criteo and Industrial datasets. "M" means million and "K" means thousand.

Datasets	# Users	# Items	# Categorical features	# Continuous features	# Conversions	# Samples	# Average CVR	Duration
Criteo	-	5443	9	8	3619801	15898883	0.2269	60 days
Industrial	20M	52K	4	21	769K	29M	0.0265	11 days

4.3.3 Advantages Compared to Importance Weighting Methods. Importance weighting [1] methods are widely employed in previous delayed feedback studies to weigh the loss [3, 4, 12, 34, 36]. However, most of these methods typically assume that $p(\mathbf{x}) = q(\mathbf{x})$, which is always not satisfied in prior works when conflating both the newly observed data and duplicated data together. In contrast, our approach does not make this assumption and treats newly observed streaming data and duplicated early data separately.

Furthermore, importance weighting methods often suffer from high variance because the importance weight $\frac{p(y|\mathbf{x})}{q(y|\mathbf{x})}$ is the division of two probabilities. On the contrary, we directly predict $\hat{p}_{rn}(\mathbf{x})$ and $\hat{p}_{fn}(\mathbf{x})$ for the estimation of $\hat{z}_1$ and $\hat{z}_2$ with only addition, rather than multiplication or division, in which way the error will not be amplified. Additionally, $\hat{p}_{rn}(\mathbf{x})$ and $\hat{p}_{fn}(\mathbf{x})$ are bounded within the range of $[0, 1]$, which further control the error deviation.

5 EXPERIMENTS

In this section, we conduct extensive experiments on two large-scale real-world datasets to verify the effectiveness of DDFM. The source codes are shared at https://github.com/KID-22/DDFM.

5.1 Experimental Settings

5.1.1 Datasets. We evaluate the effectiveness of our proposed DDFM on both large-scale public and industrial datasets. The statistics of the two datasets are shown in Table 2.

Criteo[1]: Criteo dataset consists of 16 million Criteo live traffic data over a period of 60 days. Each sample records click and conversion (if occurs) timestamps, hashed categorical features, and continuous features. For the attribution window w_a, we follow the common practice in previous works and set $w_a = 30$ days to get the true conversion label [2–4, 34].

Industrial: Industrial dataset contains about 29 million interactions in 11 days from a real advertising system. The attribution window w_a was set as 3 days to set the true label for each sample.

5.1.2 Streaming Evaluation Protocols. To better verify the performance of different methods for the streaming CVR prediction task, we adopt the streaming evaluation protocols commonly employed in prior studies [3, 4, 34]. Specifically, we first divide all the datasets into two parts based on the click timestamp. The first part of the data is utilized for pre-training a well-initialized model for the following streaming training and evaluation. To prevent label leakage, we assign labels as 0 to instances where the conversion occurs in the second part of the data. The second part is further divided into multiple subsets by hour according to the click timestamp. Following the online streaming manner of industrial systems, the models are trained on the t-th hour data and then tested on the subsequent $t+1$-th hour. Consequently, we report the weighted average metrics across different hours on streaming data.

[1] https://labs.criteo.com/2013/12/conversion-logs-dataset/

5.1.3 Baselines. We compare our proposed DDFM with the following state-of-the-art (SOTA) baseline methods:

Vanilla is trained on the streaming data without duplicated samples. The samples converted in the waiting window are labeled as positives; otherwise negatives.

FSIW [36] is trained on the steaming data without duplicated samples and using the FSIW loss.

Vanilla-DP is trained on the streaming data but with duplicated delayed positive samples. The delayed positives are duplicated and ingested into the training pipeline upon conversion.

FNW [12] is trained on the streaming data without a waiting window and using the fake negative weighted loss. Samples are immediately trained with negative labels and then duplicated with positives upon conversion.

FNC [12] is trained on the same steaming data as FNW but using the fake negative calibration loss.

ES-DFM [34] is trained on the steaming data with the same duplicated mechanism as Vanilla-DP but using the ES-DFM loss.

DEFER [4] is trained on the streaming data but with duplicated both delayed positive and real negative samples and using the DEFER loss.

DEFUSE [3] is trained on the streaming data with the same duplicated mechanism as Vanilla-DP but using the DEFUSE loss.

We also report the performance of **Oracle** as the *upper bound* for delayed feedback modeling. The Oracle is trained using the true label instead of the observed label, i.e., it knows whether the conversion will occur in the future at the click timestamp, which is infeasible in real-world industrial systems. All the methods are continuously fine-tuned on top of the pre-trained model that is trained on the first part of the data for the streaming evaluation.

5.1.4 Metrics. We adopt **AUC**, **PRAUC** and **NLL** to evaluate the performance, which are widely-used evaluation metrics for CVR estimation in previous studies [2–5, 31, 34]. Following [26], we also report Kolmogorov-Smirnov (**KS**) score at the best threshold on the ROC curve for comprehensive evaluation.

To further clearly demonstrate the relative improvement of each model over the Vanilla, following previous works [3, 4, 32, 34, 40], we also report the $\mathrm{RI}_{\mathrm{AUC}}$, $\mathrm{RI}_{\mathrm{KS}}$, $\mathrm{RI}_{\mathrm{PRAUC}}$, and $\mathrm{RI}_{\mathrm{NLL}}$ as:

$$\mathrm{RI}_{Metric} = \frac{Metric(\text{evaluated model}) - Metric(\text{Vanilla})}{Metric(\text{Oracle}) - Metric(\text{Vanilla})} \times 100\%,$$

where $Metric$ could be AUC, KS, PRAUC, and NLL. Obviously, a higher relative improvement (closer to 100%) indicates better performance of the evaluated model.

5.1.5 Implementation Details. The period of the first part of data for pre-training is set as 30 days and 5 days for the Criteo and Industrial datasets, respectively. To ensure a fair comparison, we carefully

Table 3: The overall performance of DDFM and the baselines on Criteo and Industrial datasets. Note that Oracle is infeasible in practice and only serves as an upper bound. The best performance and the second best performance methods are denoted in bold and underlined fonts, respectively. '$*$' indicates statistical significance with p-value< 0.05 compared to the best baseline under the two-sided t-test. '$\downarrow$' denotes that lower is better for the metric NLL, while higher is better for the remaining metrics.

Dataset	Metric	Methods without duplicating			Methods with duplicating						
		Oracle	Vanilla	FSIW	Vanilla-DP	FNW	FNC	ES-DFM	DEFER	DEFUSE	DDFM (Ours)
Criteo	AUC	84.24%	81.09%	81.52%	83.02%	83.55%	83.51%	83.77%	83.74%	83.78%	**84.13%***
	RI_{AUC}	100.00%	0.00%	13.65%	61.27%	78.10%	76.83%	85.08%	84.13%	85.40%	**96.51%***
	KS	53.13%	47.76%	48.55%	51.31%	52.10%	52.04%	52.44%	52.37%	52.47%	**52.92%***
	RI_{KS}	100.00%	0.00%	14.71%	66.11%	80.82%	79.70%	87.15%	85.85%	87.71%	**96.09%***
	PRAUC	64.31%	59.59%	60.22%	62.13%	63.03%	62.63%	63.51%	63.24%	63.49%	**64.14%***
	RI_{PRAUC}	100.00%	0.00%	13.35%	53.81%	72.88%	64.41%	83.05%	77.33%	82.63%	**96.40%***
	NLL $\downarrow$	38.93%	54.99%	52.49%	41.26%	40.48%	41.78%	39.63%	41.49%	39.67%	**39.46%***
	RI_{NLL}	100.00%	0.00%	15.57%	85.49%	90.35%	82.25%	95.64%	84.06%	95.39%	**96.70%***
Industrial	AUC	81.29%	76.64%	79.56%	77.42%	79.79%	79.77%	79.82%	79.73%	79.87%	**80.50%***
	RI_{AUC}	100.00%	0.00%	62.80%	16.77%	67.74%	67.31%	68.39%	66.45%	69.46%	**83.01%***
	KS	47.25%	41.83%	44.72%	43.22%	44.96%	44.91%	44.99%	44.95%	45.08%	**45.91%***
	RI_{KS}	100.00%	0.00%	53.32%	25.65%	57.75%	56.83%	58.30%	57.56%	59.96%	**75.28%***
	PRAUC	10.90%	9.84%	10.05%	10.12%	10.13%	10.12%	10.14%	10.10%	10.16%	**10.36%***
	RI_{PRAUC}	100.00%	0.00%	19.81%	26.42%	27.36%	26.42%	28.30%	24.53%	30.19%	**49.06%***
	NLL $\downarrow$	10.25%	12.50%	10.77%	11.52%	10.47%	10.48%	10.46%	10.55%	10.45%	**10.35%***
	RI_{NLL}	100.00%	0.00%	76.89%	43.56%	90.22%	89.78%	90.67%	86.67%	91.11%	**95.56%***

tune the hyper-parameters for all compared models. Following the previous studies [3, 4, 33, 34], all the model architecture is a simple MLP with fixed hidden units of $[256, 256, 128]$, and Leaky ReLU [17] activation functions are applied. We use the Adam [11] optimizer for optimizing all methods. The learning rate is tuned among $\{1e{-}3, 5e{-}4, 1e{-}4, 5e{-}5, 1e{-}5, 5e{-}6, 1e{-}6\}$, and the weight decay is searched from $\{1e{-}5, 1e{-}6, 1e{-}7\}$. We set waiting window w_o as 0.25 hour as suggested in previous methods [3, 4, 34].

5.2 Experimental Results

We conduct all experiments five times with different random seeds and report the averaged results of DDFM and all compared methods on Criteo and Industrial datasets in Table 3. Based on the results, we have the following observations and conclusions:

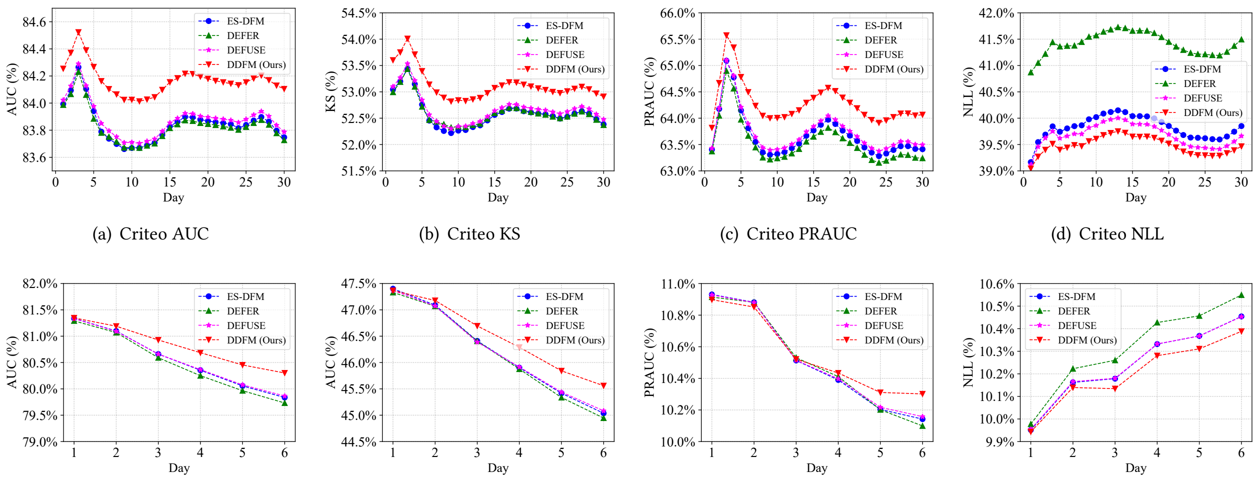
All the methods show significant improvements over Vanilla on both the Criteo and Industrial datasets, demonstrating their effectiveness in mitigating fake negatives caused by delayed feedback. Among these methods, our proposed DDFM consistently outperforms all the baselines across all evaluation metrics on both datasets. Particularly, compared to the best baseline, DDFM achieves average gains of 0.45% in AUC and 11.11% in RI_{AUC} on the Industrial dataset. Similarly, on the Industrial dataset, DDFM achieves average gains of 0.63% in AUC and 13.55% in RI_{AUC}. It is worth noting that even a 0.1% offline AUC gain is considered remarkable and can lead to a significant online promotion for the industrial scenarios [3, 16, 28]. Therefore, these results highlight the superior performance of DDFM compared to the other baselines.

Compared to methods without duplicate mechanisms (Vanilla and FSIW), almost all the methods that utilize duplicating sample mechanisms (Vanilla-DP, FNW, FNC, ES-DFM, DEFER, DEFUSE, and DDFM) demonstrate significantly improved performance on both datasets. This finding confirms the effectiveness of incorporating early samples, which provide more reliable certain information. In particular, ES-DFM, DEFER, and DEFUSE consistently outperform other baselines. Their better performance can be attributed to the inclusion of various techniques that mitigate the deviation between the observed data distribution and the ground-truth data distribution. Our proposed DDFM takes it a step further by treating newly observed data and duplicated data separately, resulting in dual unbiased CVR estimators. Additionally, DDFM assigns different weights to the two data sources, achieving a better trade-off between label correctness and model freshness. As a result, our proposed DDFM approaches the performance upper bound (i.e., Oracle) more closely, with RI metrics reaching closer to 100%.

5.3 Further Analysis

5.3.1 Performance w.r.t. the Streaming Evaluation Process. To evaluate the stability of our proposed DDFM, we analyze the cumulative averaged results of all metrics for DDFM and other SOTA baselines (ES-DFM, DEFER, and DEFUSE) during the streaming evaluation process on both Criteo and Industrial datasets. As illustrated in Figure 4, we can clearly observe that DDFM consistently outperforms the SOTA baselines in terms of all metrics throughout the entire streaming process on both datasets. Notably, as the evaluation period extends, we observe a widening performance improvement gap between our proposed DDFM and the other methods, especially on the Industrial dataset. This indicates that DDFM becomes increasingly more advantageous over time, outperforming the SOTA baselines by a larger margin.

5.3.2 Performance w.r.t. the Estimation of $\hat{z}_1$ and $\hat{z}_2$. In our proposed DDFM, there are two latent variables $\hat{z}_1$ and $\hat{z}_2$ that need to be estimated. In Section 4.2.1, we derive an addition-only estimation formula in Eq. (13) and Eq. (14) and propose a shared-bottom network with double classifiers to estimate these two latent variables. As discussed in Section 4.3.2, there are two other estimation methods for latent variables that require only a single classifier. To analyze the performance of different estimation methods, we compare our method (denoted as "Shared") with its three variants: (i) estimating with double classifiers but without the shared-bottom architecture (denoted as "w/o Shared"), (ii) estimating using Eq. (16)



(a) Criteo AUC (b) Criteo KS (c) Criteo PRAUC (d) Criteo NLL

(e) Industrial AUC (f) Industrial KS (g) Industrial PRAUC (h) Industrial NLL

Figure 4: Performance comparison of cumulative averaged metrics w.r.t. the streaming process on Criteo and Industrial datasets. The streaming evaluation period are 30 days and 6 days for the Criteo dataset and Industrial dataset, respectively.

Table 4: Performance comparisons between different estimation methods for latent variables $\hat{z}_1$ and $\hat{z}_2$ on Criteo and Industrial datasets. '*' indicates the best performance among different estimation methods.

Estimation Method		Single Classifier		Double Classifiers	
Dataset	Metric	$\hat{p}_{rn}$	$\hat{p}_{fn}$	w/o Shared	Shared
Criteo	AUC	84.05%	84.07%	84.08%	**84.13%***
	RI_{AUC}	93.97%	94.60%	94.92%	**96.51%***
	KS	52.81%	52.86%	52.87%	**52.92%***
	RI_{KS}	94.04%	94.97%	95.16%	**96.09%***
	PRAUC	63.99%	63.93%	64.08%	**64.14%***
	RI_{PRAUC}	93.22%	91.95%	95.13%	**96.40%***
Industrial	AUC	80.17%	80.18%	80.32%	**80.50%***
	RI_{AUC}	75.91%	76.13%	79.14%	**83.01%***
	KS	45.24%	45.22%	45.77%	**45.91%***
	RI_{KS}	62.92%	62.55%	72.69%	**75.28%***
	PRAUC	10.26%	10.25%	10.31%	**10.36%***
	RI_{PRAUC}	39.62%	38.68%	44.34%	**49.06%***

and Eq. (17) via a single classifier for $\hat{p}_{rn}$ (denoted as "$\hat{p}_{rn}$"), (iii) estimating using Eq. (18) and Eq. (19) via a single classifier for $\hat{p}_{fn}$ (denoted as "$\hat{p}_{fn}$"). This set of experiments can also be regarded as ablation studies for our latent variable estimation method.

Table 4 shows the evaluation results on the Criteo and Industrial datasets. As we can see, the double classifiers with shared-bottom architecture achieve the best performance in all cases, particularly on the Industrial dataset. Additionally, the double classifiers methods outperform the single classifier methods. These observations further verify our discussions in Section 4.3.2 , where we emphasized that our derived double classifiers estimation formula utilizes addition operations instead of multiplication or division, thereby avoiding high variance issues. Moreover, the shared-bottom network architecture helps improve the estimation, as the shared embedding can mitigate data sparsity.

Table 5: Performance comparisons of different methods enhanced by applying our unbiased learning objective for observed data on Criteo dataset.

Method	AUC	RI_{AUC}	KS	RI_{KS}	PRAUC	RI_{PRAUC}
Vanilla	81.09%	0.00%	47.76%	0.00%	59.59%	0.00%
+ DDFM	**82.90%**	**57.46%**	**51.16%**	**63.31%**	**62.20%**	**55.30%**
Vanilla-DP	83.02%	61.27%	51.31%	66.11%	62.13%	53.81%
+ DDFM	**83.06%**	**62.54%**	**51.38%**	**67.41%**	**62.52%**	**62.08%**

5.3.3 Applying Unbiased Loss of DDFM to Other Methods. In this section, we investigate whether our unbiased learning objective $\mathcal{L}^{obs}_{DDFM}$ can enhance existing methods with different sampling duplicated mechanisms. We conduct experiments to evaluate the performance improvement of Vanilla and Vanilla-DP when their losses are replaced with our unbiased learning objective $\mathcal{L}^{obs}_{DDFM}$ for the observed data. For Vanilla, which is trained on streaming data without duplicated samples, we directly replace its loss with $\mathcal{L}^{obs}_{DDFM}$. For Vanilla-DP, which has duplicated delayed positives, we apply $\mathcal{L}^{obs}_{DDFM}$ for the observed data, while keeping the weight of the loss for duplicated data unchanged. These enhanced methods are denoted as "+DDFM". The results on the Criteo dataset are presented in Table 5. We observe that both Vanilla and Vanilla-DP achieve improved performance across all metrics after applying our unbiased learning objective for the observed data. This demonstrates the effectiveness of our unbiased loss, which utilizes a soft label z_1 to assign weights to observed negatives, thereby mitigating the impact of fake negatives caused by delayed feedback.

5.3.4 Performance w.r.t. the Co-efficient α. As described in Section 4.1.3, DDFM incorporates a co-efficient $\alpha \in [0, 1]$ in Eq. (10) to balance the loss between newly observed data and duplicated early data, thereby achieving a trade-off between model freshness and label correctness. Specifically, a larger value of α (i.e., > 0.5) assigns

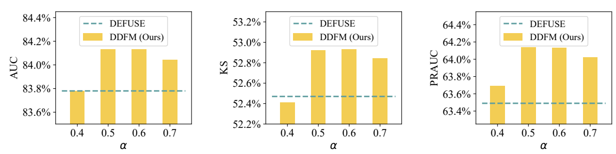


Figure 5: The impact of co-efficient α on Criteo dataset.

more weight to newly observed data, improving model freshness. However, as the coefficient α increases, delayed positives introduce more false information, leading to a loss in label correctness. To validate this analysis, we conduct experiments by varying α in the range of $\{0.4, 0.5, 0.6, 0.7\}$ on the Criteo dataset. Figure 5 presents the AUC, PRAUC, and KS scores in terms of different α. As expected, we observe that both larger and smaller values of α resulted in reduced performance, particularly when α was less than 0.5. This observation confirms the importance of model freshness in streaming CVR prediction, given the dynamic shifts in data distribution. It further validates the necessity of continuous training to ensure model freshness, which is a common practice in industrial systems.

6 CONCLUSION

In this paper, we study the delayed feedback problem in streaming conversion rate prediction. Compared with previous works, our proposed DDFM distinguishes between newly observed streaming data and duplicated early data, enabling a more accurate and fine-grained delayed feedback modeling. DDFM introduces dual unbiased CVR estimators in the same form via re-weighting with latent variables. To efficiently estimate these latent variables, we employ a shared-bottom network architecture. Moreover, we provide theoretical analysis for the unbiasedness and convergence of our proposed DDFM and discuss the advantages compared to existing methods. Extensive experiments conducted on two large-scale datasets from real-world advertising systems demonstrate the superior performance of our approach.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program of China (2019YFE0198200), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Intelligent Social Governance Interdisciplinary Platform, the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No.22XNH027), Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, and Public Computing Cloud, Renmin University of China. Supported by funds for building world-class universities (disciplines) of Renmin University of China.

A APPENDIX: PROOFS

For brevity, the conditions on $\mathbf{x}$ are all omitted in the following proofs, e.g., $p(y=1|\mathbf{x})$ is substituted as $p(y=1)$.

A.1 Proof of Theorem 4.1

Proof. Taking into account the definition of z_1 in Eq. (5) and the relationship equations given by Eq. (2) and Eq. (3), the expectation of $\mathcal{L}^{obs}_{DDFM}$ under the observed data distribution $q^{obs}(\mathbf{x}, y)$ is

$$
\begin{aligned}
&\mathbb{E}_{q^{obs}}\left[\mathcal{L}^{obs}_{DDFM}\right] = -\Big[\mathbb{E}_{q^{obs}}[y]\log(f_\theta(\mathbf{x})) \\
&\quad + \mathbb{E}_{q^{obs}}[1-y]\Big((1-z_1)\log(f_\theta(\mathbf{x})) + z_1\log(1-f_\theta(\mathbf{x}))\Big)\Big] \\
&= -\Big[q^{obs}(y=1)\log(f_\theta(\mathbf{x})) \\
&\quad + q^{obs}(y=0)\Big((1-z_1)\log(f_\theta(\mathbf{x}) + z_1\log(1-f_\theta(\mathbf{x}))\Big)\Big] \\
&= -\Big(p(y=1, d\le w_o)\log(f_\theta(\mathbf{x})) + p(y=1, d> w_o)\log(f_\theta(\mathbf{x})) \\
&\quad + p(y=0)\log(1-f_\theta(\mathbf{x}))\Big) \\
&= -(p(y=1)\log(f_\theta(\mathbf{x})) + p(y=0)\log(1-f_\theta(\mathbf{x}))) = \mathcal{L}_{ideal}.
\end{aligned}
$$

□

A.2 Proof of Theorem 4.2

Proof. The proof is similar to the proof of Theorem 4.1 in Appendix A.1. Based on the definition of z_2 in Eq. (9) and the relationship equations given by Eq. (6) and Eq. (7), we have

$$
\begin{aligned}
&\mathbb{E}_{q^{dup}}\left[\mathcal{L}^{dup}_{DDFM}\right] = -\Big[\mathbb{E}_{q^{dup}}[y]\log(f_\theta(\mathbf{x})) \\
&\quad + \mathbb{E}_{q^{dup}}[1-y]\Big((1-z_2)\log(f_\theta(\mathbf{x})) + z_2\log(1-f_\theta(\mathbf{x}))\Big)\Big] \\
&= -\Big(\frac{p(y=1, d>w_o)}{p(y=0)+p(y=1,d>w_o)}\log(f_\theta(\mathbf{x})) \\
&\quad + \frac{p(y=0)(p(y=1)-p(y=1,d>w_o))}{p(y=0)+p(y=1,d>w_o)}\log(f_\theta(\mathbf{x})) \\
&\quad + p(y=0)\log(1-f_\theta(\mathbf{x}))\Big) \\
&= -\Big(\frac{p(y=1,d>w_o)(1-p(y=0))+p(y=0)p(y=1)}{p(y=0)+p(y=1,d>w_o)}\log(f_\theta(\mathbf{x})) \\
&\quad + p(y=0)\log(1-f_\theta(\mathbf{x}))\Big) \\
&= -(p(y=1)\log(f_\theta(\mathbf{x})) + p(y=0)\log(1-f_\theta(\mathbf{x}))) = \mathcal{L}_{ideal}.
\end{aligned}
$$

□

A.3 Proof of Theorem 4.3

Proof. Let $\ell^+ := \log(f_\theta(\mathbf{x})$ and $\ell^- := \log(1-f_\theta(\mathbf{x}))$, the bias of $\mathcal{L}^{obs}_{DDFM}$ under the estimated latent variable $\hat{z}_1$ can be calculated as:

$$
\begin{aligned}
&\mathrm{Bias}\left[\mathcal{L}^{obs}_{DDFM} \mid \hat{z}_1\right] \\
&:= \left|\mathbb{E}\left[\widehat{\mathcal{L}}^{obs}_{DDFM} \mid \hat{z}_1\right] - \mathcal{L}_{ideal}\right| = \Big| - \Big(p(y=1,d\le w_o)\ell^+ \\
&\quad + (p(y=0)+p(y=1,d>w_o))\left((1-\hat{z}_1)\ell^+ + \hat{z}_1\ell^-\right)\Big) \\
&\quad + \Big(p(y=1)\ell^+ + p(y=0)\ell^-\Big)\Big| = \left|\Big(\frac{\hat{z}_1}{z_1}-1\Big)p(y=0)(\ell^+-\ell^-)\right|.
\end{aligned}
$$

Similar to the above, we have

$$
\begin{aligned}
&\mathrm{Bias}\left[\mathcal{L}^{dup}_{DDFM} \mid \hat{z}_2\right] \\
&:= \left|\mathbb{E}\left[\widehat{\mathcal{L}}^{dup}_{DDFM} \mid \hat{z}_2\right] - \mathcal{L}_{ideal}\right| = \Big| - \Big(\frac{p(y=1,d>w_o)}{p(y=0)+p(y=1,d>w_o)}\ell^+ \\
&\quad + \frac{p(y=0)}{p(y=0)+p(y=1,d>w_o)}\left((1-\hat{z}_2)\ell^+ + \hat{z}_2\ell^-\right)\Big) \\
&\quad + \Big(p(y=1)\ell^+ + p(y=0)\ell^-\Big)\Big| = \left|\Big(\frac{\hat{z}_2}{z_2}-1\Big)p(y=0)(\ell^+-\ell^-)\right|.
\end{aligned}
$$

□

REFERENCES

[1] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

[2] 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.

[3] 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.

[4] 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.

[5] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1569–1577.

[6] Yilin Hou, Guangming Zhao, Chuanren Liu, Zhonglin Zu, and Xiaoqiang Zhu. 2021. Conversion prediction with delayed feedback: A multi-task learning approach. In *2021 IEEE International Conference on Data Mining (ICDM)*. 191–199.

[7] 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.

[8] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*.

[9] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. 2013. Online learning under delayed feedback. In *International Conference on Machine Learning*. 1453–1461.

[10] Michael Jugovac, Dietmar Jannach, and Mozhgan Karimi. 2018. Streamingrec: a framework for benchmarking stream-based news recommenders. In *Proceedings of the 12th ACM conference on recommender systems*. 269–273.

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

[12] 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.

[13] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[14] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past erformance data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 768–776.

[15] 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.

[16] 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.

[17] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, Vol. 30. 3.

[18] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. 2015. The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[19] Panayotis Mertikopoulos, Zhengyuan Zhou, et al. 2020. Gradient-free online learning in games with delayed rewards. In *International Conference on Machine Learning*.

[20] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. 2018. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*. 4105–4113.

[21] Yuta Saito, Gota Morisihta, and Shota Yasui. 2020. Dual learning algorithm for delayed conversions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1849–1852.

[22] Yumin Su, Liang Zhang, Quanyu Dai, Bo Zhang, Jinyao Yan, Dan Wang, Yongjun Bao, Sulong Xu, Yang He, and Weipeng Yan. 2021. An attention-based model for conversion rate prediction with delayed feedback via post-click calibration. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 3522–3528.

[23] Surya T Tokdar and Robert E Kass. 2010. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 1 (2010), 54–60.

[24] Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*. 437–448.

[25] Claire Vernade, Olivier Cappé, and Vianney Perchet. 2017. Stochastic Bandit Models for Delayed Conversions. In *Conference on Uncertainty in Artificial Intelligence*.

[26] Hao Wang, Tai-Wei Chang, Tianqiao Liu, Jianmin Huang, Zhichao Chen, Chao Yu, Ruopeng Li, and Wei Chu. 2022. ESCM2: Entire Space Counterfactual Multi-Task Model for Post-Click Conversion Rate Estimation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 363–372.

[27] Jun Wang, Weinan Zhang, Shuai Yuan, et al. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. *Foundations and Trends® in Information Retrieval* 11 (2017), 297–435.

[28] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-Scale Learning to Rank Systems. In *Proceedings of the Web Conference 2021*. 1785–1797.

[29] Zenan Wang, Carlos Carrion, Xiliang Lin, Fuhua Ji, Yongjun Bao, and Weipeng Yan. 2022. Adaptive Experimentation with Delayed Binary Feedback. In *Proceedings of the ACM Web Conference 2022*. 2247–2255.

[30] 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.

[31] 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.

[32] Ling Yan, Wu-jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale ctr prediction in display advertising. In *International Conference on Machine Learning*. PMLR, 802–810.

[33] 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.

[34] 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.

[35] Shota Yasui and Masahiro Kato. 2022. Learning Classifiers under Delayed Feedback with a Time Window Assumption. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2286–2295.

[36] Shota Yasui, Gota Morishita, Fujita Komei, and Masashi Shibata. 2020. A feedback shift correction in predicting conversion rates under delayed feedback. In *Proceedings of The Web Conference 2020*. 2740–2746.

[37] Yuya Yoshikawa and Yusaku Imai. 2018. A nonparametric delayed feedback model for conversion rate prediction. *arXiv preprint arXiv:1802.00255* (2018).

[38] Xiao Zhang, Sunhao Dai, Jun Xu, Zhenhua Dong, Quanyu Dai, and Ji-Rong Wen. 2022. Counteracting user attention bias in music streaming recommendation via reward modification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2504–2514.

[39] Xiao Zhang, Haonan Jia, Hanjing Su, Wenhan Wang, Jun Xu, and Ji-Rong Wen. 2021. Counterfactual reward modification for streaming recommendation with delayed feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 41–50.

[40] 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.

[41] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2191–2200.