UKD: Debiasing Conversion Rate Estimation via Uncertainty-regularized Knowledge Distillation

Zixuan Xu[‡], Penghui Wei[‡], Weimin Zhang, Shaoguo Liu, Liang Wang and Bo Zheng[*]

Alibaba Group

{xuzixuan.xzx,wph242967,dutan.zwm,shaoguo.lsg,liangbo.wl,bozheng}@alibaba-inc.com

ABSTRACT

In online advertising, conventional post-click conversion rate (CVR) estimation models are trained using clicked samples. However, during online serving the models need to estimate for all impression ads, leading to the sample selection bias (SSB) issue. Intuitively, providing reliable supervision signals for unclicked ads is a feasible way to alleviate the SSB issue. This paper proposes an uncertainty-regularized knowledge distillation (UKD) framework to debias CVR estimation via distilling knowledge from unclicked ads. A teacher model learns click-adaptive representations and produces pseudo-conversion labels on unclicked ads as supervision signals. Then a student model is trained on both clicked and unclicked ads with knowledge distillation, performing uncertainty modeling to alleviate the inherent noise in pseudo-labels. Experiments on billion-scale datasets show that UKD outperforms previous debiasing methods. Online results verify that UKD achieves significant improvements.

CCS CONCEPTS

• **Information systems → Online advertising**.

KEYWORDS

CVR Estimation, Debiasing, Distillation, Uncertainty Modeling

1 INTRODUCTION

In online advertising systems, post-click conversion rate (CVR) estimation is to predict the probability of conversion after an ad click event, and predicted CVR score is a key factor in many applications such as the ranking procedure and smart bidding. In many marketing scenarios, conversion is the ultimate goal of advertisers, and thus CVR estimation plays an important role.

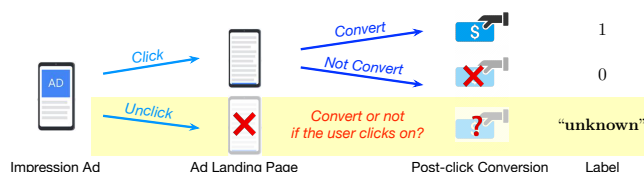---

[‡]Co-first authorship. [*]Corresponding author.

Figure 1: User click and conversion behaviors in advertising.

Figure 1 shows user click and conversion behaviors in online advertising. If users click on an impression ad, they will arrive at a landing page that shows the detailed information of the ad, and then users might take conversion actions or not. Obviously, only clicked ads have post-click conversion labels, and for the ads that users do *not* click on, we will never know whether post-click conversion actions will happen. Due to the lack of ground-truth labels for unclicked ads, conventional CVR estimation models are typically trained using clicked ads only, but the models need to predict CVR for entire impression ads (including both clicked and unclicked ones) during online serving. The problem that there is a gap between training space (i.e., *click space*) and inference space (i.e., *entire impression space*) is called sample selection bias (SSB) [30]. These models may be *biased* for unclicked ads, because their training procedures do not learn much knowledge from unclick ads.

The representative methods probing into the SSB issue can be divided into two categories: 1) auxiliary task learning based [16, 27], and 2) counterfactual learning based methods [9, 31]. Ma et al. [16] propose to incorporate two auxiliary tasks that can be trained in entire impression space to indirectly learn an entire space CVR estimator. However, for unclicked ads, the model tends to optimize the predicted CVR scores to zeros (see § 2.3.3 for proof) while their actual conversion labels are "unknown". Zhang et al. [31] employ counterfactual learning to produce a theoretically unbiased CVR estimator, but the training procedure of CVR task does not explicitly take unclicked ads into account. In all, current methods cannot essentially debias CVR models for unclicked ads, and the SSB issue in CVR estimation is still an open problem.

To learn entire space CVR models that can accurately estimate CVR for all impression ads, a feasible way is to provide reliable pseudo-conversion labels for unclicked ads as supervision signals. In this way, the training procedure of CVR models can explicitly utilize both clicked ads (with labels from logs) and unclicked ads (with pseudo-labels). Thus, these models can benefit from learning with unclicked ads compared to the ones trained on clicked ads only. To achieve this, the key is how to produce reliable pseudo-conversion labels for unclicked ads when we can only access ground-truth labels of clicked ads, as well as how to learn an accurate entire space CVR estimator with both ground-truth labels and pseudo-labels. For the former, consider that there is a discrepancy between

the data distributions of clicked and unclicked ads (which causes the SSB issue) [31], we propose to formulate pseudo-conversion label generation as an unsupervised domain adaptation problem. Click/unclick space is regarded as source/target domain, and our goal is to obtain pseudo-labels for unlabeled unclicked ads (target domain) based on labeled clicked ads (source domain). For the latter one, consider that the confidence of unclicked ads' pseudo-labels is inferior to clicked ads' ground-truth labels, we propose to reduce the negative impact of inherent noise existed in pseudo-labels by modeling their uncertainty during training.

Motivated by the above considerations, in this paper we propose **U**ncertainty-regularized **K**nowledge **D**istillation (**UKD**), which aims to debias CVR estimation via distilling knowledge from unclicked ads. UKD's overall workflow contains a click-adaptive teacher model that produces pseudo-conversion labels for unclicked ads, and an uncertainty-regularized student model that can effectively distill the knowledge in unclicked ads learned by the teacher. Specifically, to produce supervision signals for unclicked ads, the teacher learns click-adaptive representations for impression ads using domain adaptation, and its predicted CVR scores on unclicked ads are taken as their pseudo-conversion labels. Then the student can learn from both clicked ads (with ground-truth labels) and unclicked ads (with pseudo-labels from teacher), and also performs uncertainty estimation to pseudo-labels for alleviating the inherent noise in them. For each unclicked ad, our student estimates its pseudo-label's uncertainty and dynamically adjust the weight of its CVR loss during training to weaken its negative impact. Experimental results on billion-scale datasets show that UKD outperforms previous state-of-the-art methods. We have deployed UKD in Alibaba advertising platform, and online results verify that UKD achieves significantly improvements. The main contributions of this work are:

- We propose uncertainty-regularized knowledge distillation (UKD) to debias CVR models via learning from unclicked ads. It employs a click-adaptive teacher to generate pseudo-conversion labels for unclicked ads, and then trains a student model that takes both clicked and unclicked ads into account.
- Our student model performs uncertainty estimation to pseudo-labels generated by the teacher, alleviating the inherent noise to reduce the negative impact during distillation.
- Experimental results on public and large-scale production datasets show that UKD outperforms the state-of-the-art methods. Online experiments further verify that it achieves significantly improvements on core metrics.

2 PREREQUISITES

2.1 Problem Definition

In online advertising systems, we can log user feedbacks on impression ads to train models for estimating CTR (click-through rate), CVR and CTCVR (click-through conversion rate). Let $\mathcal{D} = \{(x, y_{click}, y_{conv}, y_{pv\text{-}conv})\}$ denote the collected dataset of impression ads. For each impression sample, x denotes the feature information, which is usually a high-dimensional vector consisting of one-hot encodings from user, ad and context fields. y_{click}, y_{conv} and $y_{pv\text{-}conv}$ are the binary labels of click event, post-click conversion event and post-view conversion event respectively.

According to the values of click labels, we divide all samples in $\mathcal{D}$ into two subsets: clicked samples $\mathcal{D}_{click} = \{\mathcal{D} \mid y_{click} = 1\}$ (their conversion labels y_{conv} are observed) and unclicked samples $\mathcal{D}_{unclick} = \{\mathcal{D} \mid y_{click} = 0\}$ (all conversion labels are "unknown").

CTR estimation is to predict the probability of click event, i.e., $p_{CTR} = p(y_{click} = 1 \mid x)$. CVR estimation is to predict the probability of conversion if a user has clicked on an ad, i.e., $p_{CVR} = p(y_{pv\text{-}conv} = 1 \mid y_{click} = 1, x) = p(y_{conv} = 1 \mid x)$. And for CTCVR estimation, we have $p_{CTCVR} = p(y_{pv\text{-}conv} = 1 \mid x) = p_{CTR} \cdot p_{CVR}$.

Conventional CVR estimation models employ similar techniques as in CTR estimation task, such as logistic regression [18], factorization machines [12, 17] and deep neural networks (DNN) [4, 8]. Next we introduce both conventional models and entire space models.

2.2 Base CVR Models Trained in Click Space

2.2.1 ***Single-Task CVR Model***. Conventional CVR estimation models are trained on the clicked data $\mathcal{D}_{click}$. Let $\hat{p}_{CVR} = F_v(x)$ denotes a single-task CVR model, where $\hat{p}_{CVR} \in (0, 1)$ is the predicted CVR score for the impression ad x. $F_v(\cdot)$ represents a network that consists of a feature embedding layer and several dense layers. The objective is formulated based on cross-entropy loss $\ell(\cdot, \cdot)$:

$$\min_{F_v} \frac{1}{|\mathcal{D}_{click}|} \sum_{\mathcal{D}_{click}} \ell\left(y_{conv}, F_v(x)\right) . \tag{1}$$

2.2.2 ***Joint Estimation of CVR and CTR***. To alleviate the data sparsity issue in CVR task, jointly optimizing CVR and CTR estimation tasks is a commonly-used way, because the CTR task is trained on impression ads $\mathcal{D}$ and has much richer samples than the CVR task [16, 25]. The joint model contains a shared feature embedding layer, as well as two separate dense blocks to predict CVR and CTR scores respectively. Let $\hat{p}_{CVR} = F_v(x)$ and $\hat{p}_{CTR} = F_c(x)$ denote the predicted CVR and CTR scores of the joint model (note that $F_v(\cdot)$ and $F_c(\cdot)$ share the feature embedding layer), the objective of the joint model is:

$$\min_{F_v, F_c} \frac{1}{|\mathcal{D}_{click}|} \sum_{\mathcal{D}_{click}} \ell\left(y_{conv}, F_v(x)\right) + \gamma \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \ell\left(y_{click}, F_c(x)\right) \tag{2}$$

where γ is a trade-off hyperparameter.

2.2.3 ***Limitations***. The training process of the single-task CVR model does not learn from unclicked ads, and the joint model only incorporates such information by means of the shared embeddings from CTR task. Thus their predicted CVR scores in unclicked space may have a non-negligible deviation because there is a discrepancy between the data distributions of click and unclick ads.

2.3 Entire Space CVR Estimation Models

2.3.1 ***Auxiliary Task Learning based Models***. Ma et al. [16] incorporate two auxiliary tasks, click-through rate (CTR) and click-through conversion rate (CTCVR), that can be trained in entire impression space to indirectly learn an entire space CVR estimator. The model has the same architecture as the joint model (i.e., $\hat{p}_{CVR} = F_v(x)$ and $\hat{p}_{CTR} = F_c(x)$), while the objective is to minimize the cross-entropy loss on CTCVR and CTR estimation:

$$\min_{F_v, F_c} \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \left(\ell\left(y_{pv\text{-}conv}, F_c(x) \cdot F_v(x)\right) + \gamma \ell\left(y_{click}, F_c(x)\right) \right) \tag{3}$$

where $F_c(x) \cdot F_v(x) = \hat{p}_{CTR} \cdot \hat{p}_{CVR}$ is the predicted CTCVR score. With the help of learning CTCVR estimation, the network $F_v(x)$ for CVR can learn from unclicked ads, alleviating the SSB issue.

2.3.2 Counterfactual Learning based Models. Counterfactual learning offers a way to tackle the missing-not-at-random problem [1, 2, 5, 15, 21, 23, 26, 29]. Several recent studies [9, 31] employ counterfactual learning, such as inverse propensity score (IPS) and doubly robust (DR) estimators, to debias CVR estimation. An IPS-based method utilizes the predicted CTR score as propensity of the CVR loss on clicked ads to achieve an theoretically unbiased CVR estimator. The optimization objective is:

$$\min_{F_v, F_c} \frac{1}{|\mathcal{D}_{click}|} \sum_{\mathcal{D}_{click}} \frac{1}{F_c(x)} \ell\left(y_{conv}, F_v(x)\right) + \gamma \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \ell\left(y_{click}, F_c(x)\right)$$

(4)

A DR-based method further learns an imputation model $F_i(\cdot)$ that estimates the CVR loss of each unclicked ad. The CVR task and imputation task are alternately trained, where the $F_i(\cdot)$ is trained on clicked data $\mathcal{D}_{click}$. Refer to [9, 31] for details.

2.3.3 Limitations. Although the auxiliary task learning based models can learn from unclicked ads with the learning of CTCVR estimation task, they have two main limitations:

- For a clicked ad (i.e., $y_{click} = 1$), if its post-view conversion label $y_{pv\text{-}conv} = 0$, this ad is a positive sample of CTR task as well as an negative sample of CTCVR task, which may result in gradient conflict to the two learning tasks.
- For an unclicked ad (i.e., $y_{click} = 0$ and y_{conv} is "unknown"), the models tend to optimize the predicted CVR scores to zeros. The proof is given here: The loss of CTCVR estimation is $\ell = -y_{pv\text{-}conv} \log(\hat{p}_{CTR} \cdot \hat{p}_{CVR}) - (1 - y_{pv\text{-}conv}) \log(1 - \hat{p}_{CTR} \cdot \hat{p}_{CVR})$. For an unclicked ad whose $(y_{click}, y_{conv}, y_{pv\text{-}conv}) = (0, \text{unknown}, 0)$, the gradient to $\hat{p}_{CVR}$ is $\frac{\partial \ell}{\partial \hat{p}_{CVR}} = \frac{\hat{p}_{CTR}}{1 - \hat{p}_{CTR} \cdot \hat{p}_{CVR}}$. Note that $\hat{p}_{CTR} \in (0, 1)$ and $\hat{p}_{CVR} \in (0, 1)$, thus the gradient is always positive, which means that it tends to optimize $\hat{p}_{CVR}$ of unclicked ads to 0, but the actual label is "unknown".

The counterfactual learning based models have achieved state-of-the-art performance. However, the limitations of them contain:

- For IPS-based models, the training procedure of the CVR task is on clicked data and does not explicitly take unclicked ads into account.
- For DR-based models, although the imputation task is used to estimate CVR loss of each unclicked ad, its learning procedure still utilizes clicked data only, and thus the imputation task is lack of accurate supervision.

3 PROPOSED METHOD

We propose an **U**ncertainty-regularized **K**nowledge **D**istillation (**UKD**) framework, which aims to debias CVR estimation via distilling knowledge from unclicked ads. The basic idea is that we build an entire space CVR estimation model by producing reliable pseudo-conversion labels for unclicked ads as supervision signals.

Fig. 2 illustrates the overall workflow of UKD, which consists of a click-adaptive teacher model that produces pseudo-conversion labels for unclicked ads, and an uncertainty-regularized student model that distills the valuable knowledge learned by the teacher

to perform entire space CVR estimation. Next, we elaborate our UKD from the details of teacher and student respectively.

3.1 Click-Adaptive Teacher Model

The goal of the teacher in UKD is to produce pseudo-conversion labels for unclicked ads $\mathcal{D}_{unclick}$ under the condition that we can only access ground-truth conversion labels of clicked ads $\mathcal{D}_{click}$, facilitating the entire space training of CVR estimation task.

There is a discrepancy between the feature distributions of clicked and unclicked samples. To possess the ability of accurate inference on unclicked ads, the teacher model may not learn feature representations specific to clicked samples, but learn click-adaptive representations. We propose to tackle pseudo-conversion label generation from the perspective of unsupervised domain adaptation, where the source/target domain is clicked/unclicked space, inspired by [3]. In this way, the problem is formulated as producing reliable pseudo-conversion labels for unlabeled unclicked ads ($\mathcal{D}_{unclick}$, as target domain) given labeled clicked ads ($\mathcal{D}_{click}$, as source domain).

3.1.1 Click-Adaptive Representation Learning. Specifically, our click-adaptive teacher model adopts adversarial learning [7, 24] that introduces a click discriminator to mitigate inconsistent feature distributions of clicked/unclicked samples during training.

Model Architecture As illustrated in the left part of Fig. 2, the click-adaptive teacher model consists of a feature representation learner $T_f(\cdot)$, a CVR predictor $T_p(\cdot)$ and a click discriminator $T_d(\cdot)$. Formally, $T_f(\cdot)$ takes each sample's feature x as input to learn its dense representation $\boldsymbol{h}^{(T)}$, where $T_f(\cdot)$ contains a feature embedding layer and several dense layers. $T_p(\cdot)$ intends to estimate the sample's CVR score $\hat{p}_{CVR}^{(T)}$. It consists of several dense layers and a softmax function on top to produce probability distribution.

With the aim of making the feature representation $\boldsymbol{h}^{(T)}$ more click-adaptive to facilitate pseudo-conversion label generation on unclicked ads, the teacher model introduces a click discriminator $T_d(\cdot)$ to classify each sample's domain (i.e., clicked or unclicked) based on the sample's representation $\boldsymbol{h}^{(T)}$. The intuition is that if a strong click discriminator cannot predict a sample's domain label correctly, its representation $\boldsymbol{h}^{(T)}$ is click-adaptive.

Overall, the forward process of the teacher model is:

$$\boldsymbol{h}^{(T)} = T_f(x)$$
$$\boldsymbol{p}_{conv}^{(T)} = \text{softmax}\left(T_p\left(\boldsymbol{h}^{(T)}\right)\right) = \left(\hat{p}_{CVR}^{(T)}, 1 - \hat{p}_{CVR}^{(T)}\right) \quad (5)$$
$$\boldsymbol{p}_d = \text{softmax}\left(T_d\left(\boldsymbol{h}^{(T)}\right)\right)$$

where $\boldsymbol{p}_{conv}^{(T)}$ is the predicted CVR distribution ($\hat{p}_{CVR}^{(T)}$ is the predicted CVR score). $\boldsymbol{p}_d$ is the predicted domain distribution.

Adversarial Learning To learn click-adaptive representations, given an impression ad x, its representation $\boldsymbol{h}^{(T)}$ learned by $T_f(\cdot)$ aims to confuse the click discriminator and maximize the domain classification loss, while the click discriminator $T_d(\cdot)$ itself aims to minimize the domain classification loss to be a strong classifier. The teacher is optimized via an adversarial learning procedure:

$$\min_{T_f, T_p} \mathcal{L}_{CVR}^{(T)} = \frac{1}{|\mathcal{D}_{click}|} \sum_{\mathcal{D}_{click}} \ell(y_{conv}, \boldsymbol{p}_{conv}^{(T)}) \quad (6)$$
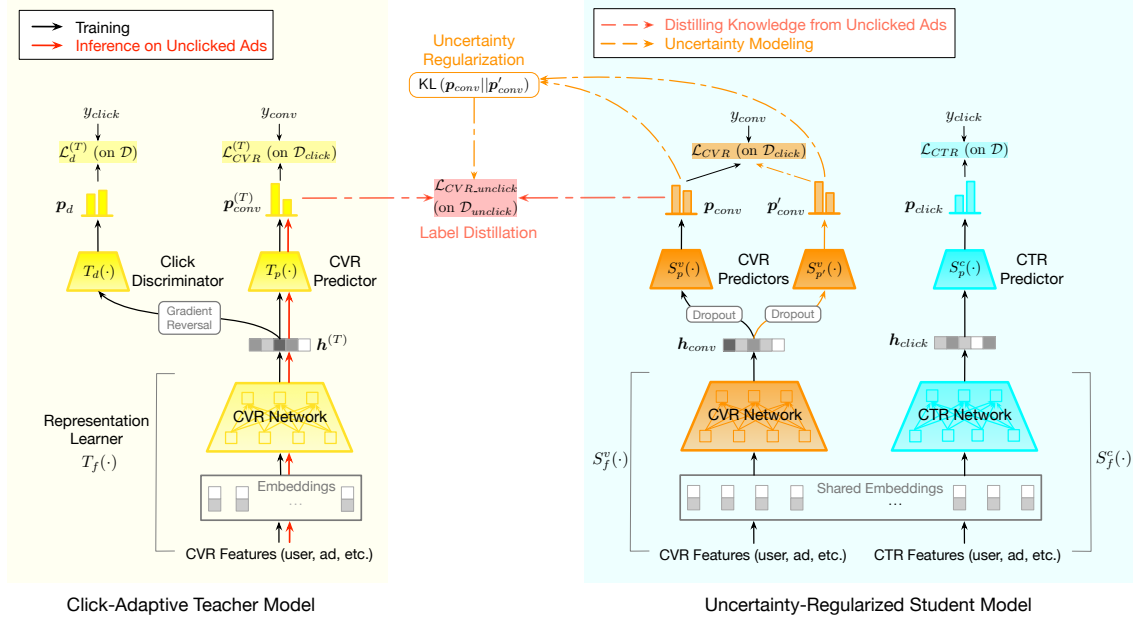
Figure 2: Overview of uncertainty-regularized knowledge distillation (UKD) for debiasing post-click conversion rate estimation (better viewed in color). It contains a click-adaptive teacher model that provides pseudo-conversion labels for unclicked ads, and an uncertainty-regularized student model that is trained on entire impression space.

$$\max_{T_f} \min_{T_d} \mathcal{L}_d^{(T)} = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \ell(y_{click}, \boldsymbol{p}_d) \quad (7)$$

The first equation minimizes the loss of CVR estimation to optimize the learner $T_f(\cdot)$ and the predictor $T_p(\cdot)$. The second one means that the learner $T_f(\cdot)$ makes the representations of clicked and unclicked ads indistinguishable, while the click discriminator $T_d(\cdot)$ is optimized to better distinguish clicked ads from the unclicked ones. In practice we implement it via gradient reversal [7].

The learned representations from two domains are effectively aligned when a well-trained discriminator cannot distinguish them. Therefore, based on the click-adaptive representations, the teacher is able to make reliable CVR estimation on unclicked ads.

3.1.2 **Produce Pseudo-Conversion Labels for Unclicked Ads.** The trained teacher model performs inference on each unclicked ad in $\mathcal{D}_{unclick}$ to produce the predicted CVR distribution $\boldsymbol{p}_{conv}^{(T)}$ as the pseudo-conversion label, where the forward process of inference only includes $T_f(\cdot)$ and $T_p(\cdot)$, without the need of $T_d(\cdot)$.

We use $\widetilde{\mathcal{D}}_{unclick} = \{(x, \boldsymbol{p}_{conv}^{(T)})\}$ to denote the unclicked samples coupled with pseudo-conversion labels, which will be utilized to train an entire space CVR model.

3.2 Uncertainty-Regularized Student Model

Based on unclicked ads' pseudo-conversion labels learned by the click-adaptive teacher model, our UKD framework further builds a student model based on knowledge distillation [11], which learns from both clicked ads (with ground-truth labels) and unclicked ads (with pseudo-labels) to perform entire space CVR estimation. Compared to the models that are trained using clicked samples only, our model alleviates the SSB issue via explicitly taking unclicked samples into account during training.

We elaborate distillation strategy that can guide the student model to mine the valuable knowledge learned by the teacher. Due to the inherent noise existed in teacher predictions, the confidence of unclicked ads' pseudo-labels is inferior to clicked ads' ground-truth conversion labels. To address this, we propose an uncertainty-regularized student that reduces the negative impact of noise by modeling pseudo-labels' uncertainty during distillation. Next we detail the distillation strategy of our student model with its two key modules: label distillation and uncertainty regularization.

3.2.1 **Base Student Model: Label Distillation.** We start from introducing a base student model, which is jointly learned with both CVR and CTR estimation tasks as the joint model in § 2.2.2.

Model Architecture The base student consists of two feature representation learners (i.e., $S_f^v(\cdot)$ for CVR task and $S_f^c(\cdot)$ for CTR task), a CVR predictor $S_p^v(\cdot)$ that outputs the predicted CVR score, and a CTR predictor $S_p^c(\cdot)$ that outputs the predicted CTR score.

Formally, the two representation learners $S_f^v(\cdot)$ and $S_f^c(\cdot)$ share the feature embedding layer, and each learner has several dense layers to learn representation $\boldsymbol{h}_{conv}$ / $\boldsymbol{h}_{click}$ w.r.t. the CVR / CTR task. Further, each of the two predictors $S_p^v(\cdot)$ and $S_p^c(\cdot)$ consists of several dense layers with a softmax function to produce probability distribution for estimating CVR / CTR score. The forward process of the base student model is:

$$\boldsymbol{h}_{conv} = S_f^v(x), \quad \boldsymbol{h}_{click} = S_f^c(x)$$
$$\boldsymbol{p}_{conv} = \mathsf{softmax}\left(S_p^v(\boldsymbol{h}_{conv})\right) = (\hat{p}_{CVR}, 1 - \hat{p}_{CVR}) \quad (8)$$
$$\boldsymbol{p}_{click} = \mathsf{softmax}\left(S_p^c(\boldsymbol{h}_{click})\right) = (\hat{p}_{CTR}, 1 - \hat{p}_{CTR})$$

where $\boldsymbol{p}_{conv}$ denotes the predicted CVR distribution ($\hat{p}_{CVR}$ is the predicted CVR score). $\boldsymbol{p}_{click}$ and $\hat{p}_{CTR}$ can be similarly defined.

Distilling Knowledge from Unclicked Ads   With the help of unclicked ads' pseudo-conversion labels learned by the teacher, our student is optimized in entire impression space to alleviate the SSB issue. The objective of CVR estimation task is:

$$\mathcal{L}_{CVR} = \underbrace{\sum_{\mathcal{D}_{click}} \ell(y_{conv}, \boldsymbol{p}_{conv})}_{\mathcal{L}_{CVR\_click}} + \alpha \underbrace{\sum_{\widetilde{\mathcal{D}}_{unclick}} \ell\left(\boldsymbol{p}_{conv}^{(T)}, \boldsymbol{p}_{conv}\right)}_{\mathcal{L}_{CVR\_unclick}} \quad (9)$$

where $\mathcal{L}_{CVR\_click}$ and $\mathcal{L}_{CVR\_unclick}$ are the CVR task losses on clicked and unclicked ads, and the hyperparameter α balances the weight of two terms.[1] The base student model's optimization objective is the sum of two losses about CVR task and CTR task:

$$\mathcal{L}_{student} = \mathcal{L}_{CVR} + \gamma \mathcal{L}_{CTR} \quad (10)$$

where $\mathcal{L}_{CTR} = \sum_{\mathcal{D}} \ell(y_{click}, \boldsymbol{p}_{click})$.

3.2.2 ***Uncertainty-regularized Student: Alleviate Noise***. It is expected that the confidence of unclicked ads' pseudo-conversion labels is inferior to that of clicked ads' ground-truth conversion labels, because the latter is obtained from user feedback logs while the former is produced by the teacher model. Due to inherent noise existed in teacher's predictions, the unclicked samples having noisy pseudo-labels mislead the student model's training procedure.

For effective knowledge distillation from unclicked ads, the key is two-fold: (i) identify noisy and unreliable unclicked samples, and (ii) reduce their negative impacts during distillation. To identify them, we resort to estimate the uncertainty of unclicked samples' pseudo-labels, where a higher uncertainty value indicates worse reliability. By using *high uncertainty* as a measure of noisy unclicked samples, we can reduce the negative impacts of such samples via simply assigning *low weights* to their CVR losses, which avoids misleading the student model's distillation procedure. Based on the above considerations, we propose an uncertainty-regularized student model. It estimates uncertainty to each unclicked ad's pseudo-label, and dynamically adjusts weights to unclicked ads' CVR losses according to uncertainty levels, reducing the negative impact of noise. [2]

How to Identify Noisy Samples   To design a student model that possesses the ability of identifying noisy pseudo-labels, we first conduct an experiment to explore: *on clean samples and noisy samples, what difference will a CVR model perform?*

We use clicked dataset $\mathcal{D}_{click} = \{(x, y_{conv})\}$ to run such experiment, because all labels in it are known and we can synthesize noisy dataset as well as controlling the proportion of $\frac{\# \ clean \ samples}{\# \ noisy \ samples}$. We add noise to obtain a noisy dataset $\mathcal{D}'_{click}$ by randomly choosing $k\%$ of positive samples in $\mathcal{D}_{click}$ and converting the labels from 1 to 0 (to keep the ratio of positive samples unchanged, we also convert the same size of negative samples' labels from 0 to 1).

The studies of learning from noisy data [10, 13, 22, 32] reveal that the inconsistency of two neural models' predictions on noisy training samples is usually larger than that on clean samples. Based on such guidance, we use the noisy dataset $\mathcal{D}'_{click}$ to train a CVR model, which contains an embedding layer, a representation learner
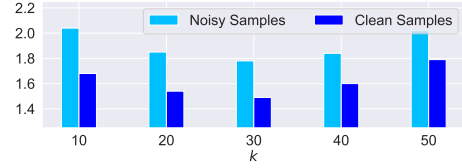
Figure 3: KL-divergence of two predictors' outputs ($\times 10^{-3}$).

and *two seperate CVR predictors* on top (the objective is the average of two predictors' cross-entropy losses). After training, we observe the averaged KL-divergence of two predictors' outputs on (1) noisy samples and (2) clean samples, respectively. Figure 3 shows the results with different k, and the KL-divergence on noisy samples is larger than that on clean samples for each value of k. This phenomenon can be explained that if a sample's label is clean, both two predictors easily fit the label during training, where the corresponding two loss values are small and the two predictions are similar. In contrast, if a label is noisy, the fitting procedures of two predictors will be hard to be consistent, and they tend to produce inconsistent predictions that result in a large variance. The experimental results verify that the inconsistency of two CVR predictors can be utilized to identify noisy training samples.

Uncertainty Modeling   Motivated by the above experiment, our uncertainty-regularized student model contains two CVR predictors $S_p^v(\cdot)$ and $S_{p'}^v(\cdot)$ to simultaneously estimate CVR scores (as illustrated in the right part of Fig. 2), and then models the uncertainty as the inconsistency of them. Formally, let $\boldsymbol{p}_{conv}$ and $\boldsymbol{p}'_{conv}$ denote the predicted distributions from the two CVR predictors. We formulate the uncertainty as the KL-divergence of two predictions:

$$\boldsymbol{p}_{conv} = S_p^v\left(\text{dropout}(\boldsymbol{h}_{conv})\right), \ \boldsymbol{p}'_{conv} = S_{p'}^v\left(\text{dropout}'(\boldsymbol{h}_{conv})\right)$$

$$\text{KL}\left(\boldsymbol{p}_{conv}||\boldsymbol{p}'_{conv}\right) = \boldsymbol{p}_{conv} \log \frac{\boldsymbol{p}_{conv}}{\boldsymbol{p}'_{conv}}$$

$$(11)$$

where we apply independent dropout operations to the learned representation $\boldsymbol{h}_{conv}$ to increase the discrepancy of two predictors.

Distillation with Uncertainty Regularization   Based on the estimated uncertainty for each unclicked sample, we reduce the negative impacts of noisy unclicked samples during distillation via dynamically adjusting uncertainty-based weights to CVR losses. Compared to the base student model, now the distillation procedure from unclicked ads is regularized by pseudo-label's uncertainty, alleviating the inherent noise existed in the teacher's predictions.

For each unclicked sample, we weight its original CVR loss with a factor $\exp\left(-\lambda \cdot \text{KL}\left(\boldsymbol{p}_{conv}||\boldsymbol{p}'_{conv}\right)\right) \in (0, 1]$ as uncertainty regularization. The factor is inversely related to uncertainty (λ controls its scale). Thus, the loss $\mathcal{L}_{CVR\_unclick}$ is reformulated as:

$$\sum_{\widetilde{\mathcal{D}}_{unclick}} \exp\left(-\lambda \cdot \text{KL}\left(\boldsymbol{p}_{conv}||\boldsymbol{p}'_{conv}\right)\right) \cdot \ell\left(\boldsymbol{p}_{conv}^{(T)}, \boldsymbol{p}_{conv}\right) \quad (12)$$

If a sample has high uncertainty, the factor returns a small value to down-weigh its CVR loss. If the uncertainty is close to 0, the factor tends to 1, and such student devolves to the base student model.

We also add a loss term $\sum_{\widetilde{\mathcal{D}}_{unclick}} \text{KL}\left(\boldsymbol{p}_{conv}||\boldsymbol{p}'_{conv}\right)$ that acts as a regularization for uncertainty estimation. Without minimizing such term, a large KL-divergence leads to small label distillation loss, which will make the model poorly optimized.

Table 1: Statistics of five datasets.

Dataset	# Impression	# Click	# Conversion	# Time Interval
Ali-CCP	84M	3.4M	18k	N/A
EC-Small	0.18B	26M	88k	39 Days
EC-Large	0.51B	95M	0.3M	72 Days
LS-Small	0.28B	15M	0.2M	31 Days
LS-Large	0.75B	37M	0.5M	92 Days

4 EXPERIMENTS

In this section, we conduct both offline and online experiments, and intend to answer the following research questions:

- **RQ1** (offline performance): Does our proposed UKD outperform the state-of-the-art CVR estimation models?
- **RQ2** (teacher's utility): Is the teacher model necessary for debiasing CVR estimation models, and does the choice of teacher model affect the performance of our UKD?
- **RQ3** (student's utility): Does the distillation strategy of uncertainty regularization effectively help the student model?
- **RQ4** (model analysis): Does our UKD benefit from incorporating more unclicked samples during distillation?
- **RQ5** (online performance): Does our UKD achieve improvements when we deploy it to industrial advertising system?

4.1 Experimental Setup

4.1.1 Datasets. Offline experiments are conducted on a public available dataset Ali-CCP, and four large-scale production datasets from a leading advertising platform. Table 1 lists the statistics.

Public Dataset The Ali-CCP dataset [16] is the benchmark of CVR estimation, collected from user traffic logs in Taobao.

Production Datasets We further collect four large-scale production datasets from Alibaba advertising platform. Specifically, we collected 3-month consecutive user feedback logs from two different marketing scenarios: the first scenario is named *EC*, which contains e-commerce ads for attracting potential customers. The second scenario is named *LS*, in which the ads are about local life services. We organize them into four datasets: *EC-Small/Large* and *LS-Small/Large*. See Table 1 for their statistics.

4.1.2 Competitors of CVR Estimation. We compare our *UKD* with the following strong baselines. Except the first model, all the rest models are trained with impression dataset $\mathcal{D}$:

- *SingleCVR* (§ 2.2.1) is a single-task network that estimates $\hat{p}_{CVR}$, and is trained on clicked samples $\mathcal{D}_{click}$.
- *Joint* (§ 2.2.2) is a multi-task model that estimates both CVR and CTR. The CTR task is trained on impression data $\mathcal{D}$, and the CVR task is trained on clicked data $\mathcal{D}_{click}$.
- *ESMM* (§ 2.3.1) is an entire space model that learns CTR and CTCVR estimation [16]. It is optimized in impression space $\mathcal{D}$, where the predicted CTCVR is equal to $\hat{p}_{CTR} \cdot \hat{p}_{CVR}$.
- *Division* is a variant of *ESMM*, which formulates the CVR estimation task as $p_{CVR} = p_{CTCVR}/p_{CTR}$ [16]. Compared to *ESMM*, its two dense blocks produce $\hat{p}_{CTR}$ and $\hat{p}_{CTCVR}$.
- *CFL* (§ 2.3.2) employs counterfactual learning and achieves the state-of-the-art performance [9, 31]. Here we implement the model in [31] as our competitor because we find that its performance is superior and stable among counterfactual learning based CVR models.

4.1.3 Evaluation Metrics. We use AUC and NLL (a.k.a., LogLoss) as evaluation metrics, where the former reflects ranking ability on candidates and the latter measures fitting ability of predicted scores. Specifically, (i) AUC_{CVR} and NLL_{CVR} denote the metrics of CVR estimation on clicked samples in test set, because only clicked ads have post-click conversion labels y_{conv} for evaluation. (ii) AUC_{CTCVR} and NLL_{CTCVR} denote the metrics of CTCVR estimation on entire impression samples (i.e., the whole test set), because all samples have post-view conversion labels $y_{pv\text{-}conv}$ for evaluation. Following [16], we use this metric to reflect the CVR model performance of alleviating the SSB issue. For each competitor, we compute predicted CTCVR score as $\hat{p}_{CTCVR} = \hat{p}_{CTR} \cdot \hat{p}_{CVR}$, where $\hat{p}_{CVR}$ is estimated by the competitor, and $\hat{p}_{CTR}$ is from the same independently trained CTR model.

We further design two new metrics, named Debiased-AUC and Debiased-NLL (D-AUC and D-NLL for short), to evaluate the performance of entire space CVR estimation using clicked samples only. Formally, the conventional AUC_{CVR} metric on clicked samples is:

$$\frac{\sum_{i \in \mathcal{D}_{click}^+, j \in \mathcal{D}_{click}^-} \mathbb{I}(\hat{p}_{CVR}(i) > \hat{p}_{CVR}(j))}{|\mathcal{D}_{click}^+| \cdot |\mathcal{D}_{click}^-|} \quad (13)$$

where $\mathcal{D}_{click}^+$ and $\mathcal{D}_{click}^-$ denote the sets of positive and negative samples respectively, $\hat{p}_{CVR}(i)$ is the predicted CVR score of sample i, and $\mathbb{I}(\cdot) \in \{0, 1\}$ is indicator function. Inspired by the idea of inverse propensity score estimator [20, 21], which utilizes $\hat{p}_{CTR}$ as propensity score to induce an unbiased estimate of true prediction error (i.e., $\mathbb{E}_{\mathcal{D}_{click}}\left[\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \frac{1}{\hat{p}_{CTR}} \cdot y_{click} \cdot \ell\left(y_{conv}, \hat{p}_{CVR}\right) \right]$), we assign each sample i with a weight $\frac{1}{\hat{p}_{CTR}(i)}$ to compute D-AUC_{CVR}:

$$\frac{\sum\limits_{i \in \mathcal{D}_{click}^+, j \in \mathcal{D}_{click}^-} \frac{1}{\hat{p}_{CTR}(i)} \cdot \frac{1}{\hat{p}_{CTR}(j)} \cdot \mathbb{I}(\hat{p}_{CVR}(i) > \hat{p}_{CVR}(j))}{\left(\sum_{i \in \mathcal{D}_{click}^+} \frac{1}{\hat{p}_{CTR}(i)}\right) \cdot \left(\sum_{j \in \mathcal{D}_{click}^-} \frac{1}{\hat{p}_{CTR}(j)}\right)} \quad (14)$$

We can similarly define the metric D-NLL_{CVR} by modifying NLL_{CVR} (equations are listed in Appendix A.2).

4.2 Performance Comparison (RQ1)

Table 2 and Table 3 show the comparative results of all CVR models on production and public datasets respectively. On all datasets, *SingleCVR* performs poorly, which indicates that the models trained using clicked samples only may not be suitable for CVR estimation. The training procedure of the *Joint* model incorporates unclicked samples via a shared embedding layer between CVR and CTR tasks, and it consistently outperforms *SingleCVR*. Thus CVR estimation models can benefit from sharing embeddings between the two tasks.

ESMM is a strong baseline, which alleviates the SSB issue via learning CTR and CTCVR estimation in impression space as auxiliary tasks to indirectly produce an entire space CVR estimator. It outperforms *Joint* on production datasets and achieves the second-best performance on the *LS* datasets. This demonstrates that learning with CTCVR estimation task is an effective way to exploit unclicked samples. Although *Division* has the same motivation with *ESMM*, it performs worse than *Joint*. We suggest that the reason is the instability of division operation, which may introduce unstable numerical range that does harm to model optimization. *CFL* is the state-of-the-art by producing a theoretically unbiased

Table 2: Results on four large-scale production datasets. Bold/Underlined values denote the best/second-best results.

| Method | Dataset: EC-Small | | | | | | Dataset: EC-Large | | | | | |
	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$	NLL_{CTCVR}	NLL_{CVR}	$D\text{-}NLL_{CVR}$	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$	NLL_{CTCVR}	NLL_{CVR}	$D\text{-}NLL_{CVR}$
SingleCVR	0.7401	0.6531	0.6558	0.00393	0.02095	0.02372	0.7454	0.6634	0.6623	0.003908	0.02087	0.02347
Joint	0.7445	0.6584	0.6582	0.00391	0.02091	0.02355	0.7470	0.6685	0.6705	0.003908	0.02086	0.02323
Division	0.7434	0.6559	0.6572	0.00392	0.02104	0.02371	0.7471	0.6635	0.6625	0.003905	0.02096	0.02335
ESMM	0.7441	0.6584	0.6585	0.00391	__0.02086__	__0.02349__	0.7480	__0.6686__	0.6697	0.003900	0.02093	0.02359
CFL	__0.7453__	__0.6600__	__0.6587__	__0.00391__	0.02110	0.02381	__0.7486__	0.6685	__0.6722__	__0.003899__	__0.02067__	__0.02321__
UKD	**0.7513**	**0.6699**	**0.6732**	**0.00389**	**0.02077**	**0.02347**	**0.7531**	**0.6741**	**0.6752**	**0.003890**	**0.02066**	**0.02319**

| Method | Dataset: LS-Small | | | | | | Dataset: LS-Large | | | | | |
	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$	NLL_{CTCVR}	NLL_{CVR}	$D\text{-}NLL_{CVR}$	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$	NLL_{CTCVR}	NLL_{CVR}	$D\text{-}NLL_{CVR}$
SinlgeCVR	0.7801	0.6773	0.6711	0.00413	0.08505	0.10098	0.7833	0.6835	0.6723	0.00412	0.08521	0.10225
Joint	0.7856	0.6927	0.6792	0.00411	0.08556	0.10174	0.7861	0.6911	0.6774	0.00410	0.08487	0.10241
Division	0.7822	0.6851	0.6725	**0.00405**	**0.08417**	**0.10001**	0.7839	0.6741	0.6638	0.00414	**0.08236**	**0.09993**
ESMM	__0.7864__	__0.6936__	__0.6801__	0.00406	0.08487	__0.10064__	__0.7876__	__0.6924__	0.6791	__0.00406__	__0.08418__	0.10205
CFL	0.7839	0.6823	0.6783	0.00409	__0.08438__	0.10268	0.7849	0.6869	__0.6825__	0.00409	0.08466	0.10271
UKD	**0.7937**	**0.6958**	**0.6831**	__0.00406__	0.08498	0.10086	**0.7955**	**0.7001**	**0.6872**	**0.00405**	0.08448	__0.10204__

Table 3: Results on the public Ali-CCP dataset.

Method	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$	NLL_{CTCVR}	NLL_{CVR}	$D\text{-}NLL_{CVR}$
SingleCVR	0.6156	0.6514	0.6447	0.00211	**0.04420**	**0.05507**
Joint	0.6206	0.6699	0.6644	0.00205	0.04553	0.05599
Division	0.6172	0.6647	0.6598	0.00205	0.04607	0.05667
ESMM	0.6290	0.6711	0.6627	0.00206	__0.04493__	__0.05535__
CFL	__0.6371__	__0.6789__	__0.6738__	**0.00205**	0.04510	0.05577
UKD	**0.6493**	**0.6919**	**0.6864**	**0.00204**	0.04553	0.05627

Table 4: Necessity of the click-adaptive teacher.

w/ teacher?	Method	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$
✓	*UKD-base*	**0.7485**	**0.6664**	**0.6689**
✗	*Joint+D*	0.7375	0.6464	0.6511
✗	*Joint*	0.7445	0.6584	0.6582
✗	*CFL*	0.7453	0.6600	0.6587

CVR estimator. On *EC* datasets, it is superior to *ESMM* on most metrics and achieves the second-best performance, indicating that counterfactual learning has advantages to alleviate the SSB issue.

Our *UKD* outperforms all competitors by a large margin on both the public and production datasets. Specifically, compared to the second-best results, *UKD* consistently shows around 5‰ improvement on AUC_{CTCVR} and $D\text{-}AUC_{CVR}$, which is a large uplift on billion-scale dataset. This demonstrates that *UKD* is an effective debiasing framework for CVR estimation, which benefits from pseudo-conversion labels learned by the click-adaptive teacher model and distillation strategy of the uncertainty-regularized student model.

In the next three sections, we further conduct detailed experiments on *EC-Small* dataset to verify the effectiveness of *UKD* from three perspectives: the teacher's utility (§ 4.3), the student's utility (§ 4.4), and effects of unclicked samples (§ 4.5).

4.3 Utility of the Click-Adaptive Teacher (RQ2)

4.3.1 Necessity of Click-Adaptive Teacher. The teacher model in *UKD* is responsible to learn click-adaptive representation and produce pseudo-conversion labels for unclicked ads, aiming to alleviate the SSB issue via explicitly taking unclicked samples into account during training CVR models (i.e., the student in *UKD*).

To verify the necessity of the knowledge distillation paradigm (i.e., incorporating such a teacher model for pseudo-labels), we compare the base version of *UKD* with the models that utilize unclicked samples but do not follow knowledge distillation:

- *UKD-base* (§ 3.2.1) is the base version of *UKD*, which contains a click-adaptive teacher model and a base student model (equation 9) that does not utilize uncertainty modeling.
- *Joint+D* directly embodies domain adaptation into the *Joint* model by adding a click discriminator to the dense block $F_v(\cdot)$ of CVR task, without incorporating a teacher model.

Results are shown in Table 4. *UKD-base* beats all other models (including the state-of-the-art model *CFL*), indicating that learning click-adaptive representations for producing pseudo-conversion labels is an ideal solution for entire space CVR estimation. *Joint+D* performs worse than our *UKD-base*. According to the performance drop on AUC_{CVR}, the poor results can be attributed the reason that adding an discriminator hurts the optimization on clicked samples. The superiority of *UKD-base* verifies that the knowledge distillation paradigm is necessary to alleviate the SSB issue.

4.3.2 Effectiveness of Click-Adaptive Teacher. A well-trained teacher model can provide powerful guidance for distilling unclicked samples' knowledge to the student model.

To verify the effectiveness of our click-adaptive teacher model, we compare *UKD-base* to a variant that replaces the teacher from our click-adaptive model with a *SingleCVR* model (i.e., a naive teacher that does not learn any information from unclicked ones) and keeps the student model unchanged (i.e., a base student model in § 3.2.1). By comparing the variant's performance with our *UKD-base*, we can verify the effectiveness of the click-adaptive teacher. Table 5 shows the comparison results. We observe that equipping our click-adaptive teacher can boost the performance on all metrics around 3‰, demonstrating the effectiveness of unsupervised domain adaptation for producing pseudo-conversion labels on unclicked ads.

We also evaluate the click discriminator in our teacher model. Its output $\boldsymbol{p}_d$ predicts the domain of an impression ad, which can be regarded as the predicted CTR distribution. We use $\boldsymbol{p}_d$ to calculate CTR AUC with click labels y_{click}. We observe that at both training and test phrases, CTR AUC is always around 0.50, indicating that the learned representations are indeed click-adaptive because they fools the well-trained click discriminator. Thus, our click-adaptive teacher model eliminates the discrepancy between the representations of clicked and unclicked ads.

Table 5: Effectiveness of the click-adaptive teacher.

Teacher	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$
Click-adaptive Model	**0.7485**	**0.6664**	**0.6689**
SingleCVR	0.7462	0.6633	0.6657
No (i.e., $\mathcal{J}oint$)	0.7445	0.6584	0.6582

Table 6: Comparisons of different uncertainty strategies.

Uncertainty Strategy	AUC_{CTCVR}	AUC_{CVR}	$D\text{-}AUC_{CVR}$
Ours	**0.7513**	**0.6699**	**0.6732**
Monte-Carlo dropout	0.7490	0.6678	0.6695
No (i.e., *UKD-base*)	0.7485	0.6664	0.6689

4.4 Utility of the Uncertainty-Regularized Student (RQ3)

To alleviate noisy pseudo-conversion labels produced by the teacher, our student model employs the variance of two CVR predictors to estimate uncertainty during distillation. To verify this strategy's effectiveness, we compare it with Monte-Carlo dropout [6], a representative method for uncertainty estimation, which employs the variance of repeated predictions from the same model (but with different dropout at inference) as the uncertainty for a sample.

To adopt Monte-Carlo dropout into our knowledge distillation framework, for each unclicked ad, the trained teacher model (after adding dropout with rate 0.2) performs inference 10 times to produce pseudo-labels. The mean of 10 predictions is used as the pseudo-label, and the variance is used as its uncertainty. We then rank all unclicked samples in ascending sort order based on their uncertainty, and retain the top 80% samples (i.e., lower uncertainty) to train a base student model for CVR estimation.

Table 6 lists the comparison results. Compared to Monte-Carlo dropout, our strategy shows around 2~3‰ improvement, indicating that uncertainty regularization is more effective for alleviating label noise. Besides, repeated predictions in Monte-Carlo dropout consume much more computing resources (for reference, the time cost of performing CVR estimation on *EC-Small* dataset is over 30 minutes, and we need to perform 10 times to estimate the uncertainty). In contrast, our uncertainty regularization strategy only employs an additional CVR predictor $S^v_{p'}(\cdot)$ following the representation learner, thus the introduced resource consumption is negligible.
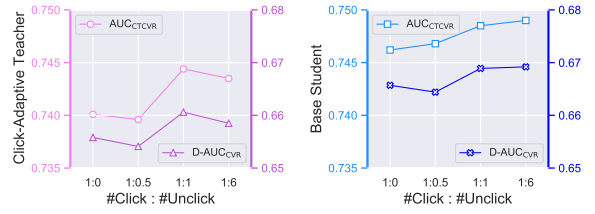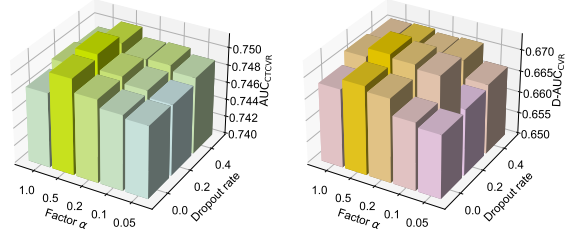
4.5 Model Analysis of *UKD* (RQ4)

We give more analysis including model performance w.r.t. the size of unclicked samples, and hyperparameter sensitivity.[3]

*4.5.1 **Effect of Unclicked Samples' Size**.* In *UKD*, the core intuition of alleviating the SSB issue is to explicitly incorporate unclicked samples during training. To show the effect of unclicked samples' size used in the teacher model, we vary the ratio of #clicked ads : #unclicked ads to 1:0 (no unclicked samples), 1:0.5 (less than clicked samples), 1:1 (equal to clicked samples) and 1:6 (all unclicked samples) to train different teacher models, and then train corresponding base student models to compare their performance.

Figure 4 shows the results. As the size of unclicked samples increases, the student model's performance generally gains. Thus our *UKD-base* benefits from incorporating more unclicked samples.

---
[3]Hyperparameters used in our experiments can be found in Appendix A.1.

Figure 4: Impact of unclicked samples' size.

Figure 5: Impacts of dropout rate after the learned representation, and factor α on unclicked samples' losses.

*4.5.2 **Hyperparameter Sensitivity Analysis**.* In the uncertainty-regularized student model, the dropout rate after the learned representation (equation 9) and the balance factor α of unclicked samples' losses (equation 11) are two key hyperparameters. Figure 5 illustrates the sensitivity analysis of them.

We can observe that the model performs better when dropout rate is not 0.0, and the best result is achieved at 0.2 rate, indicating that the dropout operation is crucial to our student model. We also see that the performance usually gains with the increasing of factor α and the best result is achieves at $\alpha = 0.5$, thus tuning the balance factor can contribute to the performance.

4.6 Online Experimental Results (RQ5)

We deploy *UKD* to the *LS* scenario on Alibaba advertising platform and conduct online A/B test for one-week. To make a fair comparison, we follow the same configuration with the best model deployed online, such as feature set and model size. The online metrics include CVR ($\frac{\#conversion}{\#click}$), CTCVR ($\frac{\#conversion}{\#impression}$) and cost per action (i.e., CPA, equal to $\frac{total\ cost}{\#conversion}$, lower is better).

We observe that *UKD* achieves **+3.4%** lift on CVR, **+5.0%** lift on CTCVR and **-4.3%** lift on CPA, thus *UKD* improves the important online metrics and promotes the performance of advertising system.

5 CONCLUSION

In this paper, we propose an uncertainty-regularized knowledge distillation framework named *UKD* for debiasing CVR estimation via distilling knowledge from unclicked ads. It employs a click-adaptive teacher to produce pseudo-conversion labels for unclicked ads, and then trains a student model in entire space by taking both clicked and unclicked samples into account. Moreover, our student model performs uncertainty estimation to alleviate the inherent noise in pseudo-labels to improve the distillation performance. Experimental results on large-scale production datasets strongly demonstrate the superiority of UKD for CVR estimation. Online experiments further verify that it achieves significantly improvements on core online metrics including CVR, CTCVR and CPA.

REFERENCES

[1] Elias Bareinboim, Jin Tian, and Judea Pearl. 2014. Recovering from selection bias in causal and statistical inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[2] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 21–30.

[3] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 579–588.

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[5] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.

[6] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning*. 1050–1059.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[8] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*.

[9] Siyuan Guo, Lixin Zou, Yiding Liu, Wenwen Ye, Suqi Cheng, Shuaiqiang Wang, Hechang Chen, Dawei Yin, and Yi Chang. 2021. Enhanced Doubly Robust Learning for Debiasing Post-click Conversion Rate Estimation. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[10] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[12] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 43–50.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2016. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st Conference on Neural Information Processing Systems*.

[14] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1910–1918.

[15] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 831–840.

[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 and Development in Information Retrieval*.

[17] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[18] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

[19] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* (2014).

[20] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[21] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of International Conference on Machine Learning*. 1670–1679.

[22] Tong Shen, Dong Gong, Wei Zhang, Chunhua Shen, and Tao Mei. 2019. Regularizing proxies with multi-adversarial training for unsupervised domain-adaptive semantic segmentation. *arXiv preprint arXiv:1907.12282* (2019).

[23] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 713–722.

[24] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.

[25] Qi Wang, Zhihui Ji, Huasheng Liu, and Binqiang Zhao. 2019. Deep Bayesian Multi-Target Learning for Recommender Systems. *arXiv preprint arXiv:1902.09154* (2019).

[26] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*. PMLR, 6638–6647.

[27] Penghui Wei, Weimin Zhang, Zixuan Xu, Shaoguo Liu, Kuang-chih Lee, and Bo Zheng. 2021. AutoHERI: Automated Hierarchical Representation Integration for Post-Click Conversion Rate Estimation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3528–3532.

[28] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4133–4141.

[29] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving ad click prediction by considering non-displayed events. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 329–338.

[30] Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21st International Conference on Machine Learning*. 114.

[31] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *Proceedings of The Web Conference 2020*. 2775–2781.

[32] Zhedong Zheng and Yi Yang. 2021. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision* 129, 4 (2021), 1106–1120.

A SUPPLEMENTARY MATERIAL

A.1 Implementation Details

All competitors use the same feature set and embedding sizes, as well as the same model architecture for fair comparison. Each dense block contains four fully-connected layers with the output sizes of [1024, 512, 256, 2], where the first three layers belong to representation learner and the last layer belongs to predictor. During optimization, we set the batch size to 128, and adopt Adam optimizer with 0.005 learning rate. For training the teacher model of *UKD*, we randomly sample unclicked ads to keep the ratio of #clicked ads : #unclicked ads as 1:1 for optimizing click discriminator. For the student model, the rate of dropout operation is set to 0.2. Other hyperparameters are set as follows: $\gamma = 0.2, \alpha = 0.5, \lambda = 100$. In practice, we treat the two predictors $S_p^v(\cdot)$ and $S_{p'}^v(\cdot)$ equally, therefore the objective $\mathcal{L}_{CVR}$ also contains the symmetrical terms $\sum_{\mathcal{D}_{click}} \ell(y_{conv}, \boldsymbol{p}'_{conv})$ and $\alpha \sum_{\widetilde{\mathcal{D}}_{unclick}} \exp\left(-\lambda \cdot \mathsf{KL}\left(\boldsymbol{p}'_{conv} || \boldsymbol{p}_{conv}\right)\right) \cdot \ell\left(\boldsymbol{p}_{conv}^{(T)}, \boldsymbol{p}'_{conv}\right)$. During inference, we use the average of the two predictors' outputs for estimating CVR., i.e., $(\boldsymbol{p}_{conv} + \boldsymbol{p}'_{conv})/2$. For each production dataset, we use the data at the penultimate/last day as validation/test set, and all the rest data as training data.

A.2 Definition of D-NLL$_{CVR}$

Let $\mathrm{NLL}_{CVR}(i)$ denote the NLL of the sample i, and the metric NLL_{CVR} is defined as:

$$\frac{1}{|\mathcal{D}_{click}|} \sum_{i \in \mathcal{D}_{click}} \mathrm{NLL}_{CVR}(i) \tag{15}$$

Our metric D-NLL$_{CVR}$ is defined as:

$$\frac{1}{\sum_{i \in \mathcal{D}_{click}} \frac{1}{\hat{p}_{CTR}(i)}} \sum_{i \in \mathcal{D}_{click}} \frac{1}{\hat{p}_{CTR}(i)} \cdot \mathrm{NLL}_{CVR}(i). \tag{16}$$