

# Decision-Focused Learning for Bidding Optimization in Online Advertising

Qilong Lin  
Shanghai Jiao Tong University  
Shanghai, China  
edersnow@sjtu.edu.cn

Zhenzhe Zheng\*  
Shanghai Jiao Tong University  
Shanghai, China  
zhengzhenzhe@sjtu.edu.cn

Zixiao Wang  
Tencent Inc.  
Beijing, China  
zzzxwang@tencent.com

Yan Jin  
Tencent Inc.  
Beijing, China  
hankjin@tencent.com

Tianxu Ji  
Tencent Inc.  
Beijing, China  
tianxuji@tencent.com

Wei Wang  
Tencent Inc.  
Beijing, China  
pierrowang@tencent.com

Fan Wu  
Shanghai Jiao Tong University  
Shanghai, China  
fwu@cs.sjtu.edu.cn

Guihai Chen  
Shanghai Jiao Tong University  
Shanghai, China  
gchen@cs.sjtu.edu.cn

## Abstract

In large-scale networked systems such as online marketplaces and advertising exchanges, real-time decision-making under dynamic environments is a fundamental challenge. A widely adopted paradigm is the predict-then-optimize workflow, where the prediction stage estimates key factors for decisions, and the optimization stage optimizes objectives based on these factors. However, existing prediction models are typically trained only for prediction accuracy, without awareness of the downstream optimization objective, leading to suboptimal results. We investigate this problem in the context of first-price auction bidding, a canonical decision problem in networked platforms. Specifically, we first propose the Decision-Focused Bidding (DFB) approach, which retains the predict-then-optimize workflow, but integrates the downstream objective into the training of the prediction model. To overcome the zero-gradient issue in back-propagation, we develop a smoothed decision mapping that provides usable gradients while preserving clear interpretability in the bidding context. We further show that DFB's decision-learning objective and the classical objective of learning winning probabilities are complementary under non-i.i.d. environments, and accordingly propose Enhanced DFB (EDFB), which jointly optimizes both objectives to achieve robust generalization in real-world bidding environments. Finally, our EDFB approach has been deployed in a production auto-bidding system, where it achieves a 2.3% GMV lift in a 3-day online A/B test over about 100 billion requests.

\*Zhenzhe Zheng is the corresponding author.

## CCS Concepts

• **Networks** → **Network economics**; • **Theory of computation** → **Machine learning theory**.

## Keywords

Online Marketplaces, Auto-Bidding, Decision-Focused Learning

## ACM Reference Format:

Qilong Lin, Zhenzhe Zheng, Zixiao Wang, Yan Jin, Tianxu Ji, Wei Wang, Fan Wu, and Guihai Chen. 2026. Decision-Focused Learning for Bidding Optimization in Online Advertising. In *The Twenty-seventh International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '26)*, November 23–26, 2026, Tokyo, Japan. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3842721.3849660>

## 1 Introduction

In online networked systems, the “predict-then-optimize” workflow is widespread across decision-making pipelines [3, 26]. In this workflow, the prediction stage is responsible for estimating **key factors** for decision-making, such as demand forecasts and outcome probabilities, etc., which are typically performed by machine learning models [16, 27, 40]. The optimization stage, on the other hand, focuses on optimizing a decision objective in tasks such as resource allocation [15], scheduling [12], and pricing in networked systems [38], and is highly dependent on the key factors derived in the prediction stage.

Among these networked systems, online advertising exchanges represent one of the largest real-time networked systems, where demand-side platforms (DSPs) interact with ad exchanges through auction protocols to compete for billions of ad impressions daily. We focus on the auto-bidding problem [21] in first-price auctions, which serves as a canonical and representative instance of the predict-then-optimize workflow in such networked systems. This scenario is viewed from DSP, where the DSP participates in auctions through bidding decisions, and seeks to maximize her expected surplus [30]. However, computing this expected surplus



requires knowing the winning probability, which is not directly observable and must be estimated from data. Consequently, existing bidding strategies naturally follow a predict-then-optimize workflow, where the learned model predicts the winning probability, and a downstream solver selects the bid that maximizes expected surplus [35, 41].

However, existing prediction objectives often focus solely on the accuracy of key factors and lack awareness of downstream decision-making, which can lead to suboptimal results. Specifically, the prediction models in bidding mainly use a loss similar to maximum likelihood [31, 35], where accuracy is equally optimized across all samples. However, from a decision-making perspective, the DSP should place greater emphasis on the high-value samples and prioritize their prediction accuracy. This misalignment of prediction and optimization objectives can result in poor decisions. An existing solution [7, 9] is the end-to-end idea, which bypasses the modular workflow and directly learns the optimal bids through black-box models. However, by discarding the structural priors of bidding, such end-to-end models must learn the bid-to-surplus relation purely from data, making it hard to avoid the generalization issue in the real-world non-i.i.d. task. This can cause the end-to-end approach to perform worse than existing predict-then-optimize solutions, as our experiments show.

To resolve this misalignment while retaining the structural priors that end-to-end methods discard, we propose the decision-focused bidding (DFB) approach. Inspired by decision-focused learning [4, 19], DFB keeps the two-stage workflow intact but closes the loop by back-propagating the downstream surplus objective into the prediction model, so that the predictor is explicitly trained to produce predictions that lead to high-surplus bids rather than merely accurate probabilities. The main obstacle is the zero-gradient issue: as a function of the prediction, the surplus objective is flat everywhere except at jump points, preventing classical gradient descent methods from working. We resolve this by introducing a smoothed decision mapping based on a softmax relaxation, which yields usable gradients and, unlike prior smoothing techniques [4, 33], carries a concrete bidding-level meaning: it is equivalent to an entropy-regularized surplus maximization, a form already widely adopted in industrial auto-bidding systems. This grounded practice translates into empirical gains, with our method outperforming alternative smoothing techniques in our experiments (Section 5).

We also observe that real-world networked systems are inherently non-stationary: auction dynamics, user behavior, and competition patterns shift continuously over time. Under such distribution shift, DFB's exclusive focus on decision quality can lead to poor generalization on unseen data, degrading its practical performance. Building on learnability analysis under distribution shift, we compare two learning objectives in bidding: learning decisions (as in DFB) versus learning factors (as in existing prediction-focused bidding, PFB). We show that these two objectives exhibit complementary strengths: in some data distributions only one is learnable, while in others the situation reverses, with the relative advantage determined by which mapping is simpler to learn—features-to-winning-price (for PFB) or features-to-optimal-bid (for DFB). Building on these insights, we further propose the enhanced DFB (EDFB) approach, which jointly optimizes both

objectives so that whichever mapping happens to be learnable in the deployed environment, the model inherits its generalization guaranty-yielding robust performance under the unpredictable distribution shifts in the real-world bidding environment.

Our contributions in this work can be summarized as follows:

- We take a decision-focused view of the predict-then-optimize paradigm in networked platforms. We argue that aligning the prediction and optimization stages is key to improving solution quality across a broad class of networked decision problems, and instantiate this view in first-price auction bidding.
- We propose DFB, which resolves the zero-gradient issue of applying decision-focused learning to bidding via a smoothed decision mapping with a clear bidding-level interpretation.
- We formally characterize a complementarity between learning decisions and learning factors under non-stationary environments, and based on this, further propose EDFB, which jointly optimizes both objectives to inherit whichever one is learnable at deployment. This is the first generalization analysis for decision-focused learning in networked platforms.
- On the Tencent dataset, EDFB improves offline surplus by 16% over PFB. Deployed in Tencent Ads, it improves GMV by 2.3% in a 3-day online A/B test over about 100 billion requests.

## 2 Related Work

**Prediction-Focused Bidding (PFB).** Since the winning probability is crucial for bidding decisions, a natural idea is the PFB approach, where the DSP trains the prediction model solely based on accuracy. Following this idea, a series of studies focused on improving the accuracy of prediction models [10, 25, 34, 35], which serves subsequent bidding decisions. However, in PFB, the prediction stage is decision-agnostic and thus cannot focus on the parts of predictions that most influence correct decisions.

**End-to-End Bidding (EEB).** Another idea is the EEB approach, where the DSP directly uses the model to predict the optimal decisions. In this line of work, Gligorijevic et al. [7] introduced factorization machines as the model structure, while Gong et al. [9] considered multi-task and multi-slot scenarios. However, such bidding strategies lack an interpretable structure, which can result in poor performance in non-i.i.d. online environments.

**Decision-Focused Learning.** Decision-focused learning introduces guidance from downstream decisions for the prediction model, thus aligning the objectives of the entire workflow. The challenge is that the optimization step typically has zero or undefined gradients, making it difficult to apply standard gradient descent methods. Many techniques have been proposed to address this, such as introducing a convex surrogate loss [4], and smoothing [18, 33] or randomizing [2, 11, 20] the decision mapping. However, existing work overlooks the generalization issue on non-i.i.d. data, which may lead to suboptimal online bidding.

## 3 Preliminaries

### 3.1 Bidding Process Formulation

We first provide a standard formulation of the bidding process as a predict-then-optimize workflow. In online advertising, the DSP needs to make bidding decisions for each auction. We assume that the candidate bids are discrete, forming a finite set

$\mathcal{B} = \{B_1, B_2, \dots, B_M\}$  of size  $M^1$ . We adopt a general formulation, where for any auction, the DSP's *decision* is denoted as  $b \in \Delta$ , which is an  $M$ -dimensional vector belonging to the probability simplex  $\Delta = \{b \succeq 0, \mathbf{1}^\top b = 1 \mid b \in \mathbb{R}^M\}$ , and the  $m$ -th element  $b_m$  represents the probability that the DSP selects bid  $B_m$ .

With this setup, we next model the DSP's *prediction* process. For each ad auction, the DSP first receives the feature  $x$ , which contains relevant information about the auction. For selecting the optimal bid, an important reference is the winning price, i.e., the minimum bid needed to win the auction. It is usually modeled as a random variable that follows a distribution  $p \in \Delta$ , which is also an  $M$ -dimensional vector belonging to the probability simplex  $\Delta$ , and the  $m$ -th element denotes the probability that the winning price falls in  $[B_{m-1}, B_m)$ , where we denote  $B_0 = 0$  for the completeness of the definition. In practice, these probabilities are estimated by the DSP using machine learning models, and we abstract this prediction process as  $\hat{p} = g_\theta(x)$ , where  $g$  represents the model with parameters  $\theta$ , and we use  $\hat{p}$  to denote the predicted result in order to distinguish it from the ground truth  $p$ .

Based on predictions, we further model the *optimization* process, which produces the bidding decisions. We use the predicted expected surplus as the optimization objective, which is widely adopted in bidding practice [37, 41]. To model this, we first introduce two auxiliary variables. The first is the predicted winning probability  $\hat{q} \in \mathbb{R}^M$ , which is the cumulative version of the interval probability  $\hat{p}$ , and the  $m$ -th element  $\hat{q}_m$  represents the probability of winning the auction when bidding  $B_m$ . Mathematically, the relation between these two probabilities can be written as  $\hat{q} = L\hat{p}$ , where  $L \in \mathbb{R}^{M \times M}$  is a lower triangular matrix with elements  $L_{ll'} = 1, \forall l \geq l'$ . The second is the winning surplus, denoted as a diagonal matrix  $S = \text{diag}(s_1, s_2, \dots, s_M)$ , where the  $m$ -th element  $s_m$  represents the surplus when bidding  $B_m$  and winning the auction. Generally, the DSP estimates the value  $v$  of winning each auction, and since payment in a first-price auction equals the bid, the surplus can be given by  $s_m = v - B_m$ . Note that in the product of these two variables,  $S\hat{q}$ , the  $m$ -th element represents the predicted expected surplus when bidding  $B_m$ . Therefore, given a decision  $b$  and the prediction  $\hat{p}$ , the predicted expected surplus of the DSP in the auction can be written as:

$$f(b, \hat{p}) = (S\hat{q})^\top b = (SL\hat{p})^\top b. \quad (1)$$

The DSP's objective is to maximize the predicted expected surplus. Therefore, given the predicted probability  $\hat{p}$ , the decision process for the DSP can be represented as the following *decision mapping*:

$$b^*(\hat{p}) = \arg \max_{b \in \Delta} f(b, \hat{p}). \quad (2)$$

Thus, based on the prediction  $\hat{p}$ , the decision made by the DSP can be written as  $\hat{b} = b^*(\hat{p})$ .

## 3.2 Performance Metric under Censorship

We now explain the *evaluation* of the resulting decisions  $\hat{b}$ . Our main metric is the expected surplus the DSP actually obtained. Note that the decision mapping  $b^*$  already evaluates the *predicted* expected surplus by the function  $f$  in Equation (1). Therefore, if the

DSP could access the ground truth label  $p$ , the expected surplus of the decision  $\hat{b}$  could similarly be written as  $f(\hat{b}, p)$ .

However, due to the nature of the bidding environment, the information available to the DSP is limited, making it difficult to fully obtain the ground truth  $p$ . This is referred to as the *censorship problem* in the literature [31, 35], which makes the DSP's evaluation difficult. Formally, we follow the standard censorship setting [34, 35, 41], where we denote the set of auctions used for training or evaluation by  $\mathcal{A}$ . Specifically,  $\mathcal{A}$  can be divided into a winning auction set  $\mathcal{W}$  and a losing auction set  $\mathcal{L}$ , and the DSP can only observe the exact winning price  $B_{m_i}$  in auction  $i \in \mathcal{W}$ . For the remaining auctions  $i \in \mathcal{L}$ , the DSP only knows that the winning price is greater than some bid  $B_{m_i-1}$ . Thus, the data available to the DSP in each auction  $i$  can be denoted as  $(x_i, m_i, S_i)$ , where the meaning of  $m_i$  differs between the two sets  $\mathcal{W}$  and  $\mathcal{L}$ . As can be seen, under the censorship setting, the DSP cannot directly obtain the ground-truth label  $p$ , which poses a challenge for the DSP in evaluating its bidding decisions.

To address this challenge, the key is to design a reasonable mapping  $\pi$  that can construct a label  $p_i$  based on the information available in auction  $i$ ,  $(x_i, m_i, S_i)$ . This label should capture the objectives of existing methods while also being interpretable in bidding practice. We defer the construction of the mapping  $\pi$  as a contribution to Section 4.1. Here, we simply use such a mapping to define our evaluation metrics. Formally, for the bidding decision set  $\{\hat{b}_i\}_{i \in \mathcal{A}}$  and the constructed label set  $\{p_i\}_{i \in \mathcal{A}}$ , our main metric is the *mean expected surplus* (MES) across the auctions in  $\mathcal{A}$ :

$$\text{MES} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} f_i(\hat{b}_i, p_i) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} (S_i L p_i)^\top \hat{b}_i. \quad (3)$$

## 3.3 Existing Approaches

**3.3.1 Prediction-Focused Bidding.** The *prediction-focused bidding* (PFB) idea has been widely adopted for bidding in related works [17, 22, 41]. This approach trains and evaluates the prediction model  $g_\theta$  based solely on the accuracy objective, while overlooking the subsequent decision-making process. Based on the available data under censorship  $(x_i, m_i, S_i)$ , the widely adopted PFB loss  $L_{PF}$  [34, 35, 41] on the auction set  $\mathcal{A}$  can be written as:

$$L_{PF}(\theta) = -\frac{1}{|\mathcal{W}|} \sum_{i \in \mathcal{W}} \log(\hat{p}_i)_{m_i} - \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} \log \sum_{m=m_i}^M (\hat{p}_i)_m. \quad (4)$$

This loss can be understood from the spirit of maximum likelihood: for winning auctions  $i \in \mathcal{W}$ , the DSP can observe the winning price label  $m_i$ , and the loss coincides with the standard maximum likelihood loss; for the other auctions  $i \in \mathcal{L}$ , the DSP can only optimize the probability over the region to which the label belongs. In Section 4.1, we will show that this loss is equivalent to a generalized maximum likelihood loss. Note that this loss is oblivious to the subsequent decision-making. Therefore, it does not leverage the decision-related data  $S_i$ , although they are accessible to the DSP.

**3.3.2 End-to-End Bidding.** We next introduce the *end-to-end bidding* (EEB) idea. This method does not depend on the prediction  $\hat{p}$ ; instead, it directly learns the optimal bids through a black-box model. To distinguish this from the prediction model  $g_\theta$ , we denote the end-to-end model as  $\hat{b} = h_\theta(x, S)$ . In particular, note that the

<sup>1</sup>In practice, the prediction model outputs a distribution over a finite set of price intervals [17, 37], so bids are chosen from the interval endpoints.

winning surplus matrix  $S$  is necessary for decision-making, so we include it as an input to the model  $h_\theta$ . Thus, following the evaluation metric above, we can naturally use the expected surplus as the EEB loss  $L_{EE}$  for model  $h_\theta$ :

$$L_{EE}(\theta) = - \sum_{i \in \mathcal{A}} f_i(\hat{b}_i, p_i) = - \sum_{i \in \mathcal{A}} f_i(h_\theta(x_i, S_i), p_i). \quad (5)$$

Recall that the data available to the DSP is  $(x_i, m_i, S_i)$ , which does not include the ground truth  $p_i$ . However, existing EEB approaches do not discuss this [7, 9]. In fact, Gligorijevic et al. [7] did not consider the censorship environment and thus avoided this problem, while Gong et al. [9] introduced an additional prediction model and used its output as the ground truth  $p_i$ . These methods lack an in-depth examination of the PFB loss (4), and do not answer what the prediction label  $p_i$  should be under censorship. Therefore, instead of adopting their solutions, we implement EEB with the labels constructed in Section 4.1.

## 4 Design

### 4.1 A Unified Prior for Censorship

Due to the censorship problem, existing work has employed a specialized PFB loss  $L_{PF}$  for prediction. However, the ground truth  $p$  within it lacks a well-defined formulation. This creates challenges in fairly evaluating the EEB loss  $L_{EE}$  (as well as our losses). To address this challenge, we reformulate the PFB loss and uncover the implicit “prior” it contains, which can be further used to construct reasonable labels. These constructed labels enable us to calculate the surplus for the DSP, thus allowing the comparison of different methods under a unified prior. The details are as follows.

We first introduce a generalized maximum likelihood loss: *cross-entropy loss with noisy labels* (CELN), which was originally proposed for classification tasks with uncertain labels [8, 28]:

**DEFINITION 1 (CELN).** *Given the model predictions  $\hat{p}_i = g_\theta(x_i)$  and the labels  $p_i$  for each auction  $i \in \mathcal{A}$ , the cross-entropy loss with noisy labels (CELN)  $L_{CE}$  on the dataset  $\mathcal{A}$  is defined as:*

$$L_{CE}(\theta) = - \sum_{i \in \mathcal{A}} \log((\hat{p}_i)^\top p_i) = - \sum_{i \in \mathcal{A}} \log(g_\theta(x_i)^\top p_i). \quad (6)$$

We can see that the CELN  $L_{CE}$  is similar to the widely adopted PFB loss  $L_{PF}$  (4). In fact, by constructing appropriate labels  $p_i$  from the available data under censorship, the CELN  $L_{CE}$  is actually equivalent to the PFB loss  $L_{PF}$ . Formally, we refer to this process of constructing corresponding labels  $p_i$  from the known data  $m_i$  as the *prior*, defined as follows:

**DEFINITION 2 (PRIOR).** *For historical auctions  $\mathcal{A}$  under censorship, a prior refers to a mapping  $\pi$  from the bid index set  $\{m_i\}_{i \in \mathcal{A}}$  to a label set  $\{p_i\}_{i \in \mathcal{A}}$ . That is, given the prior  $\pi$ , for each auction  $i \in \mathcal{A}$ , the label can be constructed from the known bid index as  $p_i = \pi(m_i)$ .*

**PROPOSITION 1.** *Minimizing the PFB loss  $L_{PF}$  (4) is equivalent to minimizing a CELN  $L_{CE}$  (6), when the labels  $p_i$  in  $L_{CE}$  for each auction  $i \in \mathcal{A}$  are constructed by the following prior  $\pi_{PF}$ :*

$$(\pi_{PF}(m_i))_m = \begin{cases} \frac{1}{M-m_i+1}, & \text{if } m \in [m_i, M] \text{ and } i \in \mathcal{L}, \\ 1, & \text{if } m = m_i \text{ and } i \in \mathcal{W}, \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

where  $(\pi_{PF}(m_i))_m$  is the  $m$ -th element of the constructed label  $p_i$ .

Proposition 1 reveals that the widely used PFB loss can be reduced to a generalized likelihood loss CELN, where the labels can be constructed based on the available data and the prior  $\pi_{PF}$ . In fact, subsequent variants of the PFB loss can be similarly reduced to the (weighted) CELN through analogous prior definitions [17, 25], thus our core methodology can be easily transferred. This demonstrates that our approach also exhibits advanced extensibility.

We further discuss the meaning of prior  $\pi_{PF}$  in the context of a classification task. For the auction  $i \in \mathcal{W}$ , the DSP can precisely know the “class” of winning price as  $B_{m_i}$ . According to the maximum likelihood principle, the loss on auction  $i$  can be directly written as  $-\log(\hat{p}_i)_{m_i}$ , which aligns with the PFB loss  $L_{PF}$ . Therefore, the construction in  $\pi_{PF}$  for the winning auctions  $i \in \mathcal{W}$  is natural. What is special are the remaining losing auctions  $i \in \mathcal{L}$ , where the DSP only knows that the “class” of the winning price lies within the set  $\{B_m | m \in [m_i, M]\}$  (i.e., bids not less than  $B_{m_i}$ ). According to Proposition 1, the PFB loss actually introduces an assumption, treating the uniform distribution over this set  $\{B_m | m \in [m_i, M]\}$  as the ground truth label. This provides a new interpretation for the existing PFB loss under censorship. Therefore, to distinguish it from the true information accessible to the DSP, we refer to it as the prior in this paper.

Given these equivalent labels, we can calculate existing losses  $L_{PF}$ ,  $L_{EE}$ , and loss  $L_{DF}$  we propose later, and fairly evaluate them using the MES metric in Equation (3). In fact, both EEB and our methods directly optimize this metric. However, our decision-making pipeline differs from EEB by introducing an interpretable decision-mapping structure. On the one hand, it provides our method with better performance and interpretability; on the other hand, this also introduces additional challenges to our model training. We provide a detailed discussion for this in the next section.

### 4.2 Decision-Focused Bidding

We first introduce the overall idea of our proposed *decision-focused bidding* (DFB) approach. Based on the preceding groundwork, the key of this approach is to directly use the decision surplus to guide the update of the prediction model  $g_\theta$ . Thus, the DFB loss  $L_{DF}$  in auctions  $\mathcal{A}$  can be naturally written as:

$$L_{DF}(\theta) = - \sum_{i \in \mathcal{A}} f_i(\hat{b}_i, p_i) = - \sum_{i \in \mathcal{A}} f_i(b_i^*(g_\theta(x_i)), p_i). \quad (8)$$

**4.2.1 Principle Analysis.** We first highlight the advantages of DFB through comparison. Compared to EEB, DFB explicitly incorporates the bidding decision process. Intuitively, this endows DFB with better generalization performance, which is also supported by the results of our experiments (see Section 5 for details).

Therefore, our focus lies in the comparison with PFB. We consider a simple example, as shown in Figure 1, where we focus only on the decisions between two bid indices,  $m_1$  and  $m_2$ . We assume that the feature is one-dimensional and unrelated to winning surplus  $s_{m_1}$  and  $s_{m_2}$ , but is related to winning probabilities  $q_{m_1}$  and  $q_{m_2}$ , that is, to expected surplus  $s_{m_1} q_{m_1}$  and  $s_{m_2} q_{m_2}$ . Specifically, we assume that the true relation between the feature and the expected

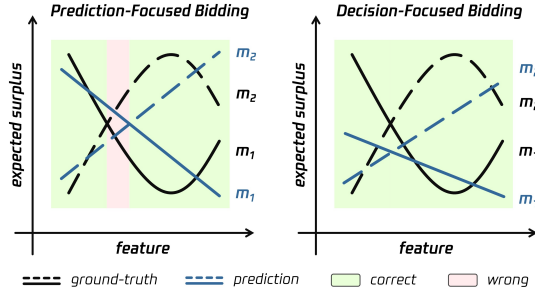


**Figure 1: A comparison between the outputs of the prediction models in the PFB approach and the DFB approach under a special case with i.i.d. data.**

surpluses of the two bids is represented by the black curve in Figure 1. Since  $s_{m_1}$  and  $s_{m_2}$  are constants, this curve also reflects the relation between the feature and the probabilities  $q_{m_1}$  and  $q_{m_2}$ .

Thus, we conduct the comparison in a simple setting where the prediction model  $g_\theta$  is linear. As a result, it can only capture a linear relation between the feature and the predicted probabilities  $\hat{q}_{m_1}$ ,  $\hat{q}_{m_2}$ , as well as the expected surplus  $s_{m_1}\hat{q}_{m_1}$  and  $s_{m_2}\hat{q}_{m_2}$ , as illustrated by the blue line in Figure 1. For PFB, its objective is prediction accuracy, so its predicted expected surplus will largely follow the ground truth, as shown in the left half of Figure 1. However, the DSP's objective is to select the bid with the highest surplus, and we can see that the decision of PFB in the red-shaded area does not match the ground truth. In contrast, DFB directly targets optimal decisions, and in this simple example, it can produce decisions that perfectly match the ground truth, as shown in the right half of Figure 1. This illustrates the advantage of DFB.

It should be noted that the advantage of DFB depends on the model complexity [5]. If the model is sufficiently expressive to learn the decision factor, e.g., capable of representing the non-linear relation indicated by the black curves in Figure 1, then the advantage of DFB would diminish, as confirmed by the experiments in Section 5.2. Furthermore, our example does not consider the generalization performance under non-i.i.d. data, that is, model performance on unseen feature values. However, this is quite common in bidding and directly affects the effectiveness of applying DFB. We will discuss this issue and our solutions in later sections.

**4.2.2 Challenge Analysis.** To illustrate the challenge of applying DFB, we use the chain rule to derive the derivative of the DFB loss  $L_{DF}$  with respect to the model parameters  $\theta$ , which is essential for updating the model using gradient descent methods:

$$\frac{\partial L_{DF}(\theta)}{\partial \theta} = - \sum_{i \in \mathcal{A}} \frac{\partial f_i(\hat{b}_i, p_i)}{\partial \hat{b}_i} \cdot \frac{\partial b_i^*(\hat{p}_i)}{\partial \hat{p}_i} \cdot \frac{\partial g_\theta(x_i)}{\partial \theta}. \quad (9)$$

We examine the three factors of the product. The first factor is the partial derivative of the function  $f_i$  w.r.t. its input  $\hat{b}_i$ , which is easy to compute due to the linearity of  $f_i$  as defined in (1). The third factor is the derivative of the prediction model  $g_\theta$  w.r.t. its parameters  $\theta$ , which is provided by the differentiable structure of the model. Therefore, the key lies in the second factor: the derivative of the decision mapping  $b_i^*$  w.r.t. its input  $\hat{p}_i$ . To study this derivative, we

need to examine the decision mapping  $b_i^*$  in (2) more clearly. We first show that it has a more intuitive form:

**PROPOSITION 2.** *The range of the decision mapping  $b^*$  defined in Equation (2) is the set of standard basis vectors  $\{e_1, e_2, \dots, e_M\}$  in  $\mathbb{R}^M$ . In fact, the mapping can be written as:*

$$b^*(p) = e_{m^*}, \quad \text{where } m^* = \arg \max_m (SLp)_m. \quad (10)$$

Proposition 2 aligns with practical understanding, as it provides a more natural interpretation of the bidding process: namely, computing the expected surplus  $Sq = SLp$  for all the bids, selecting such a bid index  $m'$  that has the highest expected surplus  $(SLp)_{m'}$ , and choosing the corresponding optimal bid  $B_{m'}$  without randomness, which is making decision  $e_{m'}$ . This result is also consistent with intuition, as linear programming theory guarantees that an optimal solution must exist among the extreme points of the feasible domain [6]. Overall, this proposition helps us to better understand the form of the decision mapping  $b^*$ .

However, its derivative is zero almost everywhere:

**PROPOSITION 3.** *For any  $S$  within the mapping  $b^*$  and any input  $p$ , if the partial derivative  $\partial b^*(p)/\partial p$  exists, it must be 0.*

The result in Proposition 3 represents the main challenge of our DFB idea: it causes the gradients to vanish during back propagation. To see this, we continue with the majority cases above, where the output of  $b^*$  remains  $e_{m'}$  after any small perturbation of  $\hat{p}'$ . This means that even if our prediction  $\hat{p}'$  changes slightly, the calculated loss  $L_{DF}$  remains unchanged. In other words, in most cases, the loss function is flat around  $\hat{p}'$ . Since gradient descent only seeks a local descent direction, a flat region naturally prevents it from making progress. We refer to this as the *zero-gradient issue*.

**4.2.3 Learnable Implementation.** To address the zero-gradient issue, we propose the **softmax mapping** approach, whose key idea is to avoid the zero-gradient issue by leveraging the softmax approximation commonly used in practical systems [14, 23, 24]. Formally, it rewrites the original decision mapping (2) as:

$$\tilde{b}^{so}(\hat{p}) = \frac{1}{1^\top \exp(SL\hat{p})} \cdot \exp(SL\hat{p}), \quad (11)$$

in which case the key second factor in the derivative expression (9) is non-zero, and thus the overall derivative is usable for learning. In particular, we note that the effect of this approximation can be formalized: it is equivalent to replacing the original optimization problem (2) with the following one [1]:

$$\tilde{b}^{so} = \arg \max_{b \in \Delta, b > 0} f(b, \hat{p}) - b^\top \log(b). \quad (12)$$

Intuitively, this differs from the original problem in that the new objective adds an entropy term over the bids, whereas the original objective only considers the outcome of the optimal bid. This provides a practical understanding of using the softmax mapping.

### 4.3 Enhanced DFB for Online Environment

Previously, we have provided an overview of DFB and proposed a smoothed decision mapping to address the zero-gradient issue. However, in the online context with non-i.i.d. data, we find that

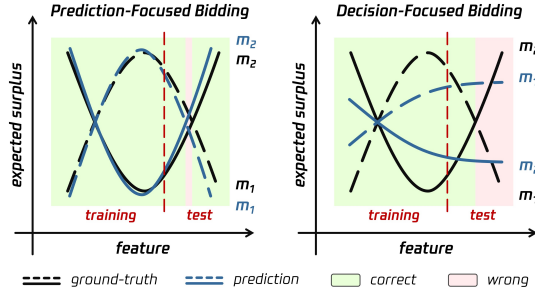


**Figure 2: A comparison between the outputs of the prediction models in the PFB approach and the DFB approach under a special case with non-i.i.d. data.**

DFB still faces the generalization issue, which may significantly reduce the DSP's surplus. Specifically, in practice, the DSP can only train the prediction model using historical data and apply it to future bidding. Meanwhile, the environment of online ad auctions is constantly changing. This places high demands on the model's generalization ability beyond the classical i.i.d. data settings. To our knowledge, this generalization issue under non-i.i.d. data has been underexplored in decision-focused learning studies.

To address this challenge, we conduct an in-depth analysis of the generalization performance of PFB, DFB, and EEB on non-i.i.d. tasks, as well as their respective advantages. Based on insights from the theoretical analysis, we further propose an Enhanced DFB (EDFB) approach tailored to online bidding tasks. Specifically, we first perform a case analysis by extending the example in Figure 1 from the i.i.d. setting to the non-i.i.d. setting, and observe that the original advantage of DFB disappears. Instead, PFB exhibits better performance on unseen data, as it more effectively learns the decision-dependent factors. Furthermore, leveraging theoretical frameworks for non-i.i.d. environments, we compare PFB and DFB and find that the two methods have complementary strengths: in some non-i.i.d. environments, PFB tasks are learnable while EEB tasks are not, whereas in other non-i.i.d. environments the opposite holds. In particular, we identify that the relative complexity of **learning key factors** versus **learning decisions** is important for driving the differences in learnability. Finally, based on these theoretical insights, we empirically combine the DFB and PFB approach to propose the EDFB approach. Details are as follows.

**4.3.1 PFB versus DFB.** We first conduct the comparison between DFB and PFB on non-i.i.d. data under a special case, and show that PFB may be better at learning the underlying decision-dependent factors, thereby making correct decisions on unseen data. To illustrate this, we extend the example on i.i.d. data described in Figure 1 by splitting the data into training and test sets, as indicated by the red dashed line in Figure 2. This time, we assume that the model is sufficiently expressive to learn the underlying factors on the training set, that is, the blue curve that represents the model's output can closely follow the black curve that represents the ground truth. We compare the generalization performance as follows:

- Since PFB aims to learn the underlying factors accurately, it can efficiently learn the relation between the feature and the

expected surplus, that is, efficiently learn the shape of the black curves, and generalize this relation well to the test set. Since the optimal decision is to choose the bid with the highest expected surplus, accurate expected surplus predictions on the test set enable PFB to make correct decisions (Figure 2, left).

- DFB only learns to make correct decisions on the training set and lacks an objective of prediction accuracy. As a result, it can learn the correct bids on the training set, that is, to identify which of the solid black curve or the dashed curve is higher; however, it does not learn the underlying relation between the expected surplus and the features, that is, to fit the shape of the black curves. Therefore, its model can make correct decisions on the training set, but since it fails to capture the underlying decision-dependent factors, it has no guidance when faced with unseen data on the test set and would make incorrect decisions, as shown in the right half of Figure 2.

This intuitively reveals the positive role of PFB's accuracy objective in improving the generalization ability for non-i.i.d. data, while also highlighting the generalization issues of DFB. Moreover, since EEB shares the same learning objective as DFB and neither incorporates an accuracy metric, we note that EEB will also suffer from the same issue in our example.

**4.3.2 PFB versus EEB.** Beyond the preceding case study, we further compare PFB and EEB from a theoretical perspective and show that the two methods have complementary strengths. To this end, we adopt the theoretical framework of out-of-distribution generalization and introduce several additional definitions [36]. Formally, this framework considers non-i.i.d. tasks  $(\mathcal{Z}, \mathcal{C}, \mathcal{E}_{train}, \mathcal{E}_{eval})$ , where  $\mathcal{Z}$  is the set of feature components,  $\mathcal{C}$  is the label space,  $\mathcal{E}_{train}$  is the dataset of auctions available to the DSP, and  $\mathcal{E}_{eval}$  is the dataset of auctions the DSP aims to generalize to, with  $\mathcal{E}_{train} \subseteq \mathcal{E}_{eval}$ . For any auction set  $\mathcal{A} \in \mathcal{E}_{eval}$ , we denote by  $P_{\mathcal{A}}(Z|c)$  the conditional probability distribution of the feature component  $Z \in \mathcal{Z}$  as a random variable given the label  $c \in \mathcal{C}$ . To characterize the generalization task, we focus on the variation and informativeness of different feature components, defined below:

**DEFINITION 3 (VARIATION AND INFORMATIVENESS).** The variation  $\mathcal{V}$  and the informativeness  $\mathcal{I}$  of a feature component  $Z$  over the label space  $\mathcal{C}$  and dataset  $\mathcal{E}$  are defined as:

$$\mathcal{V}(Z, \mathcal{C}, \mathcal{E}) = \max_{c \in \mathcal{C}, \mathcal{A} \in \mathcal{E}, \mathcal{A}' \in \mathcal{E}} \rho(P_{\mathcal{A}}(Z|c), P_{\mathcal{A}'}(Z|c)), \quad (13)$$

$$\mathcal{I}(Z, \mathcal{C}, \mathcal{E}) = \sum_{c \neq c' \in \mathcal{C}} \min_{\mathcal{A} \in \mathcal{E}} \rho(P_{\mathcal{A}}(Z|c), P_{\mathcal{A}}(Z|c')). \quad (14)$$

Here,  $\rho$  is the total variation distance [29] of two distributions. Intuitively, the variation  $\mathcal{V}(Z, \mathcal{C}, \mathcal{E})$  measures the extent to which a feature component  $Z$  varies across different auction distributions in  $\mathcal{E}$ , while the informativeness  $\mathcal{I}(Z, \mathcal{C}, \mathcal{E})$  captures the ability of the feature component  $Z$  to distinguish the labels across the auction distributions in  $\mathcal{E}$ .

**DEFINITION 4 (EXPANSION FUNCTION).** A function  $u$  is an expansion function, if it is monotonically increasing, and  $u(a) \geq a$ ,  $\forall a$ , and  $\lim_{a \rightarrow 0} u(a) = u(0) = 0$ .

**DEFINITION 5 (LEARNABILITY).** A task  $(\mathcal{Z}, \mathcal{C}, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$  is learnable, if there exists an expansion function  $u$  and an informativeness degree  $\delta \geq 0$  such that  $u(\mathcal{V}(\mathcal{Z}, \mathcal{C}, \mathcal{E}_{\text{train}})) \geq \mathcal{V}(\mathcal{Z}, \mathcal{C}, \mathcal{E}_{\text{eval}})$  for all feature components  $Z \in \mathcal{Z}$  satisfying  $\mathcal{I}(\mathcal{Z}, \mathcal{C}, \mathcal{E}_{\text{train}}) \geq \delta$ .

To compare the learnability of PFB and DFB, we adopt a simplified setting, assuming that all auctions in  $\mathcal{A} \in \mathcal{E}_{\text{eval}}$  are winning auctions  $\mathcal{W}$ . In this setting, the factor learned in PFB—namely, the winning probability  $p_i$ —reduces to the deterministic winning price  $m_i$ ; meanwhile, the decision learned in EEB—namely, the optimal decision  $b_i$ —also reduces to the deterministic optimal bid  $m_i^*$ . Specifically, recalling our previous modeling:

- For PFB, the feature components  $\mathcal{Z}_P$  consist of the components of the feature vector  $x$ . In auction  $i \in \mathcal{A}$ , the component  $Z_P \in \mathcal{Z}_P$  realizes its value as the corresponding component of feature  $x_i$ , and the label  $c_P \in \mathcal{C}_P$  corresponds to the winning price  $m_i$ .

- For EEB, the feature components  $\mathcal{Z}_E$  consist of the components of the feature vector  $x$  together with the expected surplus matrix  $S$ . In auction  $i \in \mathcal{A}$ , the component  $Z_E \in \mathcal{Z}_E$  realizes its value as the corresponding component of  $x_i$  or  $S_i$ , and the label  $c_E \in \mathcal{C}_E$  corresponds to the optimal bid  $m_i^*$ , where  $m_i^*$  is defined as in Proposition 2. We note that  $\mathcal{Z}_P \subset \mathcal{Z}_E$  and  $m_i^* = \arg \max_m (S L e_{m_i^*})_m$ .

Intuitively, compared to PFB, the key distinction of EEB lies in the additional surplus matrix  $S$  that introduces extra correlations between features and labels. This leads to complementary strengths between the two approaches. In some cases, even when the factor-learning task in PFB is challenging, i.e., the relationship between the winning-price labels and the features fails to generalize, the relationship between the optimal bid labels obtained through the matrix  $S$  and the features can be much simpler, making the decision-learning task in EEB easy. Conversely, in other cases, even when the factor-learning task in PFB is simple, if the relationship between the matrix  $S$  and the features does not generalize well, the decision-learning task in EEB can also become challenging. We formally present these conclusions below.

**THEOREM 1.** *There exists some  $\mathcal{E}_{\text{train}} \subseteq \mathcal{E}_{\text{eval}}$  such that the task for EEB  $(\mathcal{Z}_E, \mathcal{C}_E, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$  is learnable while the task for PFB  $(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$  is not learnable.*

**PROOF.** First, we need to construct an unlearnable PFB task  $(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$ . Suppose the feature is one-dimensional, i.e.,  $|\mathcal{Z}_P| = 1$ , meaning there is only a single component  $Z_P$ , and the winning price takes two possible values  $\mathcal{C}_P = \{1, 2\}$ . In the available set  $\mathcal{E}_{\text{train}}$ , we assume that only the cases  $(Z_P = 1, c_P = 1)$  and  $(Z_P = 0, c_P = 2)$  occur, whereas in the remaining set  $\mathcal{E}_{\text{eval}} \setminus \mathcal{E}_{\text{train}}$ , the situation is reversed, with only  $(Z_P = 1, c_P = 2)$  and  $(Z_P = 0, c_P = 1)$  occurring. Clearly, in this case, the variation and informativeness of  $Z_P$  over the available set  $\mathcal{E}_{\text{train}}$  are zero, i.e.,  $\mathcal{V}(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}) = \mathcal{I}(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}) = 0$ , while the variation over  $\mathcal{E}_{\text{eval}}$  is nonzero, i.e.,  $\mathcal{V}(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{eval}}) > 0$ . Since the expansion function  $u$  satisfies  $u(0) = 0$ , we have  $\mathcal{V}(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{eval}}) > u(\mathcal{V}(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}))$ , which violates the requirement in Definition 5.

We next construct a learnable EEB task  $(\mathcal{Z}_E, \mathcal{C}_E, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$ . In fact, it suffices to let all the matrices  $S$  in the set  $\mathcal{E}_{\text{eval}}$  be  $\text{diag}(0, -1)$ . In this case, the optimal bidding label is always  $c_E = 1$ , and thus

for any feature  $Z_E$ , both the variation and the informativeness over  $\mathcal{E}_{\text{train}}$  and  $\mathcal{E}_{\text{eval}}$  are zero. Clearly, under this construction, the EEB task is learnable.  $\square$

**THEOREM 2.** *There exists some  $\mathcal{E}_{\text{train}} \subseteq \mathcal{E}_{\text{eval}}$  such that the task for PFB  $(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$  is learnable while the task for EEB  $(\mathcal{Z}_E, \mathcal{C}_E, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$  is not learnable.*

**PROOF.** Similar to the previous proof, we first construct a learnable PFB task  $(\mathcal{Z}_P, \mathcal{C}_P, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$ . We again assume that the feature is one-dimensional, i.e.,  $|\mathcal{Z}_P| = 1$ , and that the winning price takes two possible values  $\mathcal{C}_P = \{1, 2\}$ . The difference is that, across all auctions in  $\mathcal{E}_{\text{eval}}$ , we assume that only the case  $(Z_P = 0, c_P = 2)$  occurs. Clearly, the variation of  $Z_P$  on both datasets  $\mathcal{E}_{\text{train}}$  and  $\mathcal{E}_{\text{eval}}$  is zero, and the task is learnable.

We next construct an unlearnable EEB task  $(\mathcal{Z}_E, \mathcal{C}_E, \mathcal{E}_{\text{train}}, \mathcal{E}_{\text{eval}})$ . This time, we assume that the matrix  $S$  differs between the sets  $\mathcal{E}_{\text{train}}$  and  $\mathcal{E}_{\text{eval}} \setminus \mathcal{E}_{\text{train}}$ . For the available set  $\mathcal{E}_{\text{train}}$ , we assume  $S = \text{diag}(2, 1)$ , in which case the optimal bid label  $c_E$  coincides with the winning price, i.e.,  $c_E = 2$ . For the remaining set  $\mathcal{E}_{\text{eval}} \setminus \mathcal{E}_{\text{train}}$ , we assume  $S = \text{diag}(0, -1)$ , in which case the optimal bid label is  $c_E = 1$ . Note that for any feature component  $Z_E$ , both the informativeness and the variation over the available set  $\mathcal{E}_{\text{train}}$  are zero. However, if  $Z_E$  corresponds to a diagonal element of  $S$ , its variation over  $\mathcal{E}_{\text{eval}}$  is nonzero. For this component, since the expansion function requires  $u(0) = 0$ , we have  $\mathcal{V}(\mathcal{Z}_E, \mathcal{C}_E, \mathcal{E}_{\text{eval}}) > u(\mathcal{V}(\mathcal{Z}_E, \mathcal{C}_E, \mathcal{E}_{\text{train}}))$ , which violates the requirement in Definition 5.  $\square$

Overall, in non-i.i.d. environments, PFB and EEB exhibit complementary strengths, with the primary driver being the relative difficulty of learning key factors versus learning decisions. When learning factors is relatively easy, PFB has an advantage; conversely, when learning decisions is easy, EEB tends to be more advantageous. Notably, we believe that our conclusion has broad implications. On the one hand, if PFB is learnable, then the bidding task based on PFB is also learnable, since the optimal bid can be derived from  $b^*$ . On the other hand, since DFB and EEB share the same objective, their performance in non-i.i.d. environments is similar. Therefore, the above analysis uses EEB as the representative.

**4.3.3 Enhanced DFB Approach.** Combining the above insights, we further propose a simple yet effective approach, which we refer to as enhanced DFB. The idea of this approach is straightforward: combine the existing DFB loss with the PFB loss for the model:

$$L_{ED}(\theta; \alpha) = \alpha L_{DF}(\theta) + (1 - \alpha) L_{PF}(\theta), \quad (15)$$

where  $\alpha$  is the weighting factor that balances the two losses. Note that this weighted loss does not simply produce a performance average of the original methods. The key insight is that as long as either the PFB or EEB task is learnable, the performance of this weighted approach on the corresponding task can be derived based on the work of Ye et al. [36], thereby obtaining generalization guarantees under non-i.i.d. environments. As a result, it can achieve better performance than any single method alone.

Note that although EEB also targets surplus maximization, it does not adopt a predict-then-optimize framework. As a result, although our experiments in Section 5 show that EEB also suffers from generalization issues, it cannot address these problems by

combining with PFB in the way that DFB does. This also highlights the advantages of DFB over EEB.

## 5 Experiment Results

### 5.1 Experimental Setup

**5.1.1 Datasets.** Our experiments are conducted on two datasets:

- **The iPinYou dataset** is a public dataset widely used in related work [39]. It contains data from 9 ad campaigns over 10 days. Following previous work [31], we preprocessed the data and used the first 7 days as the training set, the remaining as the test set.

- **The Tencent dataset** is collected by Tencent Ads with daily traffic of tens of billions. It contains 3.36 billion records from June 29 to July 2, 2025, spanning 4 days. We used the first 3 days of data as the training set and the remaining data as the test set.

Note that the format of the data is provided in Section 3.2. In addition, we numerically scale the surplus matrix  $S_i$  so that the EEB and DFB losses are on the same magnitude as the PFB loss.

**5.1.2 Methods.** We compare the following methods in our work:

- **EEB** is the end-to-end bidding method, as introduced in Section 3.3.2. This is the latest solution reported by Alibaba [9].

- **PFB** is the prediction-focused method, which is our main baseline for comparison, as introduced in Section 3.3.1. This is the latest solution in Tencent [10, 32].

- **DFB** is our softmax mapping proposed in Section 4.2.3; We also evaluate classic decision-focused methods proposed by Wilder et al. and Elmachetou and Grigas; we denote these alternative DFB implementations as **SCA** [33] and **SPO** [4].

- **EDFB** is our further proposed enhanced decision-focused bidding approach, which is detailed in Section 4.3. Empirically, EDFB adopts the implementation of DFB, with the weight parameter  $\alpha$  set to 0.5 on both the iPinYou dataset and the Tencent dataset.

It should be noted that, according to Section 4.2 and Section 4.3, the prediction model's expressive capability is also an important setting. We use the model from Li et al. [17], which primarily consists of a three-layer fully connected network, with the number of discrete decision outputs  $M$  set to 300. We also use the Adam optimizer [13] with a learning rate of  $10^{-4}$  for model training.

**5.1.3 Metrics.** Our primary metric is the DSP objective:

- **MES** is the mean expected surplus defined in Equation (3).

In addition, we consider two auxiliary metrics. The first is the prediction accuracy of the model, which is the objective of PFB and reflects the interpretability of the model's predictions:

- **MLL** is the mean likelihood loss, which is the mean value of CELN loss (i.e., the loss of PFB), as defined in Equation (6).

We also consider the DSP's cost in participating in the auctions. Note that in a first-price auction, the payment upon winning is equal to the bid, so we can compute the mean cost in a way similar to MES. Let the winning cost matrix be denoted as  $C = \text{diag}(B_1, B_2, \dots, B_M)$ ; replacing the winning surplus matrix  $S_i$  with  $C$  in MES yields:

- **MEC** is the mean expected cost of the DSP over the auctions  $\mathcal{A}$ . Given the decisions  $\{\hat{b}_i\}_{i \in \mathcal{A}}$  and the constructed labels  $\{p_i\}_{i \in \mathcal{A}}$ , the MEC is calculated as  $\sum_{i \in \mathcal{A}} (CL p_i)^T \hat{b}_i / |\mathcal{A}|$ .

Note that, in general, a higher MES is better, while lower MLL and MEC values are preferred.

### 5.2 Evaluation on i.i.d. Data

We first compare different methods on i.i.d. data. Here, i.i.d. means that both training and evaluation are conducted using sufficient data from the training set. This represents an ideal setting where the model's generalization ability has minimal impact on performance. Following this, we conducted experiments on the iPinYou dataset by ad campaign, and the MES results are shown in Table 1.

**Table 1: The MES Performance on the iPinYou Training Set**

| Camp. | EEB   | PFB   | DFB   | SCA   | SPO   | EDFB  |
|-------|-------|-------|-------|-------|-------|-------|
| 1458  | 19.52 | 26.92 | 26.56 | 27.03 | 26.98 | 27.03 |
| 2259  | 14.43 | 23.48 | 23.57 | 23.75 | 23.67 | 23.75 |
| 2261  | 15.96 | 23.77 | 24.46 | 24.53 | 24.58 | 24.54 |
| 2821  | 15.67 | 23.30 | 23.37 | 23.53 | 23.48 | 23.37 |
| 2997  | 17.23 | 23.28 | 22.71 | 23.62 | 24.05 | 24.16 |
| 3358  | 12.90 | 19.58 | 19.61 | 19.79 | 19.78 | 19.81 |
| 3386  | 18.32 | 25.57 | 25.32 | 25.71 | 25.61 | 25.71 |
| 3427  | 14.48 | 20.97 | 20.85 | 21.13 | 21.07 | 21.12 |
| 3476  | 14.57 | 21.36 | 21.29 | 21.53 | 21.45 | 21.53 |

Overall, PFB, DFB implementations (including the DFB, SCA and SPO columns), and EDFB all achieve significant improvements over EEB, demonstrating the advantage of the predict-then-optimize paradigm compared to black-box end-to-end models. However, the performance differences among PFB, DFB, and EDFB are minor. Recalling the analysis in Section 4.2, this may be attributed to the data distribution of iPinYou as well as the model we use, that is, our model is sufficiently expressive to capture the true relation between features and labels of the PFB task.

We also performed experiments on the Tencent dataset. In particular, based on the significantly worse performance of EEB in the iPinYou experiments, we omit EEB and compare the DFB implementations and EDFB with PFB on the Tencent dataset. The results are shown in Table 2. It can be seen that both EDFB and DFB achieve higher MES than PFB on the Tencent training set, with EDFB achieving an **11%** improvement over PFB, which clearly demonstrates the effectiveness of our approach.

**Table 2: Evaluation on the Tencent Training Set**

| Metric     | PFB   | DFB   | SCA   | SPO   | EDFB         |
|------------|-------|-------|-------|-------|--------------|
| <b>MES</b> | 1.626 | 1.798 | 1.412 | 1.587 | <b>1.811</b> |
| MLL        | 0.23  | 0.61  | 11.24 | 0.51  | 0.25         |
| MEC        | 0.474 | 0.458 | 0.117 | 1.031 | 0.510        |

In summary, under i.i.d. data, DFB directly optimizes the surplus, which gives it an advantage over PFB, whose goal is the prediction accuracy. However, when the underlying ground truth relation is simple enough for the model to capture well, the advantage of DFB would diminish, as shown by the iPinYou results in Table 1.

### 5.3 Evaluation on Non-i.i.d. Data

We further conduct experiments on the non-i.i.d. data, which better reflects the DSP's situation in practice. We similarly train the

prediction models on the training set, but evaluate them on the test set. The results are shown in Table 3 and Table 4, respectively.

**Table 3: The MES Performance on the iPinYou Test Set**

| Camp. | EEB   | PFB          | DFB   | SCA   | SPO   | EDFB         |
|-------|-------|--------------|-------|-------|-------|--------------|
| 1458  | 18.72 | 19.96        | 19.57 | 12.07 | 16.72 | <b>20.02</b> |
| 2259  | 11.26 | 13.82        | 13.72 | 10.31 | 10.43 | <b>13.97</b> |
| 2261  | 16.34 | 18.50        | 18.01 | 16.02 | 15.55 | <b>18.55</b> |
| 2821  | 13.02 | 14.31        | 13.89 | 10.73 | 10.71 | <b>14.41</b> |
| 2997  | 18.11 | 19.58        | 19.48 | 16.38 | 15.31 | <b>19.67</b> |
| 3358  | 10.99 | <b>12.10</b> | 11.35 | 8.70  | 10.80 | 12.03        |
| 3386  | 17.43 | 18.93        | 18.39 | 15.20 | 15.55 | <b>18.96</b> |
| 3427  | 14.01 | 15.59        | 15.26 | 12.60 | 13.00 | <b>15.61</b> |
| 3476  | 14.25 | 15.19        | 15.16 | 11.60 | 11.87 | <b>15.21</b> |

**Table 4: Evaluation on the Tencent Test Set**

| Metric | PFB   | DFB   | SCA   | SPO   | EDFB         |
|--------|-------|-------|-------|-------|--------------|
| MES    | 1.508 | 1.659 | 1.363 | 1.494 | <b>1.759</b> |
| MLL    | 0.24  | 0.67  | 11.37 | 0.49  | 0.27         |
| MEC    | 0.481 | 0.413 | 0.110 | 0.954 | 0.468        |

We can see that on non-i.i.d. data, the performance of SCA and SPO declines to some extent. Relatively, our DFB appears to generalize better: its performance on the iPinYou dataset is close to PFB, while it outperforms PFB on the Tencent dataset. Given this empirical performance, we adopt DFB to implement EDFB.

Finally, our EDFB method combines the advantages of both the PFB and DFB methods, achieving essentially the best performance on the iPinYou dataset, while delivering significantly superior results on the Tencent dataset: a 16% improvement over PFB and 6% over DFB. It is worth noting that on the previous Tencent training set, EDFB achieves only a 0.7% improvement over DFB. As a comparison, the 6% improvement on the test set demonstrates the effectiveness of EDFB’s generalization capability in the non-i.i.d. environment. Overall, these results support the advantage of EDFB in real-world scenarios.

#### 5.4 Impact of the Weight in EDFB

We evaluate the impact of the weight parameter  $\alpha$  on the performance of EDFB. We selected portions of Tencent training and test sets and evaluated EDFB with different values of  $\alpha$ . The results are shown in Figure 3, where  $\alpha = 0$  corresponds to EDFB reducing to PFB, and  $\alpha = 1$  corresponds to EDFB reducing to DFB. It can be seen that when  $\alpha$  is chosen as an intermediate value, the objectives of learning factors and learning decisions are appropriately combined, resulting in better performance than either method alone. Although we noticed that improvement does not appear to follow a simple unimodal relationship w.r.t. the weight  $\alpha$ , which may limit the search for the optimal parameter, Figure 3 shows that  $\alpha$  can bring significant improvements across a fairly wide range. This demonstrates the robustness and stability of EDFB’s advantage and

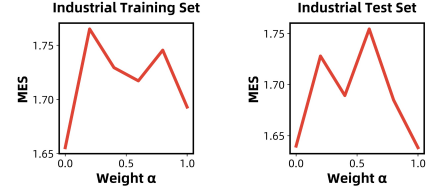


**Figure 3: The MES of the EDFB method with different weight  $\alpha$  on the Tencent dataset.**

facilitates its practical application. Based on these, we simply set  $\alpha = 0.5$  in all our experiments.

#### 5.5 Online Experiment

Our EDFB method has been fully deployed in Tencent Ads. During deployment, we conducted an A/B test comparing EDFB with the PFB baseline, and the details are presented as follows.

- *Deployment.* Our method only modifies the loss for training the winning price prediction model. In particular, the winning price model structure is consistent with that described in Section 5.1.

- *Scope.* Our A/B test was conducted from Nov. 26, 2025 to Nov. 28, 2025, including about 100 billion requests over 3 days. During this period, incoming bidding requests to the DSP were randomly split between the EDFB and PFB methods with an 80%–20% ratio.

- *Metrics.* We adopt common metrics in bidding systems [7, 22], including gross merchandise volume (GMV), cost per mille (CPM), cost per click (CPC), and cost per acquisition (CPA). Among these, GMV can be viewed as the realized outcome of the expectation in MES, serving as the main metric; while CPM, CPC, and CPA measure the costs associated with impressions, clicks, and conversions in bidding, serving as auxiliary metrics. In particular, since we modify the loss computation, we additionally monitor the average backward-pass time (ABT) to assess the training overhead.

- *Statistics.* We report the relative improvement (RI) for each metric, which is the percentage gain of EDFB over PFB.

**Table 5: The Results of Online A/B Test**

|    | GMV   | CPM   | CPC   | CPA   | ABT   |
|----|-------|-------|-------|-------|-------|
| RI | 2.30% | 1.58% | 1.69% | 1.14% | 0.07% |

The results of our A/B experiment are shown in Table 5. As shown, EDFB significantly improves GMV by 2.3%, while incurring only a minor burden in ABT. In addition, the CPM, CPC, and CPA metrics indicate that the cost for the DSP to obtain a unit of ad performance increases, which may be because the surplus-maximizing objective of EDFB places greater emphasis on requests with high surplus, and winning such requests typically requires higher bids. We also observe that the online improvements are smaller than the offline ones. This may be due to the discrepancy between our modeling and the complex real-world environment. We leave further improvements for future work.

## 6 Conclusion

In this work, we investigate bidding in first-price auctions, a canonical instance of the predict-then-optimize workflow. We address the misalignment between the prediction and decision objectives by proposing the DFB approach, in which we use a softmax relaxation of the decision mapping to train the prediction model on the downstream objective. Furthermore, we integrate the accuracy objectives into DFB to develop EDFB to further improve the model's generalization ability in non-i.i.d. environments. Our results offer valuable insights for predict-then-optimize workflows in large-scale networked platforms.

## Acknowledgments

This work was supported in part by China NSF grant No. U2268204, 62322206, 62432007, 62441236, 62132018, 62025204, 62272307, 62372296, U25A6024. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

## References

- [1] Brandon Amos. 2019. *Differentiable Optimization-Based Modeling for Machine Learning*. Ph.D. Dissertation. Carnegie Mellon University.
- [2] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. 2020. Learning with Differentiable Perturbed Optimizers. In *NeurIPS*. 9508–9519.
- [3] Hemant K Bhargava. 2022. The creator economy: Managing ecosystem supply, revenue sharing, and platform design. *Management Science* 68, 7 (2022), 5233–5251.
- [4] Adam N Elmachtoub and Paul Grigas. 2022. Smart “predict, then optimize”. *Management Science* 68, 1 (2022), 9–26.
- [5] Adam N. Elmachtoub, Henry Lam, Haofeng Zhang, and Yunfan Zhao. 2025. Estimate-Then-Optimize versus Integrated-Estimation-Optimization versus Sample Average Approximation: A Stochastic Dominance Perspective. arXiv:2304.06833
- [6] Shu-Cherng Fang and Sarat Puthenpura. 1993. *Linear optimization and extensions: theory and algorithms*. Prentice-Hall, Inc.
- [7] Djordje Gligorijevic, Tian Zhou, Bharatbhusan Shetty, Brendan Kitts, Shengjun Pan, Junwei Pan, and Aaron Flores. 2020. Bid Shading in The Brave New World of First-Price Auctions. In *CIKM*. 2453–2460.
- [8] Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *ICLR*.
- [9] Zhen Gong, Lvyin Niu, Yang Zhao, Miao Xu, Haoqi Zhang, Zhenzhe Zheng, Zhilin Zhang, Rongquan Bai, Chuan Yu, Jian Xu, Bo Zheng, and Fan Wu. 2023. MEBS: Multi-task End-to-end Bid Shading for Multi-slot Display Advertising. In *CIKM*. 4588–4594.
- [10] Jiani Huang, Zhenzhe Zheng, Yanrong Kang, and Zixiao Wang. 2024. From Second to First: Mixed Censored Multi-Task Learning for Winning Price Prediction. In *WSDM*. 295–303.
- [11] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.
- [12] Xin Jin, Hongqiang Harry Liu, Rohan Gandhi, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Jennifer Rexford, and Roger Wattenhofer. 2014. Dynamic scheduling of network updates. In *SIGCOMM*. 539–550.
- [13] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
- [14] Matt J. Kusner and José Miguel Hernández-Lobato. 2016. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. arXiv:1611.04051
- [15] Mathieu Leconte, Georgios S. Paschos, Panayotis Mertikopoulos, and Ulas C. Kozat. 2018. A Resource Allocation Framework for Network Slicing. In *INFOCOM*. 2177–2185.
- [16] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *KDD*. 768–776.
- [17] Xu Li, Michelle Ma Zhang, Zhenya Wang, and Youjun Tong. 2022. Arbitrary Distribution Modeling with Censorship in Real-Time Bidding Advertising. In *KDD*. 3250–3258.
- [18] Jayanta Mandi and Tias Guns. 2020. Interior Point Solving for LP-based prediction+optimisation. In *NeurIPS*. 7272–7282.
- [19] Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. 2024. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research* 80 (2024), 1623–1701.
- [20] Mathias Niepert, Pasquale Minervini, and Luca Franceschi. 2021. Implicit MLE: Backpropagating Through Discrete Exponential Family Distributions. In *NeurIPS*. 14567–14579.
- [21] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press.
- [22] Shengjun Pan, Brendan Kitts, Tian Zhou, Hao He, Bharatbhusan Shetty, Aaron Flores, Djordje Gligorijevic, Junwei Pan, Tingyu Mao, San Gultekin, and Jianlong Zhang. 2020. Bid Shading by Win-Rate Estimation and Surplus Maximization. arXiv:2009.09259
- [23] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In *SIGIR*. 859–868.
- [24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. In *ICML*. 8821–8831.
- [25] Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, and Yong Yu. 2019. Deep Landscape Forecasting for Real-time Bidding Advertising. In *KDD*. 363–372.
- [26] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.
- [27] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *WWW*. 521–530.
- [28] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training Convolutional Networks with Noisy Labels. In *ICLR Workshop*.
- [29] Alexandre B. Tsybakov. 2009. Lower bounds on the minimax risk. In *Introduction to Nonparametric Estimation*. Springer New York, 77–135.
- [30] Jun Wang, Weinan Zhang, and Shuai Yuan. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. *Foundations and Trends® in Information Retrieval* 11, 4-5 (2017), 297–435.
- [31] Yuchen Wang, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Functional Bid Landscape Forecasting for Display Advertising. In *ECML PKDD*. 115–131.
- [32] Zixiao Wang, Zhenzhe Zheng, Yanrong Kang, and Jiani Huang. 2024. Cost-Effective Active Learning for Bid Exploration in Online Advertising. In *WSDM*. 788–796.
- [33] Bryan Wilder, Bistra Dilkina, and Milind Tambe. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *AAAI*. 1658–1665.
- [34] Wush Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2018. Deep Censored Learning of the Winning Price in the Real Time Bidding. In *KDD*. 2526–2535.
- [35] Wush Chi-Hsuan Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2015. Predicting Winning Price in Real Time Bidding with Censored Data. In *KDD*. 1305–1314.
- [36] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. 2021. Towards a Theoretical Framework of Out-of-Distribution Generalization. In *NeurIPS*. 23519–23531.
- [37] Wei Zhang, Brendan Kitts, Yanjun Han, Zhengyuan Zhou, Tingyu Mao, Hao He, Shengjun Pan, Aaron Flores, San Gultekin, and Tsachy Weissman. 2021. MEOW: A Space-Efficient Nonparametric Bid Shading Algorithm. In *KDD*. 3928–3936.
- [38] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *KDD*. 1077–1086.
- [39] Weinan Zhang, Shuai Yuan, Jun Wang, and Xuehua Shen. 2015. Real-Time Bidding Benchmarking with iPinYou Dataset. arXiv:1407.7073
- [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 *KDD*. 1059–1068.
- [41] Tian Zhou, Hao He, Shengjun Pan, Niklas Karlsson, Bharatbhusan Shetty, Brendan Kitts, Djordje Gligorijevic, San Gultekin, Tingyu Mao, Junwei Pan, Jianlong Zhang, and Aaron Flores. 2021. An Efficient Deep Distribution Network for Bid Shading in First-Price Auctions. In *KDD*. 3996–4004.