Joint Optimization of Ranking and Calibration with Contextualized Hybrid Model

Xiang-Rong Sheng
xiangrong.sxr@alibaba-inc.com
Alibaba Group
Beijing, China

Jingyue Gao
jingyue.gjy@alibaba-inc.com
Alibaba Group
Beijing, China

Yueyao Cheng
chengyueyao.cyy@alibaba-inc.com
Alibaba Group
Beijing, China

Siran Yang
siran.ysr@alibaba-inc.com
Alibaba Group
Beijing, China

Shuguang Han*
shuguang.sh@alibaba-inc.com
Alibaba Group
Beijing, China

Hongbo Deng
dhb167148@alibaba-inc.com
Alibaba Group
Beijing, China

Yuning Jiang
mengzhu.jyn@alibaba-inc.com
Alibaba Group
Beijing, China

Jian Xu
xiyu.xj@alibaba-inc.com
Alibaba Group
Beijing, China

Bo Zheng
bozheng@alibaba-inc.com
Alibaba Group
Beijing, China

ABSTRACT

Despite the development of ranking optimization techniques, pointwise loss remains the dominating approach for click-through rate prediction. It can be attributed to the **calibration ability** of the pointwise loss since the prediction can be viewed as the click probability. In practice, a CTR prediction model is also commonly assessed with the **ranking ability**. To optimize the ranking ability, ranking loss (e.g., pairwise or listwise loss) can be adopted as they usually achieve better rankings than pointwise loss. Previous studies have experimented with a direct combination of the two losses to obtain the benefit from both losses and observed an improved performance. However, previous studies break the meaning of output logit as the click-through rate, which may lead to sub-optimal solutions. To address this issue, we propose an approach that can **J**ointly optimize the **R**anking and **C**alibration abilities (**JRC** for short). JRC improves the ranking ability by contrasting the logit value for the sample with different labels and constrains the predicted probability to be a function of the logit subtraction. We further show that JRC consolidates the interpretation of logits, where the logits model the joint distribution. With such an interpretation, we prove that JRC approximately optimizes the contextualized hybrid discriminative-generative objective. Experiments on public and industrial datasets and online A/B testing show that our approach improves both ranking and calibration abilities. Since May 2022, JRC has been deployed on the display advertising platform of Alibaba and has obtained significant performance improvements.

*Shuguang Han is the corresponding author.

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

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599851

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

Click-Through Rate Prediction, Calibration, Hybrid Model

ACM Reference Format:
Xiang-Rong Sheng, Jingyue Gao, Yueyao Cheng, Siran Yang, Shuguang Han, Hongbo Deng, Yuning Jiang, Jian Xu, and Bo Zheng. 2023. Joint Optimization of Ranking and Calibration with Contextualized Hybrid Model. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599851

1 INTRODUCTION

Click-through rate (CTR) prediction has played a vital role in many industrial applications, such as recommender systems and online advertising. A CTR prediction model is commonly evaluated from two perspectives: whether the prediction aligns with the actual click-through rate (i.e., calibration ability) [20, 31], and whether the prediction leads to a correct ranking (i.e., ranking ability) [20, 28].

Despite recent progress in ranking optimization techniques, the pointwise model [10, 16, 23, 43, 46] remains the dominating approach for CTR prediction. The wide use of the pointwise model is mainly attributed to its calibration ability since the prediction can be treated as the click probability [22]. However, in the pointwise model, direct optimization of the cross-entropy loss may fail to improve the ranking ability. Such a loss function does not consider the discriminative ability among items, which, on the contrary, is the primary measurement for the ranking ability [30]. In addition, the identical and independently distributed (i.i.d.) assumption in the pointwise model may deviate from real-world applications. For example, users typically see a list of items and make click decisions. Therefore, researchers have proposed various pairwise and listwise approaches to directly optimize ranking metrics such as AUC (Area Under the Curve) [13], nDCG (normalized discounted cumulative

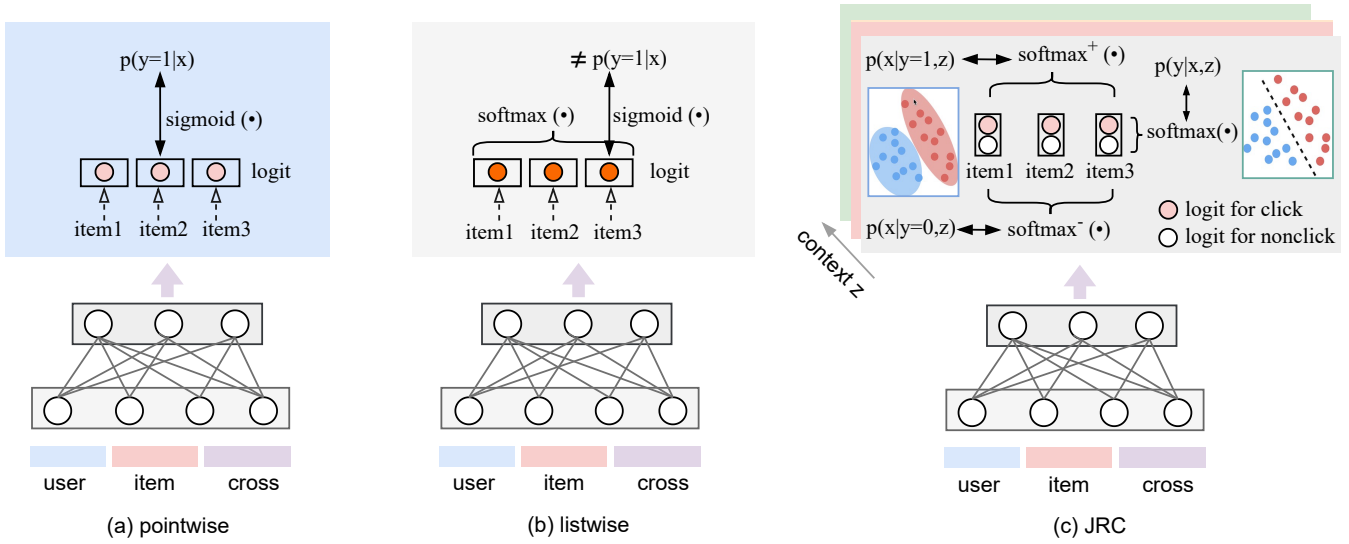


Figure 1: A comparison among the three approaches: (a) pointwise model, (b) listwise model, and (c) our proposed JRC model.

gain) [24], MAP (Mean Average Precision) [1]. They improve the ranking-related metrics to a large extent but introduce new issues.

With a pairwise or listwise approach, the predictions will no longer correspond to click probability [3, 30]. In general, logits[1] in the pointwise and pairwise/listwise methods are defined based on different underlying assumptions. As illustrated by Figure 1(a), the logit in a pointwise model can be easily converted (after sigmoid transformation) to the click probability, whereas the logit in a listwise model, e.g., ListNet [7], only measures the degree of relevance since the softmax function is conducted over all of the items within the same list, lacking a direct connection to the click probability. To preserve the benefits from both pointwise and pairwise/listwise approaches, an intuitive way, is to combine the two loss functions [28, 39]. Despite being effective, the meaning of the integrated logit becomes unclear as it is adopted to define two different probabilities, i.e., the probability of click and the probability of being ranked on the top.

The abovementioned issues call for the redesign of logit to optimize ranking and calibration within the same framework. To this end, we propose a **J**oint optimization approach that maximizes the **R**anking and **C**alibration abilities (**JRC** for short) at the same time. To avoid carrying different meanings using one single logit, we extend the one-dimensional logit to two-dimensional logits, i.e., adding another degree of freedom to the model output. Let f_θ denote the model parameterized by θ, the output of this model is a vector of two values $f_\theta(\mathbf{x})[0]$ and $f_\theta(\mathbf{x})[1]$. $f_\theta(\mathbf{x})[0]$ and $f_\theta(\mathbf{x})[1]$ represent the logits corresponding to the non-click state and click state, respectively. Hereafter, we refer them as to the click-logit and the nonclick-logit.

With these two logits, we may simply adopt a multi-task modeling approach that uses one for ranking optimization and the other for calibration optimization. However, in this way, such two logits are connected indirectly, making it hard to fuse them for the final prediction. On contrary, as shown in Equation 1, JRC computes the predicted click probability based on the subtraction $f_\theta(\mathbf{x})[1] - f_\theta(\mathbf{x})[0]$, in which both of the two logits contributed to the final prediction. Here, the predicted click probability is obtained by applying the sigmoid function on $f_\theta(\mathbf{x})[1] - f_\theta(\mathbf{x})[0]$, which is equivalent to applying the softmax function over the two logits.

$$\hat{p}(y=1|\mathbf{x}) = \frac{1}{1+\exp(-(f_\theta(\mathbf{x})[1] - f_\theta(\mathbf{x})[0]))} \tag{1}$$

As for the optimization objective, we firstly introduce a pointwise loss, using the two logits, to preserve the calibration ability:

$$\begin{aligned}\ell_{\text{calib}} &= -\sum_{\mathbf{x},y} \log \hat{p}(y|\mathbf{x}) \\ &= -\sum_{\mathbf{x},y} \log \frac{\exp(f_\theta(\mathbf{x})[y])}{\exp(f_\theta(\mathbf{x})[0]) + \exp(f_\theta(\mathbf{x})[1])}.\end{aligned} \tag{2}$$

To further improve the ranking ability, we adopt a listwise-like loss. For each positive sample $(\mathbf{x}, y = 1)$, the listwise-like loss contrasts its click-logit to the click-logits of all other samples within the same context z. For each negative sample $(\mathbf{x}, y = 0)$, the listwise-like loss encourages its nonclick-logit to be larger than the nonclick-logits of all other samples. Such a process can be illustrated by Equation 3,

$$\ell_{\text{rank}} = -\sum_{\mathbf{x},y,z} \log \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{\mathbf{x}_i \in X_z} \exp(f_\theta(\mathbf{x}_i)[y])}, \tag{3}$$

where z indicates the current context, e.g., the current session [28]. X_z denote the set of samples that share the same context z. The final objective of JRC is defined as

$$\ell_{\text{final}} = \alpha \ell_{\text{calib}} + (1-\alpha)\ell_{\text{rank}}, \tag{4}$$

in which α is the hyper-parameter that balances the importance of different loss functions.

[1] In this research, the logit refers to the model output *before* the last activation layer (e.g., sigmoid or softmax function).

By introducing an additional degree of freedom, the above formulation alleviates the conflict of optimizing ranking and calibration with only one single logit. We further show that JRC consolidates the interpretation of logits for both pointwise loss and listwise-like loss. Particularly, JRC can be seen as an energy-based model (EBM) [27], in which logits are treated as the energy values assigned to each state (click state or non-click state). EBM defines the joint distribution of ($\mathbf{x}$, y, z) with the energy values: $\hat{p}(\mathbf{x}, y, z) = \frac{\exp(f_\theta(\mathbf{x})[y])}{Z(\theta)}$, where $Z(\theta)$ is the unknown normalization constant. With such a definition, we prove that (in Section 3.3) ℓ_{rank} is approximately optimizes the contextualized generative objective

$$\ell_{\text{rank}} \approx -\sum_{\mathbf{x},y,z} \log \hat{p}(\mathbf{x}|y,z), \tag{5}$$

in which $\hat{p}(\mathbf{x}|y,z)$ stands for the probability of $\mathbf{x}$ occurring in the click samples (non-click samples) under the context z. Hence, JRC is approximately optimizing the contextualized hybrid discriminative and generative objective,

$$-\sum_{\mathbf{x},y,z} \alpha \log \hat{p}(y|\mathbf{x}) + (1-\alpha) \log \hat{p}(\mathbf{x}|y,z). \tag{6}$$

The detailed analysis is offered in Section 3.4. In this sense, the new definition offers a unified interpretation for logit, and the click-through rate can be derived from the two logits.

Our later experiments validate the effectiveness of JRC in both offline and online experiments. By introducing the listwise-like generative loss, we see an improvement of ranking ability. Furthermore, consistent with previous studies on generative modeling [17, 29], we also observe an improvement of calibration ability. One particularly interesting point is that, compared to the simple loss combination [28, 39], our proposed approach further improves the model performance with the unified logit interpretation.

To summarize, the main contributions of our work are as follows:

- We propose the JRC approach for joint optimization of ranking and calibration in the task of click-through rate prediction. JRC extends the model output from the one-dimensional logit to two-dimensional logits, alleviating the conflict of multiple objectives. By optimizing the ranking of the click and non-click logits while constraining their subtraction to be the predicted probability, JRC effectively improves the ranking ability and keeps it well-calibrated.
- We provide a deep analysis of the effectiveness of JRC and show that it unifies the interpretation of logits. By defining the joint distribution $\hat{p}(\mathbf{x}, y)$ with the new logits, we demonstrate that JRC approximately optimizes the contextualized hybrid discriminative and generative objective. This unification provides a theoretical explanation for the effectiveness of JRC and also helps explain the efficacy of previous attempts on direct loss combination [28, 39].
- We verify the effectiveness of JRC in industrial datasets and examine the model performance through online A/B testing on the display advertising platform of Alibaba. The experiment results show that JRC simultaneously improves the ranking and calibration abilities. During online A/B testing, JRC has also exhibited significant performance improvement, leading to a 4.4% increase in CTR and a 2.4% increase in RPM.

2 RELATED WORK

We briefly review related studies from three aspects: click-through rate prediction, learning-to-rank, and hybrid modeling.

Most research efforts on **click-through rate prediction** have been devoted to improving model architectures. Wide & Deep [10] and deepFM [21] combine low-order and high-order features to improve model expressiveness. PNN [37] introduces a product layer to capture interactive patterns between inter-field categories. Considering the importance of historical user behaviors, DIN [46] employs the attention mechanism to activate historical behaviors locally w.r.t. the target item and captures the diversity of user interest. Moreover, inspired by the success of the self-attention architecture in the tasks of sequence to sequence learning, Transformer is also introduced for feature aggregation [14]. Despite the recent advancement, less attention has been paid to defining a better loss function. Up to now, the pointwise model with a LogLoss has been the dominating paradigm for CTR prediction [46]. The pointwise loss often yields well-calibrated predictions but may lead to a sub-optimal ranking performance [30]. However, in practical industrial systems, both the calibration and ranking quality are essential when measuring the performance of a CTR prediction model [28, 39]. This requirement calls for research on direct optimization of the ranking.

A **Learning-To-Rank** (LTR) algorithm generally addresses the problem of ranking ability optimization. The main goal is to learn a scoring function for computing the degree of relevance, which further induces a ranking list [4, 30, 41]. In its most straightforward format, a ranking problem can be viewed as correctly estimating the relevance score, and thus being named as the pointwise approach [15]. Another line of work has been focusing on correctly predicting ordered pairs by optimizing a pairwise loss [6, 18], which simplifies the learning-to-rank problem while misaligning with the ranking utilities. A ranking system is commonly evaluated with a set of ranking metrics such as Area Under the Curve [13], Normalized Discounted Cumulative Gain [24] or MAP (Mean Average Precision) [1]. Considering the gap between the optimization goal and evaluation metric, researchers have proposed a set of ranking-metric-based optimization approaches [5, 7, 36].

One critical issue for the non-pointwise LTR algorithms is that the prediction score no longer aligns with the click probability. To preserve both calibration and ranking quality, prior studies have examined a linear combination of pointwise and non-pointwise loss [28, 39, 42]. Despite being effective, the underlying meaning of the output is not well-understood as the two losses correspond to different goals. Recently, some studies have proposed to model the uncertainty of neural rankers and yield better uncertainty estimation [11, 35, 42]. Unlike these works, this research focuses on the problem of jointly optimizing the ranking and calibration ability of CTR prediction models. To this end, we propose JRC and derive the pointwise and listwise-like losses. JRC offers a unified view of multiple losses, redefining logits as the energy of click/non-click states.

Under the above definition, we show that JRC is approximately modeling the contextualized **hybrid discriminative-generative** [29, 38] objectives. In concrete, hybrid model combines generative and discriminative models to get the best of both paradigms. It has been observed that generative models benefit downstream problems such

as calibration [17, 38], semi-supervised learning [9, 25] and the handling of missing data [32]. By contrast, discriminative models are usually more accurate when labeled training data is abundant. By combining them, hybrid modeling achieves superior performance and produces more calibrated output. However, context information is largely ignored in previous studies. The proposed JRC method extends the idea of hybrid modeling with contextualization for CTR prediction. Incorporating context information further enables our model to be personalized and adaptable, which better meets online system requirements. In this work, we observe that the contextualized hybrid model JRC simultaneously improves the ranking quality and calibration, and the result is also consistent with prior studies [17, 29, 38].

3 METHODOLOGY

In the following, we give a brief description about the background firstly and then introduce the proposed JRC approach.

3.1 Background

In CTR prediction, the pointwise model takes the input as $(\mathbf{x}, y) \sim (X, Y)$, and learns a function f with parameter θ, $f_\theta : \mathbb{R}^D \to \mathbb{R}^1$, that maps $\mathbf{x}$ to a logit. Here, $\mathbf{x}$ denotes the feature and $y \in \{0, 1\}$ indicates the click label. The logit will then be used to compute the predicted click probability (pCTR for short) with the below sigmoid function:

$$\hat{p}(y = 1|\mathbf{x}) = \frac{\exp(f_\theta(\mathbf{x}))}{1 + \exp(f_\theta(\mathbf{x}))}. \tag{7}$$

In industrial applications like online advertising, calibrated predicted click probability is often required [28]. Thus pointwise models are widely adopted as their predictions can be treated as the click probability [22], in which the LogLoss is minimized:

$$-\sum_{\mathbf{x},y} \log \hat{p}(y|\mathbf{x}). \tag{8}$$

Pointwise models are well-calibrated but have inferior ranking ability. In this study, we propose a Joint optimization of Ranking and Calibration approach (JRC).

3.2 Joint Optimization of Ranking and Calibration

As mentioned above, previous attempts to improve ranking for CTR prediction model directly combines pointwise and pairwise losses [28, 39]. Despite being effective, the meaning of the logit becomes unclear. To avoid representing different meanings with the same logit, the proposed JRC extends the output logit from 1 dimension to 2 dimensions, $f_\theta : \mathbb{R}^D \to \mathbb{R}^2$, as shown in Figure 1(c). The intuition of introducing an additional degree of freedom is to alleviate the conflict between the optimization of ranking and calibration.

Let $f_\theta(\mathbf{x})[y]$ indicates the y-th index of $f_\theta(\mathbf{x})$. In JRC, $f_\theta(\mathbf{x})[1]$ is the logit corresponding the click state and $f_\theta(\mathbf{x})[0]$ is the logit corresponds to the non-click state. JRC computes the predicted probability based on the subtraction $f_\theta(\mathbf{x})[1] - f_\theta(\mathbf{x})[0]$:

$$\hat{p}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(f_\theta(\mathbf{x})[1] - f_\theta(\mathbf{x})[0]))}. \tag{9}$$

Given the predicted probability, we firstly introduce the pointwise loss to preserve the calibration ability:

$$\begin{aligned} \ell_{\text{calib}} &= -\sum_{\mathbf{x},y} \log \hat{p}(y|\mathbf{x}) \\ &= -\sum_{\mathbf{x},y} \log \frac{\exp(f_\theta(\mathbf{x})[y])}{\exp(f_\theta(\mathbf{x})[0]) + \exp(f_\theta(\mathbf{x})[1])}. \end{aligned} \tag{10}$$

Note that Equation 10 is equivalent to the standard cross-entropy loss. In recommendation, every item is displayed to the user in a specific context, e.g., the specific spot that item presented to users [40]. To improve the relative ranking of the same context, we add a listwise-like loss aiming to learn the ranking in a specific context,

$$\ell_{\text{rank}} = -\sum_{\mathbf{x},y,z} \log \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{\mathbf{x}_i \in X_z} \exp(f_\theta(\mathbf{x}_i)[y])}, \tag{11}$$

where z indicates the current context and X_z denote the set of samples that share the same context z.

The final objective of JRC can be written as:

$$\ell_{\text{final}} = \alpha \ell_{\text{calib}} + (1 - \alpha)\ell_{\text{rank}}, \tag{12}$$

where α is the hyper-parameter between [0, 1]. In doing so, JRC optimizes the ranking ability through contrasting the logit value for a sample x from the logit values of other samples with different labels, and constrains the predicted probability to be a monotonic function of the logit subtraction for calibration.

3.3 Contextualized Hybrid Discriminative-Generative Model

We further show that in JRC, the logits have a unified interpretation. In particular, JRC can be seen as an energy-based model (EBM) [27]. In concrete, we treat the two-dimensional logits as energy values of $(\mathbf{x}, y, z)$ as Equation 13:

$$\hat{p}(\mathbf{x}, y, z) = \hat{p}(\mathbf{x}, y) = \frac{\exp(f_\theta(\mathbf{x})[y])}{Z(\theta)}, \tag{13}$$

where $Z(\theta)$ is the unknown normalization constant. The equation $\hat{p}(\mathbf{x}, y, z) = \hat{p}(\mathbf{x}, y)$ holds since the context feature can be added to $\mathbf{x}$ thus we can absorb z into $\mathbf{x}$. Given a specific context z, $\hat{p}(\mathbf{x}, y|z)$ is calculated as Equation 14,

$$\begin{aligned} \hat{p}(\mathbf{x}, y|z) &= \frac{\hat{p}(\mathbf{x}, y, z)}{\hat{p}(z)} = \frac{\hat{p}(\mathbf{x}, y)}{\hat{p}(z)} \\ &= \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{\mathbf{x}_i \in \mathcal{X}_z} \sum_{y'} \exp(f_\theta(\mathbf{x}_i)[y'])}. \end{aligned} \tag{14}$$

$\mathcal{X}_z$ denote the full set of samples that includes z. Subsequently, we can compute the probability $\hat{p}(y|z)$ by marginalizing over all $\mathbf{x}$ with context z:

$$\hat{p}(y|z) = \frac{\sum_{\mathbf{x}_i \in \mathcal{X}_z} \exp(f_\theta(\mathbf{x}_i)[y])}{\sum_{\mathbf{x}_i \in \mathcal{X}_z} \sum_{y'} \exp(f_\theta(\mathbf{x}_i)[y'])}. \tag{15}$$

By marginalizing out y in Equation 13, we can also obtain the probability density $p(\mathbf{x})$:

$$\hat{p}(\mathbf{x}) = \hat{p}(\mathbf{x}, z) = \frac{\sum_{y'} \exp(f_\theta(\mathbf{x})[y'])}{Z(\theta)}. \tag{16}$$

Algorithm 1 A Tensorflow-style Pseudocode of our proposed JRC model.

```
# B: batch size, label: [B, 2], context_index: [B, 1]
# Feed forward computation to get the 2-dimensional logits
# and compute LogLoss -log p(y|x, z)
logits = feed_forward(inputs)
ce_loss = mean(CrosssEntropyLoss(logits, label))

# Mask: shape [B, B], mask[i,j]=1 indicates the i-th sample
# and j-th sample are in the same context
mask = equal(context_index, transpose(context_index))

# Tile logits and label: [B, 2]->[B, B, 2]
logits = tile(expand_dims(logits, 1), [1, B, 1])
y = tile(expand_dims(label, 1), [1, B, 1])

# Set logits that are not in the same context to -inf
y = y * expand_dims(mask, 2)
logits = logits + (1-expand_dims(mask, 2))*-1e9
y_neg, y_pos = y[:,:,0], y[:,:,1]
l_neg, l_pos = logits[:,:,0], logits[:,:,1]

# Compute listwise generative loss -log p(x|y, z)
loss_pos = -sum(y_pos * log(softmax(l_pos, axis=0)), axis=0)
loss_neg = -sum(y_neg * log(softmax(l_neg, axis=0)), axis=0)
ge_loss = mean((loss_pos+loss_neg)/sum(mask, axis=0))

# The final JRC model
loss = alpha*ce_loss + (1-alpha)*ge_loss
```

Table 1: A comparison of listwise softmax loss and JRC generative loss with several toy examples.

list (context)	listwise softmax loss [7]	listwise generative loss (in JRC)
condition a): (d_i^+, d_j^+)	$-(\log \frac{\exp s_i}{\exp s_i+\exp s_j} + \log \frac{\exp s_j}{\exp s_i+\exp s_j})$	$-(\log \frac{\exp t_i^1}{\exp t_i^1+\exp t_j^1} + \log \frac{\exp t_j^1}{\exp t_i^1+\exp t_j^1})$
condition b): (d_i^-, d_j^-)	0	$-(\log \frac{\exp t_i^0}{\exp t_i^0+\exp t_j^0} + \log \frac{\exp t_j^0}{\exp t_i^0+\exp t_j^0})$
condition c): (d_i^+, d_j^-)	$-\log \frac{\exp s_i}{\exp s_i+\exp s_j}$	$-(\log \frac{\exp t_i^1}{\exp t_i^1+\exp t_j^1} + \log \frac{\exp t_j^0}{\exp t_i^0+\exp t_j^0})$

Then we can compute $\hat{p}(y|\mathbf{x})$ via $\hat{p}(\mathbf{x}, y)/\hat{p}(\mathbf{x})$ by dividing Equation 13 to Equation 16:

$$\hat{p}(y|\mathbf{x}) = \frac{\hat{p}(\mathbf{x}, y)}{\hat{p}(\mathbf{x})} = \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{y'} \exp(f_\theta(\mathbf{x})[y'])}. \tag{17}$$

We can see it yields the standard softmax parameterization.

Similarly, we can compute the $\hat{p}(\mathbf{x}|y, z)$ via $\hat{p}(\mathbf{x}, y|z)/\hat{p}(y|z)$ by dividing Equation 14 to Equation 15 as follows:

$$\hat{p}(\mathbf{x}|y, z) = \frac{\hat{p}(\mathbf{x}, y|z)}{\hat{p}(y|z)} = \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{\mathbf{x}_i \in \mathcal{X}_z} \exp(f_\theta(\mathbf{x}_i)[y])}. \tag{18}$$

The denominator $\sum_{\mathbf{x}_i \in \mathcal{X}_z} \exp(f_\theta(\mathbf{x}_i)[y])$ of Equation 18 require summation over all sample with context z that is intractable. Note that if we approximate the generative loss by replacing the full set of samples $\mathcal{X}_z$ with samples in the mini-batch X_z, then $-\log \hat{p}(\mathbf{x}|y, z)$ has the same form as the listwise-like loss in JRC! Thus ℓ_{rank} is actually approximately optimizing the contextualized generative objective, i.e., $\ell_{\text{rank}} \approx -\sum_{\mathbf{x},y,z} \log p(\mathbf{x}|y, z)$. In the later experiment section, we compare the different context choices and give a detailed analysis.

To sum up, we show that JRC has a novel objective function for CTR prediction that consists of both discriminative loss and context-dependent generative loss. Unlike non-pointwise LTR approaches, JRC first defines the joint probability and then derives the conditional probabilities, preserving the meaning of output $\hat{p}(y|\mathbf{x})$ as click-through rate for calibration. In the later experiment section, we show that the proposed JRC gains the benefits of both discriminative and generative approaches, improving both ranking and calibration abilities. The TensorFlow-like [12] pseudo code of JRC is presented in Algorithm 1.

3.4 Understanding JRC

For the discriminative loss in Equation 17, the subtraction between the two logits of JRC is equivalent to the one-dimensional logit in the pointwise approach. Interestingly, we observe that the listwise generative loss in Equation 18 is structurally similar to the listwise softmax loss. They share the same goal: contrasting the logit value for $\mathbf{x}$ with label y from the logit values of other training samples in the same context list.

To better illustrate the connections and distinctions between the listwise softmax loss and the generative loss of JRC, we provide several toy examples in Table 1. For each candidate d_i, we represent the corresponding listwise softmax logit as s_i, and the JRC logits as (t_i^0, t_i^1). Here, t_i^0 (and t_i^1) refers to the logit for the non-click (and click) state. We further represent the item with a positive (click) or a negative (non-click) label as d^+ and d^-, respectively. We take into account all of the three possible conditions for a list: a) all items are clicked, b) all items are non-clicked, and c) partial items are clicked. The list (for the listwise model) and the context (for our JRC model) are aligned for better comparison.

Table 1 shows that in list a), the listwise softmax loss is structurally similar to the listwise generative loss. Meanwhile, in list c), the listwise loss is also an essential component of the generative loss. However, they differ substantially in list b) as JRC explicitly models

Table 2: Statistics of the public and production datasets.

Dataset	#Training samples	#Testing samples	#Features
Avazu	28M	8M	22
Taobao	0.7M	0.3M	5
Production	1.8B	30M	286

the likelihood of generating the non-click data, which is overlooked in listwise models. One potential benefit of this approach is that it helps smooth the predictions of non-clicked samples, providing regularization for such data, thus improving the calibration. In addition, this also makes it possible to utilize non-click information that is usually abundant for industrial systems. It is worth noting that despite similar, the underlying definition for logit remains to be different between JRC and listwise models.

4 EXPERIMENT

In this section, we conduct extensive experiments to understand the effectiveness of the JRC model.

4.1 Experiment Setup

4.1.1 Datasets. Our experiments are conducted on two widely-adopted public datasets [48] and one production dataset. The basic statistics of these datasets are summarized in Table 2.

- **Avazu**[2]. Avazu is a Kaggle challenge dataset for CTR prediction. The data comes from a mobile advertising platform named Avazu and includes 22 features, such as device information, advertisement category, and other attributes. It consists of ten days data for training and validation and one day of data for testing. We use the benchmark data version provided by Zhu et al. [48].
- **Taobao**[3]. This dataset provides various user behavior information, including click, purchase, add-to-cart, and favorite, for about one million users from the Taobao recommendation system. It contains eight days of click-through data, with the first seven days for training and validation and the rest one day for testing. A few pre-processing steps are carried out by following [2].
- **Production**. The production dataset is sampled from the impression log of Alibaba online system. The full impression and click data from 2021/11/24 is adopted for training, and a subset of data from the next day is used for testing. The training set consists of 1.8 billion samples with 286 features, and the testing set contains 30 million samples.

4.1.2 Baselines. We include five baseline methods for comparison. According to the adopted loss function, we divide them into three groups. The first group (**G1**) is the pointwise model optimizing the LogLoss for each sample, and as a matter of fact, this is also the one deployed in the production system. The second group (**G2**) involves a pairwise model and a listwise model that directly optimize the ranking loss. Here, we utilize RankNet [5] for pairwise loss and ListNet [7] for listwise loss. They are adopted due to their superior performance and simplicity for implementation [3, 34]. The third group (**G3**) considers both ranking loss and calibrated pointwise loss, but fuses them with a linear combination [28, 39].

- **G1: Pointwise** model to optimize the standard LogLoss [30]. This is also our production baseline.
- **G2: RankNet** [6] that employs a pairwise loss to maximize the probability of correct ranking for pairs of samples.
- **G2: ListNet** [7] which defines a listwise loss to maximize the likelihood of generating the correct ranking list.
- **G3**: The **Combined-Pair** approach that derives a new loss by a linear combination of pointwise and pairwise loss [28].
- **G3**: The **Combined-List** approach that combines the pointwise and the ListNet [7].

4.1.3 Evaluation Metrics. For evaluating the ranking performance, we adopt the standard **AUC** metric (Area Under Receiver Operating Characteristic Curve). A larger AUC indicates a better ranking ability. Note that in the production dataset, we further compute the Group AUC (**GAUC**) to measure the goodness of intra-user ranking ability, which has shown to be more consistent with the online performance [40, 46, 47]. GAUC is also the top-line metric in the production system. It can be calculated with Equation 19, in which U represents the number of users, $\#\text{impression}(u)$ denotes the number of impressions for the u-th user, and AUC_u is the AUC computed only using the samples from the u-th user. However, due to the lack of user information in the Avazu dataset, and limited samples per user in Taobao, we cannot effectively compute the GAUC metric; therefore, the AUC metric is reported for the public datasets instead.

$$\text{GAUC} = \frac{\sum_{u=1}^{U} \#\text{impressions}(u) \times \text{AUC}_u}{\sum_{u=1}^{U} \#\text{impressions}(u)} \tag{19}$$

To measure the calibration performance of each method, we exploit a set of metrics including the averaged **LogLoss**, the expected calibration error (**ECE**), and the predicted CTR over the true CTR (**PCOC**) [20, 33]. They are all defined in Equation 20. For the i-th sample, let $\hat{p}_i$ represent the pCTR. For ECE, we first partition the range $[0, 1)$ equally into K buckets. $\mathbb{1}(\hat{p}_i \in B_k)$ is an indicator with a value of 1 if the predicted probability locates in the k-th bucket B_k, and otherwise 0. LogLoss measures the sample-level calibration error, whereas PCOC and ECE provide a dataset-level and a subset-level calibration measurement. A lower LogLoss or ECE implies a better performance; for PCOC, our goal is to obtain a value close to 1.0.

$$\begin{aligned}
\text{LogLoss} &= -\frac{1}{\mathcal{D}} \sum_{i=1}^{\mathcal{D}} \left(y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \right) \\
\text{ECE} &= \frac{1}{\mathcal{D}} \sum_{k=1}^{K} \left| \sum_{i=1}^{\mathcal{D}} (y_i - \hat{p}_i)\ \mathbb{1}(\hat{p}_i \in B_k) \right| \\
\text{PCOC} &= \frac{1}{\sum_{i=1}^{\mathcal{D}} y_i} \sum_{i=1}^{\mathcal{D}} \hat{p}_i
\end{aligned} \tag{20}$$

4.1.4 Implementation Details. In both Taobao and the production datasets, we utilize **CAN** [2] as the neural network structure for all methods. CAN has been shown to be effective for CTR prediction with user behavior sequence. However, due to a lack of

[2]https://www.kaggle.com/c/avazu-ctr-prediction/data
[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

Table 3: A comparison of model performance on two widely-adopted public datasets. The best results are highlighted in bold. A number with * indicates that the improvement over the *Combined-Pair* and *Combined-List* methods is significant with p-value < 0.05 by pairwise t-tests.

Dataset		Avazu				Taobao			
		AUC	LogLoss	PCOC	ECE	AUC	LogLoss	PCOC	ECE
G1	Pointwise	0.7620	0.3683	**1.0163**	**0.0042**	0.8723	0.2223	1.019	0.0217
G2	RankNet [5]	0.7639	0.5939	2.8256	0.2784	0.8749	0.2210	0.9632	0.0233
	ListNet [7]	0.7646	1.1393	4.4817	0.5309	0.8733	0.4931	1.7427	0.3899
G3	Combined-Pair [28]	0.7644	0.3670	1.0330	0.0068	0.8740	0.2209	1.025	0.0145
	Combined-List	0.7643	0.3694	1.0513	0.0078	0.8729	0.2232	0.9947	0.0178
Ours	JRC	**0.7649***	**0.3667**	1.0276*	0.0065*	**0.8765***	**0.2189***	**0.9953**	**0.0120***

user behavior sequence information, for the Avazu dataset, we instead employ the **DeepFM** [21] structure, which has also shown to achieve an on-par performance as CAN [48]. For all datasets, each method is trained with one epoch, following [45]. For the production dataset, all of our experimental models are trained with the XDL platform [44].

For JRC, we adopt the Adam optimizer with the initial learning rate of 0.001 for Avazu and 0.004 for Taobao. For hyper-parameters, we experiment with the weight ratio $(1-\alpha)/\alpha$ rather than a direct tuning of α. The weight ratio is set to 0.01 for Avazu and 1.0 for Taobao. Again, due to the lack of user information and the sparsity of the same-user data samples in Taobao and Avazu, we cannot offer a meaningful, fine-grained definition for the context X_z. Therefore, we set X_z as sample of the same min-batch for the public datasets. Whereas for the production dataset, we will experiment with the influence of different definitions of the context.

4.2 Performance on Public Datasets

4.2.1 Overall Performance. Table 3 provides the model performance for all the compared methods. Overall, we observe an improved ranking ability by adopting the pairwise or listwise model in both datasets. However, this causes a significant deterioration of calibration-related metrics, mainly ascribed to the fact that ranking loss-based optimization only focuses on the relative order of samples and pays no attention to the absolute value of the prediction. The prediction no longer corresponds to the click probability, thus yielding unsatisfying calibration results. The Combined-Pair and Combined-List approaches provide a rescue that brings back the calibration ability while maintaining the ranking performance at the same level. Note that this result is also consistent with prior studies [3, 28, 34, 39].

For the JRC model, we observe a further improvement of ranking and calibration performance over the Combined-Pair and Combined-List methods, and most of the metric improvement is statistically significant. The result validates the effectiveness of our unified hybrid framework. Compared with the combined methods, JRC offers a consolidated view of the pointwise loss and listwise-like generative loss. It also effectively utilize the huge amount of non-click data. By contrast, the combined method imposes irrelevant assumptions on the same logit, limiting the optimization and leading to sub-optimal solutions.

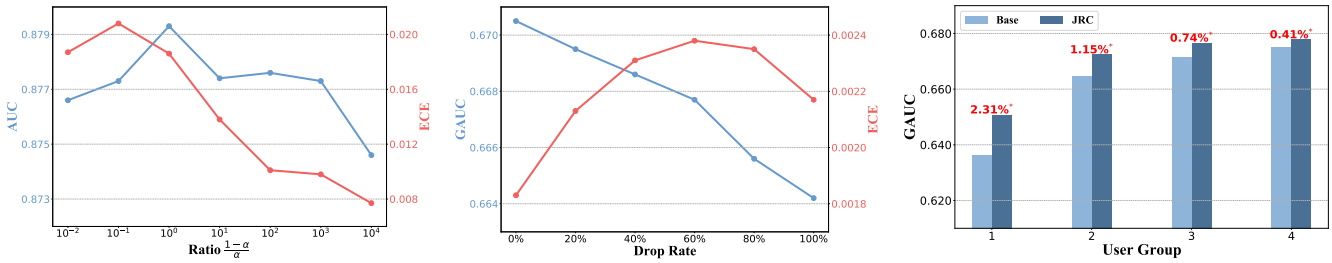
Table 4: A comparison of model performance for different methods on the production dataset. The best results are highlighted in bold.

Dataset		Production			
		GAUC	LogLoss	PCOC	ECE
G1	Pointwise	0.6642	0.3005	1.1156	0.0021
G2	RankNet (session) [5]	0.6710	2.8118	25.016	0.4517
	ListNet (session) [7]	**0.6714**	3.1603	27.163	0.4884
G3	Combined-Pair (session) [28]	0.6673	0.3004	1.1279	0.0024
Ours	JRC (session)	0.6705	**0.3004**	**1.0975**	**0.0018**

4.2.2 Impact of hyper-parameter. We investigate how the ranking and calibration abilities change with various weighting ratios between the discriminative loss and the listwise generative loss of JRC. The result is illustrated in Figure 2(a). For simplicity, we only experiment with the Taobao dataset. To be specific, we first re-scale the two loss values to the same level and then vary $\frac{1-\alpha}{\alpha}$ in $[10^{-2},10^{-1},...,10^{4}]$ with all of the other hyper-parameters fixed. We find that an extremely large or small value of $\frac{1-\alpha}{\alpha}$ would hurt the ranking ability as one loss largely dominates over the other, and JRC degenerates to a pointwise model or a listwise-like model. A weighting ratio in between help achieve the best performance, which again validates the importance of hybrid modeling. We also observe that ECE generally decreases with the increase of $\frac{1-\alpha}{\alpha}$, indicating that a more emphasis on the generative loss can yield improved calibration performance.

4.3 Performance on Production Dataset

4.3.1 Overall Performance. In the production dataset, we compare JRC with the pointwise model (the one deployed in the production system), RankNet, ListNet, and one representative combined method - Combined-Pair. Instead of AUC, we use Group AUC (GAUC) to measure the goodness of intra-user ranking quality, which is more relevant to online performance in our production system. Table 4 offers the comparison of model performance. Note that for RankNet, ListNet, and Combined-Pair, we choose samples in the same *session* to construct the pairs/list. We define a *session*



(a) (b) (c)

Figure 2: (a) Performance of JRC w.r.t. different ratios $\frac{1-\alpha}{\alpha}$ on the Taobao dataset. (b) The influence of context size for the proposed JRC method. (c) Performances of Baseline (pointwise) and JRC model on different user groups. Here, we compute the relative performance improvement over the baseline, $*$ denotes the improvement is statistically significant.

Table 5: A comparison of different context for the JRC model on the production dataset. The best results are highlighted in bold.

Method	GAUC	LogLoss	PCOC	ECE
Pointwise	0.6642	0.3005	1.1156	0.0021
JRC (batch)	0.6654	0.3006	1.1575	0.0029
JRC (gender)	0.6651	0.3005	1.1293	0.0024
JRC (domain)	0.6661	0.3005	1.1344	0.0025
JRC (session)	**0.6705**	**0.3004**	**1.0975**	**0.0018**

as all of the ads a user receives within a specific time window (e.g., ten minutes).

Like the public datasets, RankNet and ListNet outperform the Pointwise model for ranking ability, but at the expense of hurting well-calibrated probabilities. The negative impact of RankNet and ListNet on calibration makes them improper for CTR prediction. The Combined-Pair method improves the ranking ability while maintaining the comparable calibration performance to the pointwise model. The result again demonstrates the effectiveness of combining the pointwise and pairwise losses. JRC (session) improves ranking and calibration metrics over the Combined-Pair method. A detailed analysis of the context definition for JRC is provided in the following.

4.3.2 Defining Context. We show that the definition of contexts, i.e., the context list to compute the listwise generative loss, is essential to the final performance of JRC. We consider four different context definitions – batch, gender (of the user), domain (i.e., different spots an ad can be placed on), and session for the JRC method. We first experiment with the batch context, in which samples from the same mini-batch are treated as being in the same context. In terms of other contexts, we treat data samples from the same mini-batch and within the same user gender/domain/session as the same context.

Table 5 offers a detailed comparison. JRC (batch) does not improve much compared with the pointwise approach, demonstrating that a too broad context might introduce irrelevant samples. By defining the context as user gender, we do not see much performance improvement, implying that users of the same gender may not share similar interests. Besides, JRC (domain) achieves better performance than JRC (batch), suggesting the application domain is a meaningful context for consideration. A finer-grained context definition with JRC (session) achieves the best performance, illustrating that samples within the same user are more proper to represent a user's current interest and used as the context.

4.3.3 Influence of the Context Size. We also study the effect of context size, i.e., number of samples per context, on the listwise generative loss. Here we conduct experiments on JRC (session). The average number of samples in each session is 6.8 on the production dataset. For each session, we randomly drop {20%, 40%, 60%, 80%, 100%} of samples, and compute the generative loss to train the model. Note that the random drop only affects the listwise generative loss, whereas the pointwise loss keeps intact.

The result is shown in Figure 2(b). We observe that the best ranking and calibration metrics are obtained when the drop rate is 0%. Meanwhile, the ranking ability gradually decreases when we raise the drop ratio, suggesting that larger context size help improve the ranking ability. The finding is consistent with previous work [28] that more effective pairs help induce a better ranker for the pairwise loss.

A particularly interesting observation is that the calibration ability gradually decreases and then rises with the increase of the drop rate. We hypothesis that this is because when the drop rate is small, the noise during the computation of generative loss determines the calibration ability. However, when the drop ratio increases further, the generative loss becomes smaller compared with the discriminative loss, which means the discriminative loss becomes dominant. The dominant discriminative loss makes the calibration better and becomes similar to the pointwise model gradually.

4.3.4 Performance on Different User Groups. To further understand the behavior of the proposed JRC model, we split users into four groups based on their activity levels (which is measured by the numbers of clicks in the past 14 days). Users in Group 1 have the least historical behaviors, and users in Group 4 have the most behaviors. It is worth noting that users are grouped in this way to ensure that different user groups contain the same amount of data

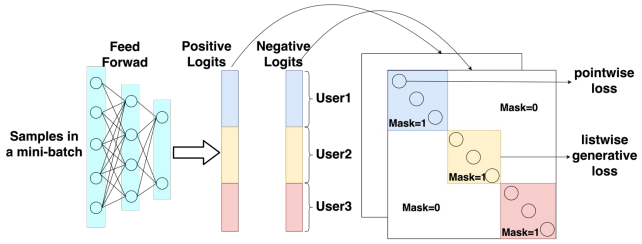


Figure 3: Samples from the same user are grouped within the time window of 10 minutes and fed to the model to calculate the logits. These logits are used to compute the pointwise loss and listwise generative loss.

samples. Afterward, we compute the relative GAUC improvement (GAUC_{JRC}-GAUC_{Base})/GAUC_{Base} over the pointwise model for each group.

As shown in Figure 2(c), JRC consistently outperforms the pointwise model over all four user groups, demonstrating the effectiveness of our model over different types of users. Meanwhile, we see more improvement for users with few behaviors, whereas there is less improvement for users with more behaviors. This result is consistent with previous studies on generative modeling, which often yields better performance when the training data is limited [32, 38]. In our case, samples for inactive users are more sparse. The generative part of JRC adds more training signals for users with few click behaviors, as it includes an objective that contrasts the positive samples with negative samples. Note that we can also connect this finding to a previous similar study conducted in Twitter Timeline [28] observing that the combination of pointwise loss and ranking loss helps alleviate the sparsity of training signal in CTR prediction.

4.4 Online Deployment

4.4.1 Online Training. One major challenge for industrial recommendation systems is dealing with the constant shifting of data distribution. To keep up with data change, industrial systems need to update the model continuously [8, 19, 26]. Due to the large amount of data, industrial CTR models are usually trained in a distributed manner. In this manner, samples of the same context might be sent to different workers as the training data comes as a stream. Note that the modification of the data organization will not affect the pointwise model but may greatly influence the proposed JRC method.

To support the **context-aware online training**, we redesign the training data organization so that samples of the same context are grouped into one mini-batch. Specifically, we first set a time window, and samples in the same context and occurring in the same time window will be placed into one mini-batch. Note that there is a trade-off – an extended time window increases the context size, while the data delay is severe, and the resources to cache samples also increase dramatically. In practice, **samples of the same user are grouped within the time window of 10 minutes**, the illustration of the loss computation during online training is shown in Figure 3.

Table 6: The performance lift of the proposed JRC model compared with the production baseline.

Metric	CTR	RPM	LogLoss
Lift	+4.4%	+2.4%	-0.27%

In addition, to serve the JRC model training, the context-sensitive organization of training samples also helps reduce data redundancy. Features of the same users will only be stored once and not repeatedly multiple times in different samples. Such an organization can compress our training data to a great extent, largely relieving the burden of system storage and computation.

4.4.2 Online Performance. The proposed JRC model is also deployed online for A/B testing in the display advertising system of Alibaba. Our experiment buckets are determined by randomly hashing the unique identifier of a user. The production baseline is a pointwise model. Compared with the pointwise model, JRC utilizes the same set of features and MLP structures, thus adding no additional computation cost for online serving. We illustrate the performance lift of JRC in Table 6. Through a five-day of online experiments (from 2022/05/21 - 2022/05/25), JRC obtains a significant performance gain over the production baseline, with a +4.4% increase in CTR, a +2.4% increase in RPM (Revenue Per Mille), and a -0.27% drop of LogLoss.

5 CONCLUSION AND DISCUSSION

Ranking ability and calibration ability are two essential aspects of CTR prediction. Pointwise models yield calibrated outputs but fall short of ranking ability. The pairwise and listwise models lift the ranking performance but hurt the calibration ability. A direct combination of the two models balances ranking and calibration. However, it breaks the meaning of output as the click probability, leading to suboptimal solutions.

In contrast to the above models, we propose the JRC method, which employs two logits corresponding to click and non-click states and jointly optimizes the ranking and calibration abilities. We further showed that JRC unified the logit interpretation as the energy-based model of the joint distribution. On top of that, conditional probabilities for discriminative and listwise generative losses can be derived. Our experiments on both public datasets, production datasets, and online A/B testing proved the effectiveness of JRC. Moreover, the proposed model further improves the ranking and calibration performance compared to a linear combination of multiple losses.

REFERENCES

[1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval.* Vol. 463. ACM press New York.
[2] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, Xinchen Luo, Shiming Xiang, Guorui Zhou, Xiaoqiang Zhu, and Hongbo Deng. 2022. CAN: Feature Co-Action Network for Click-Through Rate Prediction. In *The 15th ACM International Conference on Web Search and Data Mining*. 57–65.
[3] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. 2020. A stochastic treatment of learning to rank scoring functions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 61–69.

[4] Sebastian Bruch, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2019. An Analysis of the Softmax Cross Entropy Loss for Learning-to-Rank with Binary Relevance. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 75–78.

[5] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[6] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, Vol. 119. 89–96.

[7] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the Twenty-Fourth International Conference*, Vol. 227. 129–136.

[8] Zhangming Chan, Yu Zhang, Shuguang Han, Yong Bai, Xiang-Rong Sheng, Siyuan Lou, Jiacen Hu, Baolin Liu, Yuning Jiang, Jian Xu, and Bo Zheng. 2023. Capturing Conversion Rate Fluctuation during Sales Promotions: A Novel Historical Data Reuse Approach. *arXiv preprint arXiv:2305.12837* (2023).

[9] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (Eds.). 2006. *Semi-Supervised Learning*. The MIT Press.

[10] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[11] Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. 2021. Not All Relevance Scores are Equal: Efficient Uncertainty and Calibration Modeling for Deep Retrieval Models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. 654–664.

[12] Martín Abadi et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[13] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognit. Lett.* 27, 8 (2006), 861–874.

[14] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.

[15] Fredric C Gey. 1994. Inferring probability of relevance using the method of logistic regression. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. 222–231.

[16] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on Machine Learning*. Haifa, Israel., 13–20.

[17] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 2020. Your classifier is secretly an energy based model and you should treat it like one. In *Proceedings of the 8th International Conference on Learning Representations*.

[18] Bin Gu, Victor S. Sheng, KengYeow Tay, Walter Romano, and Shuo Li. 2015. Incremental Support Vector Learning for Ordinal Regression. *IEEE Transactions on Neural networks and learning systems* 26, 7 (2015), 1403–1416.

[19] Siyu Gu, Xiang-Rong Sheng, Ying Fan, Guorui Zhou, and Xiaoqiang Zhu. 2021. Real Negatives Matter: Continuous Training with Real Negatives for Delayed Feedback Modeling. In *Proceedings of The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2890–2898.

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 1321–1330.

[21] 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*. Melbourne, Australia., 2782–2788.

[22] Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

[23] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the 8th International Workshop on Data Mining for Online Advertising*. ACM, 5:1–5:9.

[24] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[25] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems 27*. 3581–3589.

[26] Sofia Ira Ktena, Alykhan Tejani, Lucas Theis, Pranay Kumar Myana, Deepak Dilipkumar, Ferenc Huszár, Steven Yoo, and Wenzhe Shi. 2019. Addressing Delayed Feedback for Continuous Training with Neural Networks in CTR Prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 187–195.

[27] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

[28] Cheng Li, Yue Lu, Qiaozhu Mei, Dong Wang, and Sandeep Pandey. 2015. Click-through prediction for advertising in twitter timeline. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1959–1968.

[29] Hao Liu and Pieter Abbeel. 2020. Hybrid discriminative-generative training via contrastive learning. *arXiv preprint arXiv:2007.09070* (2020).

[30] Tie-Yan Liu. 2011. Learning to rank for information retrieval. (2011).

[31] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the Calibration of Modern Neural Networks. In *Advances in Neural Information Processing Systems 34*. 15682–15694.

[32] Andrew Y. Ng and Michael I. Jordan. 2001. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*. 841–848.

[33] Feiyang Pan, Xiang Ao, Pingzhong Tang, Min Lu, Dapeng Liu, Lei Xiao, and Qing He. 2020. Field-aware Calibration: A Simple and Empirically Strong Method for Reliable Probabilistic Predictions. In *Proceedings of The Web Conference 2020*. 729–739.

[34] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. 2019. Tf-ranking: Scalable tensorflow library for learning-to-rank. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2970–2978.

[35] Gustavo Penha and Claudia Hauff. 2021. On the Calibration and Uncertainty of Neural Learning to Rank Models for Conversational Search. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. 160–170.

[36] Tao Qin, Tie-Yan Liu, and Hang Li. 2010. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval* 13, 4 (2010), 375–397.

[37] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *Proceedings of the16th International Conference on Data Mining*. IEEE, 1149–1154.

[38] Rajat Raina, Yirong Shen, Andrew Y. Ng, and Andrew McCallum. 2003. Classification with Hybrid Generative/Discriminative Models. In *Advances in Neural Information Processing Systems 16*. 545–552.

[39] David Sculley. 2010. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 979–988.

[40] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, and Xiaoqiang Zhu. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *Proceedings of The 30th ACM International Conference on Information and Knowledge Management*. 4104–4113.

[41] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, Vol. 307. 1192–1199.

[42] Le Yan, Zhen Qin, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2022. Scale Calibration of Deep Ranking Models. (2022).

[43] Yujing Zhang, Zhangming Chan, Shuhao Xu, Weijie Bian, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. KEEP: An Industrial Pre-Training Framework for Online Recommendation via Knowledge Extraction and Plugging. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3684–3693.

[44] Yuanxing Zhang, Langshi Chen, Siran Yang, Man Yuan, Huimin Yi, et al. 2022. PICASSO: Unleashing the Potential of GPU-centric Training for Wide-and-deep Recommender Systems. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE.

[45] Zhao-Yu Zhang, Xiang-Rong Sheng, Yujing Zhang, Biye Jiang, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. Towards Understanding the Overfitting Phenomenon of Deep Click-Through Rate Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2671–2680.

[46] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1059–1068.

[47] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized Cost per Click in Taobao Display Advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2191–2200.

[48] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open Benchmarking for Click-Through Rate Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2759–2769.