Adaptive Structure Learning with Partial Parameter Sharing for Post-Click Conversion Rate Prediction

Chunyuan Zheng
Peking University
Beijing, China
cyzheng@stu.pku.edu.cn

Hang Pan
University of Science and Technology of China
Hefei, China
hungpaan@mail.ustc.edu.cn

Yang Zhang
National University of Singapore
Singapore, Singapore
zyang1580@gmail.com

Haoxuan Li*
Peking University
Beijing, China
hxli@stu.pku.edu.cn

Abstract

The post-click conversion rate (CVR) prediction task aims to predict the probability of a conversion after a click, which is essential in many fields. There are two widely-recognized challenges for CVR prediction: selection bias and data sparsity. Many previous methods focus on addressing selection bias by unbiasedly estimating the ideal loss based on the doubly robust estimator, which incorporates the error imputation model and propensity model to help CVR prediction model learning. However, they struggle with unreasonable knowledge transfer between the prediction model and imputation model and inflexible network structure design under sparse data. To this end, we introduce a novel principled adaptive structure learning approach, named Adap-SL, to adaptively learn the optimal network structure, adjust the number of activated (non-zero) parameters, and determine which knowledge needs to be transferred between the prediction model and the imputation model. Specifically, we start with an over-parameterized base network, where we adaptively extract partially overlapped subnetworks for the imputation model and the prediction model. Extensive experiments are conducted on three real-world recommendation datasets, demonstrating that our method consistently improves performance while requiring fewer parameters. The code is available at https://github.com/ChunyuanZheng/sigir25-sparse-sharing.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Recommender systems, Adaptive structure learning, Debiased post-click conversion rate prediction

*Haoxuan Li is the corresponding author.

SIGIR '25, Padua, Italy.

ACM ISBN 979-8-4007-1592-1/25/07
https://doi.org/10.1145/3726302.3729887

ACM Reference Format:
Chunyuan Zheng, Hang Pan, Yang Zhang, and Haoxuan Li. 2025. Adaptive Structure Learning with Partial Parameter Sharing for Post-Click Conversion Rate Prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729887

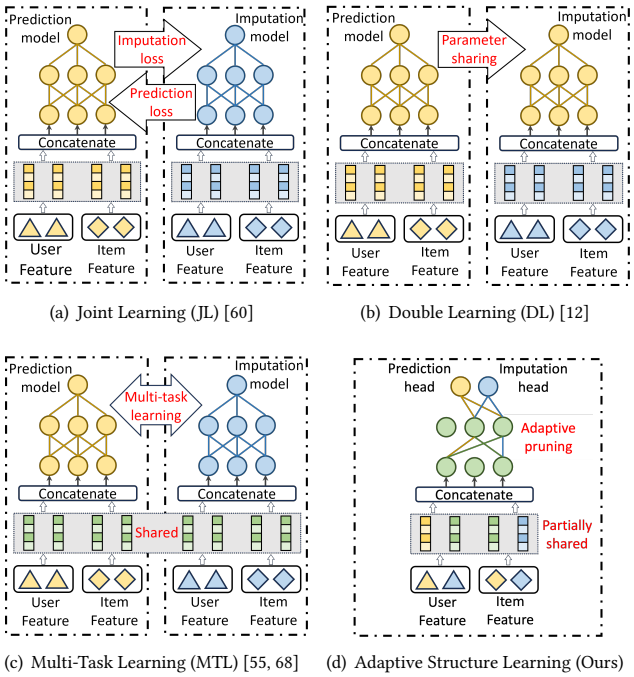
1 Introduction

To develop a recommender system for e-commerce [31, 56, 67, 69], the post-click conversion rate (CVR) prediction is one of the essential tasks, which aims to predict the probability of conversion after a click. There are two widely-recognized challenges:

• **Selection Bias:** Due to the self-selection behavior of users and the item selection process in the system, the collected conversion distribution is not a representative for all users and items [3, 33, 64]. Taking the movie recommendation system as an example, users might be more likely to see movies they think they will like, and then make ratings for the movies they have seen [38, 48]. This would create a systematic bias towards observing ratings with higher values, which poses a critical challenge for unbiased evaluation and learning of the prediction models [21, 22, 48, 58].

• **Data Sparsity:** We have no chance to collect the conversion for unclicked data, and the proportion of clicked data is always small in RS datasets, such as 2% for Yahoo! R3 [38], 4% for Movielens-1M [60], and 8% for Coat [48] datasets. Thus, the number of training samples may not be sufficient for the large parameter space [55, 68].

Causality-based techniques provide a promising direction for addressing these issues [43, 52, 53, 57]. Many previous methods focus on addressing selection bias by unbiasedly estimating the ideal loss, which is defined as the average prediction error across all user-item pairs [20, 26, 44, 51]. Among them, the most popular one is the doubly robust (DR) based methods [24, 46, 60], which incorporates the error imputation model and propensity model to help CVR prediction model learning, and results in unbiased prediction when either the imputation model or the propensity model is accurate for all users and items. In DR-based methods, the imputed errors are defined upon the prediction model, and the prediction model is trained by minimizing the DR loss affected by the imputation model, resulting in a correlation between the prediction and imputation models. In addition, DR-based methods contain multiple models that are difficult to be sufficiently learned under sparse data, which may lead to inappropriate model fitting. Thus, developing a learning



(a) Joint Learning (JL) [60] (b) Double Learning (DL) [12]

(c) Multi-Task Learning (MTL) [55, 68] (d) Adaptive Structure Learning (Ours)

Figure 1: Illustration of the learning paradigms between the prediction model and the imputation model in DR learning.

algorithm that achieves both parameter efficiency and appropriate knowledge transfer between the prediction and imputation models is important for accurate CVR prediction.

Many algorithms have been proposed, such as joint learning (JL) [60] and double learning (DL) [12]. Specifically, as shown in Figure 1(a), JL alternatively updates the prediction model and the imputation model with separate losses, embedding tables, and network structures to minimize both prediction and imputation inaccuracies. Based on JL, DL further regularly copies the prediction model parameters to the imputation model for enhancing knowledge transfer, as shown in Figure 1(b). However, these methods may still suffer from over-parametrization. In addition, the knowledge transfer is unreasonable due to hard copy parameters. There are also many multi-task learning-based methods, such as Multi-DR [68], ESCM^2-DR [55], and DCMT [71], as shown in Figure 1(c). They share the embedding tables between the prediction and the imputation model and learn two models simultaneously. Nevertheless, without data information, the pre-specified sharing mechanism cannot effectively capture the prediction model and imputation correlation, especially under sparse data.

To this end, inspired by the lottery ticket hypothesis [9], we introduce a novel principled adaptive structure learning approach, named Adap-SL (as shown in Figure 1(d)), which adaptively learns the optimal network structure, adjusts the number of activated (non-zero) parameters, and determines which knowledge needs to be transferred between prediction and imputation models. Without prior knowledge, we start with an over-parameterized base network, from which we adaptively extract partially overlapped subnetworks for prediction and imputation models. In particular, the overlapped part can efficiently transfer knowledge between the two models, while the non-overlapped part accounts for the task-specific knowledge. The contributions are summarized below.

- We reveal the problem that previous DR-based methods struggle with unreasonable knowledge transfer between the prediction model and imputation model and inflexible network structure design under sparse data.
- We propose an adaptive structure learning approach to adaptively learn the optimal network structure, adjust the number of activated parameters, and determine which knowledge needs to be transferred between the prediction model and imputation model by combining adaptive pruning and partial sharing mechanisms together in model training.
- Extensive experiments conducted on three real-world datasets demonstrate the effectiveness of our method.

2 Related Work

2.1 Debiased Recommendation

Selection bias is one of the most common bias in CVR prediction task [45, 59, 64, 65]. To address the bias, many debiasing methods are proposed [14, 19, 29, 63]. For example, error imputation-based (EIB) methods are proposed to estimate the prediction error for missing events [1, 16, 49], while inverse propensity score (IPS) methods re-weight the observed prediction error based on the inverse probability of being observed [17, 34, 47, 48]. Doubly robust (DR) methods combine both error imputation and inverse propensity re-weighting models, offering unbiased estimation when either model is accurate [54, 60]. Moreover, Li et al. [23] validate that balancing property is important for propensity learning, and Li et al. [25] further explore which function should be balanced. With a few randomized controlled trial (RCT) data available in the training stage, prior works propose to perform model selection through bi-level optimization to further debias [2, 28, 61]. However, previous methods struggle with unreasonable knowledge transfer between the prediction model and imputation model, and inflexible network structure design under sparse data. Our approach proposes a novel adaptive structure learning approach to address the above concerns.

2.2 Lottery Ticket Hypothesis

The lottery ticket hypothesis (LTH) is one of the most influential hypotheses in the field of neural network pruning [5, 9, 27, 32]. For a network, LTH iteratively removes a certain percentage of parameters based on their size. After pruning, the remaining parameters are retrained from scratch using their original initialization to achieve the same performance as the original network. Recently, some studies [8, 30, 37, 41] improve the theoretical foundation of LTH. Ma et al. [35] provide a more rigorous definition of LTH for precisely identifying winning tickets, while Zhang et al. [66] offer a formal proof for the improved generalization of winning tickets observed in LTH experiments. Diffenderfer and Kailkhura [7] introduce the Multi-Prize Tickets (MPT) algorithm to find MPT in binary neural networks. Other research [10, 40] extends LTH to a broader range of application tasks. For example, Mehta [39] proposes the lottery ticket transfer hypothesis in the image classification domain and transfers winning tickets between different image classification datasets. Prasanna et al. [42] explore the existence of winning tickets in fine-tuned language models and identify sub-networks

that match the model's performance. Chen et al. [4] extend LTH to Graph Neural Networks in node classification and link prediction tasks. In our paper, we extend the sharing mechanism in LTH by partially sharing the parameter between the prediction model and imputation model to maintain both the common and specific knowledge of each model.

2.3 Post-Click Conversion Rate Prediction

In industrial applications, the click-through rate (CTR) and post-click conversion rate (CVR) prediction are regarded as the two most fundamental tasks. Unlike debiased recommendations, the challenge in the CVR prediction task lies not only in addressing selection bias in the data but also in overcoming the data sparsity challenges in model training. To address both selection bias and data sparsity, Ma et al. [36] propose to model the product of CTR and CVR, referred to as CTCVR, to estimate CVR in the entire space, thereby mitigating the selection bias issue. Additionally, a parameter-sharing strategy is introduced to tackle the data sparsity challenge. Building on this, Zhang et al. [68] further propose an unbiased estimator and introduces multi-task learning to achieve efficient learning on industrial large-scale datasets. On the other hand, Wang et al. [55] consider incorporating IPS or DR loss, which serves as an unbiased estimator for CVR loss, into the ESMM objective function to further reduce bias. Moreover, Guo et al. [12] propose to reduce the variance of the DR estimator in the CTCVR task and introduce double learning for imputation model training. Dai et al. [6] consider the trade-off between the bias and variance of the DR estimator, aiming for lower generalization error, and Zhou et al. [70] propose using additional regularization to constrain the learning of the propensity model in IPS and DR estimators. Zhu et al. [71] propose to predict both factual and counterfactual CVR under the soft constraint of a counterfactual prior knowledge. However, without data information, the pre-specified sharing mechanism of previous methods cannot effectively and efficiently capture the prediction model and imputation correlation, especially under sparse data. We learn a sharing mechanism and network structure that adaptively addresses these issues in this paper.

3 Preliminaries

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ be the set of m users, $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ be the set of n items, and $\mathcal{D} = \mathcal{U} \times \mathcal{I}$ be the set of all user-item pairs. Denote $\mathbf{R} \in \{0,1\}^{m \times n}$ as the post-click conversion label matrix of user-item pairs, where $r_{u,i} = 1$ indicates that user u converted on item i after click and $r_{u,i} = 0$ otherwise. Denote $x_{u,i}$ as the feature of user-item pair (u, i), $\hat{\mathbf{R}} \in \mathbb{R}^{m \times n}$ as the prediction matrix for CVR, where $\hat{r}_{u,i} = f(x_{u,i}; \theta) \in [0, 1]$ is the predicted CVR by a prediction model parameterized by θ. If $\mathbf{R}$ is fully observed, the CVR prediction model $f(x_{u,i}; \theta)$ can be trained by minimizing the ideal loss:

$$\mathcal{L}_{ideal}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} e_{u,i}, \tag{1}$$

where $e_{u,i}$ is the prediction error, such as the cross entropy loss $e_{u,i} = CE(r_{u,i}, \hat{r}_{u,i}) = -r_{u,i} \log \hat{r}_{u,i} - \left(1 - r_{u,i}\right) \log \left(1 - \hat{r}_{u,i}\right)$. In real recommendation scenarios, some users do not click on some items, resulting in the challenge of unbiased estimation of the ideal loss. Let $o_{u,i}$ be the indicator of user u clicking on item i, then $r_{u,i}$ with $o_{u,i} = 0$ are not directly observable. Therefore, directly optimizing the ideal loss is not feasible. The Naive estimator optimizes the prediction model by minimizing the average prediction error corresponding to the click event:

$$\mathcal{L}_{Naive}(\theta) = \frac{1}{|\mathcal{O}|} \sum_{(u,i) \in \mathcal{O}} e_{u,i} = \frac{1}{|\mathcal{O}|} \sum_{(u,i) \in \mathcal{D}} o_{u,i} e_{u,i}, \tag{2}$$

where $\mathcal{O} = \{(u, i) \mid (u, i) \in \mathcal{D}, o_{u,i} = 1\}$ is the set of clicked user-item pairs. The Naive estimator is unbiased when the click event is missing at random (MAR). However, the presence of selection bias makes the data missing not at random (MNAR) and the clicked events are no longer representative of all events.

To address this problem, many debiased recommendation methods have been proposed. The EIB method directly fits the prediction error $e_{u,i}$ corresponding to unclicked events. The estimator is:

$$\mathcal{L}_{EIB}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} o_{u,i} e_{u,i} + (1 - o_{u,i}) \hat{e}_{u,i}, \tag{3}$$

where $\hat{e}_{u,i} = m(x_{u,i}; \phi)$ is the imputed prediction error given by an imputation model. The EIB estimator is unbiased when imputed errors are accurate, i.e., $\hat{e}_{u,i} = e_{u,i}$.

The IPS method reweights the clicked samples by $1/p_{u,i}$, where $p_{u,i} = \Pr(o_{u,i} = 1 | x_{u,i})$ denotes probability of a user u clicking on an item i, i.e., the propensity or click-through rate (CTR) in CVR prediction. The IPS estimator is given as:

$$\mathcal{L}_{IPS}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \frac{o_{u,i} e_{u,i}}{\hat{p}_{u,i}}, \tag{4}$$

where $\hat{p}_{u,i} = \pi(x_{u,i}; \psi)$ is the learned propensity given by a CTR prediction model. The IPS estimator is unbiased when the propensities of all user-item pairs are accurately estimated, i.e., $\hat{p}_{u,i} = p_{u,i}$.

The doubly robust (DR) estimator and its variants have demonstrated superior performance in debiasing, which is given by:

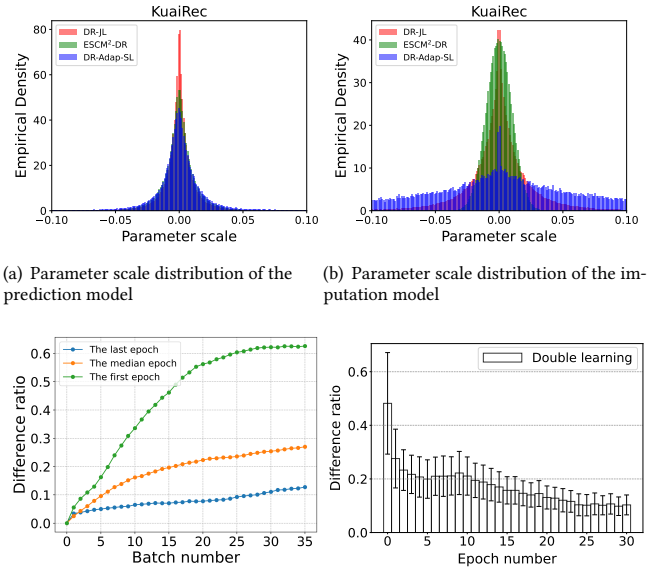
$$\mathcal{L}_{DR}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \left[\hat{e}_{u,i} + \frac{o_{u,i}(e_{u,i} - \hat{e}_{u,i})}{\hat{p}_{u,i}} \right], \tag{5}$$

which is an unbiased estimate of the ideal loss when either the imputed errors $\hat{e}_{u,i} = m(x_{u,i}; \phi)$ or the learned propensities $\hat{p}_{u,i} = \pi(x_{u,i}; \psi)$ are correctly estimated, i.e., $\hat{e}_{u,i} = e_{u,i}$ or $\hat{p}_{u,i} = p_{u,i}$.

4 Proposed Methods

4.1 Motivation

We first conduct the experiment to illustrate the insufficient knowledge transfer and unreasonable parameter-sharing mechanisms of previous methods. The JL methods adopt separate embedding tables and neural network structures for the prediction and imputation. Figures 2(a) and 2(b) show the histograms of the parameter weights for the prediction model and the imputation model, respectively, after the convergence of the training process. We find that many parameters in the prediction model and imputation model are around 0, which reveals that these two models might suffer over-parametrization, especially given the close relationship between the error imputation task and the unbiased prediction task. In addition, we found a similar phenomenon in the MTL-based method, such as ESCM^2-DR. Note that ESCM^2-DR performs better than DR-JL, because ESCM^2-DR shares the embedding table between



(a) Parameter scale distribution of the prediction model

(b) Parameter scale distribution of the imputation model

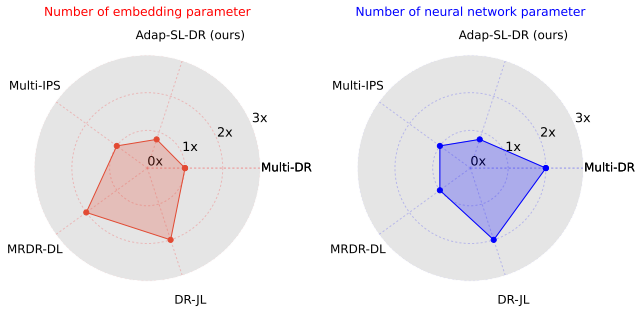
(c) Parameter difference ratio of DL within an epoch

(d) Parameter difference ratio of DL before parameter sharing during training

Figure 2: Over-parametrization of previous learning approach (a-b) and unreasonable sharing mechanism (c-d).

two models. Based on JL, DL further regularly copies the prediction model parameters to the imputation model for enhancing knowledge transfer, but raises another concern. From Figure 2(c) and Figure 2(d), the direct copying of all neural network weights results in significant differences ($|\theta-\phi|/|\phi| > 1$) in neural network weights of the imputation model before and after periodically copying all weights from the prediction model, resulting in greater imputation loss. Though the difference ratio decreases during the model training process, there are still around 10% of weights significantly different. Therefore, it is beneficial to develop a learning approach for more effective knowledge transfer between the prediction and imputation models, and more reasonable sharing mechanism and network structure design with data information.

Specifically, we propose an adaptive structure learning approach to adaptively learn the optimal network structure and determine which knowledge needs to be transferred for debiasing, which consists of two components: partial sharing and adaptive pruning. The overview of the proposed method is shown in Figure 4. The motivation behind partial sharing is to partially share parameters to enhance flexibility of knowledge transfer between the imputation and prediction models. Specifically, the imputation and prediction models partition their parameters into specific and common parts: the common part is shared with the other model, and the specific part remains as the model's private parameter. Adaptive pruning aims to answer the question of "what parameters need to be shared" and to achieve parameter efficiency by pruning parameters in each model to prevent overfitting under sparse data while maintaining performance. In addition, Figure 3 demonstrates the advantage of our method in terms of the parameter number, which helps to prevent overfitting under sparse data. Before further elaborating these two components, we first provide the theoretical guarantees for our adaptive structure learning approach.



(a) Number of embedding parameter

(b) Number of neural network parameter

Figure 3: The parameter numbers of different methods.

4.2 Theoretical Guarantee

Suppose there is a network G with width n and depth l and a masked-subnetwork $\tilde{G}$ of G with the same width and depth. Denote the weight of i-th layer in G and $\tilde{G}$ as $W_{G(i)}$ and $W_{\tilde{G}(i)}$, respectively. $W_{\tilde{G}(i)} = W_{G(i)} \odot M_i$, where $M_i \in \{0,1\}^{n_{in} \times n_{out}}$ is a binary mask matrix, where n_{in} and n_{out} are the width of input layer and output layer. The main theoretical result in this section is to show that for every target network of depth l with bounded weights, a random network of depth $2l$ and polynomial width with high probability contains a subnetwork that approximates the target network. We formally state the results in the following theorem.

Theorem 1 (Existence of Sparse Subnetwork Structure [37]). *Fix some $\epsilon, \delta \in (0,1)$. Let F be some target network of depth l such that for every $i \in [l]$, we have $\|W_{F(i)}\|_2 \leq 1, \|W_{F(i)}\|_{\max} \leq \frac{1}{\sqrt{n_{in}}}$ (where $n_{in} = d$ for $i = 1$ and $n_{in} = n$ for $i > 1$). Let G be a network of width $poly(d, n, l, \frac{1}{\epsilon}, \log\frac{1}{\delta})$ and depth $2l$, where $W_{G(i)}$ is from $U([-1,1])$. Then, with probability at least $1-\delta$ there exists a weight-subnetwork $\tilde{G}$ of G such that:*

$$\sup_{x \in \mathcal{X}} |\tilde{G}(x) - F(x)| \leq \epsilon. \tag{6}$$

Moreover, the number of active (non-zero) weights in $\tilde{G}$ is $O(dn+n^2l)$.

This theorem guarantees that in an over-parameterized network, there exists a subnetwork structure that can approximate the ground-truth network structure with the same order of non-zero parameters compared to the ground-truth network, which ensures the existence of a sparse converged network.

4.3 Partial Sharing of Model Parameters

Partial sharing aims to facilitate the transformation and sharing of information between models while maintaining model flexibility. Formally, the parameters θ of the prediction model f_θ and the parameters ϕ of the imputation model g_ϕ are divided into θ^{self} and θ^{share}, and ϕ^{self} and ϕ^{share}, respectively. The shared parameters θ^{share} and ϕ^{share} are kept equal during optimization, while θ^{self} and ϕ^{self} are optimized independently during training. During the training process, both θ and ϕ are pruned to reduce the scale of the prediction model and the imputation model, and we only share the parameters that remained in both the prediction and imputation models. We will discuss this in detail below.

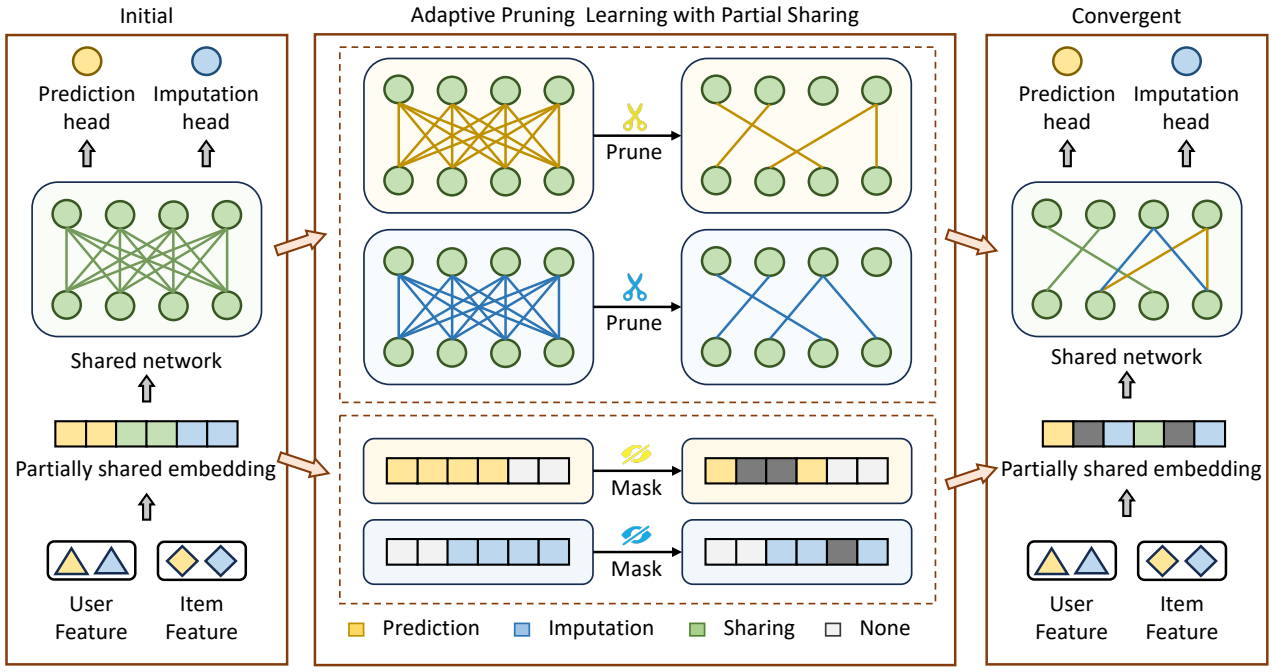


Figure 4: The overview of our end-to-end adaptive structure learning framework is as follows: (1) The shared network is initially fully connected, with adaptive pruning applied to both the prediction and imputation networks during model training. (2) For partially shared embeddings, common and task-specific embeddings are then separately masked for each model after removing irrelevant embeddings. The final result is a convergent network structure.

4.3.1 Partially Shared Embedding Layers. Given a user-item pair (u, i), the collaborative filtering model will first obtain a user embedding $p_u \in \mathbb{R}^k$ and an item embedding $q_i \in \mathbb{R}^k$ to the pair where k is the dimension of the embedding and p_u and q_i are learnable parameters. The imputation model g_ϕ assigns $p_{u,\phi}$ and $q_{i,\phi}$ and the prediction model f_θ assigns $p_{u,\theta}$ and $q_{i,\theta}$ respectively. To share the embedding, without loss of generality, we define $p_u^{share} \in \mathbb{R}^{k/2}$ and $q_i^{share} \in \mathbb{R}^{k/2}$ as the sharing embedding for the user-item pair (u, i) and $p_{u,\theta}^{self} \in \mathbb{R}^{k/2}$, $q_{i,\theta}^{self} \in \mathbb{R}^{k/2}$, $p_{u,\phi}^{self} \in \mathbb{R}^{k/2}$ and $q_{i,\phi}^{self} \in \mathbb{R}^{k/2}$ as the private embedding for the corresponding model respectively for the user-item pair (u, i), which means we share half parameters for each model. Then, the partially shared embedding for the imputation model g_ϕ is defined as:

$$p_{u,\phi}^{PS} = \begin{bmatrix} p_u^{share} \\ p_{u,\phi}^{self} \end{bmatrix} \text{ and } q_{i,\phi}^{PS} = \begin{bmatrix} q_i^{share} \\ q_{i,\phi}^{self} \end{bmatrix}, \tag{7}$$

and the embedding for the prediction model f_θ is defined as:

$$p_{u,\theta}^{PS} = \begin{bmatrix} p_u^{share} \\ p_{u,\theta}^{self} \end{bmatrix} \text{ and } q_{i,\theta}^{PS} = \begin{bmatrix} q_i^{share} \\ q_{i,\theta}^{self} \end{bmatrix}. \tag{8}$$

Then, we can construct the shared embedding matrix by considering a user embedding matrix $P \in \mathbb{R}^{|\mathcal{U}|\times k}$ contains the embedding all the users and an item embedding matrix $Q \in \mathbb{R}^{|\mathcal{I}|\times k}$ contains the embedding all the items. Thus, we define the private embedding matrix $P_\phi^{self} \in \mathbb{R}^{|\mathcal{U}|\times k/2}$ and $Q_\phi^{self} \in \mathbb{R}^{|\mathcal{I}|\times k/2}$ for the imputation model as:

$$P_\phi^{self} = [p_{u_1,\phi}^{self}, \ldots, p_{u_m,\phi}^{self}]^T \text{ and } Q_\phi^{self} = [q_{i_1,\phi}^{self}, \ldots, q_{i_n,\phi}^{self}]^T, \tag{9}$$

and $P_\theta^{self} \in \mathbb{R}^{|\mathcal{U}|\times k/2}$ and $Q_\theta^{self} \in \mathbb{R}^{|\mathcal{I}|\times k/2}$ for the prediction model as:

$$P_\theta^{self} = [p_{u_1,\theta}^{self}, \ldots, p_{u_m,\theta}^{self}]^T \text{ and } Q_\theta^{self} = [q_{i_1,\theta}^{self}, \ldots, q_{i_n,\theta}^{self}]^T. \tag{10}$$

Then, we define the partially shared user embedding matrix $P^{share} \in \mathbb{R}^{|\mathcal{U}|\times k/2}$ and item embedding matrix $Q^{share} \in \mathbb{R}^{|\mathcal{I}|\times k/2}$ as:

$$P^{share} = [p_{u_1}^{share}, \ldots, p_{u_m}^{share}]^T \text{ and } Q^{share} = [q_{i_1}^{share}, \ldots, q_{i_n}^{share}]^T. \tag{11}$$

Last, we construct the final user embedding matrix $P_\phi^{PS} \in \mathbb{R}^{|\mathcal{U}|\times k}$ and item embedding matrix $Q_\phi^{PS} \in \mathbb{R}^{|\mathcal{I}|\times k}$ for the imputation model by stacking the corresponding partially shared embedding matrix and the private embedding matrix as:

$$P_\phi^{PS} = \left[P^{share}, P_\phi^{self}\right] \text{ and } Q_\phi^{PS} = \left[Q^{share}, Q_\phi^{self}\right]. \tag{12}$$

Similarly, we construct the final user embedding matrix $P_\theta^{PS} \in \mathbb{R}^{|\mathcal{U}|\times k}$ and item embedding matrix $Q_\theta^{PS} \in \mathbb{R}^{|\mathcal{I}|\times k}$ for the prediction model as:

$$P_\theta^{PS} = \left[P^{share}, P_\theta^{self}\right] \text{ and } Q_\theta^{PS} = \left[Q^{share}, Q_\theta^{self}\right]. \tag{13}$$

4.3.2 Sharing Linear Layers. Given the user embedding p_u and the item embedding q_i, the collaborative filtering model first concatenates p_u and q_i and then passes the concatenated vector through

MLP layers, where each MLP contains several linear layers followed by the activation layer. To implement our proposed Adap-SL method, we construct a partially shared MLP by replacing the linear layers in the MLP with partially shared linear layers. Formally, we define the partially shared linear layer of our Adap-SL method. Given the impute dimension k_{in} and the output demension k_{out}, we first define the shared linear layer $LN^{share}(\cdot)$ as:

$$LN^{share}(x) = W^T_{share}x + b_{share}, \tag{14}$$

where $x \in \mathbb{R}^{k_{in}}$ is the input vector, $W_{share} \in \mathbb{R}^{k_{in} \times k_{out}/2}$ is the weight matrix and $b_{share} \in \mathbb{R}^{k_{out}/2}$ is the bias. Similarly, we define the private linear layer $LN^{self}(\cdot)$ as:

$$LN^{self}(x) = W^T_{self}x + b_{self}, \tag{15}$$

where $x \in \mathbb{R}^{k_{in}}$ is the input vector, $W_{self} \in \mathbb{R}^{k_{in} \times k_{out}/2}$ is the weight matrix and $b_{self} \in \mathbb{R}^{k_{out}/2}$ is the bias.

Then we define the partially shared linear layer for the imputation model g_ϕ as:

$$LN^{PS}_\phi(x) = concat(LN^{share}(x), LN^{self}_\theta(x)), \tag{16}$$

and the partially shared linear layer for the prediction model f_θ as:

$$LN^{PS}_\theta(x) = concat(LN^{share}(x), LN^{self}_\theta(x)). \tag{17}$$

Thus, the imputation model and the prediction model share the parameters in $LN^{share}(\cdot)$ while keeping the parameters in $LN^{self}_\phi(\cdot)$ and $LN^{self}_\theta(\cdot)$ independent.

Then we combine the constructed partially shared embedding and the partially shared MLP. For a user-item pair (u, i), the imputation model with L layers of partially shared MLP is as below:

$$z_1 = concat(p^{PS}_{u,\phi}, q^{PS}_{i,\phi}), z_2 = a_2(LN^{PS}_{2,\phi}(z_1)), \ldots, \tag{18}$$

$$z_L = a_L(LN^{PS}_{L,\phi}(z_{L-1})), \hat{r}_{u,i} = \sigma(h^T z_L), \tag{19}$$

where a_l is the activation layer of the l-th MLP. The prediction model shares the same structure as the imputation model.

4.4 Adaptive Sparse Pruning of Model Parameters

In this section, we prune the imputation model and the prediction model to overcome the data sparsity problem and to decide which knowledge needs to be shared. Inspired by [50], we introduce a hard mask $M_\phi \in \{0,1\}^{|\phi|}$ for g_ϕ and a mask $M_\theta \in \{0,1\}^{|\theta|}$ for f_θ. Instead of the entire network g_ϕ and f_θ, sub-networks $g_{\phi \odot M_\phi}$ and $f_{\theta \odot M_\theta}$ are used when making the imputation or the prediction, where $\odot$ denotes element-wise product. Specifically, taking the imputation model as an example, if the j-th element $M[j] = 1$, then the corresponding parameter $\phi[j]$ is active when making the imputation. On the other hand, if the j-th element $M[j] = 0$, then the corresponding parameter $\phi[j]$ is inactive. The pruning rule is to prune the node with a minimum value of the weighted normalized node weight and gradient. As the pruning process goes on, the masks M_ϕ and M_θ contain more 0, and the imputation and prediction models become sparse. Note that the mask will be nested during the training phase. We also try the soft masks $M^S_\phi \in [0,1]^{|\phi|}$ and $M^S_\theta \in [0,1]^{|\theta|}$. However, this kind of mask cannot guarantee the nested model and cannot tackle the data sparsity issue because the number of learnable parameters is not reduced, leading to a sub-optimal performance.

Algorithm 1: Adaptive Structure Learning with Partial Sharing

Input: Pre-trained propensity model π_ψ; Imputation model g_ϕ; Prediction model f_θ; Pruning rate $\alpha_\phi, \alpha_\theta$; Minimal sparsity S_ϕ, S_θ; Warm up epoch E_ϕ, E_θ; Datasets $\mathcal{D}$.

```
Randomly initialize φ, θ;
Initialize mask M_φ = 1^{|φ|}, M_θ = 1^{|θ|};
Epoch ← 0;
while not convergent do
    Train g_{φ⊙M_φ}, f_{θ⊙M_θ} simultaneously for k steps using Eq 21 and Eq 22;
    if Epoch ≥ E_φ and ||M_φ||_0 / |φ| > S_φ then
        Prune α_φ percent of the remaining parameters with the lowest magnitudes from φ. That is, let M_φ[j] = 0 if φ[j] is pruned;
    end
    if Epoch ≥ E_θ and ||M_θ||_0 / |θ| > S_θ then
        Prune α_θ percent of the remaining parameters with the lowest magnitudes from θ. That is, let M_θ[j] = 0 if θ[j] is pruned;
    end
    Epoch ← Epoch + 1;
end
```

Output: $g_\phi, f_\theta, M_\phi, M_\theta$.

Different from [50], where masks were trained and fixed at first and then the backbone model was trained, we combine the pruning and sharing procedure in an end-to-end manner. The insights are because the two tasks (predict CVR and impute error) are related, but also have some intrinsic differences. Specifically, we first define the sparse rate for each model as $\frac{||M_\phi||_0}{|\phi|}$ for g_ϕ and $\frac{||M_\theta||_0}{|\theta|}$ for f_θ where $||M||_0$ denotes the number of 0 in M. In each epoch, we first train the models $g_{\phi \odot M_\phi}$ and $f_{\theta \odot M_\theta}$ simultaneously. Then we prune each model with a prune rate α if its sparse rate does not meet the predefined sparsity criteria S_ϕ and S_θ. That is, we let $M[j] = 0$ for all j such that the j-th parameter is among the α percent of the remaining parameters with the lowest magnitudes. We also adopt the warm-up techniques. That is, we first train E_ϕ epochs for the imputation model and E_θ epochs for the prediction model before we start pruning. In doing so, we can make the initialized parameters meaningful enough before the pruning.

Finally, for pre-train propensity model $\hat{p}_{u,i} = \pi(x_{u,i}; \psi)$, we use the cross-entropy loss as below:

$$\mathcal{L}_p(\psi) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \left[-o_{u,i} \log(\hat{p}_{u,i}) - (1 - o_{u,i}) \log(1 - \hat{p}_{u,i}) \right]. \tag{20}$$

Table 1: Performance on AUC, Recall@K, and NDCG@K on the unbiased test set of Coat, Yahoo! R3, and KuaiRec. The best results are bolded and the best baseline is underlined. * means statistical significance with p-value ≤ 0.05 under pairwise-t test.

	Coat			Yahoo! R3			KuaiRec		
Method	AUC	R@5	N@5	AUC	R@5	N@5	AUC	R@50	N@50
NCF	$0.757_{\pm 0.005}$	$0.476_{\pm 0.006}$	$0.686_{\pm 0.008}$	$0.684_{\pm 0.001}$	$0.450_{\pm 0.002}$	$0.684_{\pm 0.003}$	$0.835_{\pm 0.001}$	$0.705_{\pm 0.002}$	$0.643_{\pm 0.002}$
IPS	$0.769_{\pm 0.004}$	$0.473_{\pm 0.007}$	$0.686_{\pm 0.007}$	$0.688_{\pm 0.003}$	$0.450_{\pm 0.004}$	$0.671_{\pm 0.003}$	$0.838_{\pm 0.002}$	$0.715_{\pm 0.003}$	$0.643_{\pm 0.003}$
SNIPS	$0.771_{\pm 0.003}$	$0.475_{\pm 0.003}$	$0.684_{\pm 0.005}$	$0.688_{\pm 0.001}$	$0.452_{\pm 0.002}$	$0.679_{\pm 0.002}$	$0.837_{\pm 0.001}$	$0.712_{\pm 0.001}$	$0.647_{\pm 0.001}$
AS-IPS	$0.763_{\pm 0.004}$	$0.477_{\pm 0.004}$	$0.688_{\pm 0.005}$	$0.687_{\pm 0.003}$	$0.454_{\pm 0.003}$	$0.677_{\pm 0.005}$	$0.834_{\pm 0.002}$	$0.712_{\pm 0.004}$	$0.642_{\pm 0.002}$
IPS-V2	$0.770_{\pm 0.004}$	$0.476_{\pm 0.004}$	$0.684_{\pm 0.005}$	$0.692_{\pm 0.004}$	$0.455_{\pm 0.003}$	$0.675_{\pm 0.005}$	$0.838_{\pm 0.003}$	$0.714_{\pm 0.002}$	$0.652_{\pm 0.002}$
Multi-IPS	$0.761_{\pm 0.003}$	$0.482_{\pm 0.005}$	$0.682_{\pm 0.005}$	$0.691_{\pm 0.004}$	$0.452_{\pm 0.006}$	$0.678_{\pm 0.005}$	$0.839_{\pm 0.003}$	$0.716_{\pm 0.002}$	$0.648_{\pm 0.002}$
ESCM2-IPS	$0.768_{\pm 0.004}$	$0.483_{\pm 0.006}$	$0.682_{\pm 0.009}$	$0.692_{\pm 0.005}$	$0.453_{\pm 0.005}$	$0.683_{\pm 0.006}$	$0.841_{\pm 0.002}$	$0.716_{\pm 0.002}$	$0.651_{\pm 0.002}$
DR-JL	$0.767_{\pm 0.002}$	$0.477_{\pm 0.004}$	$0.682_{\pm 0.004}$	$0.696_{\pm 0.003}$	$0.453_{\pm 0.004}$	$0.678_{\pm 0.004}$	$0.840_{\pm 0.001}$	$0.714_{\pm 0.003}$	$0.651_{\pm 0.003}$
MRDR-DL	$0.769_{\pm 0.003}$	$0.479_{\pm 0.005}$	$0.676_{\pm 0.009}$	$0.695_{\pm 0.002}$	$0.453_{\pm 0.003}$	$0.679_{\pm 0.004}$	$0.838_{\pm 0.002}$	$0.713_{\pm 0.002}$	$0.651_{\pm 0.003}$
DR-BIAS	$0.772_{\pm 0.003}$	$0.477_{\pm 0.005}$	$0.680_{\pm 0.005}$	$0.697_{\pm 0.002}$	$0.454_{\pm 0.003}$	$0.680_{\pm 0.003}$	$0.838_{\pm 0.002}$	$0.713_{\pm 0.004}$	$0.653_{\pm 0.004}$
DR-MSE	$0.773_{\pm 0.002}$	$0.481_{\pm 0.003}$	$0.689_{\pm 0.006}$	$0.697_{\pm 0.003}$	$0.453_{\pm 0.002}$	$0.683_{\pm 0.002}$	$0.841_{\pm 0.002}$	$0.716_{\pm 0.003}$	$0.653_{\pm 0.003}$
TDR-JL	$0.772_{\pm 0.003}$	$0.483_{\pm 0.004}$	$0.689_{\pm 0.007}$	$0.696_{\pm 0.005}$	$0.454_{\pm 0.004}$	$0.684_{\pm 0.006}$	$0.840_{\pm 0.002}$	$0.715_{\pm 0.004}$	$0.656_{\pm 0.003}$
DR-V2	$0.770_{\pm 0.004}$	$0.487_{\pm 0.005}$	$0.691_{\pm 0.004}$	$0.690_{\pm 0.003}$	$0.451_{\pm 0.003}$	$0.682_{\pm 0.005}$	$0.840_{\pm 0.003}$	$0.717_{\pm 0.005}$	$0.652_{\pm 0.003}$
Multi-DR	$0.770_{\pm 0.004}$	$0.482_{\pm 0.005}$	$0.691_{\pm 0.006}$	$0.691_{\pm 0.002}$	$0.452_{\pm 0.004}$	$0.682_{\pm 0.005}$	$0.838_{\pm 0.003}$	$0.715_{\pm 0.003}$	$0.654_{\pm 0.004}$
ESCM^2-DR	$0.772_{\pm 0.003}$	$0.485_{\pm 0.003}$	$0.690_{\pm 0.009}$	$0.695_{\pm 0.004}$	$0.456_{\pm 0.002}$	$0.687_{\pm 0.003}$	$0.842_{\pm 0.002}$	$0.723_{\pm 0.002}$	$0.653_{\pm 0.002}$
KBDR	$0.772_{\pm 0.002}$	$0.486_{\pm 0.003}$	$0.693_{\pm 0.006}$	$0.692_{\pm 0.003}$	$0.453_{\pm 0.004}$	$0.683_{\pm 0.003}$	$0.841_{\pm 0.002}$	$0.720_{\pm 0.003}$	$0.655_{\pm 0.002}$
DCE-DR	$0.770_{\pm 0.003}$	$0.489_{\pm 0.003}$	$0.693_{\pm 0.006}$	$0.693_{\pm 0.004}$	$0.456_{\pm 0.004}$	$0.687_{\pm 0.002}$	$0.841_{\pm 0.001}$	$0.719_{\pm 0.003}$	$0.654_{\pm 0.002}$
DCMT	$0.771_{\pm 0.002}$	$0.490_{\pm 0.002}$	$0.697_{\pm 0.005}$	$0.698_{\pm 0.004}$	$0.457_{\pm 0.004}$	$0.685_{\pm 0.002}$	$0.841_{\pm 0.002}$	$\mathbf{0.724}_{\pm 0.002}$	$0.656_{\pm 0.003}$
Adap-SL-DR	$\mathbf{0.777}^*_{\pm 0.003}$	$\mathbf{0.493}^*_{\pm 0.002}$	$\mathbf{0.717}^*_{\pm 0.005}$	$\mathbf{0.703}^*_{\pm 0.003}$	$\mathbf{0.461}^*_{\pm 0.002}$	$\mathbf{0.690}_{\pm 0.003}$	$\mathbf{0.845}^*_{\pm 0.001}$	$0.721_{\pm 0.002}$	$\mathbf{0.661}^*_{\pm 0.003}$

For training the CVR prediction model $f_{\theta \odot M_\theta}$, we use the DR loss, which is shown below:

$$\mathcal{L}_{DR}(\theta \odot M_\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} \left[\hat{e}_{u,i} + \frac{o_{u,i}(e_{u,i} - \hat{e}_{u,i})}{\hat{p}_{u,i}} \right]. \tag{21}$$

For training the error imputation model $g_{\phi \odot M_\phi}$, we use the mean square error between the true prediction error and imputed error on observed user-item pairs:

$$\mathcal{L}_e(\phi \odot M_\phi) = \frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} \frac{o_{u,i}(\hat{e}_{u,i} - e_{u,i})^2}{\hat{p}_{u,i}}. \tag{22}$$

The above-mentioned training process is illustrated in Algorithm 1.

5 Experiment

We conduct extensive experiments on three datasets: **Coat** [48], **Yahoo! R3** [38], and an industrial dataset **KuaiRec** [11]. These datasets contain both biased data and unbiased data. Specifically, **Coat** dataset contains 6,960 biased ratings and 4,640 unbiased ratings from 290 users to 300 items, and each user self-selected 24 items to rate and randomly rates 16 items. **Yahoo! R3** dataset contains 311,704 biased ratings and 54,000 unbiased ratings from 15,400 users to 1,000 items, and **KuaiRec** dataset contains 4,676,570 video watching ratio from 1,411 users to 3,327 items. Following [20, 23, 48, 60], we binarize the ratings for **Coat** and **Yahoo! R3** dataset by denoting ratings less than three as negative conversion with label 0, and otherwise as positive conversion with label 1. For **KuaiRec**, we follow the data preprocessing in [18], and we binarize the watch ratio by setting values less than two to 0 and otherwise to 1.

5.1 Baselines and Experiment Details

5.1.1 Baselines. We compare our method with the following debiasing methods, including: (1) IPS-based methods: **IPS** [48] and **SNIPS** [48], **AS-IPS** [45], **IPS-V2** [23], **Multi-IPS** [68], and **ESCM2-IPS** [55]; (2) DR-based methods: **DR-JL** [60], **MRDR** [12], **DR-BIAS** [6], **DR-MSE** [6], **TDR-CL** [20], **DR-V2** [23], **Multi-DR** [68], **ESCM^2-DR** [55], **KBDR** [25], **DCE-DR** [18], and **DCMT** [71].

5.1.2 Experiment details. The backbone model for learning the prediction model is selected as neural collaborative filtering (NCF) [15], a highly adopted model in debiased recommendation. NCF uses a neural embedding learning approach to understand the feature embedding of users and items, modeling the user-item interaction as a combinative function with these embeddings. We use Adam as the optimizer for both the imputation and prediction models. All experiments are run on Pytorch with NVIDIA GeForce RTX 3090 as the computational resource. We tune the learning rate in $\{0.001, 0.005, 0.01, 0.05\}$, weight decay in $\{1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3\}$, the embedding size of NCF in $\{4, 8, 16, 32\}$ for **Coat** dataset and $\{16, 32, 64, 128\}$ for **Yahoo** and **KuaiRec** datasets, the pruning rate $\alpha_\phi, \alpha_\theta$ in $\{0.05, 0.1, 0.15, 0.2, 0.25\}$, the marginal sparsity threshold S_ϕ, S_θ in $\{0.1, 0.3, 0.5, 0.7\}$, and the warm-up epochs in $\{2, 4, 6, 8, 10\}$. We set the batch size to 128 for **Coat** and 2048 for both **Yahoo! R3** and **KuaiRec**, and fixed the depth of the neural network as 3. In addition, we use the logistic regression model for propensity learning, thus there are no methods requiring unbiased data for further training.

5.2 Evaluation Metrics

Following the previous studies [6, 48, 62], we adopted three metrics for ranking or prediction tasks for evaluations: AUC, Recall@K

Table 2: Ablation study on our Adap-SL method on three real-world datasets. The best results are bolded.

	Coat			Yahoo! R3			KuaiRec		
Method	AUC	R@5	N@5	AUC	R@5	N@5	AUC	R@50	N@50
Adap-SL-DR	**0.777**$_{\pm 0.003}$	0.493$_{\pm 0.002}$	**0.717**$_{\pm 0.005}$	**0.703**$_{\pm 0.003}$	**0.461**$_{\pm 0.002}$	**0.692**$_{\pm 0.003}$	**0.845**$_{\pm 0.001}$	0.721$_{\pm 0.002}$	**0.661**$_{\pm 0.003}$
w/o share	0.772$_{\pm 0.003}$	0.491$_{\pm 0.003}$	0.712$_{\pm 0.007}$	0.700$_{\pm 0.003}$	0.453$_{\pm 0.002}$	0.684$_{\pm 0.002}$	0.843$_{\pm 0.001}$	0.716$_{\pm 0.003}$	0.655$_{\pm 0.003}$
w/o prune	0.772$_{\pm 0.003}$	0.486$_{\pm 0.005}$	0.702$_{\pm 0.008}$	0.699$_{\pm 0.004}$	0.458$_{\pm 0.004}$	0.686$_{\pm 0.002}$	0.841$_{\pm 0.001}$	**0.724**$_{\pm 0.003}$	0.653$_{\pm 0.004}$
w/o share w/o prune	0.767$_{\pm 0.002}$	0.477$_{\pm 0.004}$	0.682$_{\pm 0.004}$	0.696$_{\pm 0.003}$	0.453$_{\pm 0.004}$	0.678$_{\pm 0.004}$	0.840$_{\pm 0.001}$	0.714$_{\pm 0.003}$	0.651$_{\pm 0.003}$
w/o prune LN	0.773$_{\pm 0.003}$	**0.494**$_{\pm 0.003}$	0.715$_{\pm 0.006}$	**0.703**$_{\pm 0.003}$	0.458$_{\pm 0.005}$	0.687$_{\pm 0.004}$	**0.845**$_{\pm 0.002}$	0.718$_{\pm 0.004}$	0.658$_{\pm 0.010}$
Share propensity	0.773$_{\pm 0.003}$	0.490$_{\pm 0.004}$	0.715$_{\pm 0.006}$	0.702$_{\pm 0.004}$	0.459$_{\pm 0.004}$	0.689$_{\pm 0.003}$	0.844$_{\pm 0.002}$	0.722$_{\pm 0.003}$	0.658$_{\pm 0.007}$

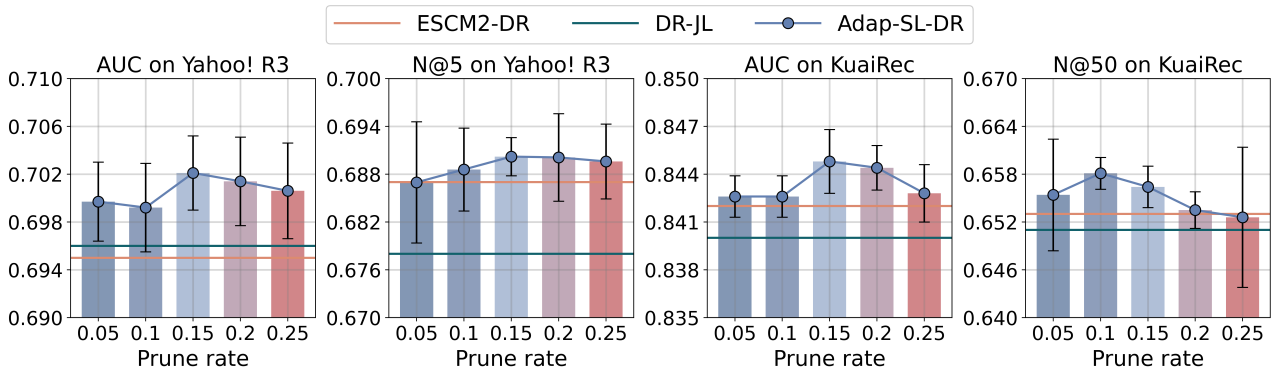


Figure 5: Performance compared to baseline with varying pruning rate on the prediction model.

(R@K), and NDCG@K (N@K). The AUC evaluates the CVR prediction accuracy and the NDCG@K and Recall@K evaluate the recommendation quality for each user. The NDCG@K can be formulated as below:

$$\mathrm{DCG}_u@\mathrm{K} = \sum_{i \in D^u_{\text{test}}} \frac{I\left(\hat{z}_{u,i} \leq K\right)}{\log\left(\hat{z}_{u,i}+1\right)}, \tag{23}$$

$$\mathrm{NDCG@K} = \frac{1}{|U|} \sum_{u \in U} \frac{\mathrm{DCG@K}}{\mathrm{IDCG@K}}, \tag{24}$$

where IDCG represents the best possible DCG, and $\hat{z}_{u,i}$ denotes the ranking of the item i among all items in the test set for the user i. With more accurate recommendations, $DCG_u@K$ would increase to be closer to $IDCG@K$, leading to the NDCG@K closer to 1. The Recall@K can be formulated as

$$\mathrm{Recall}_u@\mathrm{K} = \frac{\sum_{i \in D^u_{\text{test}}} I\left(\hat{z}_{u,i} \leq k\right)}{\min\left(K, \left|D^u_{\text{test}}\right|\right)}, \tag{25}$$

$$\mathrm{Recall@K} = \frac{1}{|U|} \sum_{u \in U} \mathrm{Recall_u@K}. \tag{26}$$

Following previous studies [13, 18, 23], because there are only 16/10 items in the unbiased data of **Coat** and **Yahoo! R3** datasets for each user, we set $K = 5$ for **Coat** and **Yahoo! R3**. In addition, for a larger dataset **KuaiRec**, we set $K = 50$.

5.3 Performance Comparison

We show the prediction accuracy for all baseline methods and our proposed method in Table 1. First, DR-based methods outperformed the IPS-based methods and the naive methods without IPS or DR. This validates the ability of the DR method in debiasing. Second, among the DR-based baselines, ESCM^2-DR performs the best for all three metrics across all datasets, showcasing its great prediction strength by using entire-space multi-task learning techniques. It demonstrates that multi-task learning is useful for improving prediction accuracy. Third, our method Adap-SL shows comparative performance with the SOTA baseline models and obtained improvements at AUC for all three datasets, which can be attributed to Adap-SL is able to effectively and efficiently capture the prediction model and imputation correlation.

5.4 In-Depth Analyses

5.4.1 Effects of pruning and sharing. We further explore the effect of pruning or sharing in our method on debiasing performance. We show the prediction accuracy of Adap-SL-DR w/o pruning, w/o sharing parameters between the prediction and imputation models in Table 2. In addition, we implement a modified method that shares the parameters of the prediction model with the propensity score model in Adap-SL-DR. We find that lacking either sharing or pruning would harm the prediction accuracy. Meanwhile, this harm is additive, leading Adap-SL w/o sharing and pruning to perform even worse than Adap-SL-DR without either sharing or pruning. Compared with the original Adap-SL-DR, Adap-SL-DR without pruning linear layers would lead to significant performance reduction at AUC on **Coat** and R@5, N@5 on both **Yahoo! R3** and **KuaiRec** datasets. It suggests that sharing linear layers is beneficial. After sharing the parameters with the propensity scores model, the method underperforms the original Adap-SL-DR. It might be attributed to the propensity is related to the exposure mechanism while the prediction/imputation is not, thus, sharing parameters between these two is irrational.

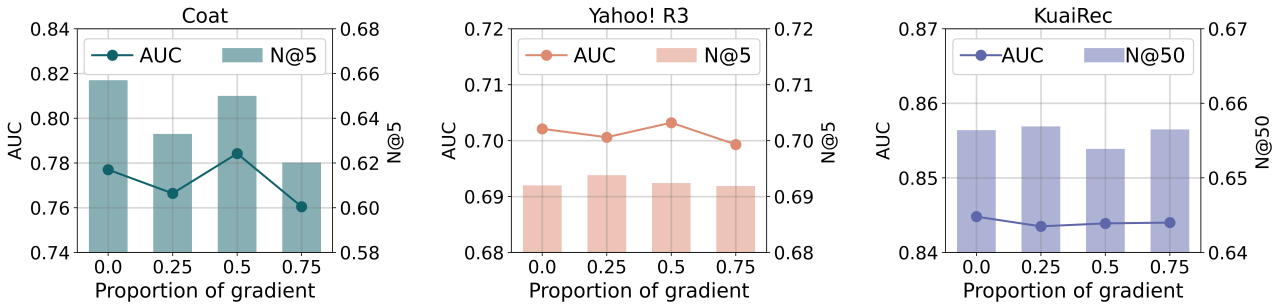


Figure 6: Performance compared to baseline with varying proportion of gradient included in pruning.

Table 3: Performance under different pruning mechanisms on the unbiased test set of KuaiRec.

	KuaiRec		
Method	AUC	R@50	N@50
Adap-SL-DR	**0.845**$_{\pm 0.001}$	0.721$_{\pm 0.002}$	**0.661**$_{\pm 0.003}$
w/o prune pred	0.841$_{\pm 0.002}$	0.722$_{\pm 0.004}$	0.657$_{\pm 0.002}$
w/o prune impu	0.843$_{\pm 0.002}$	0.719$_{\pm 0.003}$	0.658$_{\pm 0.003}$
w/o prune	0.841$_{\pm 0.001}$	**0.724**$_{\pm 0.003}$	0.653$_{\pm 0.004}$

Table 4: Percentage of remaining parameters after adaptive pruning for prediction model and imputation model.

Yahoo! R3	Total remain	Shared remain	Specific remain
Prediction	42.19%	7.97%	76.41%
Imputation	78.00%	100.00%	56.00%
KuaiRec	Total remain	Shared remain	Specific remain
Prediction	60.84%	54.32%	67.36%
Imputation	19.27%	38.01%	0.53%

5.4.2 Effects of prune rate. Besides, we explore the effect of different prune rates on debiasing performance. We show the variations of AUC and NDCG on **Yahoo! R3** and **KuaiRec** datasets with prune rates varying from 0.05 to 0.25 in Figure 5. The metrics peak at 0.1 or 0.15, suggesting a suitable pruning rate is necessary to improve the debiasing performance. Though a larger pruning rate may avoid overfitting, it also prunes some useful prediction units in the model, leading to unsatisfactory prediction accuracy.

5.4.3 Effects of pruning on only prediction and imputation model. We further evaluated the performance of Adap-SL-DR when pruning only on the prediction model or the imputation model. The model performance with **KuaiRec** is shown in Table 3. We observe that only pruning the prediction model significantly reduces the performance while only pruning the imputation model does not. It is attributed to the different roles of these two models: the prediction model plays the key role while the imputation functions as an assistant to reduce bias and variance.

5.4.4 Effects of proportions of gradient. We show the AUC and NDCG under different proportions of gradient in pruning for all three datasets in Figure 6. For the case with the proportion of gradient as 0, Adap-SL-DR only considers the weight for pruning, while for other cases, Adap-SL-DR considers the weighted average of the weight and gradient of each parameter for pruning. We find that the different proportions of gradient used for pruning has little impact on the model performance on **Yahoo! R3** and **KuaiRec** datasets. While it greatly influences the performance of Adap-SL-DR in **Coat**. It is perhaps because the batch size of **Yahoo! R3** and **KuaiRec** is much larger than that of **Coat**. The gradient in a small batch is much more influential compared with that in a large batch, thus leading to the performance of Adap-SL-DR on **Coat** being more vulnerable to different proportions of gradient.

5.4.5 Percentage of Remaining Parameters after Adaptive Pruning. We present statistics on the percentage of remaining parameters in the shared portion, specific portion, and total remaining parameters of the prediction and imputation models after adaptive pruning. As shown in Table 4, on **Yahoo! R3**, the percentage of total remaining parameters of the imputation model is much larger than that of the prediction model, while on **KuaiRec** the opposite is true. Looking into the statistics, it can be seen that the reason for this observation is the difference in the percentage of remaining parameters in the shared and exclusive portions of the two datasets. In addition, for a larger dataset **kuaiRec**, more knowledge may need to be shared.

6 Conclusion

In this paper, we introduced Adap-SL, a novel principled adaptive structure learning approach for post-click conversion rate estimation, which can adaptively learn the optimal network structure, adjust the number of activated (non-zero) parameters, and determine which knowledge needs to be transferred between prediction model and the imputation model. By leveraging adaptive pruning and a partial sharing mechanism, Adap-SL starts with an over-parameterized base network, where we adaptively extract partially overlapped subnetworks for the imputation model and the prediction model. Through extensive experimentation on three real-world recommendation datasets, Adap-SL consistently demonstrates superior performance with fewer parameters. One of the potential limitations is how to design a more reasonable algorithm to prune and to find an optimal structure instead of using the parameter scale or gradient as the pruning rule.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (623B2002).

References

[1] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. 2010. Training and Testing Low-Degree Polynomial Data Mappings via Linear SVM. *Journal of Machine Learning Research* 11, 4 (2010).

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

[3] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Transactions on Information Systems* (2023).

[4] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. 2021. A Unified Lottery Ticket Hypothesis for Graph Neural Networks. In *International Conference on Machine Learning*.

[5] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. 2024. A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, And Recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[6] Quanyu Dai, Haoxuan Li, Peng Wu, Zhenhua Dong, Xiao-Hua Zhou, Rui Zhang, Xiuqiang He, Rui Zhang, and Jie Sun. 2022. A Generalized Doubly Robust Learning Framework for Debiasing Post-Click Conversion Rate Prediction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[7] James Diffenderfer and Bhavya Kailkhura. 2021. Multi-prize Lottery Ticket Hypothesis: Finding Accurate Binary Neural Networks by Pruning a Randomly Weighted Network. In *International Conference on Learning Representations*.

[8] Utku Evci, Yani Ioannou, Cem Keskin, and Yann Dauphin. 2022. Gradient Flow in Sparse Neural Networks and How Lottery Tickets Win. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[9] Jonathan Frankle and Michael Carbin. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

[10] Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. 2022. Playing Lottery Tickets with Vision and Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[11] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *International Conference on Information and Knowledge Management*.

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

[13] Mingming Ha, Xuewen Tao, Wenfang Lin, Qiongxu Ma, Wujiang Xu, and Linxun Chen. 2024. Fine-grained dynamic framework for bias-variance joint optimization on data missing not at random. In *Advances in Neural Information Processing Systems*.

[14] Honglei Zhang Zhengnan Li Licheng Pan Haoxuan Li Hao Wang, Zhichao Chen and Mingming Gong. 2025. Debiased Recommendation via Wasserstein Causal Balancing. *ACM Transactions on Information Systems* (2025).

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *International World Wide Web Conference*.

[16] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-random Missing Data. In *International Conference on Machine Learning*.

[17] Guido W. Imbens and Donald B. Rubin. 2015. *Causal Inference For Statistics Social and Biomedical Science*. Cambridge University Press.

[18] Wonbin Kweon and Hwanjo Yu. 2024. Doubly Calibrated Estimator for Recommendation on Data Missing Not At Random. In *International World Wide Web Conference*.

[19] Haoxuan Li, Quanyu Dai, Yuru Li, Yan Lyu, Zhenhua Dong, Xiao-Hua Zhou, and Peng Wu. 2023. Multiple Robust Learning for Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[20] Haoxuan Li, Yan Lyu, Chunyuan Zheng, and Peng Wu. 2023. TDR-CL: Targeted Doubly Robust Collaborative Learning for Debiased Recommendations. In *International Conference on Learning Representations*.

[21] Haoxuan Li, Kunhan Wu, Chunyuan Zheng, Yanghao Xiao, Hao Wang, Zhi Geng, Fuli Feng, Xiangnan He, and Peng Wu. 2023. Removing Hidden Confounding in Recommendation: A Unified Multi-Task Learning Approach. In *Advances in Neural Information Processing Systems*.

[22] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, and Peng Wu. 2023. Balancing Unobserved Confounding with a Few Unbiased Ratings in Debiased Recommendations. In *International World Wide Web Conference*.

[23] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, Peng Wu, and Peng Cui. 2023. Propensity Matters: Measuring and Enhancing Balancing for Recommendation. In *International Conference on Machine Learning*.

[24] Haoxuan Li, Chunyuan Zheng, and Peng Wu. 2023. StableDR: Stabilized Doubly Robust Learning for Recommendation on Data Missing Not at Random. In *International Conference on Learning Representations*.

[25] Haoxuan Li, Chunyuan Zheng, Yanghao Xiao, Peng Wu, Zhi Geng, Xu Chen, and Peng Cui. 2024. Debiased Collaborative Filtering with Kernel-based Causal Balancing. In *International Conference on Learning Representations*.

[26] Meng Li and Haochen Sui. 2025. Causal Recommendation via Machine Unlearning with a Few Unbiased Data. In *AAAI Workshop on Artificial Intelligence with Causal Techniques*.

[27] Bohan Liu, Zijie Zhang, Peixiong He, Zhensen Wang, Yang Xiao, Ruimeng Ye, Yang Zhou, Wei-Shinn Ku, and Bo Hui. 2024. A arxivet Hypothesis. *arXiv:2403.04861* (2024).

[28] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[29] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *ACM Recommender Systems Conference*.

[30] Ning Liu, Geng Yuan, Zhengping Che, Xuan Shen, Xiaolong Ma, Qing Jin, Jian Ren, Jian Tang, Sijia Liu, and Yanzhi Wang. 2021. Lottery Ticket Preserves Weight Correlation: Is It Desirable or Not?. In *International Conference on Machine Learning*.

[31] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Jiajie Su, Yanchao Tan, and Fan Wang. 2024. User Distribution Mapping Modelling with Collaborative Filtering for Cross Domain Recommendation. In *International World Wide Web Conference*.

[32] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the Value of Network Pruning. *arXiv:1810.05270* (2018).

[33] Huishi Luo, Fuzhen Zhuang, Ruobing Xie, Hengshu Zhu, Deqing Wang, Zhulin An, and Yongjun Xu. 2024. A Survey on Causal Inference for Recommendation. *The Innovation* (2024).

[34] Jinwei Luo, Dugang Liu, Weike Pan, and Zhong Ming. 2021. Unbiased Recommendation Model Based on Improved Propensity Score Estimation. *Journal of Computer Applications* (2021).

[35] Xiaolong Ma, Geng Yuan, Xuan Shen, Tianlong Chen, Xuxi Chen, Xiaohan Chen, Ning Liu, Minghai Qin, Sijia Liu, Zhangyang Wang, et al. 2021. Sanity Checks for Lottery Tickets: Does Your Winning Ticket Really Win the Jackpot?. In *Advances in Neural Information Processing Systems*.

[36] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[37] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*.

[38] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *Conference on Uncertainty in Artificial Intelligence*.

[39] Rahul Mehta. 2019. Sparse Transfer Learning via Winning Lottery Tickets. *arXiv:1905.07785* (2019).

[40] Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. One Ticket to Win Them All: Generalizing Lottery Ticket Initializations Across Datasets and Optimizers. In *Advances in Neural Information Processing Systems*.

[41] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. 2020. Logarithmic Pruning Is All You Need. In *Advances in Neural Information Processing Systems*.

[42] Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When Bert Plays the Lottery, All Tickets Are Winning. In *Conference on Empirical Methods in Natural Language Processing*.

[43] Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70 (1983), 41–55.

[44] Yuta Saito. 2019. Unbiased Pairwise Learning from Implicit Feedback. In *NeurIPS Workshop on Causal Machine Learning*.

[45] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[46] Yuta Saito. 2020. Doubly Robust Estimator for Ranking Metrics with Post-Click Conversions. In *ACM Conference on Recommender Systems*.

[47] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-at-Random Implicit Feedback. In *International Conference on Web Search and Data Mining*.

[48] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *International Conference on Machine Learning*.

[49] Harald Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[50] Tianxiang Sun, Yunfan Shao, Xiaonan Li, Pengfei Liu, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. Learning Sparse Sharing Architectures for Multiple Tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[51] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems*.

[52] Fan Wang, Chaochao Chen, Weiming Liu, Tianhao Fan, Xinting Liao, Yanchao Tan, Lianyong Qi, and Xiaolin Zheng. 2024. CE-RCFR: Robust Counterfactual Regression for Consensus-Enabled Treatment Effect Estimation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[53] Fan Wang, Lianyong Qi, Weiming Liu, Bowen Yu, Jintao Chen, and Yanwei Xu. 2025. Inter- and Intra- Similarity Preserved Counterfactual Incentive Effect Estimation for Recommendation Systems. *ACM Transactions on Information Systems* (2025).

[54] Hao Wang. 2024. Improving Neural Network Generalization on Data-Limited Regression with Doubly-Robust Boosting. In *AAAI Conference on Artificial Intelligence*.

[55] Hao Wang, Tai-Wei Chang, Tianqiao Liu, Jianmin Huang, Zhichao Chen, Chao Yu, Ruopeng Li, and Wei Chu. 2022. ESCM^2: Entire Space Counterfactual Multi-task Model for Post-Click Conversion Rate Estimation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[56] Hao Wang, Zhichao Chen, Zhaoran Liu, Haozhe Li, Degui Yang, Xinggao Liu, and Haoxuan Li. 2024. Entire Space Counterfactual Learning for Reliable Content Recommendations. *IEEE Transactions on Information Forensics and Security* (2024).

[57] Hao Wang, Jiajun Fan, Zhichao Chen, Haoxuan Li, Weiming Liu, Tianqiao Liu, Quanyu Dai, Yichao Wang, Zhenhua Dong, and Ruiming Tang. 2023. Optimal Transport for Treatment Effect Estimation. In *Advances in Neural Information Processing Systems*.

[58] Haotian Wang, Haoxuan Li, Hao Zou, Haoang Chi, Long Lan, Wanrong Huang, and Wenjing Yang. 2025. Effective and Efficient Time-Varying Counterfactual Prediction with State-Space Models. In *International Conference on Learning Representations*.

[59] Jun Wang, Haoxuan Li, Chi Zhang, Dongxu Liang, Enyun Yu, Wenwu Ou, and Wenjia Wang. 2023. CounterCLR: Counterfactual contrastive learning with non-random missing data in recommendation. In *IEEE International Conference on Data Mining*.

[60] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. In *International Conference on Machine Learning*.

[61] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating Selection Biases in Recommender Systems with A Few Unbiased Ratings. In *International Conference on Web Search and Data Mining*.

[62] Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. 2020. Causal Inference for Recommender Systems. In *Advances in Neural Information Processing Systems*.

[63] Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan E. Kuruoglu, and Yefeng Zheng. 2020. Information Theoretic Counterfactual Learning from Missing-Not-At-Random Feedback. In *Advances in Neural Information Processing Systems*.

[64] Peng Wu, Haoxuan Li, Yuhao Deng, Wenjie Hu, Quanyu Dai, Zhenhua Dong, Jie Sun, Rui Zhang, and Xiao-Hua Zhou. 2022. On the Opportunity of Causal Learning in Recommendation Systems: Foundation, Estimation, Prediction and Challenges. In *International Joint Conferences on Artificial Intelligence*.

[65] Bo-Wen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving Ad Click Prediction by Considering Non-displayed Events. In *International Conference on Information and Knowledge Management*.

[66] Shuai Zhang, Meng Wang, Sijia Liu, Pin-Yu Chen, and Jinjun Xiong. 2021. Why Lottery Ticket Wins? A Theoretical Perspective of Sample Complexity on Sparse Neural Networks. In *Advances in Neural Information Processing Systems*.

[67] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *Comput. Surveys* (2019).

[68] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *International World Wide Web Conference*.

[69] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* (2020).

[70] Yuqing Zhou, Tianshu Feng, Mingrui Liu, and Ziwei Zhu. 2023. A Generalized Propensity Learning Framework for Unbiased Post-Click Conversion Rate Estimation. In *International Conference on Information and Knowledge Management*.

[71] Feng Zhu, Mingjie Zhong, Xinxing Yang, Longfei Li, Lu Yu, Tiehua Zhang, Jun Zhou, Chaochao Chen, Fei Wu, Guanfeng Liu, et al. 2023. DCMT: A Direct Entire-Space Causal Multi-Task Framework for Post-Click Conversion Estimation. In *International Conference on Data Engineering*.