HierRec: Scenario-Aware Hierarchical Modeling for Multi-scenario Recommendations

Jingtong Gao*
jt.g@my.cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Bo Chen*
chenbo116@huawei.com
Huawei Noah's Ark Lab
Shanghai, China

Menghui Zhu*
zhumenghui1@huawei.com
Huawei Noah's Ark Lab
Shanghai, China

Xiangyu Zhao†
xianzhao@cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Xiaopeng Li
xiaopli2-c@my.cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Yuhao Wang
yhwang25-c@my.cityu.edu.hk
City University of Hong Kong
Hong Kong, China

Yichao Wang
wangyichao5@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Huifeng Guo
huifeng.guo@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Ruiming Tang†
tangruiming@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Abstract

Click-Through Rate (CTR) prediction is a fundamental technique in recommendation and advertising systems. Recent studies have shown that implementing multi-scenario recommendations contributes to strengthening information sharing and improving overall performance. However, existing multi-scenario models only consider coarse-grained explicit scenario modeling that depends on pre-defined scenario identification from manual prior rules, which is biased and sub-optimal. To address these limitations, we propose a Scenario-Aware **Hier**archical Dynamic Network for Multi-Scenario **Rec**ommendations (HierRec), which perceives implicit patterns adaptively, and conducts explicit and implicit scenario modeling jointly. In particular, HierRec designs a basic scenario-oriented module based on the dynamic weight to capture scenario-specific representations. Then the hierarchical explicit and implicit scenario-aware modules are proposed to model hybrid-grained scenario information, where the multi-head implicit modeling design contributes to perceiving distinctive patterns from different perspectives. Our experiments on two public datasets and real-world industrial applications on a mainstream online advertising platform demonstrate that HierRec outperforms existing models significantly. The implementation code is available for reproducibility [1] [2].

*Co-first authors with equal contributions.
†Corresponding authors.
[1]https://github.com/Applied-Machine-Learning-Lab/HierRec
[2]https://github.com/mindspore-lab/models/tree/master/research/huawei-noah/HierRec

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679615

CCS Concepts

• **Information systems → Information retrieval**.

Keywords

Multi-Scenario Recommendation, Dynamic Weigh Network, CTR prediction

ACM Reference Format:
Jingtong Gao, Bo Chen, Menghui Zhu, Xiangyu Zhao, Xiaopeng Li, Yuhao Wang, Yichao Wang, Huifeng Guo, and Ruiming Tang. 2024. HierRec: Scenario-Aware Hierarchical Modeling for Multi-scenario Recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679615

1 Introduction

Click-Through Rate (CTR) prediction is a fundamental technique for online advertising and recommender systems [7, 25, 33, 36, 45, 47]. To improve the prediction accuracy and mitigate the data sparsity, the multi-scenario recommendation (a.k.a., multi-domain recommendation) is proposed by aggregating samples of similar scenarios [18, 28, 35] for training a unified model jointly. Specifically, samples from different pre-defined scenarios (e.g., different advertising slots or channels on the same platform) are explicitly distinguished by a newly introduced feature, i.e., *Scenario ID*. Therefore, by modeling the commonality and specialty of different scenarios, multi-scenario recommendation contributes to alleviating data sparsity, strengthening information sharing among different scenarios, and improving the prediction effect of all scenarios.

The core challenge of multi-scenario modeling is to portray scenario similarities and differences accurately. To achieve this, the existing multi-scenario models can be divided into two categories: Tower-based models and Dynamic Weight (DW) models, whose abstract structures are depicted in the left part of Figure 1. Tower-based models leverage a shared bottom network to model scenario-shared information, based on which several sub-towers are stacked to capture scenario-specific information [28, 35]. However, the

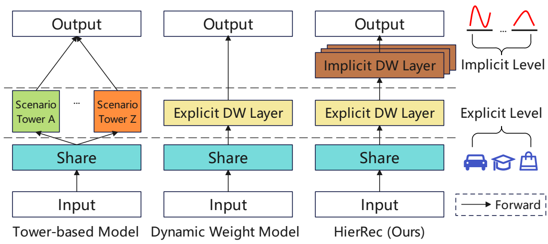


Figure 1: Comparison of different multi-scenario models.

design of complete isolation between sub-towers hinders the fine-grained modeling for scenario correlations. Besides, these methods have poor generalization and compatibility when facing a large number of scenarios. To overcome these limitations, the DW-based methods are proposed by generating dynamic parameters adaptive for each scenario in a parameter-efficient manner [44, 48], thus solving the generalization problem and facilitating the modeling of correlations between scenarios.

However, most existing multi-scenario models only consider **explicit scenario** modeling that depends on pre-defined *Scenario ID* based on manual prior rules and ignores internal data differences within each scenario, which is biased and sub-optimal [3, 35]. The initiation behind this is that the data distribution within each explicit scenario still contains many unrecognizable but important implicit patterns (e.g., diverse data distribution under feature combination "Gender" × "Age" [2, 38]), which are dynamic and difficult to pre-define manually.

Specifically, these common feature or feature combinations implicitly and meticulously divide the data into various implicit patterns where the samples in the same implicit pattern are similar in some dimensions. Distinguishing and utilizing these implicit patterns (referred to as **implicit scenarios**) for fine-grained modeling would thus greatly uncover more intricate correlations among different samples. However, existing multi-scenario models [35, 44, 48] neglect the differences in these common feature-based implicit patterns, hindering the performance. Therefore, it is crucial to excavate beneficial implicit scenarios and conduct detailed modeling for multi-scenario recommendations. To achieve this, two major challenges need to be solved: 1) A better scenario-aware modeling approach is expected to combine the human prior knowledge with the mining of implicit patterns. Therefore, the first challenge is: *How to combine explicit modeling with implicit modeling in multi-scenario recommendations?* 2) The data distribution is complex and indeterminate due to enormous features. Therefore, the second challenge is: *How to perceive implicit patterns adaptively and conduct fine-grained modeling?*

To address the challenges above, we propose a Scenario-Aware **Hier**archical Dynamic Network for Multi-Scenario **Rec**ommendations (HierRec), which is a hierarchical structure with an explicit scenario-oriented layer and multiple implicit scenario-oriented layers, shown in the right part of Figure 1. Specifically, HierRec first designs a scenario-oriented module based on the dynamic weight to capture scenario-specific representations. Based on this basic module, an explicit scenario-aware module is proposed to model coarse-grained explicit scenario information while an implicit scenario-aware module is leveraged to perceive distinctive implicit patterns and conduct fine-grained scenario modeling. Moreover, HierRec proposes a scenario-aware multi-head attention structure to identify important implicit patterns in a soft-selection manner. Subsequently, multiple implicit scenario-oriented layers are deployed parallelly to capture complicated data distributions, thus facilitating fine-grained implicit scenario modeling.

Our contributions in this paper can be summarized as follows:

- To the best of our knowledge, this is the first work that considers both explicit scenario modeling and implicit scenario modeling in multi-scenario recommendations;
- We propose a multi-scenario model HierRec based on the dynamic weight, where stacked explicit and implicit scenario-aware modules are proposed to capture explicit and implicit information, respectively. Besides, multi-head implicit modeling design contributes to perceiving complicated distribution;
- Comprehensive experiments on two public benchmark datasets and real-world applications on a mainstream online advertising platform demonstrate that HierRec outperforms existing multi-scenario recommendation models significantly.

2 Method

In this section, we first describe the problem formulation of the multi-scenario CTR prediction, and then provide an overview of HierRec and detail its key components.

2.1 Problem Formulation

Considering a training dataset $\mathcal{D} = \left\{(x_j, y_j)\right\}_{j=1}^{|\mathcal{D}|}$ with $|\mathcal{D}|$ samples, where $x_j = \{s, c_1, ...c_i, ...c_I\}$ and y_j represent the feature set and binary click label of the j_{th} sample, respectively. Feature s represents the *scenario feature* that indicates which scenario the sample comes from based on some manual prior rules explicitly. Feature c_i represents the i_{th} feature in total I *common features* $\{c_1, ...c_i, ...c_I\}$. The goal of the CTR prediction [9, 31] in the multi-scenario setting [28, 35, 44] is to learn a model $f(\cdot)$ with the provided training dataset $\mathcal{D}$ to predict the click probability $\hat{y}_j = f(x_j)$.

2.2 HierRec Overview

In this section, we present the overview architecture of HierRec with a hierarchical structure, illustrated in Figure 2. An explicit scenario-oriented layer and multiple stacked implicit scenario-oriented layers are deployed to capture explicit and implicit information, respectively. Specifically, HierRec first designs a basic **Scenario-Oriented Module** (Figure 2 (a)) based on the dynamic weight to capture scenario-specific information. Then, an **Explicit Scenario-Aware Module** (Figure 2 (b)) instantiated from the Scenario-Oriented Module is proposed to model coarse-grained explicit scenario information. HierRec takes instance x_j as input and applies an embedding layer to transform sparse one-hot features, including both scenario features and common features, into dense embeddings. The scenario feature embedding is fed into Fully Connected (FC) layers, and the output representation is used to instantiate the *Explicit Scenario-Oriented Layer*, which is leveraged to model the explicit scenario. Following an **Implicit Scenario-Aware Module** (Figure 2 (c)) is proposed to model fined-grained implicit scenario information. A scenario-aware multi-head attention network is designed to perceive distinctive implicit patterns, which are further used to instantiate multiple *Implicit Scenario-Oriented*

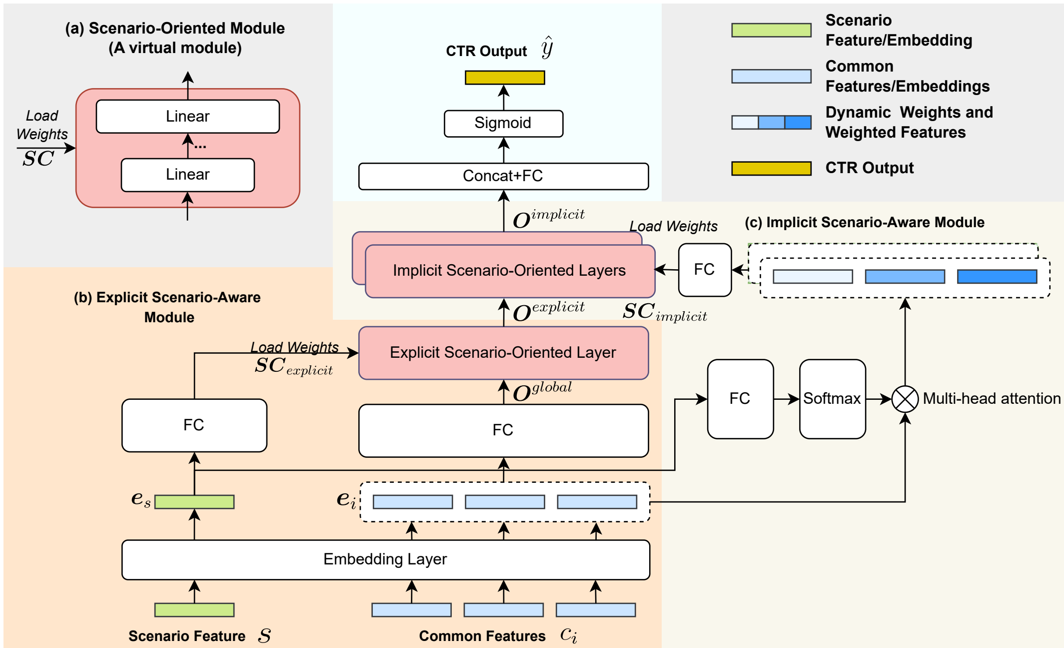


Figure 2: Overall structure of HierRec. By instantiating the Scenario-Oriented Module (a) under different scenario conditions, the Explicit Scenario-Aware Module (b) and Implicit Scenario-Aware Module (c) are applied. These modules can thus help HierRec achieve explicit and implicit scenario-aware modeling while preserving scenario extensibility.

Layers deployed parallelly for modeling complicated distribution from different perspectives. Finally, the outputs of the implicit scenario-oriented layers are concatenated and passed through the output layer for CTR prediction.

2.3 Scenario-Oriented Module (Virtual)

A key challenge in scenario modeling is how to design a unified paradigm for modeling different scenarios while portraying scenario similarities and differences. To depict different explicit and implicit scenarios delicately in a parameter-efficient manner, inspired by the dynamic weight technique [41], HierRec proposes a virtual module called scenario-oriented module based on the re-parameterization method to adaptively generate parameters depending on the given *scenario condition*. By instantiating the scenario-oriented module under different scenario conditions, HierRec can easily achieve explicit and implicit scenario-aware modeling while preserving scenario extensibility. As illustrated in Figure 2 (a), the scenario-oriented module is composed of multiple linear layers, in which the calculation of the l_{th} linear layer could be expressed as:

$$\boldsymbol{h}_{l+1} = \boldsymbol{W}_l \boldsymbol{h}_l + \boldsymbol{b}_l, \quad l \in [1, L], \tag{1}$$

where $\boldsymbol{h}_l$ and $\boldsymbol{h}_{l+1}$ are the input and output, and $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$ are its weights and bias, and L is the number of the layers.

In order to model different scenarios according to different scenario-specific information, the network weights $\{\boldsymbol{W}_l\}_{l\in[1,L]}$ and $\{\boldsymbol{b}_l\}_{l\in[1,L]}$ are adaptively generated under different scenario conditions, which can be represented as:

$$\boldsymbol{W}_l, \boldsymbol{b}_l = Reshape(\boldsymbol{SC})[l] \quad l \in [1, L], \tag{2}$$

where $\boldsymbol{SC}$ is the *scenario condition* representation expressed by a parameter vector generated from the given scenario, and *Reshape* function reshapes and splits $\boldsymbol{SC}$ into L parts whose the l_{th} part is used for parameterizing $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$. Based on the basic scenario-oriented module, the **explicit scenario-oriented layer** and **implicit scenario-oriented layers** are proposed by instantiating the scenario-oriented module with different scenario conditions $\boldsymbol{SC}$ to capture explicit and implicit information.

2.4 Explicit Scenario-Aware Module

The explicit scenario-aware module is designed to facilitate modeling based on pre-defined scenario identification. This module faces key challenges, including: (i) achieving parameter efficiency in modeling explicit scenarios and (ii) capturing the similarities and differences between explicit scenarios both effectively and efficiently. To address these challenges, we propose the explicit scenario-aware module, which leverages information from explicit scenarios to initialize the scenario-oriented module.

Specifically, the explicit scenario-aware module first embeds all features (including common features and pre-defined scenario features representing explicit scenarios) into dense embeddings with shape $\mathbb{R}^d$ (d is the embedding dimension) via an embedding layer:

$$\begin{cases} \boldsymbol{e}_i = \boldsymbol{EM}_i \cdot Onehot(c_i), & i \in [1, I] \\ \boldsymbol{e}_s = \boldsymbol{EM}_s \cdot Onehot(s), \end{cases} \tag{3}$$

where all the features are first transformed into one-hot vectors by *Onehot* function and then transformed by the embedding matrices $\boldsymbol{EM}_i$ or $\boldsymbol{EM}_s$ according to the feature fields that they belong to.

In order to save the model parameters and facilitate online inference to solve the problem (i), the common feature embeddings $\boldsymbol{E}_c = \{\boldsymbol{e}_1, ...\boldsymbol{e}_i, ...\boldsymbol{e}_I\}$ are concatenated and passed through a shared FC layer for dimension reduction and feature interaction modeling [32], obtaining global representation $\boldsymbol{O}^{global}$. Afterward, the scenario embedding $\boldsymbol{e}_s$ is further dimensionally transformed through FC layers to yield explicit scenario condition $\boldsymbol{SC}_{explicit}$ for instantiating the explicit scenario-oriented layer:

$$SC_{explicit} = FC(\boldsymbol{e}_s). \tag{4}$$

Here, FC layers contains K layers, and the k_{th} layer be:

$$\boldsymbol{h}_{k+1} = \sigma(Dropout(BN(\boldsymbol{W}_k\boldsymbol{h}_k + \boldsymbol{b}_k))), \quad k \in [1, K], \tag{5}$$

where σ is the activation function [4], $Dropout$ is the dropout function [29], BN is the batch normalization function [13] and $\boldsymbol{W}_k$ and $\boldsymbol{b}_k$ is the weight and bias of this layer. This method enables adaptive learning of scenario correlations by optimizing the FC layers, thereby addressing the problem (ii).

Then the explicit scenario condition $\boldsymbol{SC}_{explicit}$ is used to instantiate the dynamic weights of the explicit scenario-oriented layer by E.q.(2) and the global dimension-reduced representation $\boldsymbol{O}^{global}$ is fed into the explicit scenario-oriented layer for explicit modeling. By doing this, we can obtain the representation under the current explicit scenario $\boldsymbol{O}^{explicit}$.

2.5 Implicit Scenario-Aware Module

The implicit scenario-aware module aims to facilitate modeling based on implicit scenarios. It confronts several pivotal challenges: (i) uncovering valuable implicit patterns, (ii) facilitating detailed implicit scenario-aware modeling within various explicit scenarios, and (iii) integrating seamlessly with the explicit scenario-aware module in a hierarchical structure. To overcome them, we introduce the implicit scenario-aware module which utilizes data from implicit scenarios to effectively initialize the scenario-oriented module.

Specifically, given the multitude of implicit patterns based on common feature combinations and the fact that not all of them may necessarily be helpful for recommendations, it is important to identify beneficial implicit patterns effectively. To fully perceive complex data distribution and identify important implicit scenarios adaptively, addressing the problem (i) and (ii), HierRec proposes a scenario-aware multi-head attention structure to soft-select multiple beneficial implicit patterns. Specifically, the explicit scenario embedding $\boldsymbol{e}_s$ is first fed into an FC layer, whose output representation is split and reshaped into multi-group weights, which are further normalized via the Softmax function [23] to generate multi-group distributions. This process can be expressed as:

$$\begin{cases} \boldsymbol{weight}_{ori} = Reshape(FC(\boldsymbol{e}_s)) \\ \boldsymbol{weight}_{norm}[g] = Softmax(\boldsymbol{weight}_{ori}[g]), \\ g \in [1, G] \end{cases} \tag{6}$$

where $\boldsymbol{weight}_{ori} \in \mathbb{R}^{G\times I}$, $\boldsymbol{weight}_{norm} \in \mathbb{R}^{G\times I}$ are the G group weights before and after the Softmax normalization. G is the number of attention heads and I is the number of common features. By doing this, each weighted vector $\boldsymbol{weight}_{norm}[g]$ represents a kind of discovering implicit pattern over the common features, and each element in vector $\boldsymbol{weight}_{norm}[g]$ reflects the importance of the corresponding common feature under the current implicit scenario. Finally, the weighted vectors $\boldsymbol{weight}_{norm}$ are multiplied with the common feature embeddings $\boldsymbol{E}_c$ in an element-wise manner, which can be denoted as:

$$\boldsymbol{P} = \boldsymbol{weight}_{norm} \otimes \boldsymbol{E}_c, \tag{7}$$

where $\boldsymbol{E}_c \in \mathbb{R}^{Id}$ is the concatenated common feature embeddings and $\boldsymbol{P} \in \mathbb{R}^{G\times Id}$ is the G groups identified implicit scenario representations. By doing this, HierRec soft-selects multiple important implicit patterns adaptively, facilitating fine-grained modeling.

Afterward, G groups implicit scenario representations $\boldsymbol{P}$ are further dimensional transformed through a shared FC layer respectively for obtaining G disparate scenario conditions $\boldsymbol{SC}_{implicit}$, where the g_{th} scenario condition can be shown as:

$$\boldsymbol{SC}_{implicit}[g] = FC(\boldsymbol{P}[g]), \quad g \in [1, G]. \tag{8}$$

Similarly, the G implicit scenario conditions $\boldsymbol{SC}_{implicit}$ are used to instantiate the dynamic weights of G parallel implicit scenario-oriented layers by E.q.(2). Finally, to establish a hierarchical structure addressing the problem (iii), the explicit representation $\boldsymbol{O}^{explicit}$ is then fed into the implicit scenario-oriented layers for implicit modeling, and the output representation from each implicit scenario-oriented layer can be denoted as $\boldsymbol{O}_g^{implicit} (g \in [1, G])$.

2.6 Output Layer

After the explicit and implicit scenario-aware modeling, the output representations of the G implicit scenario-orient layers $\boldsymbol{O}_g^{implicit} (g \in [1, G])$ are concatenated together and passed through FC layers with sigmoid function for CTR prediction $\hat{y}$, which be expressed as:

$$\hat{y} = Sigmoid(FC(Concat(\boldsymbol{O}_1^{implicit}, ..., \boldsymbol{O}_G^{implicit}))). \tag{9}$$

Finally, the optimization of HierRec follows the commonly used settings for CTR task [39]. Specifically, the widely-used Binary Cross Entropy (BCE) loss [49, 52] is deployed to measure the CTR accuracy with the prediction score $\hat{y}$ and the ground-truth label y, which is defined as follows:

$$\mathcal{L}(\Phi) = -\frac{1}{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}\left[y_j \log \hat{y}_j + \left(1 - y_j\right)\log\left(1 - \hat{y}_j\right)\right]. \tag{10}$$

The full optimization algorithm could be found in Appendix A.1.

3 Offline Experiments

In this section, we conduct experiments on two public datasets to investigate the following questions:

- **RQ1:** How does HierRec perform in comparison with multi-scenario recommendation baselines?
- **RQ2:** Is the designed hierarchical structure helpful in making predictions for different scenarios?
- **RQ3:** Is the inference efficiency of HierRec sufficient for online deployment requirements?

In the following subsections, we first describe the evaluation settings. Afterward, we would answer the corresponding questions based on a brief review of our experiment results.

3.1 Experimental Setup

3.1.1 Dataset. We conduct experiments on two commonly-used datasets, i.e., Ali-CCP [3] [22] and KuaiRand [4] [6]. For Ali-CCP, which

[3]https://tianchi.aliyun.com/dataset/408
[4]https://kuairand.com/

Table 1: Statistics of evaluation datasets

Dataset	#Scenarios	#Features	Instances(M)		
			Train	Val	Test
Ali-CCP	3	23	38.07	4.23	43.02
KuaiRand	5	37	5.28	0.66	0.66

has a training set and a test set originally, following Xu et al. [40], we split the training set into training/validation sets with an 8:2 ratio. The classification of explicit scenarios follows the settings of the official instruction and previous work [35], which is expressed by the scenario identification feature field "*301*", indicating a categorical expression of recommendation position. For KuaiRand, to facilitate evaluation, we select the top-5 pre-defined scenarios with the most data for evaluation and split the dataset into training/validation/test sets with an 8:1:1 proportion [48]. We follow the settings of the official description [6] to divide explicit scenarios with scenario identification feature field "*tab*", which indicates the interaction scenario such as the recommendation page or main page of the Kuaishou App[5]. The statistics of the two datasets are summarized in Table 1.

3.1.2 Baseline. To verify the effectiveness of the proposed approach, we compare HierRec with the following baselines:

- **Shared Bottom** shares the embedding layer and bottom FC layers for joint modeling of different tasks, and several scenario-specific FC layers are adopted for each scenario.
- **MMoE** [22] implicitly models task relationships for multi-task learning. Here, different scenarios are treated as different tasks in modeling, and scenario-specific towers and gating networks are applied for each scenario.
- **PLE** [30] uses a progressive layer extraction for multi-task learning. Similar to MMoE, we apply scenario-specific experts and towers for each scenario.
- **STAR** [28] utilizes scenario-specific tower networks to learn scenario-specific information for each scenario, and a shared network to learn shared information.
- **APG** [42] tries to use the low-rank matrix decomposition method to parameterize dynamic weights with features of each sample for efficient recommendations.
- **AdaptDHM** [16] attempts to learn adaptive clustering through the similarity of different representations in the training stage to model implicit data distributions.
- **AdaSparse** [44] utilizes scenario embeddings as a unique input to implement scenario-aware neuron-level weighting and then adaptively learns different sparse structures for each scenario.

3.1.3 Implementation Details. The widely used metrics of AUC and Logloss are deployed for evaluation. Specifically, a higher AUC value or a lower Logloss at the "0.001" level indicates significantly better performance [9]. Besides, the RelaImpr [27, 43] metric is also applied to measure the relative improvement between HierRec and the best baselines:

$$\text{RelaImpr} = \left(\frac{AUC(\text{HierRec}) - 0.5}{AUC(\text{Best baseline}) - 0.5} - 1 \right) \times 100\%. \quad (11)$$

[5]https://www.kuaishou.com/cn

For a fair comparison, we fix the embedding size of each feature at 16, the batch size at 2000, and the optimizer is the commonly used "Adam Optimizer" [15]. The simple grid search is performed for all the adjustable hyper-parameters of HierRec and baselines. For FC layers (except the output layer), the number of layers is searched from 1 to 3, and neurons at each layer from {16, 32, 64, 128}. For explicit/implicit scenario-oriented layers, to simplify the design and reduce the number of parameters, referring to the bottleneck structure [11, 26], the number of layers L in Equation (2) is set to 2. The first linear layer (i.e., bottleneck layer) contains fewer neurons, while the second linear layer contains more neurons. To be more fair, we keep the parameter amount of HierRec similar to the baselines and set the FC layers in the output layer to a small value. The detailed parameter statistics and analysis of hyper-parameters can be found in Appendix A.2, A.3. Besides, we run each experiment 10 times with the optimal parameters searched and report the average performance. For ease of reproduction, we provide the source code[1] for the experiments conducted using the Ali-CCP and KuaiRand datasets and guidelines in Appendix A.4. Additionally, we have provided data samples from both datasets for reference purposes.

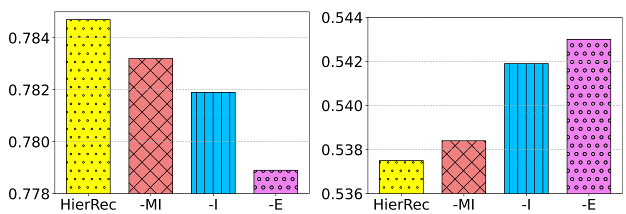
3.2 Overall Performance (RQ1)

This subsection gives an overall comparison between HierRec and different baselines, whose results are depicted in Table 2. From this we can conclude that:

- Multi-task based models (Shared Bottom, MMoE, PLE) achieve acceptable results on both datasets, which demonstrates that benefiting from the task sharing and exclusive mechanisms, multi-task learning based methods can also be applied to multi-scenario recommendations. MMoE outperforms Shared Bottom due to the modeling of task relations and better sharing design with gating networks. Besides, PLE outperforms the other two models, illustrating the effectiveness of refined information isolation in scenario-shared and scenario-specific modules and the progressive routing mechanism for information extraction.
- Multi-scenario based models (STAR, AdaSparse) achieve better performance than multi-task based models, elaborating the significance of effectively modeling the differences and associations within different explicit scenarios. In addition, from the overall performance, AdaSparse outperforms STAR due to its utilization of the DW approach for controlling neuron weights using scenario information. This allows for a more precise recognition of scenario distinctions, as well as a more detailed processing of scenario associations in every hidden layer.
- APG and AdaptDHM achieve good recommendation effects by modeling sample-level data distribution, which is similar to the concept of the implicit scenario discussed in this paper. This case demonstrates the importance of implicit scenario modeling. However, their performance is still inferior to AdaSparse and HierRec, which suggests the significance of combining explicit and implicit scenario modeling.
- HierRec outperforms all the baselines in both scenario-individual and overall performance by a significant margin, showing superior prediction capabilities and proving the effectiveness of combining explicit and implicit scenario modeling. The multi-head implicit modeling design contributes to perceiving complicated

Table 2: Performance comparison of HierRec and baselines, where sce_d indicates the evaluation in the d-th scenario. Boldface denotes the highest score and underline indicates the best result of all baselines. "*" indicates the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the best baseline. ↑: higher is better; ↓: lower is better.

Approach	Performance for Each Scenario (AUC ↑)								Overall Performance			
	Ali-CCP			KuaiRand					Ali-CCP		KuaiRand	
	sce_1	sce_2	sce_3	sce_1	sce_2	sce_3	sce_4	sce_5	AUC ↑	Logloss ↓	AUC ↑	Logloss ↓
Shared Bottom	0.6094	0.5545	0.6064	0.7298	0.7183	0.7187	0.7904	0.7565	0.6030	0.2062	0.7757	0.5453
MMoE	0.6181	0.5727	0.6123	0.7292	0.7199	0.7153	0.7794	0.7553	0.6107	0.1635	0.7776	0.5444
PLE	0.6154	0.5919	0.6126	0.7285	0.7221	0.7188	0.7902	0.7661	0.6133	0.1621	0.7784	0.5427
STAR	0.6187	0.5954	0.6132	0.7323	0.7205	0.7204	0.7903	0.7772	0.6149	0.1622	0.7802	0.5415
APG	0.6190	0.5965	0.6154	0.7324	0.7211	0.7199	0.7907	0.7788	0.6153	0.1621	0.7805	0.5407
AdaptDHM	0.6186	0.5965	0.6164	0.7320	0.7300	0.7197	0.7783	0.7875	0.6165	0.1620	0.7813	0.5399
AdaSparse	0.6192	0.5970	0.6166	0.7324	0.7301	0.7207	0.7971	0.8184	0.6171	0.1619	0.7815	0.5384
HierRec	**0.6253***	**0.6046***	**0.6228***	**0.7351***	**0.7324***	**0.7250***	**0.8005***	**0.8442***	**0.6237***	**0.1614***	**0.7847***	**0.5376***
RelaImpr	5.12%	7.84%	5.32%	1.21%	1.00%	1.95%	1.14%	8.10%	5.64%	-	1.14%	-



(a) AUC. (b) Logloss.

Figure 3: Ablation Study of overall scenarios performance on the KuaiRand dataset.

distributions and achieving fine-grained modeling. Additionally, the improvements in Ali-CCP are greater than that in KuaiRand. We attribute this distinction to the complexity of different data distributions in different datasets. For complex scenarios, the hierarchical modeling of HierRec brings superior modeling ability and uncovers more intricate correlations within scenarios, thus achieving remarkable improvements.

3.3 Ablation Study (RQ2)

This subsection presents the ablation study of our proposed HierRec model. Specifically, we compare HierRec with the following alternatives on the KuaiRand dataset:

- **w/o multi-head attention (-MI)**: with only one head in the implicit scenario-aware module;
- **w/o implicit layers (-I)**: without implicit scenario-oriented layers for implicit modeling;
- **w/o explicit layers (-E)**: without explicit scenario-oriented layers for explicit modeling.

Based on the results in Figure 3, we could conclude that both the explicit and implicit scenario modeling play an important role for HierRec. Besides, explicit modeling is more prominent, which is also the selection motivation of existing work [28, 44]. This is because data samples in different explicit scenarios often exhibit significant distribution differences, arising from their unique positions or presentation methods (e.g., advertising slot). Modeling explicit scenarios adequately allows for capturing these explicit scenario-specific differences [28]. However, the improvement brought by implicit modeling is non-negligible as it could further uncover more intricate correlations among samples through the exploration of feature-based implicit patterns. Moreover, the multi-head implicit modeling can perceive complicated data distribution sufficiently, thus conducive to fine-grained implicit modeling.

Table 3: Average inference time per batch with size 2000. The calculation of the increase percentage is based on the baseline which takes the most time.

Approach	Average Inference Time (ms)	
	Ali-CCP	KuaiRand
Shared Bottom	24.13	12.32
MMoE	23.55	12.25
PLE	25.99	13.63
STAR	23.42	11.75
APG	23.04	11.19
AdaptDHM	24.13	12.77
AdaSparse	23.74	11.87
HierRec (Ours)	26.63	13.72
Increase	2.43%	0.68%

3.4 Inference Efficiency Analysis (RQ3)

In practical applications, the inference efficiency of CTR models is a significantly important index due to the essential need for real-time response in recommender systems. Therefore, to answer RQ3, this subsection presents a comparison of inference time between HierRec and other baselines on the test set of Ali-CCP and KuaiRand. The experiments are conducted on NVIDIA GeForce RTX 3060 GPU over the entire test set (43 million samples for Ali-CCP and 1.5 million samples for KuaiRand). The batch size is set to 2000. The average inference time for one batch is reported in Table 3. Based on Table 3, it can be concluded that the inference time of HierRec increases slightly in comparison with other baselines due to the detailed multi-head implicit scenario-aware modeling. The increase in inference time is minor and acceptable for industrial applications.

4 Application: Online Advertising Platform

4.1 Scenario Description & Experimental Setting

In this section, we deploy HierRec at the ranking stage in a mainstream online advertising platform to verify its effectiveness. The overall framework of the platform is depicted in Figure 4, which

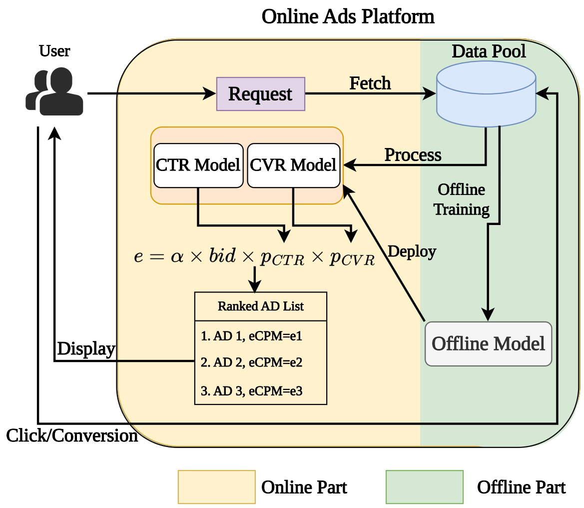


Figure 4: The overall framework of our online advertising platform, which could be roughly divided into the online part and the offline part. The recall and bidding stages are omitted since the paper focuses only on the CTR and CVR ranking prediction stage.

consists of the online part and the offline part. Specifically, when a user visits the websites, a request is sent to the online advertising platform and the platform fetches the corresponding features and feeds them to the ranking models (omitted recall stage). After that, the CVR model and CTR model predict the click rate (p_{CTR}) and conversion rate (p_{CVR}), respectively. Together with the bidding price, the platform ranks the candidate ADs by ranking function (e.g., eCPM) and displays top-ranked ADs to the user. As the final step of the loop, the click and conversion behaviors of users will be recorded and saved into the data pool, which will be used to train and update both the CTR and CVR models periodically.

The advertisers participating in the platform come from several industries, such as Automobile, Finance, and Real Estate, where *Industry Identification* is used as the explicit scenario feature to divide scenarios explicitly. Besides, more than 120 and 80 common features in CTR and CVR prediction are used to divide scenarios implicitly, including user profiles (e.g., gender), ads features (e.g., category), as well as contextual features (e.g., ad slot). For the categorical features, the feature embeddings are learned via embedding look-up, while the numerical feature embeddings are generated via the AutoDis [8]. We collect and sample about $6e8$ samples from online logs which contain two weeks of user behavior records. Specifically, the dataset for the CTR model training includes user impression records while only the clicked records are used to train the CVR model.

4.2 Experimental Results

4.2.1 Offline and Online Results. To verify the offline effectiveness of our HierRec model, we choose several compared baseline models, including single-scenario models (FiBiNet [12], DCN [32]) and multi-scenario/task models (DFFM [10], MMoE [22], PLE [30]) which are widely used in industrial recommender systems. The offline performance comparison on the large-scale industry dataset is presented in Table 4. We can observe that the multi-task baselines behave better in pCVR tasks while multi-scenario baselines are better-performing in pCTR tasks. In addition, our proposed HierRec outperforms all the baselines including single-scenario and multi-scenario/task models by a significant margin, verifying its effectiveness. Specifically, compared with the baselines, HierRec leads by a greater margin in pCVR tasks than in pCTR tasks. The reason for this may be users' conversation behaviors are deeper and more unpredictable than click behaviors, given limited training samples. Besides, the CVR models use fewer features and samples than the CTR models. Benefiting from the hierarchical structure of explicit and implicit scenario modeling, HierRec is able to perceive complicated distributions and achieves fine-grained modeling.

Table 4: Comparison of HierRec and baselines on the large-scale industry dataset on both pCTR and pCVR tasks. All experiments are repeated 5 times and we report the mean results in the table.

Tasks	Performance for Each Model (AUC ↑)					
	FiBiNet	DCN	DFFM	MMoE	PLE	HierRec
pCTR	0.8207	0.8219	0.8223	0.8217	0.8199	**0.8239**
pCVR	0.8947	0.9004	0.9010	0.9021	0.9028	**0.9084**

To further verify the performance of HierRec online, we conduct a two-week online A/B testing on the online advertising platform and deploy HierRec on both CTR and CVR ranking stages. The online CTR and CVR baselines are both highly-optimized deep multi-scenario models. For online serving, 1% of the users are randomly selected as the experimental group and recommended ads by HierRec while other users are in the control group. Compared with the baselines, the eCPM (effective cost per mile) is improved by 2.21% and 10.33% for the CTR and CVR stages, respectively. As a platform to recommend ads for the users, the higher eCPM means better online advertising effectiveness and more platform revenue, which is critical for the advertising platform. Besides, HierRec has comparable inference efficiency with other models as shown in Table 3, which demonstrates that HierRec is suitable for industrial applications. After sufficient online experimental observation, HierRec has been promoted to become the baseline model of the online advertising platform and serves billions of users by November 2023.

4.2.2 Implicit Scenario Analysis. We have visualized the weights $\boldsymbol{weight}_{norm}$ of the multi-head implicit scenario-aware modeling on the industrial dataset, as illustrated in Figure 5. From the results in Figure 5, it is evident that different attention heads assign varying weights to common features, enabling the perception and discovery of beneficial patterns in complex data distribution. In addition, several features consistently receive higher weights than others, underscoring the importance of these features for implicit scenario modeling and decision-making, thus compensating for the inadequacy of explicit modeling.

5 Related Work

This section offers a brief overview of multi-scenario recommendations [14, 19, 21, 37, 46] (a.k.a., multi-domain recommendations [5]) and Click-Through Rate prediction task.

Figure 5: Feature weights of the industrial dataset. Each count on the horizontal axis represents a common feature. Head-g represents the weights of the g_{th} attention head in implicit scenario-aware modeling.

5.1 Multi-Scenario Recommendations

Currently, existing multi-scenario models can be divided into two categories: Tower-based models and Dynamic Weight (DW) models, whose abstract structures are depicted in Figure 1. Specifically, tower-based models utilize a common network to represent scenario-shared information, upon which multiple sub-towers are built to capture scenario-specific details. Multi-task models belong to this category, such as Shared Bottom, MMoE [22], and PLE [30], which design task-sharing and task-specific networks to model task relations. Besides, STAR [28] utilizes several independent towers to learn scenario-specific information. And a shared network is applied to learn global information. It also leverages element-wise multiplication to establish connections between the tower and shared networks. CausalInt [35] further eliminates the negative transfers among different tower networks. With the design of a causal intervention method, CausalInt is able to selectively utilize the information from different scenarios to construct scenario-aware estimators in a unified model.

However, the complete isolation design between towers hinders the modeling of scenario correlations and also suffers from poor generalization and compatibility. To overcome these limitations, DW-based methods have been proposed by generating dynamic parameters adaptively for each scenario in a parameter-efficient manner [44, 48]. M2M [48] proposes a meta-unit, which uses scenario information to generate dynamic weights as parameters of different networks to realize multi-scenario and multi-task learning simultaneously. AdaSparse [44] utilizes scenario embeddings as unique input to implement scenario-aware neuron-level weighting so that it can adaptively learn sparse structures. However, all these existing multi-scenario models only consider coarse-grained explicit scenario modeling that depends on pre-defined scenario identification based on some manual prior rules, which is biased and sub-optimal. Recently, some studies have tried to consider fine-grained modeling of data distribution, which is similar to the concept of implicit scenarios. For instance, APG [41] tries to parameterize the model with input samples through low-rank matrix decomposition. Differently, AdaptDHM [16] tries to learn adaptive clusters for differential modeling among samples. However, these models ignore the guiding role of the pre-defined explicit scenarios in modeling and, therefore, cannot effectively extract and utilize beneficial implicit patterns for modeling. Therefore, to realize fine-grained modeling over complex data distribution, HierRec is proposed with a hierarchical structure to model explicit and implicit scenarios jointly.

5.2 Click-Through Rate Prediction

Click-Through Rate (CTR) prediction is a critical binary classification task in recommender systems, distinguishing user-item interactions with labels “1” (clicked) and “0” (not clicked). As a key measure of items' relevance from users' perspective, improving CTR is vital for augmenting the overall effectiveness of recommendation frameworks [45, 50, 51]. Over recent years, this area has garnered significant research interest [17, 34, 53], with methodologies evolving from logistic regression [25] through to factorization machines [9, 24], and advancing further to sophisticated deep network-based methods [20, 49]. These developments have progressively enhanced the capability to model complex feature interactions more effectively.

As the demand for more personalized and dynamic recommender systems grows, the focus has shifted towards multi-scenario CTR prediction, which tries to address data sparsity by applying data from multiple scenarios in modeling and enhancing performance across all scenarios. However, existing multi-scenario models primarily focus on targeted modeling with explicit scenario identifications, neglecting the potential impact of implicit patterns hidden in common feature combinations. To mitigate this oversight, this paper introduces HierRec, a novel approach designed to capture both explicit and implicit scenario-based patterns comprehensively. By leveraging the strengths of both types of patterns, HierRec aims to provide more accurate and context-aware recommendations, ultimately enhancing the user experience and effectiveness of recommender systems.

6 Conclusion

In this paper, we propose a scenario-aware hierarchical dynamic network HierRec to conduct explicit and implicit scenario modeling simultaneously. Specifically, a basic scenario-oriented module is first designed to capture scenario-specific information in different scenario conditions. Then the stacked explicit and implicit scenario-aware modules are proposed to model explicit and implicit scenario information in a hierarchical manner. Moreover, the multi-head implicit modeling design can perceive distinctive patterns effectively and achieve fine-grained modeling. Experiments on two public datasets and applications on a mainstream online advertising platform demonstrate the effectiveness of the proposed HierRec.

Acknowledgments

This research was partially supported by Huawei (Huawei Innovation Research Program), Research Impact Fund (No.R1015-23), APRC - CityU New Research Initiatives (No.9610565, Start-up Grant for New Faculty of CityU), CityU - HKIDS Early Career Research Grant (No.9360163), Hong Kong ITC Innovation and Technology Fund Midstream Research Programme for Universities Project, (No.ITS/034/22MS), Hong Kong Environmental and Conservation Fund (No. 88/2022), and SIRG - CityU Strategic Interdisciplinary Research Grant (No.7020046). We thank MindSpore [1] for the partial support of this work.

A Appendix

A.1 Model Optimization Algorithm

In this appendix, we delineate the optimization algorithm of HierRec, illustrated through pseudocode. The implementation, executed in PyTorch, is accessible within our code repository [1]. Algorithm 1 outlines the optimization procedure of HierRec, adhering to the standard methodology involving loss backpropagation. This sequence encompasses prediction generation (line 5), loss computation (line 6), and parameter updating (line 7). During the forward pass of the function f, for each sample, embeddings $\boldsymbol{e}_i$ and $\boldsymbol{e}_s$ are initially derived using Equation (3). Subsequently, $SC_{explicit}$ and $SC_{implicit}$ are calculated through Equations (4), (6), (7), and (8). The scenario-oriented module is instantiated using $SC_{explicit}$ and $SC_{implicit}$, facilitating the loading of weights for both explicit and implicit scenario-oriented layers. Ultimately, by forwarding $\boldsymbol{e}_i$ through the established structures, the output $\hat{y}$ could be obtained from Equation (9).

A.2 Parameter Statistics

This appendix presents statistical data regarding the parameter counts of the models utilized in the experiments described in Section 3. The parameter statistics are detailed in Table 5. As outlined in Section 3.1.3, to ensure an equitable comparison across models, we not only defined a specific parameter search space but also manually refined this search space for each model involved in the experiments. This meticulous adjustment ensures that each model operates with a comparable number of parameters, thereby neutralizing any performance disparities that might arise from variations in parameter quantities among the models.

Algorithm 1 Optimization algorithm of HierRec

Input: a training dataset $\mathcal{D} = \left\{(\boldsymbol{x}_j, y_j)\right\}_{j=1}^{|\mathcal{D}|}$ with $|\mathcal{D}|$ samples. $\boldsymbol{x}_j = \{s, c_1, ...c_i, ...c_I\}$ is user features and $y_j \in 0, 1$ is the CTR label
Output: A well-trained model f with parameters $\boldsymbol{\Phi}$

1: Randomly initialize parameters $\boldsymbol{\Phi}$ of the model f
2: **for** Epoch in 1,..., max epochs **do**
3:   **for** Batch 1,..., batch number **do**
4:     Sample a training batch DB from $\mathcal{D}$
5:     Make predictions with $f(DB)$
6:     Calculate loss $\mathcal{L}$ based on Equation 10
7:     Update $\boldsymbol{\Phi}$ via minimizing the loss $\mathcal{L}$
8:   **end for**
9:   **if** Coveraged **then**
10:     return f
11:   **end if**
12: **end for**
13: return f

14: **Function** $f(\boldsymbol{x})$**:**
15:   Calculate embeddings $\boldsymbol{e}_i$, $\boldsymbol{e}_s$ via Equation (3)
16:   Calculate $SC_{explicit}$ via Equation (4)
17:   Calculate $SC_{implicit}$ via Equation (6), (7) and (8)
18:   Forward $\boldsymbol{e}_i$ to obtain $\boldsymbol{O}^{global}$, $\boldsymbol{O}^{explicit}$ and $\boldsymbol{O}^{implicit}$
19:   Calculate $\hat{y}$ via Equation (9)
20:   return $\hat{y}$
21: **End Function**

Table 5: Parameter Statistics.

Model	Paramter Amount (M)	
	Ali-CCP	KuaiRand
Shared Bottom	20.87	43.91
MMoE	20.00	44.11
PLE	20.07	43.38
STAR	19.92	44.07
APG	18.01	41.23
AdaptDHM	21.75	44.86
AdaSparse	20.98	43.21
HierRec (Ours)	20.63	44.12

Table 6: The influence of the head number G in the implicit scenario-aware module on the KuaiRand dataset.

Head Num	1	2	4	8
AUC	0.7832	0.7841	0.7847	0.7845
Logloss	0.5382	0.5378	0.5376	0.5377

A.3 Hyper-parameter Analysis

This appendix delves into the analysis of the hyper-parameter "head num" G, which plays a pivotal role in the multi-head attention mechanism within the implicit scenario-aware module. The parameter G essentially determines the quantity of implicit scenarios selected for modeling under given explicit scenarios.

Our empirical investigation was carried out on the KuaiRand dataset, with results detailed in Table 6. The findings reveal that as the "head num" G increases, the performance of HierRec initially improves before reaching a plateau. This pattern underscores the utility of employing multi-head attention for implicit scenario modeling. An optimal "head num" G was identified to be approximately 4, suggesting that a moderate number of implicit scenarios suffices to yield substantial enhancements. This observation is attributed to the multi-head attention architecture's capability to judiciously identify and incorporate relevant implicit scenarios into the modeling process. Conversely, excessively augmenting the "head num" G risks the inclusion of irrelevant or detrimental implicit scenarios, potentially impeding the model's efficacy.

A.4 Guidelines for Reproduction

To ensure the reproducibility of HierRec, we have made the complete source code available [1]. For the datasets utilized in this study, Ali-CCP, and KuaiRand, sample datasets are available in the "data" directory for illustrative purposes. For hardware, our experiments were conducted using an NVIDIA GeForce RTX 3060 GPU. It is important to note that the hyper-parameters included in the code are provided as initial examples. The definitive hyper-parameters were determined through a comprehensive grid search, as detailed in Section 3.1.3. Additionally, due to hardware variances, results may slightly differ across different computational devices, even when using the same pseudo-random seed, because of the distinct ways in which floating-point operations are handled.

References

[1] 2020. MindSpore. https://www.mindspore.cn/
[2] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. 2020. Algorithmic effects on the diversity of consumption on spotify. In *Proc. of WWW*. 2155–2165.
[3] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, et al. 2022. CAN: feature co-action network for click-through rate prediction. In *Proc. of WSDM*. 57–65.
[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proc. of RecSys*. 191–198.
[5] Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2023. A unified framework for multi-domain ctr prediction via large language models. *arXiv preprint arXiv:2312.10743* (2023).
[6] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proc. of CIKM*. 3953–3957.
[7] Jingtong Gao, Xiangyu Zhao, Bo Chen, Fan Yan, Huifeng Guo, and Ruiming Tang. 2023. AutoTransfer: Instance Transfer for Cross-Domain Recommendations. In *Proc. of SIGIR*. 1478–1487.
[8] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In *Proc. of KDD*. 2910–2918.
[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
[10] Wei Guo, Chenxu Zhu, Fan Yan, Bo Chen, Weiwen Liu, Huifeng Guo, Hongkun Zheng, Yong Liu, and Ruiming Tang. 2023. DFFM: Domain Facilitated Feature Modeling for CTR Prediction. In *Proc. of CIKM*. 3953–3957.
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of CVPR*. 770–778.
[12] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proc. of RecSys*. 169–177.
[13] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of ICML*. 448–456.
[14] Yuchen Jiang, Qi Li, Han Zhu, Jinbei Yu, Jin Li, Ziru Xu, Huihui Dong, and Bo Zheng. 2022. Adaptive Domain Interest Network for Multi-domain Recommendation. In *Proc. of CIKM*. 3212–3221.
[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[16] Jinyun Li, Huiwen Zheng, Yuanlin Liu, Minfang Lu, Lixia Wu, and Haoyuan Hu. 2022. AdaptDHM: Adaptive Distribution Hierarchical Model for Multi-Domain CTR Prediction. *arXiv preprint arXiv:2211.12105* (2022).
[17] Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023. E4srec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443* (2023).
[18] Xinhang Li, Zhaopeng Qiu, Xiangyu Zhao, Zihao Wang, Yong Zhang, Chunxiao Xing, and Xian Wu. 2022. Gromov-wasserstein guided representation learning for cross-domain recommendation. In *Proc. of CIKM*. 1199–1208.
[19] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Hamur: Hyper adapter for multi-domain recommendation. In *Proc. of CIKM*. 1268–1277.
[20] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *Proc. of KDD*. 2636–2645.
[21] Dugang Liu, Chaohua Yang, Xing Tang, Yejing Wang, Fuyuan Lyu, Weihong Luo, Xiuqiang He, Zhong Ming, and Xiangyu Zhao. 2024. MultiFS: Automated Multi-Scenario Feature Selection in Deep Recommender Systems. In *Proc. of WSDM*. 434–442.
[22] 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 *Proc. of SIGIR*. 1137–1140.
[23] John Ashworth Nelder and Robert WM Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* (1972), 370–384.
[24] Steffen Rendle. 2010. Factorization machines. In *Proc. of ICDM*. 995–1000.
[25] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proc. of WWW*. 521–530.
[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. of CVPR*. 4510–4520.
[27] Qijie Shen, Wanjie Tao, Jing Zhang, Hong Wen, Zulong Chen, and Quan Lu. 2021. SAR-Net: A scenario-aware ranking network for personalized fair recommendation in hundreds of travel scenarios. In *Proc. of CIKM*. 4094–4103.
[28] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *Proc. of CIKM*. 4104–4113.
[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* (2014), 1929–1958.
[30] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proc. of RecSys*. 269–278.
[31] Kaan Varnali. 2021. Online behavioral advertising: An integrative review. *Journal of Marketing Communications* (2021), 93–114.
[32] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proc. of KDD*. 1–7.
[33] Tian Wang, Junming Fan, and Pai Zheng. 2024. An LLM-based vision and language cobot navigation approach for Human-centric Smart Manufacturing. *Journal of Manufacturing Systems* (2024).
[34] Xinfei Wang. 2020. A survey of online advertising click-through rate prediction models. In *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*. 516–521.
[35] Yichao Wang, Huifeng Guo, Bo Chen, Weiwen Liu, Zhirong Liu, Qi Zhang, Zhicheng He, Hongkun Zheng, Weiwei Yao, Muyu Zhang, et al. 2022. CausalInt: Causal Inspired Intervention for Multi-Scenario Recommendation. In *Proc. of KDD*. 4090–4099.
[36] Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Multi-task deep recommender systems: A survey. *arXiv preprint arXiv:2302.03525* (2023).
[37] Yuhao Wang, Xiangyu Zhao, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang. 2023. PLATE: A Prompt-Enhanced Paradigm for Multi-Scenario Recommendations. In *Proc. of SIGIR*. 1498–1507.
[38] Zhuoli Wang, Wei-Jue Huang, and Bingjie Liu-Lastres. 2022. Impact of user-generated travel posts on travel decisions: A comparative study on Weibo and Xiaohongshu. *Annals of Tourism Research Empirical Insights* (2022), 100064.
[39] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proc. of CIKM*. 2265–2268.
[40] Yu-Hao Xu, Zhen-Hai Wang, Zhi-Ru Wang, Rong Fan, and Xing Wang. 2022. A Recommendation Algorithm Based on a Self-supervised Learning Pretrain Transformer. *Neural Processing Letters* (2022), 1–17.
[41] Bencheng Yan, Pengjie Wang, Kai Zhang, Feng Li, Hongbo Deng, Jian Xu, and Bo Zheng. 2022. Apg: Adaptive parameter generation network for click-through rate prediction. *Proc. of NeurIPS* (2022), 24740–24752.
[42] Bencheng Yan, Pengjie Wang, Kai Zhang, Feng Li, Hongbo Deng, Jian Xu, and Bo Zheng. 2022. Apg: Adaptive parameter generation network for click-through rate prediction. *Proc. of NeurIPS* (2022), 24740–24752.
[43] Ling Yan, Wu-Jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale ctr prediction in display advertising. In *Proc. of ICML*. 802–810.
[44] Xuanhua Yang, Xiaoyu Peng, Penghui Wei, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. AdaSparse: Learning Adaptively Sparse Structures for Multi-Domain Click-Through Rate Prediction. In *Proc. of CIKM*. 4635–4639.
[45] Yanwu Yang and Panyu Zhai. 2022. Click-through rate prediction in online advertising: A literature review. *Information Processing & Management* (2022), 102853.
[46] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2022. A survey on cross-domain recommendation: taxonomies, methods, and future directions. *ACM Transactions on Information Systems* (2022), 1–39.
[47] Chi Zhang, Rui Chen, Xiangyu Zhao, Qilong Han, and Li Li. 2023. Denoising and prompt-tuning for multi-behavior recommendation. In *Proc. of WWW*. 1355–1363.
[48] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. 2022. Leaving no one behind: A multi-scenario multi-task meta learning approach for advertiser modeling. In *Proc. of WSDM*. 1368–1376.
[49] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584* (2021).
[50] Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, et al. 2024. M3oE: Multi-Domain Multi-Task Mixture-of Experts Recommendation Framework. In *Proc. of SIGIR*. 893–902.
[51] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering* (2024).
[52] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR prediction via knowledge distillation. In *Proc. of CIKM*. 2941–2958.
[53] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *Proc. of CIKM*. 2759–2769.