CausalInt: Causal Inspired Intervention for Multi-Scenario Recommendation

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

Huifeng Guo[1✉]
huifeng.guo@huawei.com
Huawei Noah's Ark Lab, China

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

Weiwen Liu
liuweiwen8@huawei.com
Huawei Noah's Ark Lab, China

Zhirong Liu
liuzhirong@huawei.com
Huawei Noah's Ark Lab, China

Qi Zhang
zhangqi193@huawei.com
Huawei Noah's Ark Lab, China

Zhicheng He
hezhicheng9@huawei.com
Huawei Noah's Ark Lab, China

Hongkun Zhen
zhenghongkun1@huawei.com
Huawei Technologies Co Ltd, China

Weiwei Yao
yaoweiwei4@huawei.com
Huawei Technologies Co Ltd, China

Muyu Zhang
zhangmuyu@huawei.com
Huawei Technologies Co Ltd, China

Zhenhua Dong
dongzhenhua@huawei.com
Huawei Noah's Ark Lab, China

Ruiming Tang[1✉]
tangruiming@huawei.com
Huawei Noah's Ark Lab, China

ABSTRACT

Building appropriate scenarios to meet the personalized demands of different user groups is a common practice. Despite various scenario brings personalized service, it also leads to challenges for the recommendation on multiple scenarios, especially the scenarios with limited traffic. To give desirable recommendation service for all scenarios and reduce the cost of resource consumption, how to leverage the information from multiple scenarios to construct a unified model becomes critical. Unfortunately, the performance of existing multi-scenario recommendation approaches is poor since they introduce unnecessary information from other scenarios to target scenario. In this paper, we show it is possible to selectively utilize the information from different scenarios to construct the scenario-aware estimators in a unified model. Specifically, we first do analysis on multi-scenario modeling with causal graph from the perspective of users and modeling processes, and then propose the Causal Inspired Intervention (CausalInt) framework for multi-scenario recommendation. CausalInt consists of three modules: (1) **Invariant Representation Modeling** module to squeeze out the scenario-aware information through disentangled representation learning and obtain a scenario-invariant representation; (2) **Negative Effects Mitigating** module to resolve conflicts between different scenarios and conflicts between scenario-specific and scenario-invariant representations via gradient based orthogonal regularization and model-agnostic meta learning, respectively; (3) **Inter-Scenario Transferring** module designs a novel *TransNet* to simulate a counterfactual intervention and effectively fuse the information from other scenarios. Offline experiments over two real-world dataset and online A/B test are conducted to demonstrate the superiority of CausalInt.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Transfer learning**.

KEYWORDS

Multi-Scenario, Transfer Learning, Negative Mitigating, Invariant Representation

ACM Reference Format:
Yichao Wang, Huifeng Guo[1✉], Bo Chen, Weiwen Liu, Zhirong Liu, Qi Zhang, Zhicheng He, Hongkun Zhen, Weiwei Yao, Muyu Zhang, Zhenhua Dong, and Ruiming Tang[1✉]. 2022. CausalInt: Causal Inspired Intervention for Multi-Scenario Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539221

KDD '22, August 14–18, 2022, Washington, DC, USA

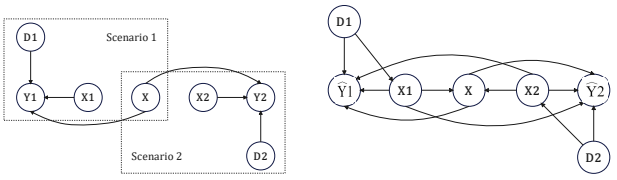
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539221

1 INTRODUCTION

Industrial commercial companies usually build multiple scenarios to meet the personalized needs of users. Take an example of travel market platform, there are hundreds of travel scenarios including parent-child theme, couple-travel theme etc [12]. In different scenarios, users will explore the preferred items with different motivations or under different situations, and meanwhile generate various behaviors, such as browses, clicks, downloads and purchases. Then, based on the user behaviors and logs, recommendation models are trained and used to serve users.

However, almost existing works in recommender systems treat the modeling processes as *Single-Scenario Modeling (SSM).* They

✉Huifeng Guo and Ruiming Tang are the co-corresponding authors



(a) User's perspective. (b) Model's perspective.

Figure 1: Causal graph of MSM from different perspectives.

train the model only with the samples collected from the corresponding scenario and then serve the users in this scenario. Although SSM can capture distinctions of user behaviors in corresponding scenarios, there are three downsides: (1) Data sparsity issue may exist in some scenarios especially in the cold-start scenarios. It is hard to train an effective model with the rare samples from individual scenario. (2) It is infeasible to construct comprehensive user profiles for each scenario without the information from other scenarios thus resulting in sub-optimal performance. (3) SSM will cause huge resource consumption and human cost since there are usually dozens or even hundreds of scenarios simultaneously in large-scale commercial platforms. To address the above issues in SSM, *Multi-Scenario Modeling (MSM)* is proposed to train a unified model so as to utilize samples from all scenarios and capture diverse user behaviors from different scenarios. The unified model serves users in all scenarios simultaneously which can significantly reduce the resource consumption. However, the complex interactions hidden in the mixed training data are hindering the performance of existing MSM models.

To better understand the principle of MSM, we present the formal definition through the causal graph [9] (as shown in Figure 1) for qualitative analysis. The nodes denote cause or effect factors, the edge $A \rightarrow B$ means the A directly affect B.

- Firstly, we illustrate the causal graph from user's perspective in Figure 1(a): Specifically, X_n, D_n and X are causes, Y_n is the effect node to denote the observed action in scenarios n, where X_n and D_n are user behaviors and scenario-aware information in scenario n, respectively. And X is scenario-invariant user portraits, Y_n denotes the observed action (e.g. click). Remarkably, node X_n, D_n and X only have direct effect on the outcome node Y_n. It is a relatively ideal causal graph since scenario-specific nodes have no effect on nodes of the other scenarios.
- Secondly, as illustrated in Figure 1(b), we further consider the causal graph from the perspective of MSM: Specially, D_1 also has direct effect on X_1 and indirect effect on X, $\hat{Y}_2$ will be affected through causal path $D_1 \rightarrow X_1 \rightarrow \hat{Y}_2$ and $D_1 \rightarrow X_1 \rightarrow X \rightarrow \hat{Y}_2$ respectively, which will lead to inaccurate estimation in *scenario* 2.

In order to recover the causal graph from Figure 1(b) to Figure 1(a) for unbiased and accurate prediction, the estimator in MSM needs to overcome three key challenges: (1) *Extracting commonalities of different scenarios*: X is the general invariant representation to be shared across all scenarios, which should affect the outcome node Y_n in different scenarios without biasing toward specific scenarios. (2) *Retaining specialities of target scenario and mitigating negative transfer*: X_n is the scenario-specific representation that contains specialities of users and items in the target scenario. Meanwhile, X_n should not have direct effect on the other outcome nodes (e.g., $X_1 \rightarrow \hat{Y}_2$ in Figure1(b)) to avoid conflicts. (3) *Exploiting and transferring scenario-aware information*: D_n is scenario-aware information (e.g. scenario indicators), which acts as *confounder* [9] and has direct or indirect effect on the outcome node in different scenarios. Exploiting and transferring the D_n effectively can promote the performance in the target scenario.

Table 1: Comparison of different MSM approaches.

Approaches		Challenge 1	Challenge 2	Challenge 3
Heuristic	Mixing	Limited	×	×
	Finetune	Limited	Limited	×
MTL	SharedBottom	Limited	Limited	×
	PLE	Limited	Limited	×
MSM	STAR	Limited	Limited	Limited
	CausalInt	High	High	High

There are three typical kinds of models for MSM: (1) Heuristic models, which are trained with heuristic strategies, e.g. Mix and Finetune. Mix is trained simply with samples from all scenarios without extra adjustment. As the characteristics of different scenarios are significantly different and Mix does not consider distinctions of different scenarios, it is difficult to train a unified model that performs well in all scenarios. Finetune further adjust the Mix model with samples from the target scenario, which can capture the characteristics of the target scenario. Whereas updating models only with samples of target scenario may lead to catastrophic forgetting issues [7] and impairs the shared parameters (e.g., $D_1 \rightarrow X_1 \rightarrow X \rightarrow \hat{Y}_2$). (2) Multi-Task Learning (MTL) based models share parameters in bottom layers and build separate tower for each scenario based on the shared layers[1, 8, 15]. MTL focuses on modeling relationship among various tasks in different label space while MSM aims at addressing tasks in same label space across different scenarios, directly adapting MTL methods to solve MSM tasks may result in sub-optimal performance. (3) Existing MSM models inherit the structure of MTL and further model the scenario-aware information explicitly [12, 13]. However, the commonalities are not efficiently extracted and the negative impact from the other scenarios is not mitigated.

To address these challenges, we propose the CausalInt inspired by the casual graph of MSM, which consists of three modules: **(1) Invariant Representation Modeling** module (cutoff $D_n \rightarrow X_n \rightarrow X$) applies disentangled representation learning approach, which squeezes out the scenario-aware information and obtains a scenario-invariant representation shared across different scenarios. **(2) Negative Effects Mitigating** module (cutoff $X_i \rightarrow Y_j$) conducts gradient based orthogonal regularization and model-agnostic meta-learning to resolve the conflict between different scenario-specific information as well as the conflict between scenario-specific and scenario-invariant information respectively. Thus eliminating negative transfer issues among scenarios and retaining the specialities in target scenario. **(3) Inter-Scenario Transferring** module starts from a counterfactual question and designs a novel *TransNet* with gating mechanism to effectively fuse the information from other scenarios. Comparisons of the existing approaches are presented in Table 1.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, it is the first work to analyze the multi-scenario modeling process from the perspective of causal intervention.

- We propose a novel CausalInt which is inspired by the causal intervention. Three modules are designed to resolve the key challenges in MSM. Invariant Representation Modeling module learns a scenario-invariant representation to be shared across scenarios. Negative Effects Mitigating module eliminates negative transfers among scenarios and retain specialities in target scenario. Inter-Scenario transferring module fuses transferable information from other scenarios to promote the performance of target scenario.
- Evaluations on both the offline and online A/B test demonstrate the effectiveness of the proposed method. CausalInt has been deployed in an online advertising platform in Huawei and serving millions of daily active users.

2 RELATED WORK

In this section, we will briefly introduce some related works of our proposed model, including Single-Scenario Recommendation, Multi-Task Learning and Multi-Scenario Recommendation.

2.1 Single-Scenario Recommendation

Most existing deep CTR models mainly focus on single scenario modeling and follow the embedding and MLP paradigm. Wide&Deep[3] and DeepFM[6] combine low-order (explicit interaction) part and deep (implicit interaction) part to improve the performance. EDCN[2] further enhances the information sharing between different interaction network in deep models with parallel structure. DIN[20] applies attention mechanism to capture interests from user behaviors with respect to target items. SIM[10] extracts user interests with two cascaded search units, which achieves better ability to model lifelong sequential behavior.

2.2 Multi-Task Learning

Multi-Task Learning (MTL) has been proposed to improving generalization by sharing transferable information across related tasks. The shared knowledge and task-specific knowledge are explored to promote the learning of different tasks. Shared Bottom[1] is the first MTL model with hard parameter sharing paradigm, which is simple but effective in industrial applications. After that, multiple studies propose soft parameter sharing paradigm to model the relationship between tasks. MMoE [8] utilizes different gate networks for each task to fuse multiple experts among tasks. PLE [15] separates shared components and task-specific components explicitly and adopts a progressive routing mechanism to extract and separate deeper semantic knowledge gradually.

2.3 Multi-Scenario Recommendation

Different from single scenario modeling, multi-scenario modelling trains an unified model with samples collected from all scenarios and serves users in all scenarios simultaneously. This brings two benefits: one is the performance promotion brought by knowledge transfer between scenarios; another is efficiency of resource and manpower for that only one model need to be maintained. Domain adaption (DA) based approaches (e.g. Finetune) [16] and multi-task learning based approaches can be used to address the multi-scenario recommendation problem. However, DA based methods transfer knowledge in one way, i.e. from the source domain to the target domain, which is inefficient in multi-scenario recommendation. Multi-task learning based methods focus on handling tasks in different label space, while the multi-scenario learning usually makes prediction for different scenarios in the same label space.

3 PRELIMINARY

In recommender system, the model takes the input as user historical behaviors, user profiles, item features and contextual features. The features is mapped by the embedding layer from high-dimensional sparse IDs into low-dimensional dense vectors. The prediction $\hat{y}$ of a user clicking one an item is calculated via: $\hat{y} = f(E)$, where E is the aggregated embedding of all features. In multi-scenario recommendation, the model take input as the (x, y, d), where x is the common features used by multiple scenarios consisting of user historical behaviors, user profiles, item features and contextual features. As we focus on Click-Through Rate(CTR) task, $y \in \{0, 1\}$ is the label indicating click or not. $d \in \{1, 2, \cdots, N\}$ is the scenario indicator that indicates which scenario the samples come from.

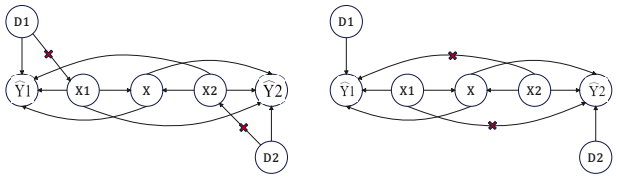
4 CAUSAL INSPIRED ANALYSIS

Causal graph is a directed acyclic graph, where a node denotes a variable and an edge indicates a causal relation between two variables [9], and helpful to the design of predictive models [19]. In this section, we conduct a causal inspired analysis for multi-scenario recommendation from the perspective of modeling and give the motivations of designing CausalInt.

4.1 Causal Graph from model perspective

As shown in Figure 1(b), we present the causal graph for multi-scenario recommendation from modeling perspective. The meaning of the nodes and edges are list as follows:

- **Node** X: the general scenario-invariant representation of user.
- **Node** X_n: the scenario-specific representation of users in scenario n, e.g. different behaviors of a user in different scenarios.
- **Node** D_n: the scenario-aware information, e.g. scenarios indicators.
- **Node** $\hat{Y}_n$: the predicted probability of a user clicking on an exposed item in scenario n, which is corresponding to the node Y_n in Figure1(a). The goal of the estimator is to minimize the discrepancy between $\hat{Y}_n$ and Y_n.
- **Edges** $\{D_n, X_n, X\} \rightarrow \hat{Y}_n$: the predicted probability in each scenario is determined by three factors: scenario-aware information, scenario-specific user representation and scenario-invariant user representation.
- **Edges** $X_n \rightarrow X$: the scenario-invariant representation is extracted from all scenario-specific representations. It is hard to model X explicitly since it is usually unobserved.
- **Edge** $X_i \rightarrow \hat{Y}_j, i \neq j$: the scenario-specific representation directly affects the predictions in other scenarios. Although sharing behaviors from other scenarios is helpful for boosting the performance in the current scenario, there may exist opposite feedbacks in different scenarios which will lead to conflicts and have negative effect on the estimator.
- **Edge** $D_i \rightarrow X_i \rightarrow \hat{Y}_j, i \neq j$: the scenario-aware information acts as a confounder, which has both the effect on user representation in the current scenario and further impact on the predictions of



(a) Cutoff $D_n \rightarrow X_n$. (b) Cutoff $X_i \rightarrow Y_j$.

Figure 2: Interventions on causal graph of MSM from the perspective of modelling.

the other scenarios through scenarios-specific representations. Such causal path causes uncontrollable effect on the prediction scores, which may lead bias towards the other scenarios.

From Section 4.1, we can find that MSM introduces several edges which will bring uncontrollable biases and conflicts thus impeding the performance of the existing estimators. To exploit the strengths and avoid the weaknesses of MSM, we need cut off these unnecessary causal paths. As shown in Figure 2(a) and 2(b), to eliminate these negative effects and bridge the gap between the predicted scores $\hat{Y}_n$ and interaction labels Y_n, we need to cut off the edges $D_i \rightarrow X_i$ and $X_i \rightarrow \hat{Y}_j, i \neq j$, respectively.

5 OVERALL FRAMEWORK

To cut off the unnecessary edges in the causal graph from modeling perspective, we propose CausalInt, which is shown in Figure 3. CausalInt consists of three modules: (1) Invariant Representation Modeling Module to learn a scenario-invariant representation to be shared across scenarios (blue area in the left of Figure 3); (2) Negative Effects Mitigating Module to eliminate negative transfers among scenarios and retain specialities in target scenario (green area in the right of Figure 3); (3) Inter-Scenarios Transferring Module to fuse transferable information from other scenarios to promote the performance in target scenario (yellow area in the middle of Figure 3). All modules are trained jointly in an end-to-end manner. In this section, we will give detailed description of these modules.

5.1 Scenario-invariant Representation Modeling

The general scenario-invariant representation X is shared across scenarios and should be unbiased to any specific scenario. However, as analyzed in Section 4, the scenario-aware information D_n acts as an confounder and has an effect on X indirectly($D_n \rightarrow X_n \rightarrow X$). It introduces biases into the estimation of other scenarios through X and degrades the accuracy of unified model.

To extract scenario-invariant representation and squeeze out the D_n (cutoff $D_n \rightarrow X_n$), we conduct disentangled representation learning method. Specifically, as shown in the blue area in Figure 3, we decompose the input embedding into two independent representations:

$$X_{IR} = Leaky\_ReLU(f_{\theta_1}(E)), \tag{1}$$

$$X_{SR} = Leaky\_ReLU(f_{\theta_2}(E)), \tag{2}$$

where E is the embedding of input features. X_{IR} is *Invariant Representation (IR)*, X_{SR} is *Scenarios Representation (SR)*, f_{θ_1} and f_{θ_2} are feature extractors of two representations parameterized by θ_1 and θ_2, respectively. To obtain X_{IR}, we decompose the problem into four tasks:

- **Task 1. Modeling with X_{IR}**: To realize scenario-invariant, Since X_{IR} is shared across scenarios, it is expected to be predictive for interactions from all scenarios. therefore, we build a prediction task for overall scenarios based on the X_{IR}:

$$\hat{y}^d = Sigmoid(g_{\phi_1}(X^d_{IR})), \tag{3}$$

$$Loss_1 = \sum_{d=1}^{N} -y^d log(\hat{y}^d) - (1-y^d) log(1-\hat{y}^d), \tag{4}$$

  where X^d_{IR} is X_{IR} filtered for scenario d. g_{ϕ_1} is the general classifier parameterized by ϕ_1. Y^d is the label of a sample in scenario d.
- **Task 2. Modeling with X_{SR}**: SR is designed to extract scenario-aware information from the shared embedding layer. To achieve this, we introduce the scenario indicator as the supervision to update X_{SR}:

$$\boldsymbol{\alpha} = Softmax(g_{\phi_2}(X_{SR})), \tag{5}$$

$$Loss_2 = \sum_{d=1}^{N} -\boldsymbol{d} log(\boldsymbol{\alpha}), \tag{6}$$

  where g_{ϕ_2} is scenario classifier parameterized by ϕ_2, $\boldsymbol{d}$ is scenario indicator with one-hot encoding.
- **Task 3. Modeling with combination of X_{IR} and X_{SR}**: Since X_{IR} is the representation that squeezes out the scenario biases, and X_{SR} extracts the scenario-aware information, the combination of these two representations is expected to be predictive for the label y^d:

$$\hat{y}^d_{concat} = Sigmoid(g_{\phi_3}([X^d_{IR}, X^d_{SR}])), \tag{7}$$

$$Loss_3 = \sum_{d=1}^{N} -y^d log(\hat{y}^d_{concat}) - (1-y^d) log(1-\hat{y}^d_{concat}), \tag{8}$$

  where g_{ϕ_3} is general classifier for concatenated representation parameterized by ϕ_3. $[\cdot,\cdot]$ means concatenation of two vectors.
- **Task 4. Decomposing X_{IR} and X_{SR}**: To effectively decompose the embedding into distinct representations, we apply gradient-based orthogonal regularization. The intuition behind this is that, moving locally along the direction of $\pm \bigtriangledown f_i(x;w)$ leads to the biggest change in model prediction $f_i(x;w)$, while moving orthogonal to $\bigtriangledown f_i(x;w)$ leads to the least change to the prediction of x [4, 14]. Specifically, we enforce the gradients of the $Loss_1$ in Eq (4) to be orthogonal to that of $Loss_2$ in Eq (6) with respect to the shared embedding layer E.

$$Loss_{orth} = \sum_{d=1}^{N} L2\_norm\left(\frac{\bigtriangledown_E Loss_1}{\| \bigtriangledown_E Loss_1 \|} \cdot \frac{\bigtriangledown_E Loss_2}{\| \bigtriangledown_E Loss_2 \|}\right). \tag{9}$$

Then, four tasks will be jointly trained to obtain the disentangled representations where the scenario-invariant representation is extracted, and the scenario-aware information is separated out to a distinct representation.

5.2 Negative Effects Mitigating

After obtaining the scenario-invariant representation , we build the scenario-specific tower with informative scenario-aware representation S^d based on the X_{IR} to capture the characteristics of different scenarios:

$$\hat{y}^d = Sigmoid(g_{\phi^d}(h_{\psi^d}(S^d, X_{IR}))), \tag{10}$$

$$Loss^d = -y^d log(\hat{y}^d) - (1-y^d) log(1-\hat{y}^d), \tag{11}$$

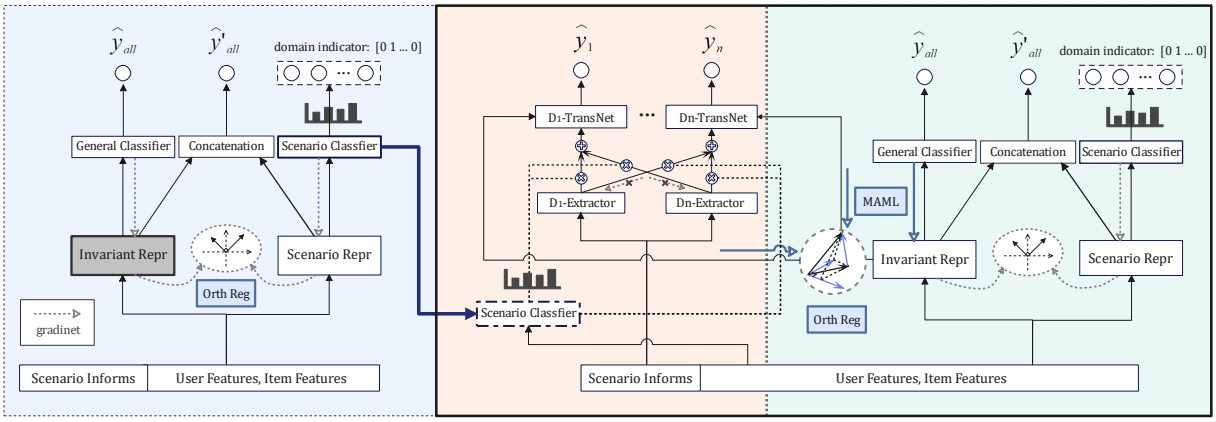


Figure 3: The overall framework of CausalInt. The right part in black solid box is the architecture of CausalInt. Left part(blue) is Invariant Representation Modeling Module, which is presented separately for convenience to display. Specifically, scenario classifier in Invariant Representation Modeling Module is directly adapt to TransNet acting as an attention module.

where h_{ψ^d} is the function to fuse scenario-aware representation S^d and scenario-invariant representation X_{IR}, it can be neural network, or parameter-free methods, such as concatenation or multiplication. g_{ϕ^d} is scenario-specific classifier parameterized by ϕ^d. The specific implementation of h^d_ψ will be detailed in Sec 5.3. As analyzed in Sec 4, scenario-specific representation X_n directly affect the predictions of other scenarios, which may result in conflicts since the same behaviors may have opposite feedback in different scenarios. To eliminate the conflicts (cutoff $X_i \rightarrow Y_j$), we first take a deep insight of the causes of conflicts. As shown in Eq (10), X_{IR} is shared across all scenarios, the gradient of $Loss^d$ from scenario d_i may be in opposite direction to that of scenario d_j with respect to X_{IR}. Besides, conflict may also exist between the $Loss_1$ and $Loss^d$ since the X_{IR} and the parameters of networks in Eq (10) are jointly trained.

5.2.1 Conflict among $Loss^d$, $d \in \{1, 2, \cdots, N\}$. Similar with the method in Task 4 in Sec 5.1, we conduct orthogonal regularization on the gradients of $Loss^d$ with respect to X_{IR}. However, cosine distance is no longer applicable to the current situation since the number of scenarios usually exceeds two. The gradient of $Loss^d$ from one scenario should be perpendicular to the *space* of all gradients from other scenarios. To solve this problem, we utilize the Gram-Schmidt procedure to compute the orthogonal basis for gradients of all scenarios:

$$\begin{aligned} b^1 &= \nabla_{X_{IR}} Loss^1, \\ b^i &= \nabla_{X_{IR}} Loss^i - \sum_{j<i} proj_{b_j}(\nabla_{X_{IR}} Loss^i) \end{aligned} \tag{12}$$

where $proj_m(n) = \frac{\langle n,m \rangle}{\langle m,m \rangle} m$ is the projection of n in the direction of m. Given the orthogonal basis $B = \{b^1, b^2, \cdots, b^N\}$, we regularize the distance between the basis and the corresponding original gradient $g^i = \nabla_{X_{IR}} Loss^i$.

$$Loss^{all}_{orth} = \sum_{i=1}^{N} L2\_norm(g^i - b^i) \tag{13}$$

5.2.2 Conflict between $Loss_1$ and $Loss^d$. Since X_{IR} is the basis of scenario-specific tower, it may be unreasonable to treat the gradient of $Loss_1$ and the gradient of $Loss^d$ as equal and utilize orthogonal regularization as before. We further transform this problem into a meta-learning problem with the intuition that X_{IR} is transferable to all scenarios. Inspired by MAML [5], we regard the training with $Loss_1$ as the initialization process of the training with $Loss^d$. Namely, we train model's initial parameters of $Loss^d$ after updating the shared parameters one step via gradients computed from $Loss_1$. The training procedure is:

$$X'_{IR} = X_{IR} - \eta \nabla_{X_{IR}} Loss_1 \tag{14}$$

We firstly update X_{IR} once with $Loss_1$ and obtain the updated representation X'_{IR}, η is the step size (learning rate). Then we substitute X'_{IR} into Eq (10) and obtain the updated prediction:

$$\hat{y}'^d = Sigmoid(g_{\phi^d}(h_{\psi^d}(S^d, X'_{IR}))) \tag{15}$$

We further evaluate the updated model on the scenario-specific task as Eq (11)

$$Loss'^d = -y^d log(\hat{y}'^d) - (1 - y^d) log(1 - \hat{y}'^d) \tag{16}$$

Finally, we minimize the $Loss'^d$ to update the X_{IR}:

$$X_{IR} = X_{IR} - \eta \nabla_{X_{IR}} Loss'^d \tag{17}$$

5.3 Inter-Scenario Transferring

The first two modules cutoff two causal paths and eliminate biases in the multi-scenario modeling process and restore the causal graph from Fig 1(b) to Fig 1(a). However, some biases still exist in the training data. We propose a counterfactual question: what the feedback would be if an user-item interaction that originally happens in scenario i occurs in scenario j ? It is infeasible to conduct such interventional experiment in practice. In this section, we simulate this intervention through a *TransNet*. Specifically, the scenario-aware representation S^p is first extracted:

$$H^p = Leaky\_ReLU(f_{\theta_p}(S^p)) \tag{18}$$

where S^p is the embedding for scenario-aware information, f_{θ_p} is feature extractor with scenario-specific parameter θ_p. Then we reshape the $H^p \in \mathcal{R}^{K^2}$ into a matrix $H'^p \in \mathcal{R}^{K \times K}$ and make it as the parameter of the TransNet, which takes the input as scenario-invariant representation X^d_{IR}. Correspondingly, the X^d_{IR} is also be

adapted for matrix multiplication via $W_2^d \in \mathcal{R}^{M\times K}$, where M is the size of X_{IR}^d:

$$\begin{aligned} V^d &= Leaky\_ReLU(X_{IR}^d W_2^d), \\ \hat{y}^{d,p} &= Sigmoid(V^d H'^p), \quad d,p \in \{1,2,\cdots,N\} \end{aligned} \tag{19}$$

where $\hat{y}^{d,p}$ is the simulated prediction of user clicking on the item in scenario p, where the user-item interactions actually happened in scenario d. Since the ground-truth cannot be obtained from scenario p, we have to use the feedback for this interaction in scenario d to supervise the training process, and regard the $\hat{y}^{d,p}$ as a vote from a expert in scenario p for interactions in scenario d. To effectively assemble votes from different experts, we reuse the scenario classifier learned in Eq (5) to justify the relevance between scenarios. It is worth noting that, directly assemble $\hat{y}^{d,p}$ from different experts may result in bias to scenario with high click rate. As a result, we adjust the simulation process as follows:

$$\begin{aligned} H^{d,p} &= Leaky\_ReLU(f_{\theta_p}(S^d)), \\ Expert^{d,p} &= \begin{cases} H^{d,p} & \text{if } d = p, \\ stop\_gradient(H^{d,p}) & \text{if } d \neq p \end{cases} \\ TransNet^d &= \sum_{o=1}^{N} \boldsymbol{\alpha}_o Expert^{d,o} \end{aligned} \tag{20}$$

where $H^{d,p}$ is extractor of scenario p which takes the input as S^d from scenario d. To avoid uncontrollable biases, we conduct $stop\_gradient$ operation to prevent the gradients of scenario d from passing into extractor of scenario p. $\boldsymbol{\alpha}$ is the scenario classifier in Eq (5) and acts as the gating weight to selectively aggregate the information of all experts. Finally, we conduct the simulated intervention and further generate the estimation through a scenario-specific classifier.

$$\hat{y}^d = Sigmoid(g_{\phi^d}(X'_{IR} TransNet^d)), \tag{21}$$

where the X'_{IR} is the scenario-invariant representation in Eq (14) (also represents the user-item interactions), g_{ϕ^d} is the classifier in scenario d parameterized by ϕ^d. Finally, all modules in CausalInt jointly trained with the following objective:

$$Loss = Loss_1 + Loss_2 + Loss_3 + Loss^d + Loss_{orth} + Loss_{orth}^{all}. \tag{22}$$

6 EXPERIMENTS

In this section, we conduct offline experiments to evaluate the proposed CausalInt and answer the following questions:

- **RQ1**: How does the CausalInt perform compared with the baseline models?
- **RQ2**: What factors will affect the performance of multi-scenario modeling?
- **RQ3**: How about the impact from different modules of CausalInt?

6.1 Experimental Setting

6.1.1 Datasets and Evaluation Protocols. We conduct experiments on both public available dataset and industrial dataset. The descriptions and statistics of two dataset are detailed in A.1

In the offline experiments, we apply the most commonly-used **AUC**(Area Under the ROC) [3, 6, 13] and **RelaImpr** [12, 18] to evaluate the performance of the proposed model and the competitors.

6.1.2 Baselines and Hyper-parameters. To demonstrate the effectiveness of our proposed model, we compare CausalInt with three typical kinds models mentioned in Table 1: Mixing, Finetune, Shared Bottom, PLE and STAR. The details of these models and the hyper-parameters are introduced in A.2.

6.2 Performance Comparison (RQ1 & RQ2)

The offline comparison on Ali-CCP and Industrial datasets between CausalInt and baseline models are shown in Table 2 and Table 3, respectively. We summarize the observations as following:

- CausalInt consistently outperforms various kinds of state-of-the-art models over all datasets by a significant margin, which demonstrates the effectiveness of CausalInt in multi-scenario recommendation task. Compared with the Single model, CausalInt utilizes samples from all scenarios which can enrich user behaviors so as to better learn user preference. Compared with the Mix, CausalInt employs the mixed data in a more elegant way which eliminates negative impacts among scenarios and retains the specialities of each scenario. Compared with Finetune, we jointly train different scenarios and alleviate the conflict hidden in transferred parameters, hence avoiding the *catastrophic forgetting* [7]. Compared with the multi-task models, CausalInt explicitly model the scenario information, which can better capture the distinct distributions between different scenarios. Compared with the STAR, CausalInt makes fine-grained design for commonalities extraction and specialities exploitation with consideration of negative effects mitigating.
- MSM can boost the performance of models on different scenarios, especially the scenarios with rare training samples. We can see, in scenario #3 of Ali-CCP dataset, scenario #2&#3&#4 in Industrial dataset, all MSM models achieves better performance than the Single model. The behind reason is that Single model on scenarios with rare samples tend to overfit. After introducing enough training sample from other scenarios, the performance is improved.
- Ignoring distinctions among different scenarios may result in sub-optimal performance in the scenario with enough data. This observation is from the fact that Mix model performs worse than the Single model in scenarios #1 and #2 on Ali-CCP dataset. After considering the distinctions of different scenarios, the performance will be improved. For example, in almost case, Finetune model achieves better performance than Mix model since it tunes all parameters for each target scenario with corresponding samples.
- Selectively aggregating information from other scenarios and explicitly exploiting the scenario information are critical for MSM. Compared with other MSM models, the STAR achieves better performance in almost scenarios since it explicitly models the scenario-aware information and considers different normalization for different scenarios. Furthermore, the proposed CausalInt obtains the best performance over all datasets since it is able to build an efficient aggregation module to exploit information from other scenarios. Besides, it cutoffs the unnecessary relations in modeling process and mitigates negative effects among different scenarios.

Table 2: The overall performance over Ali-CCP dataset. Boldface denotes the highest score and underline indicates the best result of the baselines. ⋆ represents significance level p-value < 0.05.

	Single		Mix		Shared Bottom		PLE		Finetune		STAR		CausalInt	
Scenarios	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp
#1	0.6060	-	0.6036	-0.41%	0.6100	0.65%	0.6099	0.63%	0.6091	0.50%	0.6134	1.21%	**0.6179⋆**	**1.96%**
#2	0.6136	-	0.6083	-0.87%	0.6149	0.21%	0.6160	0.39%	0.6146	0.16%	0.6176	0.65%	**0.6231⋆**	**1.55%**
#3	0.5702	-	0.5838	2.38%	0.5716	0.24%	0.5720	0.32%	0.5731	0.50%	0.5817	2.01%	**0.5996⋆**	**5.16%**

Table 3: The overall performance over Industrial dataset.

	Single		Mix		Shared Bottom		PLE		Finetune		STAR		CausalInt	
Scenarios	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp
#1	0.7474	-	0.7496	0.29%	0.7462	-0.16%	0.7495	0.28%	0.7510	0.48%	0.7502	0.38%	**0.7511⋆**	**0.50%**
#2	0.7853	-	0.7904	0.65%	0.7876	0.30%	0.7916	0.80%	0.7904	0.65%	0.7925	0.91%	**0.7930⋆**	**0.98%**
#3	0.7029	-	0.7084	0.79%	0.7054	0.37%	0.7103	1.06%	0.7097	0.98%	0.7074	0.65%	**0.7115⋆**	**1.22%**
#4	0.7073	-	0.7387	4.44%	0.7318	3.46%	0.7368	4.17%	0.7382	4.37%	0.7369	4.18%	**0.7387⋆**	**4.44%**
#5	0.6816	-	0.6887	1.04%	0.6835	0.28%	0.6882	0.97%	0.6896	1.18%	0.6885	1.01%	**0.6896⋆**	**1.18%**

Table 4: Ablation study on different module of CausalInt.

	Shared Bottom		+Do(Invar)		+Do(De-Neg)		+Do(Trans)	
Scenarios	AUC	RelImp	AUC	RelImp	AUC	RelImp	AUC	RelImp
#1	0.6149	-	0.6195	0.74%	0.6219	1.13%	0.6231	1.33%
#2	0.6100	-	0.6148	0.80%	0.6159	0.98%	0.6179	1.30%
#3	0.5716	-	0.5933	3.80%	0.5987	4.75%	0.5996	4.91%

- Multi-task based models achieve relative balanced result in comparison with the Mix model. Specifically, multi-task models outperforms the Single models in all scenarios except in scenario #1 over industrial dataset (Shared Bottom). This demonstrates that multi-task models can also be applied to multi-scenario recommendation with the help of sharing mechanism to improve generalization. PLE further separates shared components and task-specific components explicitly and applies a progressive routing mechanism to extract and separate deeper semantic knowledge gradually, which shows its superiority even in the multi-scenario recommendation.

6.3 Ablation Study (RQ3)

To verify the effectiveness of each module in CausalInt, we conduct a series of ablation studies over the public dataset. We take Shared Bottom model as the backbone of our proposed model, for that the Shared Bottom model is a common and easy-to-apply multi-task model in industrial, and it is also the backbone of many existing multi-scenario models. All modules are incrementally applied to the Shared Bottom model, and the cumulative effects of each module are reported in Table 4. From which, consistent improvement can be observed in each domain. Analysis of each module will be elaborated in the following section.

6.3.1 Scenario-invariant Representation Modeling. The Scenario-invariant Representation Modeling (SRM) module improves the performance through learning an unbiased representation to be shared across all scenarios, which remove the domain-specific bias, and as a result, capture the generality of different scenarios. Specifically, four components can account for the improvement, Table 5 shows the result of models trained with or without different components. Specifically, 1) **w/o orth** denotes the removal of orthogonal regularization between the general invariant representation and scenario representation. 2) **w/o invar** means we remove the learning of $Loss_1$ in Eq (4). 3) **w/o classifier** indicates leaving out the learning of classifier with $Loss_2$ in Eq (6). 4) **w/o concat** denotes SRM irrespective of the the $Loss_3$ for concatenation in Eq (8). 5) **SRM** means apply all the above components. We observe that, significant improvement still can be achieved in each scenario compared with the backbone (Shared Bottom) even if we only apply parts of this module to the backbone model. On the other side, degradation appears when removing the corresponding component from SRM, which means all components have consistent contribution to this module.

Table 5: Ablation study on Scenario-invariant Representation Modeling Model, w/o means remove corresponding component.

	scenario#1	scenario#2	scenario#3
w/o orth	0.6182	0.6137	0.5798
w/o invar	0.6187	0.6140	0.5854
w/o classifier	0.6179	0.6129	0.5871
w/o concat	0.6192	0.6134	0.5902
SRM	**0.6195**	**0.6148**	**0.5933**

Table 6: Ablation study on Negative Effects Mitigating module, two components are integrated into Module1 additively.

	scenario#1	scenario#2	scenario#3
SRM	0.6195	0.6148	0.5933
SRM+OR	0.6200	0.6155	0.5973
SRM+OR+MAML	**0.6219**	**0.6159**	**0.5987**

6.3.2 Negative Effects Mitigating. Negative effect Mitigating Module promote the performance of CausalInt via two gradient-based operations: (1) mitigating negative effect among scenarios through Grim-Schmidt orthogonal regularization(OR); (2) alleviating negative effect between scenario-specific and scenario-invariant parameters with gradient-based meta-learning method (MAML). We stack the two gradient-based operations incrementally to verify the effectiveness of this module. The results are shown in Table 6. Consistent improvements can be observed when the operation is appended to SRM one by one. This indirectly reflects the fact that there are conflicts between different scenarios.

6.3.3 Inter-Scenarios Transferring. Inter-Scenarios Transferring module contributes to the improvement by involving transferable information from other scenarios. We divide this module into two

Table 7: Ablation study on Inter-Scenarios Transferring Module, CausalInt-Gate denotes replacing existing scenario classifier in Module1 with an end-to-end gate network, CausalInt-Con means replacing TransNet with concatenation, CausalInt-Mul denotes replacing TransNet with element-wise multiplication.

	scenario#1	scenario#2	scenario#3
CausalInt-Gate	0.6215	0.6168	0.5991
CausalInt-Con	0.6206	0.6145	0.5995
CausalInt-Mul	0.6192	0.6165	0.5991
CausalInt	**0.6231**	**0.6179**	**0.5996**

parts and make detailed analysis respectively. Firstly, we conduct experiment to analyze the influence of different gating mechanisms. Specifically, we replace the gate in Eq (20) with an end-to-end trained gating layer just like the gate mechanism in existing works [8][11]. **CausalInt-Gate** in Table 7 records the result of this experiment, comparing with the CausalInt, we found that directly utilizing the scenario classifier in SRM as gating weight for aggregating scenario information outperforms the one trained from scratch with an end-to-end manner. We think the superiority comes from two aspects. On the one hand, the scenario classifier is explicitly supervised with scenario indicators which makes it easier to learn the distribution among different scenarios. On the other hand, the predicted attention score will gradually converge to the actual scenario with the convergence of the scenario classifier. Besides, we also compare different interaction functions between the scenario-aware information and the shared scenario-invariant representation. Concretely, we replace the TransNet with commonly-used method i.e. concatenation(denoted as **CausalInt-Con**) and multiplication(denoted as **CausalInt-Mul**). Results in Table 7 indicates that TransNet has significant superiority than the other two methods. Essentially, TransNet is a matrix multiplication with aggregated scenario information as the multiplicand, which has better interaction capacity.

6.4 Model Complexity

Since the scalability is important for industrial estimators, we compare the model parameters, training time and inference time(over the whole test set of all scenarios) of different models. All experiments are conducted on an NVIDIA Tesla P100-PCIE GPU with 16G memory. Table 9 reports the comparison results over Ali-CCP dataset. We can obeserve that, compared with the Single model, the increased model parameters of each model are negligible, this is because the most parameters come from the embedding table. The incresed training time and inference time of CausalInt are also acceptable, demonstrating that our proposed model is feasible in practical industrial applications. Notably, training time of the Finetune is almost doubled due to the fact that it requires two stages of training.

Table 8: Online A/B testing results of CausalInt compared with the base models: Single and Finetune.

	CausalInt v.s. Single				CausalInt v.s. Fintune			
Metric	CTR	eCPM	Training	Inference	CTR	eCPM	Training	Inference
RelaImp	**+3.73%**	**+2.82%**	+28.18%	+1.76%	**+0.94%**	**+1.75%**	-26.45%	+0.98%

Table 9: Time and space complexity comparison on the Ali-CCP dataset.

Model	Params($\times 10^7$)	Rel ratio	Training time	Rel ratio	Inference time	Rel ratio
Single*	1.85	-	211.77	-	144.98	-
Finetune	1.85	0.00%	578.24	173.05%	145.14	0.11%
Shared Bottom	1.87	1.08%	246.89	16.58%	174.98	20.69%
PLE	1.88	1.41%	253.22	19.57%	198.79	37.11%
STAR	1.89	2.06%	298.57	40.99%	177.53	22.45%
CausalInt	1.88	1.70%	292.43	38.09%	182.06	25.57%

* For fair comparison, the training time and inference time of the Single model are summation of three scenarios.

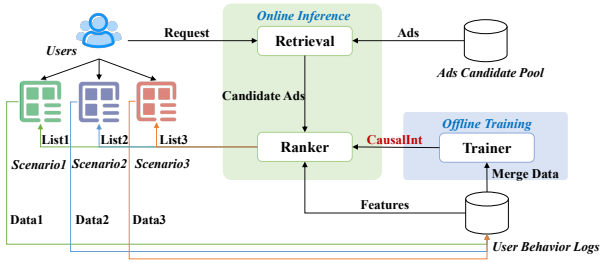


Figure 4: Overview of the Huawei online advertising platform.

7 ONLINE A/B TESTING

To evaluate the performance of CausalInt in real product environment, we conduct online A/B test in Huawei online advertising platform for consecutive two weeks.

7.1 System Overview & Scenario Description

In this section, we first briefly present the overview of the Huawei online advertising platform, which is shown in Figure 4. The platform consists of two core components: online inference module and offline training module. When a user accesses any scenarios of the Huawei advertising platform, a request with the user's attributes and contextual features is send to the online service. Then the *Retrieval* is triggered and retrieves candidate ads from a candidate pool for the *Ranker*. Afterward, *Ranker* leverages the features extracted from the historical logs and the model trained periodically by a offline *Trainer* to predict scores. Finally, the candidate ads are sorted according to a pre-defined ranking function (*e.g.*, eCPM) and the top-k ads will be inserted into some pre-determined positions for presentation. For the multi-scenario modeling, user's behavior logs across multiple scenarios will be merged before providing for model training, which is different from the single-scenario modeling. It is noteworthy that CausalInt leverages the merged data to periodically train a *unified model*, which is pushed for online serving in different scenarios. By contrast, some multi-scenario modeling methods (*e.g.*, Finetune) will further adjust the model with samples from the target scenario, resulting in *multiple models* for serving.

We deploy the CausalInt on five scenarios in Huawei online advertising, where millions of daily active users are involved and tens of millions of log events are generated. Specifically, the five scenarios consist of Browser, News Feeds, Video Page, Video Feeds and Video AppStore, which are depicted in Figure 5. These scenarios contain different display styles, including single ad card (*e.g.*, Browser) and ad list (*e.g.*, Video AppStore). Besides, the advertising contents are diverse, such as goods, services and applications. The same ads can be displayed in different scenarios with distinct materials (*e.g.*, pictures and copywriting).

Figure 5: The description of the application scenarios.

7.2 Online Experimental Results

The online A/B test is conducted for two weeks and we compare CausalInt with two methods: the Single model and Finetune model. Single model is trained over the scenario-specific data while Finetune model and CausalInt are trained over the multi-scenario merged data. Each model is trained in a single cluster, where each node contains 16 core Xeon(R) Gold 6278C CPU (2.60GHz), 32GB RAM as well as 1 NVIDIA TESLA T4 GPU cards. For online serving, all three groups of experiments are assigned with 2% random users. Similar to [2], two commonly-used online evaluation metrics in online advertising are used to evaluate the performance of CausalInt: Click Through Rate (**CTR**) and Effective Cost Per Mile (**eCPM**).

We report the overall performance of the deployed models over 5 scenarios, which is the weighted sum of CTR/eCPM according to the revenue capacity. Table 8 reports the relative improvements over two baselines: Single and Finetune. From the results, we can observe that CausalInt consistently outperforms the baseline models. Compared with the Single and Finetune, CausalInt achieves 3.72% (2.74%) and 0.94% (1.69%) improvements with respect to CTR (eCPM) respectively. For efficiency comparison, the training time of the Single is much smaller than CausalInt and Finetune due to the less training data. However, the training time of CausalInt reduces 26.45% compared with Finetune, showing the high efficiency of CausalInt. Moreover, the inference time of CausalInt is comparable with both methods, making CausalInt more practical in real-world recommendation scenarios. Besides, CausalInt serves all scenarios with a single model, which is much more efficient in manpower and resource consumption.

8 CONCLUSION

In this paper, we first summarize three key challenges in multi-scenario modeling with causal graph from perspectives of user and modeling process. Then, we propose the CausalInt to overcome the challenges in multi-scenario recommendation inspired by the causal intervention. CausalInt consists of three modules: (1) Scenario-invariant Representation Modeling module, it extracts commonalities of different scenarios through disentangled representation learning, which squeezes out the scenario-aware information and obtains a scenario-invariant representation shared across different scenarios; (2) Negative Effect Mitigating module, which retains specialities of the target scenario and mitigating negative transfer via two gradient based methods, namely orthogonal regularization and MAML; (3) Inter-Scenario Transferring module, which selectively aggregates information from other scenarios and exploits the scenario-aware information explicitly. Evaluation on both offline experiments and online A/B test demonstrate the effectiveness of our proposed model. CausalInt has also been deployed in an online advertising platform and serving millions of daily active users.

REFERENCES

[1] Rich Caruana. 1997. Multitask learning. *Machine learning* (1997).
[2] Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. 2021. Enhancing Explicit and Implicit Feature Interactions via Information Sharing for Parallel Deep CTR Models. In *Proc. of CIKM.*
[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems.*
[4] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *AISTATS.*
[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of ICML.*
[6] 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).
[7] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* (2017).
[8] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proc. of KDD.*
[9] Judea Pearl. 2009. *Causality.* Cambridge university press.
[10] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proc. of CIKM.*
[11] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).
[12] 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.*
[13] 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.*
[14] Mihai Suteu and Yike Guo. 2019. Regularizing deep multi-task networks using orthogonal gradients. *arXiv preprint arXiv:1912.06844* (2019).
[15] 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 *RecSys.*
[16] Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* (2018).
[17] Yichao Wang, Huifeng Guo, Ruiming Tang, Zhirong Liu, and Xiuqiang He. 2020. A Practical Incremental Method to Train Deep CTR Models. *arXiv preprint arXiv:2009.02147* (2020).
[18] Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the Sequential Dependence among Audience Multi-step Conversions with Multi-task Learning in Targeted Display Advertising. *arXiv preprint arXiv:2105.08489* (2021).
[19] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proc. of SIGIR.* 11–20.
[20] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proc. of KDD.*

A EXPERIMENTAL SETTINGS

A.1 Datasets

- **Ali-CCP**[1]. This dataset is collected from the traffic logs of Taobao. Each log is composed of 13 user features, 5 item features, 4 combined features, and two labels (*click* and *purchase*). In this paper, we only use the *click* as the target label. Besides, one context feature is provided, which is used to divide logs into three scenarios.
- **Industrial**. The industrial dataset is sampled from the click logs of 5 scenarios in Huawei advertising platform over 9 days. Different scenarios share the overlapping user and item space while the different user and item group are also kept. There are 31 features in this dataset including user features, item features, context features and some scenario-specific features. We set day 1-7 as training set, day 8 and day 9 as validation and test set, respectively.

The statistics of the two datasets are shown in Table 10

Table 10: The percentage of instances and average click-through rate (CTR) of each scenario.

Dataset	Ali-CCP			Industrial				
Scenario	#1	#2	#3	#1	#2	#3	#4	#5
Percentage	61.46%	37.79%	0.75%	33.60%	14.08%	12.74%	6.79%	32.79%
CTR	3.81%	4.00%	4.38%	1.10%	1.55%	4.57%	1.64%	4.08%

A.2 Baselines and Hyper-parameters

To demonstrate the effectiveness of our proposed model, we compare CausalInt with the following models:

- **Single**. The model is trained only with samples from the target scenario. Specifically, three-layer fully-connected networks are applied for the experiment on Ali-CCP dataset, the state-of-the-art EDCN [2] model is used on industrial dataset.
- **Mix**. We refer the Mix as the model trained with mixture of samples from all scenarios. The model structure is the same as the Single.
- **Finetune**. Finetune is a commonly-used and effective domain adaption (DA) training manner in industrial recommendation system[17]. It firstly trains an unified model with the mixture of samples from all scenarios(namely the Mix), then adjusts the unified model with the data of target scenario.
- **Shared Bottom**. The Shared Bottom model is widely-used multi-task model which shares parameters of the bottom layers. Specifically, we take the embedding layer as the shared part and build a specific three-layer fully-connected networks for each scenario on the shared part.
- **PLE**. The PLE[15] is a state-of-the art multi-task model. Compared with Shared Bottom model, it separates shared components and task-specific components explicitly and applies a progressive routing mechanism to extract and separate deeper semantic knowledge gradually.
- **STAR**. The STAR[13] is a state-of-the-art multi-domain model which consists of two factorized networks: one centered network shared by all domains and the domain-specific network tailored for each domain.

We use adam as the optimizer for all models with learning rate of 1e-3 and set the batch size as 6000. The embedding size is set to 20. The hidden layers of deep network are fixed to [256,128,64]. Besides, Batch Normalization is applied. The weight of $L2$ regularization is tuned from [1e-1, 1e-2, 1e-3, 1e-4, 1e-5] and dropout rate is searched from [0.1,0.2,...,0.9]. For PLE, we set 1 specific expert for each scenario.

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=408