Multi-Scenario Ranking with Adaptive Feature Learning

Yu Tian
s.braylon1002@gmail.com
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University
China

Bofang Li
bofang.lbf@alibaba-inc.com
Alibaba Group
China

Si Chen
yasui.cs@alibaba-inc.com
Alibaba Group
China

Xubin Li
lxb204722@alibaba-inc.com
Alibaba Group
China

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

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

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

Qian Wang
qianwang@whu.edu.cn
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University
China

Chenliang Li*
cllee@whu.edu.cn
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University
China

ABSTRACT

Recently, Multi-Scenario Learning (MSL) is widely used in recommendation and retrieval systems in the industry because it facilitates transfer learning from different scenarios, mitigating data sparsity and reducing maintenance cost. These efforts produce different MSL paradigms by searching more optimal network structure, such as Auxiliary Network, Expert Network, and Multi-Tower Network. It is intuitive that different scenarios could hold their specific characteristics, activating the user's intents quite differently. In other words, different kinds of auxiliary features would bear varying importance under different scenarios. With more discriminative feature representations refined in a scenario-aware manner, better ranking performance could be easily obtained without expensive search for the optimal network structure. Unfortunately, this simple idea is mainly overlooked but much desired in real-world systems.

To this end, in this paper, we propose a **m**ulti-scen**a**rio **r**anking framework with adapt**i**ve fe**a**ture learning (named Maria). Specifically, Maria is devised to inject the scenario semantics in the bottom part of the network to derive more discriminative feature representations. There are three components designed in Maria for this purpose: *feature scaling*, *feature refinement*, and *feature correlation modeling*. The purpose of feature scaling is to highlight the scenario-relevant fields and also suppress the irrelevant ones. Then, the feature refinement utilizes an automatic refiner selection subnetwork for each feature field, such that the high-level semantics with respect to the scenario can be extracted with the optimal expert. Afterwards, we further explicitly derive the feature correlations across fields as complementary signals. The resultant representations are then fed into a simple MoE structure with an additional scenario-shared tower for final prediction. Experiments on two large-scale real-world datasets demonstrate the superiority of Maria against several state-of-the-art baselines for both product search and recommendation. Further analysis also validates the rationality of adaptive feature learning under a multi-scenario scheme. Moreover, our A/B test results on the Alibaba search advertising platform also demonstrate that Maria is superior in production environments.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; **Recommender systems**.

KEYWORDS

Multi-scenario Learning, Feature Refinement, Search and Recommendation

ACM Reference Format:
Yu Tian, Bofang Li, Si Chen, Xubin Li, Hongbo Deng, Jian Xu, Bo Zheng, Qian Wang, and Chenliang Li. 2023. Multi-Scenario Ranking with Adaptive Feature Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR*

*Corresponding author. Work done when Yu Tian was an intern at Alibaba.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591736

'23), July 23–27, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591736

1 INTRODUCTION

With the rapid development of the Internet, the online business of a company is becoming more diverse and complex. Many different yet potentially relevant scenarios are developed to support various means of information seeking and cover the user's multiple intents. Figure 1 illustrates three representative business scenarios in Taobao Application: 1) Visual Search (VS): in addition to keyword search, a user can take a photo containing the desired product for retrieval; 2) Similar Search (SS): a user plans to retrieve more products that are highly similar to the target product. This time, the user's intention would be relatively more complex, such as price comparison and searching for products in the same style; 3) Interest Search (IS): This scenario is different from the above scenarios. The advertisement (ad) as an interest trigger is mainly located in the external media. When a user clicks the ads on an external media, he/she is relocated to the mobile Taobao landing page. Unlike Similar Search, the query about Interest Search combines the ads with user-interested products of more diversity at the recall stage.

Traditional methods [2, 4, 7, 12–14, 17, 22, 26–28] serve each scenario with their own data separately, each with a particularly tailored model. However, there are two defects that become more and more worrisome with this strategy: (1) some scenarios have a small amount of training data, leading to inferior performance; (2) these scenario-dependent solutions need to be optimized parallelly. These things could become troublesome when upgrading and maintaining in the future. Hence, Multi-Scenario Learning (MSL) emerges as the times require.

Recently, many methods are proposed to improve the performance of MSL. These efforts focus on how to model the commonalities and distinctions across scenarios at the same time. In general, we can summarize them into three paradigms: Auxiliary Paradigm, Expert Paradigm, and Multi-Tower Paradigm, as demonstrated in Figure 2 (a,b,c). The shared bottom layer of these models is merely simple, either an MLP layer for dimension reduction or an embedding lookup operation. That is, above the shared bottom layer for feature transformation, different network structures on top of the former are searched. While some of them calls for domain specific expertise[15, 19, 20], the others choose to learn the structure automatically[9, 21, 29]. Though fruitful performance gain is obtained by these solutions, they all take it for granted that the representations generated by the bottom layer are fair for different scenarios. In a nutshell, it is interesting but unexplored to enable adaptive feature learning for MSL. A Chinese old saying is: a nine-storeyed terrace must be constructed from its very base. Since it is the fatal bottleneck of the whole architecture and hence would eliminate the expensive structure search for the upper layer.

To this end, in this paper, we propose a **m**ulti-scen**a**rio **r**anking framework with adapt**i**ve fe**a**ture learning (named MARIA). In the bottom layer of MARIA, we design three components to enable discriminative feature learning in a scenario-aware manner: *feature scaling*, *feature refinement*, and *feature correlation modeling*. Specifically, we firstly group the attribute features into different fields like user field, product field, context field and so on. After

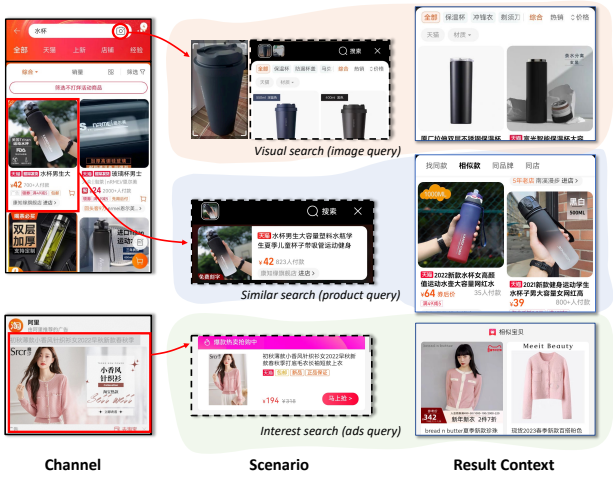
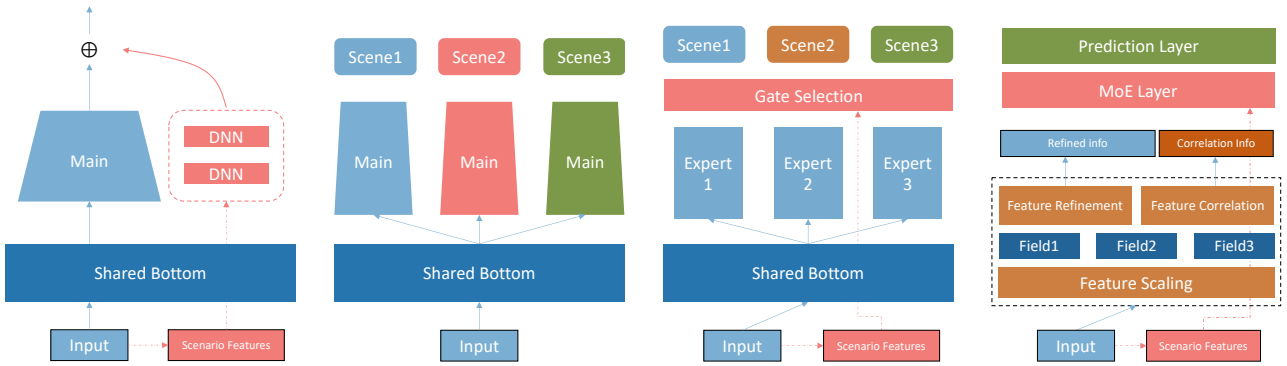


Figure 1: The example of visual search, similar search and interest search.

that, the feature scaling module is utilized to identify the importance of each feature by squeezing or magnifying the feature values. Then, in the feature refinement module, an automatic refiner selection network is utilized for each feature field to perform further high-level semantic encoding. The purpose is to pick the most effective refiner to derive more discriminative semantics at the instance level. Moreover, we further capture the semantic correlation patterns across feature fields as complementary signals. These resultant representations are then concatenated and fed into a simple mixture-of-experts (MoE) structure. It is worthwhile to mention that the above steps focus on extracting scenario-specific features. Hence, to exploit the shared knowledge, we set up an additional scenario-shared tower for final prediction. The paradigm of our proposed MARIA is demonstrated in Figure 2 (d).

We perform extensive experiments over two large-scale real-world datasets for product search and recommendation respectively. The results well demonstrate the superiority of the proposed solution against a series of SOTA alternatives. To summarize, the contributions of this paper are as follows:

- We highlight the fact that current MSL researches mainly neglect the significance of performing scenario adaptive feature learning. We argue that learning to extract discriminative features in a scenario-aware manner is simpler yet more effective than expensive network structure optimization.
- We propose a multi-scenario ranking framework that is devised with an adaptive feature learning strategy for better retrieval and recommendation. In particular, we introduce three components to derive more discriminative features for the forthcoming multi-scenario learning.
- We conduct extensive experiments on two large-scale datasets collected from real-world product search and recommendation application respectively. The experimental results demonstrate that a significant performance gain is obtained by MARIA over the state-of-the-art technique alternatives. Further analysis and experiments also validated the cost-effective nature of the MARIA.



(a) Auxiliary Paradigm **(b) Multi-tower Paradigm** **(c) Expert Paradigm** **(d) Ours**

Figure 2: The current network structures and ours.

2 RELATED WORK

2.1 Auxiliary Paradigm

Generally, scenario indicators have low dimensions. With the increase in network depth, scenario indicators have a limited impact on the final prediction results. Compared with the simplest solutions that use scenario-related features as input, auxiliary solutions build an auxiliary sub-network. In this way, affect the results of the main network by adding or multiplying operation at the output layer directly. This approach is first utilized in single-scenario learning to mitigate the selection bias in training logs, in which many works[1, 6, 24] have been proposed to solve the position bias. These methods usually need to learn a separate model to estimate the influence of bias. Affected by the above methods, DMT[3] proposed a bias deep neural network to model neighbor bias and position bias, which aims to improve the performance of multi-objective ranking. And Shen et al [19] found a seesaw phenomenon in multi-task learning, thus, applied the bias network to calculate the deviation of traffic intervention on training according to the information on scenarios. In addition to using auxiliary networks, there are also ways to add the auxiliary loss to enhance specific learning between different scenarios. For instance, MulANN[15] adds a scenario discriminator module to distinguish which scenario the samples are from. It can be concluded that the auxiliary paradigm can make the scenario characteristics directly affect the final results, but has limited impact on the intermediate layer of the main network.

2.2 Multi-Tower Paradigm

Although the auxiliary paradigm has achieved certain results to some extent, the amount of data in each scenario is seriously unbalanced. In case the whole model will be dominated by data-rich scenarios, multi-tower solutions are proposed by scientists. Each scenario corresponds to a tower and has its own independent parameter subspace, which further improves the individualization. The primary model applies shared bottom and a specific Deep Neural Network(DNN) for each scenario. In 2016, Misra et al [11] proposed a Cross-stitch unit to join two independent task networks in an end-to-end learning strategy. Furthermore, DADNN [5] utilized a routing layer splitting the samples by scenarios into respective domain layers, thus allowing for discriminative representations to be tailored for each individual scenario. More specifically, each scenario has a domain layer that only uses its own data to adjust parameters. Another classical approach STAR [20] added an extra shared tower and proposed a novel parameter fusion method. It multiplies the shared parameters of the extra tower and the customized parameter matrix of the specific scenario to obtain the final network processing parameters. It is worth mentioning that STAR also refers to the first paradigm in scenario information modeling. It can be concluded that the sharing tower can be added to learn the common information between scenarios and alleviate the problem of poor learning of small scenarios.

2.3 Expert Paradigm

In order to enhance the decoupling ability of the model, Google [18] proposed Mixture-of-Experts (MoE), which can significantly increase model capacity and capability. This structure is widely used in the MSL area. After that, MMoE [9] characterizes the task correlation and learns the function of specific tasks based on shared representation. Li et al [8] focus on recommendations in multi-country scenarios. The PLE [21] model shares experts in the share layer and refines tasks uniquely, namely, Customized Gate Control (CGC) structure. In this way, it effectively alleviates the noise caused by other scenarios and improves the effectiveness of feature extraction. Besides, Zhang et al [25] utilize expert networks to solve multi-scenario and multi-task problems on the advertiser's side and use the meta network to express the scenario information explicitly. Zou et al [29] propose a novel expert network structure with automatic selection of fine granularity. By calculating the KL divergence of the gated column vector, one-hot vector, and uniform vector, gates can select the most suitable sharing and exclusive experts. To summarize, the expert paradigm borrowed the idea of bagging, more specifically, training multiple experts to make decisions. This decision is better than one expert in terms of generalization, expression, and learning ability. At the same time, the setting of gates increases flexibility, which takes into account the different learning modes of different scenarios.

3 METHOD

In this section, we present the proposed MARIA model in detail. Figure 3 illustrates the network architecture of our model.

3.1 Problem Formulation

Let $\mathcal{V}, \mathcal{U}, \mathcal{S}$ be the set of all products (including ads), all users, and all scenarios available respectively, and $\{b_u\}_M^N$ be the behavior sequence set where $M = |\mathcal{V}|, N = |\mathcal{U}|$ and $S = |\mathcal{S}|$. Given each user u, the corresponding behavior sequence is organized by following the chronological order: $b_u = [x_1, x_2, \cdots, x_m]$, where $x_j \in \mathcal{V}$ for $1 \leq j \leq m$, and m is the predefined maximum capacity. Product field contains P attributes: $\mathcal{A}_x = [a_x^1, a_x^2, \cdots, a_x^P]$, while user field contains L attributes: $\mathcal{A}_u = [a_u^1, a_u^2, \cdots, a_u^L]$. As to the attributes having continuous numbers, we transform the value into categorical features. Similarly, let $C_{us} = \{c_u^1, c_u^2, ..., c_u^{N_c}\}$ denote N_c context attributes associated with each user u under scenario s. These context features usually describe the page and physical environments of the user side. The total number of available attribute features for users, products and contexts are denoted as N_a^u, N_a^x, and N_a^c respectively. Our task is to precisely identify the target product x_i that will be interested by user u with the above knowledge.

Note that our purpose is to devise a unified ranking framework applicable to both product search and sequential recommendation. For product search, we have an additional input query provided from the user. Since those S scenarios cover diverse services, queries could be in various heterogeneous forms including images, products, and ads, besides the texts. Hence, to keep a concise description in the rest parts, we use the notion of *trigger* to denote the queries under different application scenarios without further clarification. Moreover, without loss of generality, we can assume that each trigger t also includes O attributes: $\mathcal{A}_t = [a_t^1, a_t^2, \cdots, a_t^O]$ and N_a^t denotes the total feature number accordingly. For target scenario s and the corresponding trigger t, a model $f()$ is optimized to precisely generate the likelihood for each candidate product as follows:

$$\hat{y}_{ui} = f(x_i \mid b_u, s, t, C_{us}, \mathcal{A}_u, \mathcal{A}_t, \{\mathcal{A}_{x_1}, \mathcal{A}_{x_2}, \cdots, \mathcal{A}_{x_m}\}); \quad (1)$$

where $\hat{y}_{ui}$ is the likelihood that user u will interact with product x_i with respect to trigger t[1].

3.2 Encoder Layer

Embedding Layer. In the embedding layer, we firstly utilize seven embedding tables to encode the semantics for users, products, user attributes, product attributes, queries, query attributes, contexts, and scenarios respectively: $\mathbf{U} \in R^{N \times d_u}, \mathbf{V} \in R^{M \times d_x}, \mathbf{A}_u \in R^{N_a^u \times d_a}, \mathbf{A}_x \in R^{N_a^x \times d_a}, \mathbf{T} \in R^{N_a^t \times d_a}, \mathbf{C} \in R^{N_a^c \times d_c}, \mathbf{S} \in R^{S \times d_s}$, where d_u, d_x, d_a, d_c, and d_s denote the embedding size of the user, product, attribute, trigger, context, and scenario features respectively. Hence, for each product, user, attribute, trigger, context, and scenario, we can perform the table lookup from these matrices to obtain the corresponding embedding representations.

[1]The trigger will not exist under recommendation scenario. We consider the trigger field when describing the algorithm.

Then, we concatenate the product embedding and the corresponding attribute embeddings to fully represent the semantics of the product:

$$\mathbf{x} = [\mathbf{e}_x || \mathbf{a}_x^1 || ... || \mathbf{a}_x^P] \quad (2)$$

where $\mathbf{e}_x$ is the product embedding for product x, $\mathbf{a}_x^j$ is the corresponding attribute embedding for j-th attribute a_x^j, and $||$ denotes the vector concatenation. Consequently, we can form the feature matrix $\mathbf{B}_u = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$ for sequence b_u. Similarly, we represent the semantics of a user, a trigger, and the corresponding context as $\mathbf{u} = [\mathbf{e}_u || \mathbf{a}_u^1 || \cdots || \mathbf{a}_u^L]$, $\mathbf{t} = [* || \mathbf{a}_t^1 || \cdots || \mathbf{a}_t^O]$ and $\mathbf{c} = [\mathbf{c}_u^1 || \cdots || \mathbf{c}_u^{N_c}]$ respectively, where $*$ denotes $\mathbf{x}$ for product based trigger or image embedding for the visual trigger. Here, we use linear projection to fix the embedding size of the product-based trigger to be the same as the image embedding, which is marked as d_t. The image embedding is pretrained based on the Alimama platform.

Sequence Encoder. As for the user's consecutive behaviors modeling, we choose to extract the user's preference signals from b_u with a Transformer encoder. The built-in stacked self-attention mechanism could enable us to further exploit semantic correlations across different scenarios. In detail, we apply the Transformer network on top of $\mathbf{B}_u$ as follows:

$$\mathrm{H}_u = [\mathbf{h}_{x_1}, \mathbf{h}_{x_2}, \cdots, \mathbf{h}_{x_m}] = Trans(\mathbf{B}_u), \quad (3)$$

where function $Trans$ denotes the Transformer network and the details can be referred to [23], $\mathbf{h}_{x_j}$ is the sequence-wise representation for product x_j from the inter-scenario perspective. After that, a trigger-aware attention module is utilized to aggregate the relevant features towards trigger t as follows:

$$\mathbf{h}_b = \sum_{i=1}^{m} \alpha_i \mathbf{h}_{x_i} \quad (4)$$

$$\alpha_i = \frac{\exp\left(sim(\mathbf{t}, \mathbf{h}_{x_i})\right)}{\sum_{j=1}^{m} \exp\left(sim(\mathbf{t}, \mathbf{h}_{x_j})\right)} \quad (5)$$

$$sim(x, y) = FC([x||y]). \quad (6)$$

where $\mathbf{h}_b$ is the trigger-aware representation of the user's historical behaviors, and function $FC()$ denotes the fully-connected layer. In the recommendation scenario, the trigger representation $\mathbf{t}$ is replaced by the target representation $\mathbf{x_i}$ referring to Equation 2. Afterwards, we concatenate the representations of different fields together as input for later steps:

$$\mathbf{Q} = [\mathbf{h}_b || \mathbf{u} || \mathbf{x_i} || \mathbf{t} || \mathbf{c}], \quad (7)$$

where $\mathbf{x_i}$ is the representation of target product x_i field according to Equation 2.

3.3 Feature Scaling

Considering the heterogeneity of different scenarios, such as different triggers, it is apparent that different kinds of features have different discrimination in various scenarios. For example, visual features generally play a more significant role in the photo query

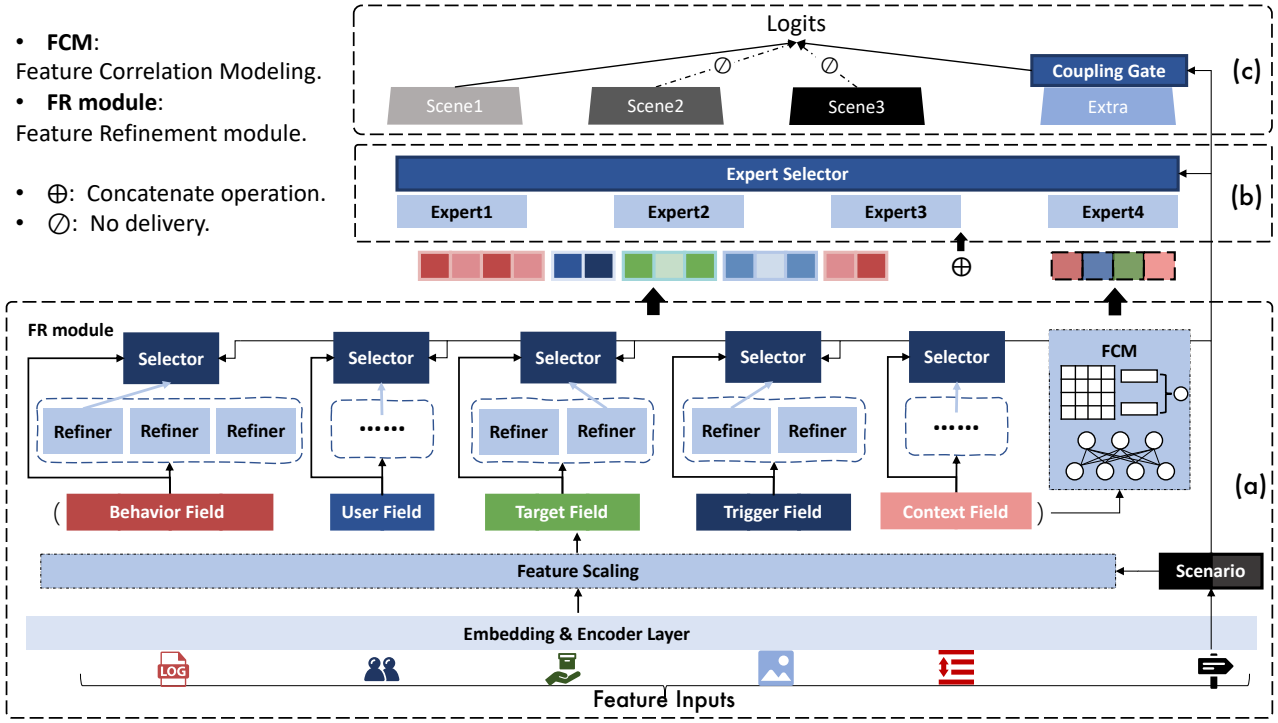


Figure 3: The network architecture of our proposed MARIA.

scenario than that in the product query scenario. What's more, we also consider the interplay between the user's interest and the target product.

Thus, the FS module is designed to squeeze or magnify each feature according to the scenario indicator. Specifically, we calculate a scaling factor for each feature element in $\mathbf{Q}$. Here, a feature element refers to an individual representation. For example, according to Equation 7, $\mathbf{h}_b$ is a feature element, $\mathbf{e}_x$ and $\mathbf{a}_x^1$ are also feature elements. Let N_Q denotes the total number[2] of feature elements included in $\mathbf{Q}$, we calculate a scaling vector $\boldsymbol{\alpha} \in R^{N_Q}$ as follows:

$$\boldsymbol{\alpha} = \lambda * sigmoid(FCN([freeze(\mathbf{Q})||\mathbf{e}_u||\mathbf{e}_{x_i}||\mathbf{e}_s])) \tag{8}$$

where $freeze()$ denotes the stop gradient operation which aims to avoid the overfitting phenomenon and gradient conflict problem [16], $\mathbf{e}_s$ is the scenario embedding. Compared with the standard attention mechanism, the scaling factor λ can be larger than one. That is, we can magnify the important signals instead of keeping them remain. Afterwards, the output of FS module $\mathbf{Q}_S$ is generated as follows:

$$\mathbf{Q}_S = [\boldsymbol{\alpha}_1\mathbf{Q}_1||...||\boldsymbol{\alpha}_{N_Q}\mathbf{Q}_{N_Q}] \tag{9}$$

$$= [\boldsymbol{\alpha}_1\mathbf{h}_b||\boldsymbol{\alpha}_2\mathbf{e}_u||...||\boldsymbol{\alpha}_{N_Q}\mathbf{c}_u^{N_c}] \tag{10}$$

where $\boldsymbol{\alpha}_j$ refers to j-th scalar value in vector $\boldsymbol{\alpha}$, and $\mathbf{Q}_j$ refers to j-th feature element in the order according to Equation 7. We can also denote the scaled representations of different fields as $\hat{\mathbf{h}}_b, \hat{\mathbf{u}}, \hat{\mathbf{x}}_i, \hat{\mathbf{t}}, \hat{\mathbf{c}}$ and $\mathbf{Q}_S = [\hat{\mathbf{h}}_b, \hat{\mathbf{u}}, \hat{\mathbf{x}}_i, \hat{\mathbf{t}}, \hat{\mathbf{c}}]$. Note that scaling vector $\boldsymbol{\alpha}$ is calculated by taking all available features of these five fields into account, which implicitly models the feature correlations.

[2] $N_Q = L + P + O + N_c + 4$

3.4 Feature Refinement

To accommodate scenario-specific features at the instance level, we design a feature refinement module, which utilizes an automatic refiner selection network to support high-level semantic encoding. Specifically, we set up for each feature field a set of feature refiners, each of which is a shallow fully-connected layer. As shown in Figure 3(a), the selection of refiners is automatically made in a scenario-aware manner. A selector is utilized to calculate weights $\boldsymbol{\beta}$ for each field that takes scenario embedding $\mathbf{e}_s$ and the field representation as input. Taking the user behavior field $\hat{\mathbf{h}}_b$ as an example, the high-level feature vector is calculated as follows:

$$\boldsymbol{\beta} = GS(sigmoid(FCN([\hat{\mathbf{h}}_b||\mathbf{e}_s]))), \ \boldsymbol{\beta} \in R^{N_b} \tag{11}$$

$$\tilde{\mathbf{h}}_b = [\boldsymbol{\beta}_1 FC_1(\hat{\mathbf{h}}_b)|| \cdots ||\boldsymbol{\beta}_{N_b} FC_{N_b}(\hat{\mathbf{h}}_b)] \tag{12}$$

where GS and $sigmoid$ denote Gumbel Softmax layer and sigmoid activation respectively, N_b denotes the number of refiners deployed for this feature field, $FC_j()$ denotes the j-th refiner for the field, and $\tilde{\mathbf{h}}_b$ is the resultant high-level feature vector. Here, GS layer is used to approximate the discretization nature of the selection process. We use $ReLU$ as the activation function for each refiner. The same procedure is performed for other feature fields, the resultant high-level features can be represented as follows:

$$\mathbf{Q}_R = [\tilde{\mathbf{h}}_b||\tilde{\mathbf{u}}||\tilde{\mathbf{x}}_i||\tilde{\mathbf{t}}||\tilde{\mathbf{c}}]. \tag{13}$$

3.5 Feature Correlation Modeling

After feature scaling, we further choose to explicitly model the semantic correlations across different feature fields. At first, the representation of a field is projected into the same dimension size d_r via an independent fully-connected layer. The resultant field representations are denoted as $\bar{\mathbf{h}}_b$, $\bar{\mathbf{u}}$, $\bar{\mathbf{x}}_i$, $\bar{\mathbf{t}}$, $\bar{\mathbf{c}}$. Then, we calculate the dot product for each field pair, and concatenate the scores as follows:

$$\mathbf{Q}_C = [\bar{\mathbf{h}}_b \cdot \bar{\mathbf{u}} || \bar{\mathbf{h}}_b \cdot \bar{\mathbf{x}}_i || \cdots || \bar{\mathbf{t}} \cdot \bar{\mathbf{c}}] \tag{14}$$

where symbol $\cdot$ denotes the dot product between two vectors. Ultimately, the final output of the adaptive feature learning by our Maria is formed by concatenating both $\mathbf{Q}_R$ and $\mathbf{Q}_C$:

$$\mathbf{Q}_f = [\mathbf{Q}_R || \mathbf{Q}_C]. \tag{15}$$

3.6 Network Layer

After the adaptive feature learning described above, we utilize a standard Mixture-of-Experts (MoE) [18] as the main structure in the network layer (as shown in Figure 3 (b)). Briefly, we set up a shared set of experts for all S scenarios, where an expert is an independent fully-connected network. A gating mechanism is utilized to aggregate the output of these experts as follows:

$$\mathbf{h}_N = \sum_{j=1}^{N_e} \mathbf{g} f_j(\mathbf{Q}_f) \tag{16}$$

$$\mathbf{g} = softmax(\mathbf{W}_g \mathbf{e}_s) \tag{17}$$

where N_e denotes the number of experts in network layer, W_g is a learnable parameter of the gating mechanism, vector $\mathbf{g}$ contains the importance weights of the experts, and f_j is the j-th expert. In detail, we use *ReLU* as the activation function.

3.7 Prediction and Model Optimization

Prediction. In the prediction layer, we draw lessons from the advantages of multi-tower structure and introduce the scenario-specific DNN tower $FCN^s_{sp}()$ in the prediction phase. In addition, an extra tower $FCN_{sh}()$ is utilized to harness scenario-shared information (as shown in Figure 3 (c)). Thus, the final representation $\mathbf{h}_f$ used for prediction is obtained from these two perspectives:

$$\mathbf{h}_f = \mathbf{h}^s_{sp} + \alpha_s \mathbf{h}_{sh} \tag{18}$$

$$\mathbf{h}^s_{sp} = FCN^s_{sp}(\mathbf{h}_N) \tag{19}$$

$$\mathbf{h}_{sh} = FCN_{sh}(\mathbf{h}_N) \tag{20}$$

where α_s is a coupling coefficient to control the impact of the shared information. Here, we expect that α_s should be small when the target scenario has little connection to other scenarios. Therefore, we calculate α_s by measuring the relevance across scenarios as follows:

$$\alpha_s = \frac{1}{N_s - 1} \sum_{j=1, s_j \neq s}^{N_s} \mathbf{e}_s \cdot \mathbf{e}_{s_j} \tag{21}$$

Finally, the likelihood that user u will interact with product x_i is calculated as follows:

$$\hat{y}_{ui} = FCN(\mathbf{h}_f), \tag{22}$$

Table 1: Statistics of the ALi-CCP datasets. #product, #user #impression and #click represent the number of products/ads, users, user views, clicks respectively.

Datasets	Ali-CCP			
Scenarios	S1	S2	S3	All
#User	91,488	2,612	154,024	244,397
#Product	535,711	198,651	537,937	538,376
#Impression	32,236,951	639,897	52,439,671	85,316,519
#Click	1,291,063	28,022	1,998,618	3,317,703

where $\hat{y}_{ui}$ denotes the prediction score, $FCN()$ denotes the fully-connected network with sigmoid activation.

Model Optimization. For the sake of ensuring the high mobility and universality of the model, all samples use the cross entropy loss, although there are multiple scenarios in the dataset. Thus, the final loss is formulated as follows:

$$\mathcal{L}_{final} = \mathcal{L}_1 + \gamma \mathcal{L}_2, \tag{23}$$

$$\mathcal{L}_1 = -\sum_{u,i} [y_{ui} ln(\hat{y}_{ui}) + (1 - y_{ui}) ln(1 - \hat{y}_{ui})], \tag{24}$$

where $y_{ui}, y_{ui} \in \{0, 1\}$ denotes the ground truth, $\mathcal{L}_2$ is the L_2 norm of all model parameters, γ is a hyperparameter to control the former.

4 EXPERIMENTS

In this section, we conduct extensive experiments over two large-scale real-world datasets to validate the efficacy of Maria in both item search and item recommendation scenarios.

4.1 Experimental Settings

Datasets. To validate the efficacy of our proposed Maria, two real-world large-scale datasets covering diverse scenarios are used for performance evaluation. The first one is the Alimama retrieval dataset collected from the Alibaba search advertising platform. As aforementioned in Section 1, this dataset covers daily search logs of the three scenarios in the period of 2022/08/25 to 2022/09/23: 1) Visual Search (VS) based on the taken photo; 2) Similar Search (SS) for relevant product search against the selected target product; and 3) Interest Search (IS) for ads search from external traffic. We take the instances of the last day, the penultimate day, and earlier ones for testing, validation, and training respectively.

The second dataset is Ali-CCP[3], which is widely used in the relevant literature [8, 10] for product recommendation, which was collected Taobao's recommender system under three scenarios. However, the whole dataset has been anonymized, we cannot describe the specific business scenario. Instead, we denote these scenarios as S1, S2, and S3. The training set, validation set, and test set based on the official split are taken for experiments [10].

Table 1 reports the statistics of Ali-CCP dataset in detail. Due to the sensitive business information involved in the data, we are allowed to report the detailed numbers. #product, #user #impression and #click is at hundred millions, ten millions, billions and hundred millions level respectively. To keep the true characteristics of the real-world scenarios, we do not include further preprocessing

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

steps. In terms of count numbers, we can see that the Alimama dataset is at least two orders of magnitude larger than Ali-CCP, though both of them contain user behaviors of large-scale. Furthermore, there are also obvious distribution discrepancies between scenarios. For example, the number of impressions in the SS scenario of the Alimama dataset is much smaller than the other two scenarios. The same imbalance also occurs in the Ali-CCP dataset. To summarize, these datasets have different data characteristics and distributions covering a broad range of real-world situations, which can effectively verify the effectiveness and universality of our model.

Baselines. We compare the proposed MARIA[4] against the following state-of-the-art methods:

- **Hard Sharing** is a multi-scenario model that shares the parameters of the bottom layer. On top of the shared bottom layer, a single DNN is used for prediction across scenarios.
- **Shared Bottom (Multi-DNN)** replaces the single DNN with multiple DNNs. That is, an individual DNN is utilized for each scenario. And we add an auxiliary tower to enhance the ability to characterize the scenario indicator.
- **MulANN** [15] introduces a domain discriminator module, which aims to distinguish which scenario the instance is from. Adversarial learning is utilized to avoid overfitting against the scenario-specific features, leading to better scenario-shared knowledge transfer.
- **MMoE** [9] implicitly models task relationships for multi-task learning, where different tasks may have different label spaces. Here we adapt MMoE for multi-scenario learning. The number of experts is equal to the number of experts of MARIA. The sum of weighted outputs from the experts are fed into the individual tower for each scenario respectively.
- **PLE** [21] is a state-of-the-art multi-scenario/multi-task model that organizes the experts into scenario-specific groups and scenario-shared groups for the purpose of avoiding negative transfer or seesaw phenomenon.
- **STAR** [20] proposes a star topology to accommodate with the scenario-specific characteristics. Specifically, a shared network works as the center node for knowledge sharing and each scenario network connects only with the center node.
- **AEMS**^{2} [29] proposes a novel MMoE-based model with automatic search towards the optimal network structure. In contrast to PLE and STAR, an expert can be either scenario-shared or scenario-specific dynamically in an instance-aware manner.

Hyperparameter Settings. For a fair comparison, all methods are implemented in the Tensorflow framework, and Adam optimizer is utilized with default parameter setting. Moreover, the number of experts N_e is set to 4 in an expert layer. The learning rate, mini-batch size, and decay rate are set to 0.05, 512, and $1e^{-2}$ respectively. Moreover, we set the number of hidden units to 256 for each expert that is instantiated with a two-layer fully-connected network. A single expert layer is used for all MoE-based models, except for PLE where two expert layers are stacked. That is, we aim to keep their unique network structures but keep the model sizes comparable. As for the prediction layer, a DNN with $128 \times 64 \times 32 \times 1$ structure is adopted for baselines and the towers in MARIA. The weight of domain loss in MulANN is 0.1 on the Ali-CCP and 0.01 on the Alimama dataset respectively.

As to MARIA, we find our model performs relatively stable when λ is set to 2, and the temperature is set to 0.01 for the Gumbel Softmax layer. Also, for the Ali-CCP dataset, γ is set to $1e^{-2}$ and the number of refiners is set to 2 for the user field, while a single refiner is used for the rest fields. For the Alimama dataset, these values are set to $1e^{-6}$ and 2 respectively. Moreover, the dimension size of the refined field representation is reduced against the counterpart in Equation 7. Because we observe little performance fluctuation in a wide range of compression ratio, say 50%-85%.

Evaluation Metric. We adopt the area under the ROC curve (AUC) for performance evaluation. For each method, we repeat the experiment five times and report the averaged results. The statistical significance test is conducted by the student t-$test$.

4.2 Performance Evaluation

The overall performance of all methods is reported in Table 2. Here, we make the following observations.

As for traditional Hard Sharing, it is very difficult to achieve satisfied performance. Compared with other models with expert network or multi-tower structure, the hard sharing solution is not suitable for complex multi-scenario learning. Shared Bottom (Multi-DNN) adds a specific DNN for each scenario in order to harness the scenario-specific knowledge. However, this simple strategy performs suboptimal in two scenarios on the Alimama dataset but performs poorly on the Ali-CCP dataset. This phenomenon is caused by the different characteristics of the two datasets. The scenarios of the Alimama are quite different, as aforementioned. On the contrary, the scenarios in the Ali-CCP dataset are relatively more similar to each other. Note that the advantage of Shared Bottom over Hard Sharing is reversed on Ali-CCP. This suggests that these two trivial methods are inferior to handling complex MSL.

By reinforcing the extraction of the scenario-shared knowledge, MulANN seems to achieve better performance than the above two baselines. But, it is obvious to observe the seesaw phenomenon across scenarios. Moreover, it is surprising that MMoE achieves the second-best performance in the VS scenario on the Alimama dataset. However, the data imbalance problem can not be well addressed by MMoE, leading to inferior performance in the SS scenario instead. As a variant of MMoE, PLE separates the experts into two groups, which alleviates the problems of MMoE somehow. The performance of PLE in various scenarios is more stable. We can see that in the Ali-CCP dataset, PLE shows a significant improvement over MMoE. Furthermore, AEMS^2 performs relatively well on the Ali-CCP dataset but worse on the Alimama. That is, AEMS^2 is possibly more desired for scenarios that are similar to each other. Also, the STAR model only achieves suboptimal performance in S3 on Ali-CCP. Overall, we can see that there is no dominating network structure that can handle complex MSL tasks in real-world scenes.

Our proposed MARIA has significant yet consistent improvement across the three scenarios and two datasets. Specifically, we also summarize the total improvement over each dataset by summing the relative performance gain over the three scenarios. The relative improvement is up to 6.06% and 12.16% for Alimama and Ali-CPP

[4]The code implementation is available at https://github.com/WHUIR/Maria.

Table 2: Performance comparison of different methods across the two datasets. Each dataset consists of three scenarios respectively. The best and second-best results are highlighted in boldface and underlined respectively. $*$ indicates that the performance difference against the best result is statistically significant at 0.05 level.

Dataset	Scenarios	Models							
		Hard Sharing	Shared Bottom	MulANN	MMoE	PLE	STAR	$AEMS^2$	MARIA
Alimama	VS	0.7415*	0.7318*	0.7423*	<u>0.7441</u>*	0.7421*	0.7323*	0.7289*	**0.7473**
	SS	0.6777*	<u>0.6842</u>*	0.6792*	0.6779*	0.6840*	0.6724*	0.6703*	**0.6927**
	IS	0.7131*	<u>0.7159</u>*	0.7129*	0.7139*	0.7126*	0.6925*	0.6994*	**0.7178**
	Total Impr.	+0.0255	+0.0259	+0.0234	+0.0219	+0.0191	+0.0606	+0.0592	–
Ali-CCP	S1	0.5747*	0.5528*	0.5625*	0.5740*	0.5744*	0.5524*	<u>0.5773</u>*	**0.5869**
	S2	0.5912*	0.5606*	0.5779*	0.5789*	0.5939*	0.5826*	<u>0.5957</u>*	**0.6114**
	S3	0.5888*	0.5710*	0.5754*	0.5768*	0.5927*	<u>0.5975</u>*	0.5916*	**0.6077**
	Total Impr.	+0.0513	+0.1216	+0.0902	+0.0763	+0.0450	+0.0735	+0.0414	–

datasets respectively. Recall that our network layer only counts on a standard MoE structure. These results suggest that the idea of performing adaptive feature learning is critical for better MSL.

4.3 Model Analysis

In this section, we perform a deep analysis of our MARIA. At first, the model complexity analysis is conducted to illustrate that the MARIA gains positive benefits with comparable computation complexity and the model size against these baseline methods. Then, a series of ablation studies are performed to validate each design choice. At last, we further dive into the working mechanism of automatic refiner selection with some visualizations.

Complexity Analysis. Since the adaptive feature learning part is unique for MARIA, we therefore analyze the additional computation cost introduced by the three modules. In detail, the FS module takes $O(D_Q \times N_Q)$ to calculate $\boldsymbol{\alpha}$, where D_Q denotes the dimension size of $\mathbf{Q}$. Then, the FR module takes at most $O(D_Q \times D_R)$ to finish the feature refinement, where D_R denotes the dimension size of the $\mathbf{Q}_R$. As to the FCM module, $O(d_r^2)$ is taken to obtain the feature correlation signals. Note that the D_Q, D_R and d_r are relatively small values and $\mathcal{D}_R$ is much smaller than $\mathcal{D}_Q$, the additional computation cost is negligible.

As to model size, on the Alimama dataset, MARIA contains 15.40 Billion (B) parameters. In contrast, this number for some representative baselines is as follows: PLE - 15.31B; $AEMS^2$ - 15.31B; MMoE - 15.29B; STAR - 15.21B. It is clear that our model has a comparable model size but consistently better performance in different multi-scenario tasks.

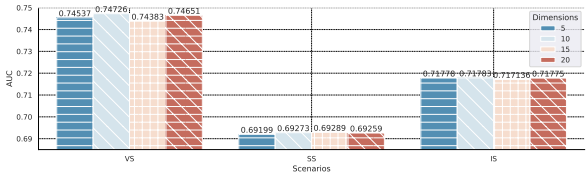
Ablation Study. We conduct further ablation study to validate each design choice: Feature Scaling (FS), Feature Refinement (FR), Feature Correlation Modeling module (FCM), Network Layer (NL), the shared tower (ST) in the prediction layer and the Gumbel Softmax (GS) respectively.

Table 3 reports the performance of these variants and the full MARIA model on both datasets. Here, we can make the following observations. On the whole, these six components have played an obvious positive role in both datasets. The results illustrate that FS, FR, and FCM modules improve the ability to model complex multi-scenario situations through refinement and information sharing at the feature level. Also, the performance reduction by removing NL and ST demonstrates that we still need to facilitate knowledge transfer on top of adaptive feature learning. But we emphasize again that scenario-aware adaptive feature learning is simpler and more efficient for MSL.

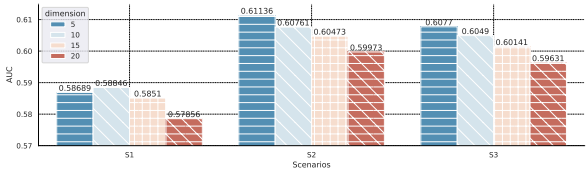
Table 3: The ablation study of MARIA on two Datasets. The best results are highlighted in boldface.

Models	Alimama			
	VS	SS	IS	Total Gain
w/o ST	0.7461	0.6873	0.7173	-0.0071
w/o FCM	0.7445	0.6922	0.7174	-0.0037
w/o FR	0.7466	0.6903	0.7150	-0.0059
w/o FS	0.7462	0.6909	0.7141	-0.0066
w/o NL	0.7447	0.6877	0.7169	-0.0085
w/o GS	0.7455	0.6912	0.7165	-0.0046
MARIA	**0.7473**	**0.6927**	**0.7178**	–
Models	Ali-CPP			
	S1	S2	S3	Total Gain
w/o ST	0.5792	0.5986	0.5952	-0.0330
w/o FCM	0.5686	0.5893	0.5877	-0.0604
w/o FR	0.5709	0.5891	0.5873	-0.0587
w/o FS	0.5844	0.6005	0.5983	-0.0228
w/o NL	0.5796	0.5989	0.5969	-0.0306
w/o GS	0.5857	0.6053	0.5997	-0.0153
MARIA	**0.5869**	**0.6114**	**0.6077**	–

Impact of d_r values. Recall that we measure the semantic correlations between different feature fields by projecting them into the same dimension size d_r. In other words, the setting of this value controls the granularity of correlation measure. Accordingly, we plot the performance patterns of varying dimension d_r values for both Ali-CCP and Alimama datasets in Figure 4. Here, it is reasonable to see that AUC scores increase when d_r becomes small. Our MARIA obtains the best performance on two out of three scenarios

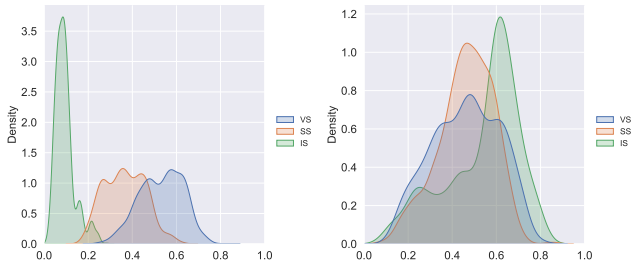


(a) On Alimama dataset



(b) On Ali-CCP dataset

Figure 4: Performance for each scenario with different d_{co}.



(a) User field. (b) User recent behavior field.

Figure 5: Distribution of the RFC weight in the FR modules.

for Ali-CCP when the dimension is 5. Nevertheless, on the Alimama dataset, considering the tremendous data volume, more complex scenarios, and other factors, MARIA is not sensitive to varying values of d_r in different scenarios but a relatively large value of 10 is more optimal.

Visualization of the FR module. A key part of MARIA is the FR module for automatic refiner selection. Hence, to gain further insight into the effectiveness of this mechanism, we sample 1,000 examples and calculate the refiner selection weights distribution on the Alimama dataset[5]. Note that we set the number of refiners to be two for the Alimama dataset, Figure 5a and Figure 5b illustrate the distribution of picking the first refiner across different scenarios on the user field and user behavior field respectively. The abscissa is the probability value, and the ordinate is the data density value obtained after the kernel function transformation. In this way, by comparing the distribution differences of the same refiner corresponding to the same feature field in different scenarios, we can prove that the adaptive feature learning mechanism has realized our expected purpose. That is, the feature-level adaptation is achieved by selecting different refiners to extract high-level semantics at an instance level. Specifically, we can see that the distribution deviation is very obvious (as shown in Figure 5a). This confirms the unique characteristics of this MSL setting: more diverse scenarios with different page environments and user intents, and so on. In this situation, the state-of-the-art solutions cannot well handle this distribution deviation using the same feature representation for all scenarios. In Figure 5b, the distribution deviation in the behavior field also significantly holds between VS and IS scenarios.

[5]The same phenomenon can be observed on the Ali-CCP dataset as well.

Table 4: Comparisons with base serving models in all scenarios on the Alimama platform.

Models	Metric	VS	SS	IS
base models	AUC	0.7344	0.5812	0.8279
MARIA		**0.7416**	**0.6832**	**0.8441**
base models	PCOC	0.0737	1.3641	0.0551
MARIA		**0.0602**	**0.0826**	**0.0226**

5 OFFLINE A/B TEST

We also perform the offline A/B test based on the traffic logs of the three scenarios in Alimama. The performance comparison between MARIA and three independent base models that were serving online is reported in Table 4. In real serving environments, VS, SS, and IS are trained using 60, 90, and 120 days of historical data respectively. While Maria applies 60 days of training data in all three scenarios, which greatly reduces the cost of training. We take the averaged results from March 11th to 16th, 2023, as shown in Table 4. Specifically, As for MARIA, we successfully obtain 12.3% of AUC revenue in total. In order to verify whether the prediction is closer to the real click-through rate (CTR), we further measure the ratio of the Predicted CTR Over the posterior CTR (PCOC). When PCOC is closer to 1.0, the more accurate the CTR prediction is. In Table 4, we can see that the PCOC of MARIA in all scenarios is more compact and concentrated around 1.0 than the three serving models, which demonstrates the superiority of our solution.

6 CONCLUSION

In this paper, we propose a multi-scenario ranking framework with adaptive feature learning, named MARIA. We design three components to enable discriminative feature learning in a scenario-aware manner: *feature scaling*, *feature refinement*, and *feature correlation modeling*. In this way, the scenario-relevant features suppress the irrelevant counterparts. And an automatic refiner selection in the FR module further refines the high-level semantics for better scenario adaptation. Moreover, the feature correlations are also exploited to enhance discrimination. The key point of our proposed MARIA is to highlight that performing adaptive feature learning could be simpler and more effective instead of conducting the structure search.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (No. 62272349); Alibaba Group through Alibaba Innovative Research Program; and Young Top-notch Talent Cultivation Program of Hubei Province. Chenliang Li is the corresponding author.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.

[2] 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*. 7–10.

[3] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.

[4] 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).

[5] Junyou He, Guibao Mei, Feng Xing, Xiaorui Yang, Yongjun Bao, and Weipeng Yan. 2020. DADNN: Multi-Scene CTR Prediction via Domain-Aware Deep Neural Network. *arXiv preprint arXiv:2011.11938* (2020).

[6] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*. 781–789.

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* (2009), 30–37.

[8] Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. 2020. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2605–2612.

[9] 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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[10] 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 *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[11] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3994–4003.

[12] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. 995–1000.

[13] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[14] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. 291–324.

[15] Alice Schoenauer-Sebag, Louise Heinrich, Marc Schoenauer, Michele Sebag, Lani F Wu, and Steve J Altschuler. 2019. Multi-domain adversarial learning. *arXiv preprint arXiv:1903.09239* (2019).

[16] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems* 31 (2018).

[17] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 255–262.

[18] 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).

[19] 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 *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4094–4103.

[20] 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 *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4104–4113.

[21] 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 *Fourteenth ACM Conference on Recommender Systems*. 269–278.

[22] Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. 2022. When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1632–1641.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[24] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.

[25] 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 *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1368–1376.

[26] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *The 38th European Conference on Information Retrieval*. 45–57.

[27] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[28] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[29] Xinyu Zou, Zhi Hu, Yiming Zhao, Xuchu Ding, Zhongyi Liu, Chenliang Li, and Aixin Sun. 2022. Automatic Expert Selection for Multi-Scenario and Multi-Task Search. In *The 45th International ACM SIGIR Conference on Research & Development in Information Retrieval*.