UFIN: Universal Feature Interaction Network for Multi-Domain Click-Through Rate Prediction

Zhen Tian[1,2], Changwang Zhang[3], Wayne Xin Zhao[1,2](✉), Xin Zhao[3], Ji-Rong Wen[1,2], and Zhao Cao[3]

[1] Gaoling School of Artificial Intelligence, Remin University of China, Beijing, China
[2] {chenyuwuxinn,batmanfly,jrwen}@ruc.edu.cn
[3] Poisson Lab, Huawei, Beijing, China

Abstract. Click-Through Rate prediction, which aims to estimate the probability of a user clicking on an item, is a key task in online advertising. Existing CTR models concentrate on modeling the feature interactions within a single domain, thereby rendering them inadequate for fulfilling the requisites of multi-domain scenarios. Some recent approaches propose intricate architectures to enhance knowledge sharing across multiple domains. However, they encounter difficulties when being transferred to new domains, owing to their reliance on the modeling of ID features. To tackle this issue, we propose the **U**niversal **F**eature **I**nteraction **N**etwork (**UFIN**) approach for CTR prediction. UFIN exploits the textual data to learn the universal feature interactions that can be effectively transferred across diverse domains. Specifically, we regard the text and feature as two different *modalities* and develop an encoder-decoder network to enforce the transference of data from text modality to feature modality. Building upon the above foundation, we devise a mixture-of-experts enhanced adaptive interaction model to learn the transferable collaborative patterns across multiple domains. As such, UFIN can effectively bridge the semantic gap to learn the common knowledge across various domains, surpassing the constraints of ID-based models. Extensive experiments conducted on eight datasets show the effectiveness of UFIN. Our code is available at https://github.com/RUCAIBox/UFIN.

Keywords: Universal Feature Interaction, CTR Prediction

1 Introduction

Click-Through Rate (CTR) prediction, which aims to predict the probability of a user clicking on an item, is an important task for online advertising and recommender systems. Various approaches have been proposed for effective CTR prediction [11, 25]. These methods mainly focus on accurately modeling the complicated feature interactions to capture the underlying collaborative patterns. Most of the existing approaches concentrate on single-domain prediction, where

✉Wayne Xin Zhao (batmanfly@gmail.com) is the corresponding author.

each model is solely trained to serve the CTR prediction of a single scenario. However, in large-scale corporate enterprises, numerous business domains frequently necessitate CTR prediction to augment user contentment and enhance commercial revenue. For instance, in the case of e-commerce enterprises, the advertising scenarios encompass a wide array of options and manifest notable disparities, encompassing domains such as motion pictures, literary works, electronic devices, and culinary delights, among others. Merely mixing all the data and training a single shared CTR model cannot yield satisfactory results across all domains owing to the substantial distribution variance among diverse scenarios (*domain seesaw phenomenon* [2]). The domain-specific modeling paradigm severely restricts the efficient utilization of extensive user behavior data in business scenarios.

Some recent studies [2, 20, 26, 27] propose conducting multi-domain CTR predictions. The core idea of these approaches is to introduce a shared neural network for learning the common knowledge across diverse domains, while simultaneously integrating multiple domain-specific sub-networks to capture the distinct characteristics of each domain. Although somewhat efficacious, the majority of these methods rely on modeling the ID features (*e.g., item_ id*) to develop the CTR prediction. A major obstacle of this paradigm is the limited transferability of the learned model to new recommendation scenarios, even when the underlying data structures remain unchanged.

Inspired by recent advancements in natural language recommendations [13], our objective is to devise a novel approach to learn universally applicable collaborative patterns by surpassing the constraints of ID features. Our fundamental concept entails transforming raw features, such as the *location* of an item, into textual data and employing Large Language Models (LLMs) to acquire transferable representations. While previous attempts have demonstrated the promise of this approach for certain recommendation tasks [8, 9], there remain several critical challenges to address in the context of multi-domain CTR predictions. First, the textual semantic space is not directly conducive to the task of CTR prediction [10]. In comparison to traditional feature interaction based methods, LLMs encounter difficulties in capturing collaborative patterns, thereby resulting in suboptimal model performance. Second, due to substantial distribution variance across different domains, effectively leveraging the collaborative knowledge from the source domain to enhance the target domain proves to be a formidable task. For instance, the interaction between features *user* and *title* proves most valuable in movie recommendations, yet its efficacy diminishes in the context of beauty recommendations.

To address these issues, in this paper, we propose the **U**niversal **F**eature **I**nteraction **N**etwork (**UFIN**) for multi-domain CTR prediction. UFIN exploits textual data to acquire knowledge of universal feature interactions that can be effectively transferred across diverse domains. To learn universal feature representations, we devise a prompt to convert the raw features into text and subsequently generate a set of *universal features* to capture the general attributes of interactions. Notably, we regard the text and feature representations as two *modalities* and devise an encoder-decoder network founded on a Large Language Model (LLM) to enforce the conversion of data from the text modality to the feature modality. This scheme can be denoted as "raw features ⇒ text ⇒ universal features". For learning universal feature interactions, we develop an MoE-enhanced adaptive feature interaction model, which can learn the generalized collaborative patterns from diverse domains. To further enhance the acquisition of collaborative knowledge, we propose a multi-domain knowledge distillation framework to supervise the training of our approach. Through these aforementioned mechanisms, UFIN can effectively bridge the semantic gap to learn common knowledge across various recommendation domains, surpassing the limitations of ID-based models.

The paper's main contributions are summarized as follows:

- We propose a novel Universal Feature Interaction Network (UFIN) for CTR prediction, intelligently acquiring the collaborative patterns across diverse domains.
- To the best of our knowledge, UFIN is the first deep CTR model to harness Large Language Models (LLMs) to adaptively learn the feature interactions for recommendations, thereby obtaining universal feature representations from textual data. This empowers UFIN to proficiently bridge the semantic gap across various domains.
- We propose a multi-domain knowledge distillation framework for enhancing the feature interaction learning. This motivates UFIN to proficiently acquire the collaborative knowledge from diverse domains, thereby improving the model performance.
- We conduct extensive experiments on eight widely used datasets. UFIN outperforms a number of competitive baselines in both multi-domain and cross-platform settings, demonstrating the effectiveness of our approach.

2 Methodology

In this section, we present a universal feature interaction network for multi-domain CTR predictions, named **UFIN**. Our goal is to learn the universal feature interactions that are able to effectively transferred to new recommendation domains. As the core idea, we consider the text and features as two *modalities*, and adopt an encoder-decoder network for learning universal feature representations. For learning universal feature interactions, we further develop a multi-experts adaptive interaction network to acquire transferable collaborative knowledge. In what follows, we introduce the details of universal feature representation learning (Section 2.1) and universal feature interaction learning (Section 2.2).

2.1 Universal Feature Representation Learning

To learn transferable feature representations, we adopt natural language text as the universal data form, which is derived from raw features through a prompt. As increasingly more evidence shows [10], text and feature representations can be regarded as two *modalities* that can be mutually transformed. Based on this idea, we employ an MoE-enhanced LLM as the encoder to enforce the text modality

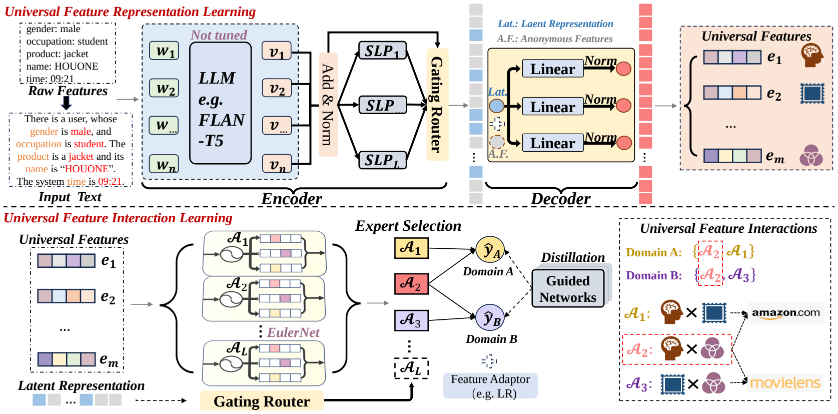


Fig. 1: The overall framework of UFIN.

into a latent space and develop a decoder that performs a mapping to the feature modality, thereby generating the universal feature representations (*universal features*). This approach can be expressed as "raw features ⇒ text ⇒ universal features", providing a means to bridge the semantic gap across different domains.

Feature Textualization The first step in learning universal feature representations is to transform the raw features into textual data, as described by a prompt. As previous work shows [5], an effective prompt should consist of *personalized* fields for different users and items. For this purpose, we design a prompt, that includes the user profile, item description and contextual information to conduct such transformation. As shown in Figure 1, given the user features (*i.e., male*, *student*), item features (*i.e., jacket*, *HOUONE*) and context features (*i.e., 09:21*), the transformed data is shown as: "There is a user, whose gender is male, and occupation is student. The product is a jacket and its name is "HOUONE". The system time is 09:21." In our prompt, different types (*i.e.,* user-side, item-side and context-side) of raw features (*i.e.,* $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$) are sequentially summarized into a sentence, where the descriptions of different sides are separated by the period `"."`, and the features are separated by the comma `","`. As such, a CTR instance can be denoted as $\{\boldsymbol{w}_1, ..., \boldsymbol{w}_n\}$, where n is the number of words in a sentence. In this way, we can obtain a universal data form (*i.e.,* natural language text) to represent the CTR instances across various domains or platforms.

Textual Feature Encoding Given the universal data form in the text modality, our goal is to transform it into feature modality to obtain the universal features. We follow most multimodal work [16] (*e.g.,* text-to-image) that employs an encoder-decoder architecture to align the representations of different modalities. Specifically, we first harness an MoE-enhanced LLM as the encoder to project the textual data into a common latent space. Given the words of textual data

$\{\boldsymbol{w}_1, ..., \boldsymbol{w}_n\}$, we feed them into the LLM:

$$\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\} = \text{LLM}(\{\boldsymbol{w}_1, ..., \boldsymbol{w}_n\}), \quad \boldsymbol{s} = \text{LayerNorm}(\sum_{j=1}^{n} \boldsymbol{v}_j), \tag{1}$$

where $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\} \in \mathbb{R}^{d_V \times n}$ is the last hidden state of LLM, d_V is the state dimension and $\boldsymbol{s}$ is the latent vector. Unlike existing studies [8], we use sum pooling to preserve token-level semantics of features and apply LayerNorm [1] to adjust the semantic distributions. **Note that the LLM is solely for text encoding, which is not tuned during training**. Therefore, we can cache the last hidden state $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\}$ to ensure the efficiency of our approach.

Multi-Domain Semantic Fusion. In the above, we obtain semantic representations from LLMs. However, recent study [8] found that the original semantic space of PLMs is not suitable for the recommendation task. To address this issue, a commonly used approach is to employ a neural network to learn the appropriate semantic space for enhancing the representations. Since different domains usually correspond to varying semantic contexts, merely learning a shared semantic space for all domains will suffer from the domain seesaw phenomenon that degrades the model capacity. To address this issue, our idea is to learn an independent semantic space for each domain and adaptively combine them based on the semantic context. Specifically, given L domain, we employ a single-layer perception (SLP) to learn the semantic representation for each domain, and incorporate them into an MoE model to enhance the domain fusion:

$$\boldsymbol{z} = \sum_{j=1}^{L} \sigma(\boldsymbol{W}_j \boldsymbol{s} + \boldsymbol{b}_j) \cdot g_j, \quad \boldsymbol{g} = \text{Softmax}(\boldsymbol{W}_g \boldsymbol{s}), \tag{2}$$

where $\boldsymbol{W}_j \in \mathbb{R}^{d_V \times d_V}$ and $\boldsymbol{b}_j \in \mathbb{R}^{d_V}$ are the weight and bias of the j-th expert, σ is the activation function, g_j is the j-th combination weight of the gating router $\boldsymbol{g}$, $\boldsymbol{W}_g \in \mathbb{R}^{L \times d_V}$ is the router weight, and $\boldsymbol{z}$ is the enhanced representations.

Fusing Anonymous Features. As there exists numerous anonymous features (such as the identifiers of the user/item) that lack semantic information, we refrain from including them in our prompt template. Nevertheless, these features may play a significant role, particularly in situations where semantic features are limited or nonexistent. For instance, in the case of the Amazon dataset, only *user_id* features are accessible on the user side. To generate more comprehensive predictions, we expand our methodology to encompass these features. There are many ways to achieve this purpose, and we follow the existing ID-Text fusion work [8], which employs a distinct embedding for each anonymous feature and merges them with the textual representations:

$$\tilde{\boldsymbol{z}} = \boldsymbol{z} + \sum_{k=1}^{c} \boldsymbol{U}_k \boldsymbol{h}_k, \tag{3}$$

where $\{\boldsymbol{h}_1, ..., \boldsymbol{h}_c\} \in \mathbb{R}^{d_A \times c}$ is the anonymous embeddings, c denotes the anonymous field number, and $\boldsymbol{U}_k$ is the projection matrix. **Note that the anonymous**

features are only auxiliary representations and are not used unless specified. For efficiency considerations, we do not employ other complex mechanisms (e.g., self-attention), which will be studied in our future work.

Universal Feature Generation With the above textual encoding procedure, we can obtain the universal representation of the instances. Previous works [14] directly feed the textual representations into a feedforward network (*e.g.,* MLP) to make predictions, resulting in suboptimal performance. As the recent study shows [19], it is challenging for an MLP to capture effective collaborative patterns compared with the feature-wise interactions (*e.g.,* FM [18]). Our proposed solution, by contrast, entails harnessing textual data to generate *universal features* that transcend various domains, thereby capturing the collective patterns that are commonly observed. To illustrate, we anticipate generating the universal attribute *"amusing"* from the textual expressions *"This movie is hilarious"* and *"This book is whimsical"* within the realms of movie and book recommendations. Based on this idea, we develop a decoder that conducts a transformation to map the latent representation of textual data to the feature modality, shown as:

$$\tilde{\boldsymbol{e}}_j = \text{LayerNorm}(\boldsymbol{V}_j \tilde{\boldsymbol{z}}), \tag{4}$$

where $j \in \{1, 2, ..., n_u\}$ and n_u is the field number of universal features. Here we incorporate a set of projection matrices $\{\boldsymbol{V}_j \in \mathbb{R}^{d \times d_V}\}_{j=1}^{n_u}$ to generate a set of universal features $\tilde{\boldsymbol{E}} = \{\tilde{\boldsymbol{e}}_1, \tilde{\boldsymbol{e}}_2, ..., \tilde{\boldsymbol{e}}_{n_u}\}$, each measuring different aspects from different representation subspaces. As such, we use the generated universal representations for subsequent feature interaction modeling.

2.2 Universal Feature Interaction Learning

The core of our proposed UFIN is to learn the universal feature interactions for intelligently acquiring the generalized collaborative knowledge across diverse domains. To this end, our approach is to model the adaptive feature interactions based on the generated universal features to capture the common collaborative patterns. Furthermore, to promote feature interaction learning, we introduce a framework for distilling knowledge to guide the model's learning process and subsequently enhance its performance.

Adaptive Feature Interaction Learning at a Single Domain For learning universal feature interactions, an important issue is how to accurately model the interactions orders/forms within each domain, as different domains typically correspond to varying feature relationships. Traditional methods manually design a maximal order and further remove the useless interactions from them, *e.g.,* FM [18] empirically enumerates all second-order feature interaction terms. These models not only result in inaccurate modeling of the underlying true feature interactions in real-world scenarios but also limits the transferability. As a promising approach, recent study EulerNet [24] proposes to model the *adaptive* feature interactions, *i.e.,* the interaction forms are automatically learned from data, allowing for arbitrary orders and a flexible number of terms. Given the input features $\tilde{\boldsymbol{E}} = \{\tilde{\boldsymbol{e}}_1, ..., \tilde{\boldsymbol{e}}_{n_u}\}$ (See Eq. (4)), the adaptive feature interaction learning function of EulerNet [24] is shown as:

$$\mathcal{F}(\tilde{\boldsymbol{E}}; \mathcal{A}) = \boldsymbol{w}^{\top} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \tilde{\boldsymbol{e}}_1^{\alpha_1} \odot \tilde{\boldsymbol{e}}_2^{\alpha_2} \odot ... \odot \tilde{\boldsymbol{e}}_{n_u}^{\alpha_{n_u}}, \tag{5}$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_{n_u}]$ denotes the learnable order parameter of each feature, $\mathcal{A}$ is the parameter set of all the learnable orders, and $\boldsymbol{w}$ is a transition vector for generating a scalar result. For a given domain, the underlying true feature interaction forms are automatically learned from the parameter $\mathcal{A}$, *e.g.,* the interactions of FMs [18] can be learned by $\mathcal{A} = \{\boldsymbol{\alpha} | \sum_{j=1}^{m} \alpha_j = 2, \forall \alpha_j \in \{0, 1\}\}$. Previous works [3] face challenges in achieving this, because when the embedding $\boldsymbol{e}_j$ contains negative values, the order α_j must be set to an integer value to avoid invalid operation (*e.g.,* $(-1)^{0.5}$). As a solution, EulerNet [24] leverages Euler's formula to learn the feature interactions in a complex vector space that enables the efficient learning of arbitrary-order feature interactions, without additional restrictions (*e.g., non-negative* embedding or *integer* order).

In our case, we employ EulerNet [24] to learn the underlying feature interaction orders/forms within each given domain. We mainly transfer the knowledge of interaction orders (*i.e.,* $\mathcal{A}$) to enhance predictions in a target domain.

Multi-Domain Feature Interaction Learning In the above, we have discussed the feature interaction learning at a single domain. In this section, we generalize the methodology for adapting the multiple domains. Intuitively, we can train distinct models independently to adapt to the distribution of each domain. However, this approach fails to grasp the interconnectedness between diverse domains, leading to challenges in the cold-start scenario. Our objective is to acquire knowledge of the universal feature interactions that encompass the shared collaborative patterns between different domains. Our proposed solution entails the introduction of multiple sets of interaction orders (*i.e.,* $\mathcal{A}$), each of which learns the underlying true feature interaction orders for a single domain, and shares some of them across different domains to acquire knowledge of the common collaborative patterns.

In practice, we employ an MoE model to implement our idea. Given L domains, we introduce L experts, each expert j is implemented by EulerNet [24] with the learnable order parameters $\mathcal{A}_j$. All experts share the input embedding $\tilde{\boldsymbol{E}} = \{\tilde{\boldsymbol{e}}_1, ..., \tilde{\boldsymbol{e}}_{n_u}\}$ (See Eq. (4)), combined by a gating router based on the semantic representation $\tilde{\boldsymbol{z}}$ (See Eq. (3)):

$$\zeta = \sum_{j=1}^{L} \mathcal{F}(\tilde{\boldsymbol{E}}; \mathcal{A}_j) \cdot \tilde{\boldsymbol{g}}_j, \quad \tilde{\boldsymbol{g}} = \text{TopK}\Big(\text{Softmax}(\tilde{\boldsymbol{W}}_g \tilde{\boldsymbol{z}})\Big) \tag{6}$$

where TopK$(\cdot)$ retains only the top-K elements of the input tensor and sets all other elements as zero, $\mathcal{A}_j$ (*learnable parameter*) is the interaction order of j-th expert, $\mathcal{F}(\cdot)$ is the feature interaction function (See Eq. (5)) of EulerNet [24], $\tilde{\boldsymbol{g}}_j$ is the j-th combination weight, $\tilde{\boldsymbol{W}}_g \in \mathbb{R}^{L\times d_V}$ is the router weight, and ζ is the output logits. For an instance in a given domain u, we use its corresponding semantic representations $\tilde{\boldsymbol{z}}$ to select K experts with the order sets $\mathcal{S}_u = \{\mathcal{A}_j^u\}_{j=1}^K$ for collaboratively learning the feature interactions. Formally we set $K > \lceil L/2 \rceil$, and we have the following finding:

Theorem 1. *Given L domains $\mathcal{D} = \{D_1, ..., D_L\}$ and L experts $\mathcal{A} = \{\mathcal{A}_1, ..., \mathcal{A}_L\}$, each domain D_u select K experts $\mathcal{S}_u = \{\mathcal{A}_1^u, \mathcal{A}_2^u, ..., \mathcal{A}_K^u\}$ from $\mathcal{A}$,* i.e., *$\mathcal{S}_u \subseteq \mathcal{A}$. If $K > \lceil L/2 \rceil$, then for any given domain D_u and D_v, the set of the selected experts $\mathcal{S}_u$ and $\mathcal{S}_v$ must have a same element,* i.e., *$\mathcal{S}_u \cap \mathcal{S}_v \neq \emptyset, \forall u \neq v$.*

Proof. Assume there exists $\mathcal{D}_u$ and $\mathcal{D}_v$ $(u \neq v)$ that satisfies $\mathcal{S}_u \cap \mathcal{S}_v = \emptyset$, we have: $|\mathcal{S}_u \cup \mathcal{S}_v| = |\mathcal{S}_u| + |\mathcal{S}_v| - |\mathcal{S}_u \cap \mathcal{S}_v| = 2K > 2 \times \lceil L/2 \rceil \geq L$ On the other hand, since $\mathcal{S}_u \subseteq \mathcal{A}$ and $\mathcal{S}_v \subseteq \mathcal{A}$, we have $\mathcal{S}_u \cup \mathcal{S}_v \subseteq \mathcal{A}$. Therefore, we have: $|\mathcal{S}_u \cup \mathcal{S}_v| \leq |\mathcal{A}| = L$. Since the assumption leads to a contradiction, our initial assumption must be **False**. Therefore, $\forall u \neq v, \mathcal{S}_u \cap \mathcal{S}_v \neq \emptyset$.

It demonstrates that for any given domains u and v, there exist at least one shared expert that learns the common feature interactions. Therefore, our approach can capture the common feature interactions between arbitrary domain pairs, thereby capable of learning the generalized feature relationship across all domains. Finally, we apply the sigmoid function on the logits to obtain the prediction, *i.e.,* $\hat{y} = \text{Sigmoid}(\zeta)$.

As mentioned in the work [20], a good multi-domain CTR model should contain the features that depict the domain-specific information. For this purpose, we can further incorporate a ***feature adaptor***, which takes as input the domain-specific features, to precisely capture the distinct characteristics of each domain. Following existing multi-domain methods [20], we add the output logits of the feature adaptor (*i.e.,* ζ_f) to our approach (*i.e.,* ζ) for prediction:

$$\hat{y} = \text{Sigmoid}(\zeta + \zeta_f). \tag{7}$$

Knowledge Distillation Enhanced Training With the above approaches, UFIN is able to learn the universal feature interactions based on the representations generated by LLMs. However, as existing study shows [10], it is challenging for the representations of LLMs to capture the feature co-occurrence correlation that results in the poor performance compared with traditional collaborative models. To facilitate our approach in capturing the underlying collaborative patterns, we propose a multi-domain knowledge distillation framework that promotes feature interaction learning.

Multi-Domain Guided Distillation. In the knowledge distillation framework, for each domain, we pre-train a feature interaction model as the *guided network* to learn the domain-specific collaborative patterns. As such, multiple guided networks from different domains are incorporated as teacher model to supervise the training of our approach. It allows flexibility to choose any teacher model for each domain, and we specifically employ the EulerNet [24] as the teacher with consideration for the consistency. Following existing studies [23], we use MSE loss to align the output logits between the guided network and our approach:

$$\mathcal{L}_{KD} = \sum_{p=1}^{M}\sum_{i=1}^{N_p} ||\zeta_{p,i}^{G} - \zeta_{p,i}||^2, \tag{8}$$

where $\zeta_{p,i}^{G}$ and $\zeta_{p,i}$ (see Eq. (6)) are the logits of the guided network and our proposed UFIN for the i-th instance in the p-th domain. Note that the guided networks are only used for auxiliary training, which are discard during inference, thus ensuring the transferability of our approach.

Multi-Task Learning. To further promote the utilization of LLMs in the CTR prediction task, we incorporate the commonly adopted binary cross-entropy loss:

$$\mathcal{L}_{CTR} = \sum_{p=1}^{M}\sum_{i=1}^{N_p} \Big(y_i^p \log(\hat{y}_i^p) + (1 - y_i^p)\log(1 - \hat{y}_i^p) \Big), \tag{9}$$

where y_i^p and $\hat{y}_i^p$ are the ground-truth label and predicted result of i-th instance in the p-th domain. For the training, we adopt a multi-task training strategy to jointly optimize the knowledge distillation loss and the binary classification loss:

$$\mathcal{L} = \mathcal{L}_{KD} + \mathcal{L}_{CTR}. \tag{10}$$

2.3 Discussion

In the literature, a number of CTR models have been proposed. To better highlight the novelty and difference of our approach, we make a brief comparison of different CTR methods. For ID-based methods, such as STAR [20], they rely on the ID features to develop the prediction, which impairs the inherent semantic of features that makes them incapable of being applied in the new platforms. While for semantic methods, such as P5 [6], attempt to leverage the world knowledge of LLMs to uplift the prediction. The primary challenge of these approaches lies in their inability to capture collaborative signals, resulting in poor performance. Although CTRL [10] proposes a contrastive learning framework to transfer the knowledge of LLMs to a domain-specific collaborative model. Due to the limited capacity of the collaborative model, it cannot adequately learn the common knowledge across different domains and suffers the seesaw phenomenon (See Section 3.2). In contrast, our model is more *universal* in the application of multi-domain or cross-platform settings, naturally integrating the semantic knowledge of LLMs and collaborative knowledge of interactions. The overall comparison is presented in Table 1.

Table 1: Comparison of different methods.

Methods	Transfer Learning		Knowledge Patterns	
	Multi-Domain	Cross-Platform	Collaborative	Semantic
STAR [20]	✓	✗	✓	✗
CTRL [10]	✗	✗	✓	✓
P5 [6]	✓	✓	✗	✓
UFIN (ours)	✓	✓	✓	✓

Table 2: The statistics of datasets.

Dataset	**# Users**	**# Items**	**# Interactions**
Amazon	1,002,827	2,530,874	7,427,505
- Movies	295,908	40,792	2,141,592
- Books	585,167	191,826	4,077,731
- Electronics	184,876	26,336	780,698
- Food	14,552	5,474	86,518
- Instruments	1,429	736	7,835
- Office	4,890	1,805	36,572
- Toys	19,231	9,142	116,559
MovieLens-1M	6,041	3,669	739,012

3 EXPERIMENTS

3.1 Experimental Settings

Datasets To evaluate the performance of our model, we conduct experiments on the Amazon and MovieLens-1M datasets in both multi-domain setting and cross-platform setting. Specifically, Amazon is a popular dataset for recommender systems research. We select seven subsets for the multi-domain setting (*i.e.,* *"Movies and TV"*, *"Books"*, *"Electronics"*, *"Office Products"*, *"Musical Instruments"*, *"Toys and Games"* and *"Grocery and Gourmet Food"*); MovieLens-1M is a movie recommendation dataset, which does not contain overlapped users or items with Amazon. We use this dataset to evaluate the performance in a cross-platform setting. The statistics of datasets are summerized in Table 2.

Compared Models We compare the UFIN with twelve state-of-the-art methods, including: **(1)** four single-domain methods: DeepFM [7], DCNV2 [25], xDeepFM [12], EulerNet [24], **(2)** six multi-domain methods: MMoE [15], HMOE [22], AESM [28], STAR [20], PEPNet [2], and **(3)** two semantic methods: P5 [6], CTRL [10]. For our approach, since there are no effective user-side textual features (*e.g.,* only *user_id* available on the Amazon dataset), we incorporate the *user_id* features into our approach (See Section 2.1). We introduce two versions: (1) $\underline{\textbf{UFIN}_t}$ denotes the model using only item text; (2) $\underline{\textbf{UFIN}_{t+f}}$ denotes the model using item text and integrating a feature adaptor (See Section 2.2).

Implementation Details The dataset is randomly split into 8:1:1 for train/val/test. Following the work [2], we separately train each single-domain model in each domain to report their best results. For multi-domain evaluation, we mix all train data of Amazon to train the multi-domain and semantic models, and evaluate their performance in each given domain. For cross-platform evaluation, we first pre-train each semantic model on the mixed Amazon dataset, and then fine-tune them on MovieLens-1M. the number of experts in the semantic fusion MoE and feature interaction MoE is equal to the number of domains (*i.e.,* $L = 7$). For the setting of UFIN, the universal feature fields n_u is set to 7 and dimension d is set to 16. The K value of $TopK$ function is 5. The SLP hidden size is 128. For the EulerNet expert, the layer number is 1, and the number of order vectors is 7.

3.2 Overall Performance

We compare the UFIN with twelve baseline methods on seven multi-domain datasets and one cross-platform dataset. The overall results are presented in Table 3, and we have the following observations:

Single-domain methods perform well on the Movies and Books datasets, but exhibit subpar performance on sparse domains with less interactions (*i.e.,* Instruments and Office). Multi-domain methods achieve comparable performance on the Instruments and Office datasets, showing the effectiveness of multi-domain information sharing. For semantic methods, P5 [6] performs poorly across all datasets due to the limitation of LLMs in capturing collaborative signals. Besides, CTRL [10] achieves better performance by aligning the representations between LLMs and collaborative models, but it becomes less effective on the Instruments dataset, showing its limitation of capturing the relatedness between different domains. UFIN achieves the best performance in almost all the cases. It signifies that our suggested universal feature interactions are better suited for the adaptation of multi-domain distributions. Further, the cross-platform evaluation results show that our approach can be effectively transferred to a new platform. Besides, UFIN$_{t+f}$ has a great improvement over UFIN$_t$, showing the effectiveness of incorporating feature adaptor to learn domain-specific collaborative patterns.

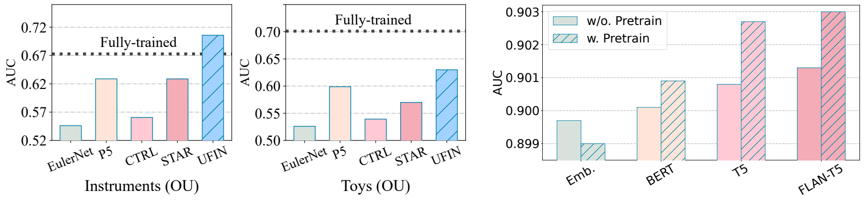
Regarding the efficiency, it can be observed that the latency of single-domain models is relatively small due to their simplistic architecture. Conversely, the latency of multi-domain models is relatively large due to their intricate architecture and learning algorithm. For semantic models, the latency of P5 [6] is much larger since the training process of PLMs is extremely time-consuming. In contrast, CTRL [10] and our approach are much more efficient. This is because we can cache the textual representations of the LLMs, and only the lightweight feature interaction backbone needs to be deployed in the inference stage, thereby preserving the efficient online inference akin to traditional recommendation models.

3.3 Further Analysis

Zero-shot Learning Analysis To show the transfer ability of UFIN, we evaluate the zero-shot performance of five methods (*i.e.,* EulerNet [24], P5 [6], CTRL [10], STAR [20] and our proposed UFIN), and compare the results to the best performance of fully-trained single-domain methods. In this setting, we train the model on three pre-trained datasets (*i.e., Movies*, *Books* and *Electronics*) and directly test them on two downstream datasets (*i.e., Instruments* and *Toys*) without further training. The downstream datasets retain only the interactions involving **overlapped users** from the pre-trained datasets. As shown in Figure 2(a), UFIN achieves the best zero-shot performance on both datasets. On the Instruments dataset, UFIN performs even better than the fully-trained single-doamin methods. These results demonstrate the strong transferability and inductive capability of our approach in learning general knowledge across different domains.

Table 3: Performance comparison of different CTR models. "LL" denotes the LogLoss "*" denotes that statistical significance for $p < 0.01$ compare to the best baseline. Note that a higher AUC or lower Logloss at 0.001-level is regarded significant, as stated in previous studies [7, 21].

Eval.	Dataset	Metric	Single-Domain Methods				Multi-Domain Methods					Semantic Methods			
			DeepFM	xDeepFM	DCNV2	EulerNet	MMoE	HMoE	AESM^2	STAR	PEPNet	P5	CTRL	UFIN$_t$	UFIN$_{t+f}$
Multi-Domain	Movies	AUC	0.8568	0.8565	0.8537	0.8550	0.8440	0.8445	0.8477	0.8448	0.8456	0.6440	0.8435	0.8515	**0.8582***
		LL	0.2670	0.2738	0.2698	0.2685	0.2751	0.2755	0.2726	0.2738	0.2746	0.3760	0.2736	0.2741	**0.2649***
	Books	AUC	0.8107	0.8094	0.8118	0.8166	0.8095	0.8097	0.8108	0.8057	0.8080	0.5740	0.8058	0.8130	**0.8215***
		LL	0.2486	0.2492	0.2484	0.2458	0.2509	0.2500	0.2489	0.2507	0.2505	0.3332	0.2549	0.2514	**0.2437***
	Elect.	AUC	0.7493	0.7504	0.7508	0.7524	0.7455	0.7420	0.7438	0.7277	0.7386	0.5762	0.7376	0.7337	**0.7532**
		LL	0.3207	0.3192	0.3186	**0.3181**	0.3217	0.3227	0.3218	0.3272	0.3272	0.3751	0.3218	0.3336	0.3185
	Food	AUC	0.7198	0.7208	0.7204	0.7188	0.7103	0.7117	0.7154	0.7139	0.7135	0.5899	0.7188	0.7231	**0.7322***
		LL	0.2938	0.2902	0.2944	0.3064	0.2841	0.2829	0.2822	0.2810	0.2836	0.3658	0.3034	0.2903	**0.2774***
	Instru.	AUC	0.5811	0.5879	0.5789	0.6056	0.6275	0.6301	0.6587	0.6618	**0.6780**	0.5623	0.5978	0.6635	0.6768
		LL	0.4921	0.4751	0.4514	0.3070	0.1809	0.1717	0.1714	**0.1711**	0.1722	0.2494	0.4929	0.1720	0.1798
	Office	AUC	0.7391	0.7405	0.7425	0.7401	0.7334	0.7524	0.7522	0.7788	0.7554	0.5755	0.7432	0.7590	**0.7812***
		LL	0.2334	0.2503	0.2077	0.2353	0.2038	0.2018	0.2050	0.1974	0.2024	0.2743	0.2721	0.2179	**0.1945***
	Toys	AUC	0.7781	0.7821	0.7856	0.7820	0.7666	0.7799	0.7806	0.7725	0.7717	0.5772	0.7623	0.7768	**0.7964***
		LL	0.2196	0.2177	0.2207	0.2277	0.2164	0.2141	0.2116	0.2124	0.2145	0.2937	0.2218	0.2262	**0.2061***
Cross-Platform	ML-1M	AUC	0.8973	0.8969	0.8989	0.9018	0.8967	0.8978	0.8963	0.8964	0.8970	0.7840	0.8970	0.9024	**0.9029***
		LL	0.3166	0.3189	0.3147	0.3086	0.3183	0.3183	0.3206	0.3167	0.3151	0.4482	0.3155	0.3096	**0.3053***
Efficiency	Amazon	Lat.	0.0238	0.0345	0.0270	0.0323	0.0357	0.1111	0.1250	0.0400	0.0370	6.5737	0.0714	0.0385	0.0476
	ML-1M	Lat.	0.0247	0.0352	0.0264	0.0349	0.0394	0.1289	0.1470	0.0564	0.0477	6.5737	0.0852	0.0453	0.0594



(a) Performance under the zero-shot setting.

(b) Cross-platform Performance

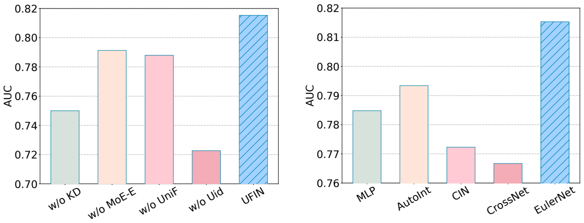
Fig. 2: Transfer learning analysis of UFIN.

Cross-Platform Learning Analysis We investigate the effectiveness of the proposed LLM based representation approach. Specially, we examine whether the model pre-trained on the Amazon datasets performs better than those without pre-training. We compare the performance of the following representation methods: (1) *Embedding Look-up*: use an embedding vector to represent each feature, (2) *PLM based encoding*: encode the textual data with PLMs.

The results are shown in Figure 2(b). We observe that PLM-based methods perform better when pre-training is applied, whereas the Embedding Look-up method is negatively affected by pre-training. It indicates that natural language is more promising as the general representations across different scenarios. Besides, the approach with T5 [17] and FLAN-T5 [4] largely outperforms other methods, showing the excellent language modeling capacity of LLMs.

Ablation Study In this part, we analyze the impact of each proposed technique or component on the model performance. We propose four variants as: (1) *w/o KD* removing the knowledge distillation procedure, (2) *w/o MoE-E* removing the MoE model of the encoder ($\boldsymbol{z} = \boldsymbol{s}$ in Eq. (2)), (3) *w/o UniF* without generating universal features ($\boldsymbol{e}_j$ in Eq.(4)), *i.e.,* directly feeding the latent vector $\tilde{\boldsymbol{z}}$ (Eq. (4)) into an MLP, and (4) *w/o Uid* without fusing *user_id* features (See Section 2.1). We show the results of ablation study in Figure 3(a). We observe that all the proposed components are effective to improve the model performance.

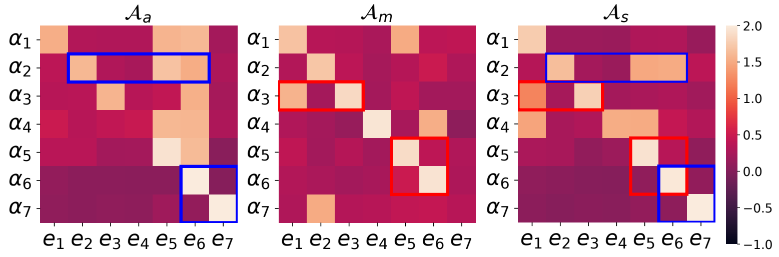
Besides, we explore the effect of the feature interaction experts (Section 2.2). We show the results of five methods (*i.e.,* MLP, AutoInt [21], CIN [12], CrossNet [25], EulerNet [24]) in Figure 3(b). We can observe that choice of EulerNet achieves the best performance, indicating that adaptive feature interaction learning is the key component to improve the model capacity for domain adaptation.



(a) Components (b) Feature Interaction Expert

Fig. 3: Ablation study of UFIN on the Amazon dataset.

Visualizing the Universal Feature Interactions As discussed in Section 2.2, UFIN captures the common interactions to learn the relatedness between different domains. To verify this, we visualize the interaction orders (*i.e.,* $\mathcal{A}_j$ in Eq.(6)) of the Top-1 expert for the Amazon_movies, MovieLens in Figure 4(a) and (b) respectively. We can observe that their learned interactions exhibit substantial differences. We visualize one of their shared experts in Figure 4(c). Notably, the orders learned in the shared expert exhibit some intersections with domain-specific expert of Amazon_movies (*i.e.,* $\boldsymbol{\alpha}_2, \boldsymbol{\alpha}_6, \boldsymbol{\alpha}_7$) and MovieLens (*i.e.,* $\boldsymbol{\alpha}_3, \boldsymbol{\alpha}_5, \boldsymbol{\alpha}_6$). These shared feature interactions enhance the model transferability and enable it to capture more generalized collaborative knowledge for CTR predictions.



(a) Amazon_movies (b) MovieLens (c) Shared Expert

Fig. 4: Visualization of the feature interactions.

4 Related work

CTR Prediction Models. Predicting the probability of users clicking ads or items is critical in online advertising and recommender systems. Traditional CTR models [7, 21] often conduct feature interactions to capture the underlying correlation within a single-domain. In real-world platforms, the user behavior data are often collected from multi-domains. Due to the *seesaw phenomenon* [2], it is limited for traditional approaches to be applied into such scenario. To address this issue, a lot of approaches [2, 20, 26, 27] are proposed to enhance the information sharing and improve the multi-domain performance. Some work [2] extends the multi-task learning (MTL) methods into multi-domain scenarios, by regarding each domain as a specific task. However, they can not exploit the domain relationship and suffer from the degradation of model capability. As a promising direction, recent studies propose to introduce the domain-shared parameters to compactly learn the common knowledge across different domains. However, they still rely on the feature IDs, which impairs the inherent semantic of features and makes it difficult to be applied in different platforms.

Semantic CTR Prediction Models. Recently, the large language models have shown excellent performance in various downstream tasks, which promises researchers to apply LLMs into the recommendation tasks. For example, P5 [6] converts different recommendation task into text generation and employs T5 [17] model to generate the result. Besides, M6-Rec [5] uses the M6 [14] model to deliver the recommendation in a prompting learning framework. Despite the progress, they can not capture the collaborative patterns that severely limits the model capacity. As a promising approach, recent study [10] proposes a contrastive learning framework to align the knowledge between LLMs and collaborative models. Due to the limited capacity of collaborative model, it can not adequately learn the common knowledge across different domains.

5 Conclusion

In this paper, we propose the Universal Feature Interaction Network (**UFIN**) for multi-domain CTR prediction. Unlike previous approaches that heavily rely on modeling ID features for developing the CTR predictions, our approach leverage the textual data to learn the universal feature interactions. Specifically, we regard the text and features as two modalities that can be mutually converted. As such, we employ a LLM-based encoder-decoder architecture to transform the data from text modality to feature modality, obtaining universal feature representations. Building upon this foundation, we design an adaptive feature interaction model enhanced by a mixture-of-experts (MoE) architecture for capturing the generlized feature interactions across different domains. To effectively learn the collaborative patterns across different domains, we propose a multi-domain knowledge distillation framework to improve the training of our approach. For future work, we will consider incorporating the Conversion Rate (CVR) prediction task into our approach to capture more effective correlation between different tasks.

Acknowledgement

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205 and 62222215, and Beijing Municipal Science and Technology Project No. Z231100010323009.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Chang, J., Zhang, C., Hui, Y., Leng, D., Niu, Y., Song, Y.: Pepnet: Parameter and embedding personalized network for infusing with personalized prior information. arXiv preprint arXiv:2302.01115 (2023)
3. Cheng, W., Shen, Y., Huang, L.: Adaptive factorization network: Learning adaptive-order feature interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3609–3616 (2020)
4. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al.: Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416 (2022)
5. Cui, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: M6-rec: Generative pretrained language models are open-ended recommender systems. arXiv preprint arXiv:2205.08084 (2022)
6. Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y.: Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In: Proceedings of the 16th ACM Conference on Recommender Systems. pp. 299–315 (2022)
7. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247 (2017)
8. Hou, Y., Mu, S., Zhao, W.X., Li, Y., Ding, B., Wen, J.R.: Towards universal sequence representation learning for recommender systems. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 585–593 (2022)
9. Li, J., Wang, M., Li, J., Fu, J., Shen, X., Shang, J., McAuley, J.: Text is all you need: Learning language representations for sequential recommendation. arXiv preprint arXiv:2305.13731 (2023)
10. Li, X., Chen, B., Hou, L., Tang, R.: Ctrl: Connect tabular and language model for ctr prediction. arXiv preprint arXiv:2306.02841 (2023)
11. Li, Z., Cheng, W., Chen, Y., Chen, H., Wang, W.: Interpretable click-through rate prediction through hierarchical attention. In: Proceedings of the 13th International Conference on Web Search and Data Mining. pp. 313–321 (2020)
12. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1754–1763 (2018)
13. Lin, J., Dai, X., Xi, Y., Liu, W., Chen, B., Li, X., Zhu, C., Guo, H., Yu, Y., Tang, R., Zhang, W.: How can recommender systems benefit from large language models: A survey (2023)
14. Lin, J., Men, R., Yang, A., Zhou, C., Ding, M., Zhang, Y., Wang, P., Wang, A., Jiang, L., Jia, X., et al.: M6: A chinese multimodal pretrainer. arXiv preprint arXiv:2103.00823 (2021)

15. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: 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. pp. 1930–1939 (2018)
16. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763 (2021)
17. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research **21**(1), 5485–5551 (2020)
18. Rendle, S.: Factorization machines. In: 2010 IEEE International conference on data mining. pp. 995–1000. IEEE (2010)
19. Rendle, S., Krichene, W., Zhang, L., Anderson, J.: Neural collaborative filtering vs. matrix factorization revisited. In: Fourteenth ACM conference on recommender systems. pp. 240–248 (2020)
20. Sheng, X.R., Zhao, L., Zhou, G., Ding, X., Dai, B., Luo, Q., Yang, S., Lv, J., Zhang, C., Deng, H., et al.: 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. pp. 4104–4113 (2021)
21. Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., Tang, J.: Autoint: Automatic feature interaction learning via self-attentive neural networks. arXiv preprint arXiv:1810.11921 (2019)
22. Tang, H., Liu, J., Zhao, M., Gong, X.: Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In: Proceedings of the 14th ACM Conference on Recommender Systems. pp. 269–278 (2020)
23. Tian, Z., Bai, T., Zhang, Z., Xu, Z., Lin, K., Wen, J.R., Zhao, W.X.: Directed acyclic graph factorization machines for ctr prediction via knowledge distillation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. pp. 715–723 (2023)
24. Tian, Z., Bai, T., Zhao, W.X., Wen, J.R., Cao, Z.: Eulernet: Adaptive feature interaction learning via euler's formula for ctr prediction. arXiv preprint arXiv:2304.10711 (2023)
25. Wang, R., Shivanna, R., Cheng, D., Jain, S., Lin, D., Hong, L., Chi, E.: Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In: Proceedings of the Web Conference 2021. pp. 1785–1797 (2021)
26. Zhang, Y., Wang, X., Hu, J., Gao, K., Lei, C., Fang, F.: Scenario-adaptive and self-supervised model for multi-scenario personalized recommendation. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 3674–3683 (2022)
27. Zhou, J., Cao, X., Li, W., Bo, L., Zhang, K., Luo, C., Yu, Q.: Hinet: Novel multi-scenario & multi-task learning with hierarchical information extraction. arXiv preprint arXiv:2303.06095 (2023)
28. Zou, X., Hu, Z., Zhao, Y., Ding, X., Liu, Z., Li, C., Sun, A.: Automatic expert selection for multi-scenario and multi-task search. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1535–1544 (2022)