

A Unified Framework for Multi-Domain CTR Prediction via Large Language Models

ZICHUAN FU*, City University of Hong Kong, China
XIANGYANG LI*, Huawei Noah's Ark Lab, China
CHUHAN WU*, Huawei Noah's Ark Lab, China
YICHAO WANG, Huawei Noah's Ark Lab, China
KUICAI DONG, Huawei Noah's Ark Lab, Singapore
XIANGYU ZHAO†, City University of Hong Kong, China
MENGCHEN ZHAO, Huawei Noah's Ark Lab, China
HUIFENG GUO, Huawei Noah's Ark Lab, China
RUIMING TANG†, Huawei Noah's Ark Lab, China

Multi-Domain Click-Through Rate (MDCTR) prediction is crucial for online recommendation platforms, which involves providing personalized recommendation services to users in different domains. However, current MDCTR models are confronted with the following limitations. Firstly, due to varying data sparsity in different domains, models can easily be dominated by some specific domains, which leads to significant performance degradation in other domains (i.e., the "seesaw phenomenon"). Secondly, when new domain emerges, the scalability of existing methods is limited, making it difficult to adapt to the dynamic growth of the domain. Traditional MDCTR models usually use one-hot encoding for semantic information such as product titles, thus losing rich semantic information and leading to insufficient generalization of the model. In this paper, we propose a novel solution Uni-CTR to address these challenges. Uni-CTR leverages Large Language Model (LLM) to extract layer-wise semantic representations that capture domain commonalities, mitigating the seesaw phenomenon and enhancing generalization. Besides, it incorporates a pluggable domain-specific network to capture domain characteristics, ensuring scalability to dynamic domain growth. Experimental results on public datasets and industrial scenarios show that Uni-CTR significantly outperforms state-of-the-art (SOTA) models. In addition, Uni-CTR shows significant results in zero shot prediction. Code is available at https://anonymous.4open.science/r/multi-domain.

CCS Concepts: • **Information systems** → **Recommender systems**; **Language models**.

*Co-first authors with equal contributions.
†Corresponding authors.

Authors' addresses: Zichuan Fu, zc.fu@my.cityu.edu.hk, City University of Hong Kong, Hong Kong, China; Xiangyang Li, lixiangyang34@huawei.com, Huawei Noah's Ark Lab, Shenzhen, Guangdong, China; Chuhan Wu, wuchuhan1@huawei.com, Huawei Noah's Ark Lab, Shenzhen, Guangdong, China; Yichao Wang, wangyichao5@huawei.com, Huawei Noah's Ark Lab, Shenzhen, Guangdong, China; Kuicai Dong, dong.kuicai@huawei.com, Huawei Noah's Ark Lab, Singapore; Xiangyu Zhao, xianzhao@cityu.edu.hk, City University of Hong Kong, Hong Kong, China; Mengchen Zhao, zhaomengchen@huawei.com, Huawei Noah's Ark Lab, Shenzhen, Guangdong, China; Huifeng Guo, huifeng.guo@huawei.com, Huawei Noah's Ark Lab, Shenzhen, Guangdong, China; Ruiming Tang, tangruiming@huawei.com, Huawei Noah's Ark Lab, Shenzhen, Guangdong, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Copyright held by the owner/author(s).
ACM 1558-2868/2024/10-ART
https://doi.org/10.1145/3698878

Just Accepted

Additional Key Words and Phrases: Click-Through Rate Prediction, Multi-Domain Learning, Large Language Model, Recommender System

1 INTRODUCTION

Click-Through Rate (CTR) prediction aims to estimate the likelihood of a person engaging with an advertisement or item by clicking on it. This estimation is essential for many online recommendation platforms, including e-commerce, advertising, social networking, and multimedia services [57, 82]. The recent advent of digitization has led to the expansion of commercial platforms, diversifying the range of services and products across numerous domains [16]. They offer services such as online shopping, ride-sharing, food delivery, and virtual event hosting, providing users with more options and greater convenience [79]. Therefore, their recommender systems need to serve various scenarios and generate accurate CTR predictions for them.

Since the data from different domains usually have quite different characteristics, simply mixing multi-domain data is often sub-optimal [45]. Thus, traditional single-domain CTR systems [20, 27, 41, 54, 62, 75] typically train separate models based on data from different domains. However, this approach often overlooks the shared characteristics inherent in data across different domains. For instance, if users exhibit a preference for purchasing sportswear in online shopping, they are also likely to be interested in sports-related content for video streaming. Ignoring such shared information can lead to sub-optimal model performance and loss of cross-domain knowledge. Furthermore, training a model for each domain would significantly increase the resource and manpower costs of the system.

To mitigate this issue, Multi-Domain CTR (MDCTR) systems [5, 31, 45, 61, 67, 81] have been developed to make accurate predictions across various domains. The ultimate goal of MDCTR systems is to provide personalized recommendation services for users in different domains. However, the current MDCTR systems encounter challenges in achieving this goal due to the following reasons:

1) **Sparsity of Domain Data.** In practice, many domains do not accumulate sufficient training data, which means these domains are lack of users, items and click feedback from user-item interactions. Therefore, models can easily be dominated by data-rich domains, which leads to significant performance degradation in other domains (i.e., the "seesaw phenomenon"). As a result, the performance of conventional MDCTR models in many domains is not satisfactory, especially on long-tailed samples.
2) **Limited Scalability.** Traditional multi-domain methods often have to either fine-tune the entire existing model or train a completely new model when a new domain is introduced [44]. STAR [61] uses topology to address this challenge, which performs an element-wise multiplication operation between the outputs of the backbone network and the domain-specific network to obtain the final output. However, such design requires the size and structure of domain-specific networks to be identical to those of the backbone network. Therefore, as the number of domains increases, the number of model parameters increases multiplicatively.
3) **Weak Generalization.** In traditional multi-domain methods, features are typically transformed into discrete IDs as inputs; such a conversion process often results in the loss of semantic information associated with these data (e.g., computer and keyboard). When encountering a cold-start recommendation situation, the MDCTR model is difficult to quickly generalize to new domains (i.e., zero-shot prediction), leading to sub-optimal results.

To address the above challenges, we propose Uni-CTR, a unified framework for Multi-Domain CTR Prediction that incorporates large language models as bridges among different domains. The foundation of Uni-CTR is using natural language as a general information carrier so that knowledge from different domains can be universally encoded and exploited. Specifically, Uni-CTR employs a well-curated *prompting strategy* to convert non-textual and textual features into a correlated prompt sequence, which preserves rich semantic and contextual information. Secondly, Uni-CTR utilizes an *LLM backbone* to capture commonalities of all domains, the incorporated *domain-specific networks* (DSN) to learn characteristics of different domains. Specifically, DSN can utilize contextual

Just Accepted

knowledge from various layers of LLM since the bottom layers of the LLM learn surface phrase-level features, while higher ones focus more on understanding more complex and semantic concepts [28]. Besides, the DSN network is pluggable and can be flexibly added or removed when new domains emerge or old domains become obsolete, greatly increasing the scalability of the model. Specifically, we design a *masked loss strategy* by masking the loss back-propagated to different parts of Uni-CTR networks. The gradient of each sample is used to update: (i) its corresponding domain-specific network only, rather than all domain-specific networks, and (ii) the LLM backbone and the general network. Our masked loss strategy ensures the decoupling between domain-specific networks. It has two advantages: (i) alleviating the domain seesaw problem because common and distinct features are modeled separately, and (ii) improving the system's scalability as each domain-specific network becomes pluggable with respect to the LLM backbone. Therefore, while training a new domain or fine-tuning a specific existing domain, the parameters of other domain-specific networks will not be affected. Lastly, a *general network* to further refine the commonalities for zero-shot prediction.

In summary, our contributions are three-fold:

- We propose Uni-CTR, an LLM-based model, for multi-domain CTR prediction. It utilizes domain-specific networks to capture domain characteristics and LLM to capture domain commonalities greatly mitigates the seesaw phenomenon. To the best of our knowledge, this is the first utilization of a large language model as the backbone for MDCTR prediction.
- We introduce a masked loss strategy to ensure the decoupling of domain-specific networks relative to the LLM backbone. This allows DSNs to be pluggable so they can be flexibly added or removed as new domains are added or old domains are removed.
- Experimental results indicate that Uni-CTR outperforms the baseline with an impressive margin regarding prediction accuracy and zero-shot prediction capabilities. This holds for both public and industrial datasets. These findings confirm the effectiveness of our approach and its potential for wider application in MDCTR prediction.

The following of this paper is organized as follows: in Section 2, we provide an overview of the existing research on traditional single-domain and multi-domain CTR prediction, and LLM-based CTR models. A preliminary background of MDCTR prediction is presented in Section 3. Section 4 elaborates on the architecture of Uni-CTR. Section 5 delves into the experimental design and a comparative analysis with existing models, underscoring our model's performance and zero-shot prediction ability. Furthermore, our model is also verified on an industrial dataset, shown in section 6. Finally, Section 7 wraps up the paper by summarizing our contributions and outlining avenues for future research.

2 RELATED WORK

2.1 Click-Through Rate (CTR) prediction

Click-through rate (CTR) prediction is a task to forecast the probability that a user will click on a given item, such as an advertisement or a product. Typically, CTR prediction models employ advanced algorithmic models [6, 7, 20, 31, 35, 36, 42, 87] to analyze user-item data and contextual information, thereby accurately estimating the likelihood of user engagement with specific content. User and item features for CTR prediction modeling typically include user personal information, browsing and purchasing history, user interaction patterns, and item attributes. These features allow for a subtle and nuanced understanding of user preferences and behaviors, improving the accuracy of personalized content recommendations. The evolution of CTR prediction models has progressed from multivariate statistical approaches [9] to factorization machines (FMs) [55], deep learning [11, 14] and hybrid techniques, reflecting advancements in capturing complex feature interactions and patterns in increasingly large industry scenario.

Just Accepted

Factorization Machines (FMs) [55] initially dominated the landscape of CTR prediction models. Their ability to capture interactions between features and computational efficiency made them particularly effective in handling sparse datasets. With the birth of deep neural networks (DNNs), there was a significant shift in CTR prediction methodologies. This era witnessed the emergence of models that synergized FMs and DNNs, harnessing the strengths of both approaches. For example, DeepFM [20] integrates factorization machines for low-order feature interactions with deep neural networks for high-order feature interactions, offering computational efficiency and superior performance across various tasks. xDeepFM [41] further innovates by introducing the Compressed Interaction Network (CIN) to capture high-order feature interactions efficiently. DIN [87] employs an attention-like module focused on user history, which they call the interest network. Subsequently, Zhou et al. [86] extended DIN to DIEN by incorporating GRU units to capture the temporal evolution of user interests.

With the increase in the number of users and items and the emergence of various business scenarios, the increasing complexity of user-item interactions places demands on finer-grained predictions. This leads to the concept of multi-domain CTR (MDCTR) prediction. MDCTR prediction addresses the need to understand and predict user behavior across various domains for more accurate recommendations.

2.2 Multi-Domain CTR (MDCTR) Prediction

In many commercial click-through rate prediction scenarios (e.g., domain), different user groups (e.g., new and old users) [37], different channel modules of APPs [22, 61], different item categories [89], etc., can be regarded as different domains. There are obvious differences in the data distribution of users and items in different domains. If each domain is built with an independent model, it may ignore the commonalities among domains [3], making it difficult to learn from long-tail domains effectively. Furthermore, training a model for each domain would significantly increase the resource and manpower costs of the system. If we directly mix the samples and train a single model, it will overlook the variability of different domains, decreasing prediction accuracy [45]. In addition, if the data samples are imbalanced across different domains, the model may be dominated by the domains with larger amounts of data, resulting in poor learning performance for the smaller domains. Therefore, adopting MDCTR prediction approaches has become the mainstream solution in the industry [83].

One of the key challenges in MDCTR is the **'Domain Seesaw Phenomenon'** [67]. This challenge arises because, in industrial applications, different domains have different amounts of data. Models are highly susceptible to being dominated by domains with large amounts of data, thus decreasing the accuracy of predictions for other domains with less data. Most existing MDCTR models typically solve this problem by separating the model into multiple domain-specific parts and shared parts to learn domain commonalities and characteristics individually. For example, Ma *et al.* introduced the MMOE model [45], which handles the trade-off between domain commonalities and distinctions using multiple experts in a shared-bottom architecture. PLE [67], a progressive layered extraction model, adjusting the balance between shared and domain-specific components, allowing for a more nuanced approach to handling inter-domain dynamics. While existing models have attempted to address the 'Domain Seesaw Phenomenon' through structural modifications, they still struggle significantly in some data-sparse domains [5]. In contrast, our Uni-CTR leverages the extensive background knowledge of LLMs about the world. This repository of pre-existing knowledge allows our model to handle a wide range of domains, avoiding the seesaw issue.

Another challenge in MDCTR is the **under-utilization of semantic information** due to the reliance on sparse feature embedding of mainstream models, leading to a loss of critical domain context and meaning. To address this challenge, various approaches represent strides in addressing the semantic information under-utilization challenge by introducing domain-aware structures and independent embeddings for each domain. DADNN [22] utilizes a domain-aware structure to preserve the unique characteristics of different domains. Tan et al. [66] explore the non-shared embeddings for each task and domain, reducing their coupling. However, these attempts

Just Accepted

still use traditional feature engineering to enhance the remaining features, which does not essentially address the loss of semantic information. Instead, our model concatenates the input features with additional background texts into the textual form of prompts, which are embedded at the token level through a tokenizer and fed into the LLM, which maximizes the preservation of the original input information.

Furthermore, the **scalability** of MDCTR models is often limited by their architectures and feature embedding methods, which can struggle to adapt to new and evolving domains. Scalability in this context refers to a model's ability to efficiently handle an increasing number of domains or adapt to a new domain without requiring extensive retraining or reconstruction of the entire model. Some existing methods attempt to improve the model's scalability by separating hard-sharing networks with different purposes [40]. For instance, MTMS [66] proposes a unified ranking model that integrates independent embedding layers and unified feature management to improve performance and scalability. STAR [61] adds a separate domain-specific tower for each domain to enable new domains to be joined independently. However, MTMS lacks the ability to model the characteristics of domains. As the number of domains increases, the parameters of STAR models must increase exponentially. Our model overcomes these shortcomings by decoupling LLMs and domain-specific networks to model the commonalities and characteristics of domains, respectively and exhibits good scalability.

2.3 LLM for CTR Prediction

Natural language processing (NLP) has witnessed rapid advancements in language model development. These models can be categorized based on their scale, ranging from **Language Models** (LMs) to **Large Language Model** (LLMs). Initial advancements are marked by models like BERT [12] and RoBERTa [43], which employ pre-training tasks such as Masked Language Model (MLM) and achieve impressive performance in various downstream text comprehension tasks [53]. As the field progressed, the emergence of LLMs like DeBERTa [24], GPT-3 [2], and LLaMA [69] represented a huge leap in language modeling. These models, characterized by their enormous size and capacity, demonstrate incomparable proficiency in text generation and understanding human conversations [85].

The introduction of LLMs in CTR prediction is motivated by the need to leverage both non-textual and textual data [73]. The CTR and MDCTR prediction models mentioned above mainly focus on capturing collaborative filtering signals [56], which typically arise from user-item interactions, such as clicks, views, ratings, or purchases. These interactions are primarily captured using unique user and item IDs, allowing recommendation models to uncover patterns in user behavior and item popularity. While these collaborative signals are highly effective in scenarios where sufficient interaction data is available, they fall short in capturing features in data-sparse or cold-start situations [76]. To overcome this limitation, LLMs offer a solution by encoding semantic knowledge directly from natural language data, such as product descriptions, user reviews, or item attributes. LLMs are extensively pre-trained on huge datasets and thus have vast world knowledge that captures deep user demands, making them highly effective in addressing CTR prediction challenges.

Incorporating BERT and its variants in CTR prediction marked a substantial development. Approaches like CTR-BERT [49] and DCAF-BERT [50] blended textual and numerical data effectively, demonstrating the potential of language modeling for CTR prediction. Similarly, CTR-BERT [49] enhances the click-through rate prediction by employing a dual-tower structure based on separate user and item language models coupled with a distillation scheme.

The latest trend in CTR prediction is marked by the use of **generative LLMs** like GPT [2], LLaMA [69], GLM [13]. M6-Rec [10] revolutionizes the representation of user behavior by treating it as plain text and introduces an innovative prompt tuning method termed "option tuning". CTRL [39] and FLIP [74] challenge the traditional one-hot feature encoding process by leveraging pre-trained language models to assimilate semantic signals and external world knowledge, offering a more nuanced approach. The S&R Multi-Domain Foundation model [19]

Just Accepted

utilizes LLMs to distill domain-invariant text features from queries and items, enhancing CTR predictions in cold start scenarios within online services. Further extending the capabilities of traditional recommendation models, KAR [77] harnesses the power of LLMs for open-world reasoning and factual knowledge extraction and adaptation. Despite significant advancements in utilizing LMs for CTR prediction, the field of multi-domain CTR prediction is still relatively unexplored.

2.4 Feature Engineering for Non-LLM based Systems.

In conventional MDCTR prediction methods that do not use language models, continuous user-item features (e.g., age, income, price) can be directly input into the model since they are numerical in nature. In contrast, categorical ones (e.g., occupation, gender, brand) typically require one-hot encoding. For example, the user-item features $\boldsymbol{x}$ (Occupation=`Doctor`, Title=`Harmonicas (12 ct)`, Brand=`Fun Express`, ...) can be represented as a series of one-hot vectors:

$$\boldsymbol{x} = \underbrace{[1, 0, 0, \ldots, 0]}_{\text{Occupation}} \underbrace{[0, 1, 0, \ldots, 0]}_{\text{Title}} \underbrace{[0, 0, 1, \ldots, 0]}_{\text{Brand}} \ldots\ldots \tag{1}$$

Conventional approaches will map these one-hot encoded features into a dense vector space through an embedding matrix.

These one-hot encoded categorical features are important in collaborative filtering-based recommendation systems. In particular, user and item IDs serve as unique identifiers that encapsulate the historical interactions between users and items, thereby preserving valuable collaborative signals. By leveraging these signals, the system can uncover latent user preferences and identify similar items based on shared interactions [59]. The collaborative information captured by user and item IDs is often challenging to derive from text-based features alone, making these categorical features indispensable for uncovering user-item relationships and enhancing the performance of recommendation systems [18].

However, conventional methods significantly loses crucial semantic information, which is often the key to understanding and distinguishing different domains. Consequently, many existing MDCTR models have not adequately considered such critical semantic information, which hinders their ability to model the commonalities and characteristics across multiple domains.

3 TASK FORMULATION

In this section, we introduce the task formulation of multi-domain CTR (MDCTR) prediction . MDCTR aims to build a model that accurately predicts the probability of users clicking on recommended items across various domains. Different user groups (e.g., new and old users) [37], different channel modules of APPs [22, 61], different item categories [89], etc., can be regarded as different domains. Given an MDCTR dataset with M distinct domains $\boldsymbol{D} = \{d_1, d_2, \ldots, d_M\}$, d_m represents user-item data from a specific domain, denoted as:

$$\mathcal{D}_{d_m} = \left\{ (\boldsymbol{x}_i^{d_m}, y_i^{d_m}), i \in \left| \mathcal{D}_{d_m} \right| \right\}, \tag{2}$$

where $\boldsymbol{x}_i^{d_m}$ represents the user-item feature set and $y_i^{d_m} \in \{0, 1\}$ is the click label, with 1 indicating a click and 0 indicating none. $\left| \mathcal{D}_{d_m} \right|$ denotes the total number of samples in the dataset corresponding to domain d_m.

For a specific domain d_m, when given feature $\boldsymbol{x}_i^{d_m}$ and corresponding binary labels $y_i^{d_m}$, the model $f(\cdot)$ predicts the click-through rate, denoted by $\hat{y}_i^{d_m}$, which can be represented as :

$$\hat{y}_i^{d_m} = P(y_i^{d_m} = 1 \mid \boldsymbol{x}_i^{d_m}, d_m) = f(\boldsymbol{x}_i^{d_m}, d_m). \tag{3}$$

Just Accepted

To align the predictions $\hat{y}_i^{d_m}$ with the actual click labels $y_i^{d_m}$, the binary cross-entropy (BCE) Loss will commonly be used to optimize the model:

$$\mathcal{L} = BCE(y_i^{d_m}, \hat{y}_i^{d_m}) = -\left[y_i^{d_m} \log(\hat{y}_i^{d_m}) + (1 - y_i^{d_m}) \log(1 - \hat{y}_i^{d_m}) \right]. \tag{4}$$

4 THE PROPOSED METHOD

In this section, we elaborate on our proposed Uni-CTR model in detail. First, we provide an overview framework of Uni-CTR in Section 4.1. Then we describe our prompting strategy for semantic encoding in Section 4.2, followed by the Uni-CTR architecture and further prediction in Section 4.3 and Section 4.4, respectively. Finally, a horizontal comparison against exiting MDCTR systems is performed in Section 4.5 to illustrate the advantage of our model design.

4.1 Framework Overview

The whole framework has mainly three parts, prompting (bottom) in Section 4.2, Uni-CTR architecture (top left) in Section 4.3 and customised loss strategy (top right) in Section 4.4, as shown in Fig. 2. During prompting, the input data points (domain, user, and product features) are first converted into natural language sequences using a pre-defined prompt template, as detailed in Fig. 1. These sequences are then passed into the Uni-CTR model:

- Firstly, the prompt sequences are fed into an **LLM backbone** like DeBERTaV3 [23] to obtain contextualized semantic representations, as mentioned in Section 4.3.1.
- Secondly, **domain-specific networks** leverage the LLM representations from different transformer layers through ladder networks (see in Section 4.3.2). Ladder networks are designed to capture fine-grained domain-specific characteristics by receiving intermediate representations from various layers of the LLM. Each domain-specific network is equipped with a unique set of parameters, allowing it to specialize in its respective domain. This modular design also ensures that adding a new domain-specific network for a new domain or updating an existing one does not impact the performance of other domain-specific networks.
- Thirdly, a **general network** utilizes the LLM representations from the final transformer layer to model the commonalities across all domains, as detailed in Section 4.3.3. The general network focuses on extracting features that are shared across all domains, providing a robust foundation for generalization. This component is crucial for zero-shot prediction, where the model encounters new, unseen domains with sparse data.

Finally, a **masked loss strategy**, introduced in Section 4.4, is designed to ensure the decoupling of domain-specific and general networks. Hence, we can incorporate new domains to Uni-CTR without affecting other domain-specific networks.

4.2 Prompt-based Semantic Modeling

For the input of the LLM, we design a prompt-based modeling approach to capture the rich semantic information of text-based features (e.g., product title and brand), instead of simply encoding them as one-hot vectors in traditional approaches [8, 26, 47, 84]. Considering the importance of ID features in capturing collaborative signals, as discussed in Section 2.4, we preserve both user and item IDs within the prompt. Including both is necessary because collaborative signals arise from the interactions between specific users and items.

As demonstrated in Fig. 1, our prompt incorporates feature information from the following three sources:

- **Domain Context (d)** preserves domain information by explicitly appending the domain type at the start of the prompt sequence. This component enables Uni-CTR to comprehend the input domain, thereby effectively learning and distinguishing features between different domains.

Just Accepted

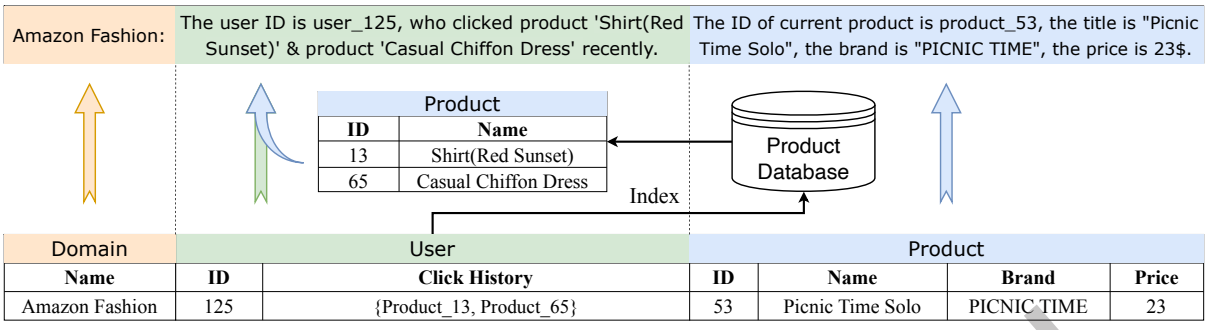


Fig. 1. Prompt Template Design to consolidate domain, user, and product features for Uni-CTR.

- **User Information (u)** captures users' behavioral patterns and preferences, including user IDs and click history. Click history lists the IDs of products recently viewed by the user. Note that we substitute product IDs with their corresponding textual descriptions extracted from the product database.
- **Product Information (p)** exhibits extensive product description, including the unique product ID, title, brand, and price.

Consequently, the input data can be represented by the following aggregation: $x = \{d, u, p\}$, representing domain, user, and product feature, respectively. We design the following prompt template to consolidate these features into textual sequences x_{text}:

$$\begin{aligned} x_{\text{text}} = & \text{[Domain Name]: The user ID is user_[User ID],} \\ & \text{who clicked product '[Product1 Title]' \& product '[Product2 Title]' \& ... recently.} \\ & \text{The ID of the current product is product_[Product ID],} \\ & \text{the title is [Product Name], the brand is [Brand], the price is [Price].} \end{aligned} \tag{5}$$

where [Product Title], [Product ID], [Brand], and [Price] are substituted by information retrieved from the product database.

In our prompt design, we continue to use IDs of users and products due to their significant benefits in traditional recommendation. Firstly, user and product IDs improve recommendation accuracy when there are historical interactions between IDs. Secondly, LLM also has an embedding layer to represent user and product IDs in the form of tokens. This process is similar to the structure of traditional recommendation models and can fully extract ID information. By maintaining these IDs in the prompt, we ensure that our model retains the advantages of traditional ID-based recommendation systems while enriching the semantic context with textual descriptions.

Using our prompt-based semantic modeling design, Uni-CTR is able to preserve all semantic features in the prompt format (textual sequence). Our LLM backbone can subsequently utilize these prompts to capture both commonalities and characteristics of multiple domains.

4.3 Uni-CTR Architecture

As shown in Fig. 2, Uni-CTR mainly comprises three components: **LLM Backbone** (Section 4.3.1), **Domain-Specific Network** (Section 4.3.2), and **General Network** (Section 4.3.3).

4.3.1 ***LLM Backbone.*** As the backbone of Uni-CTR, LLM serves to encode the rich semantic contextual information from the input sequence x_{text} (see in Equation 5).

Just Accepted

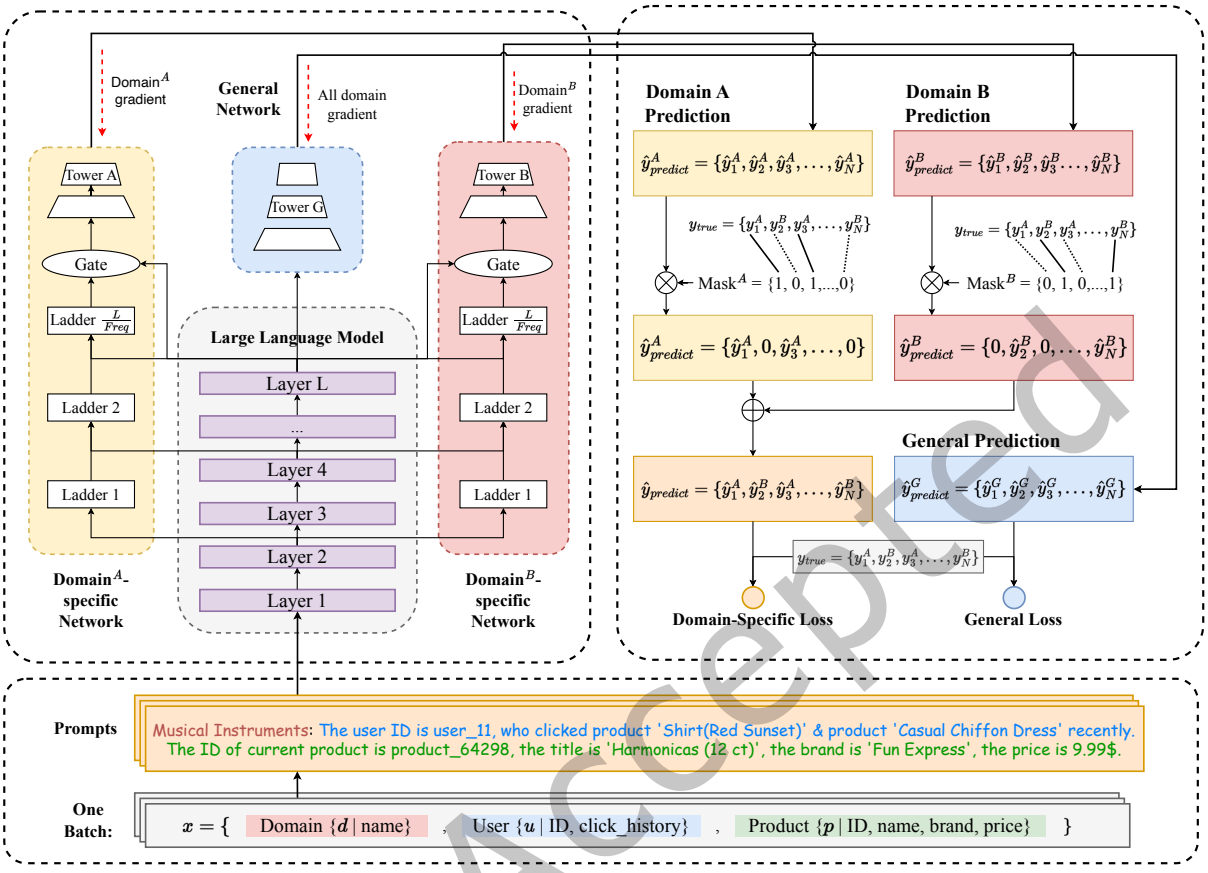


Fig. 2. The architecture of the Uni-CTR, which takes the prompts as the input and obtains semantic representations using an LLM. Among them, domain-specific networks address the characteristics of each domain, while an additional general network aims to extract the commonalities among domains.

Just Accepted

The prompt x_{text} is first tokenized into a sequence of tokens t_j:

$$\boldsymbol{x}_{\text{tokens}} = \text{Tokenizer}(x_{\text{text}}) = \{t_0, t_1, \ldots, t_J\}, \tag{6}$$

where each token t corresponds to a unique ID in a predefined token dictionary.

Next, these tokens are passed into the LLM embedding layer $\boldsymbol{E}_{embed}$, where each token t_j is mapped to a fixed-dimensional vector $\boldsymbol{e}_j$, which contains the semantic and positional information. The initialized representation is written by:

$$\boldsymbol{h}_0 = \boldsymbol{E}_{embed}(\boldsymbol{x}_{\text{tokens}}) = \{\boldsymbol{e}_0, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_J\}. \tag{7}$$

Following the embedding layer, $\boldsymbol{h}_0$ is passed to the LLM's encoder or decoder, which typically consists of multiple transformer layers. Each layer l of the encoder or decoder receives the output $\boldsymbol{h}_{l-1}$ from its preceding layer and produces a new output $\boldsymbol{h}_l$, which can be defined as follows:

$$\boldsymbol{h}_l = \text{Transformer}_l(\boldsymbol{h}_{l-1}), l \in \{1, 2, \ldots, L\}, \tag{8}$$

where we assume that the encoder or decoder of the LLM has $\boldsymbol{L}$ layers. We utilize the representations from the embedding layer and all $\boldsymbol{L}$ transformer layers. The collection of all representations $\boldsymbol{H}$ is described as:

$$\boldsymbol{H} = \{\boldsymbol{h}_0, \boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_L\}, \tag{9}$$

The semantic representations $\boldsymbol{H}$ generated by the LLM backbone can capture the commonalities and coarse-grained characteristics across all domains. In order to model more fine-grained domain characteristics, $\boldsymbol{H}$ is fed into domain-specific networks (see Section 4.3.2) for characteristics of distinct domains.

4.3.2 ***Domain-Specific Network.*** Traditional models often struggle to adapt to continuously evolving user-item features, particularly when integrating new domains. This challenge typically requires significant modifications and retraining of models. To overcome this limitation, we design the Domain-Specific Network (DSN), a modular module that laterally interfaces with the LLM backbone, akin to a ladder. As depicted in Fig. 2, each domain corresponds to a DSN to extract its fine-grained and domain-specific features. Notably, fine-tuning existing DSN or traning a new DSN does not influence other DSNs. The DSN comprises three principal components: (i) the **ladder network** to extract intermediate representations from the LLM backbone, (ii) the **gate network** to regulate the information that passes through the ladder network, and (iii) the **tower network** to make in-domain predictions.

Ladder Network. Existing MDCTR methods such as STAR [61] train multiple domain-specific networks that mirror the size and structure of the backbone network. In contrast, we train a ladder side network [65], a small and independent network that receives intermediate representations $\boldsymbol{H}$ (in Equation 9) through shortcut connections, referred to as ladders, from the backbone LLM.

The number of ladders is proportionally correlated with the number of representations within $\boldsymbol{H}$, as obtained from the LLM backbone. However, introducing one ladder for every component of $\boldsymbol{H}$ would lead to an excessive number of parameters for domain-specific networks. To reduce the size of the ladder network, we define a frequency hyper-parameter ϕ (denoted by '*Freq*' in Fig. 2), and deploy one ladder for every ϕ transformer layers of the backbone LLM. Given a ladder network with $F = \frac{L}{\phi}$ ladders, the output $\boldsymbol{lad}_f$ of the f_{th} ladder is calculated as follows:

$$\boldsymbol{lad}_f = \begin{cases} Ladder_1(\boldsymbol{h}_\phi) & \text{if } f = 1 \\ Ladder_f(\boldsymbol{h}_{f\cdot\phi} + \boldsymbol{lad}_{f-1}) & \text{if } f \in \{2, \ldots, F\} \end{cases}, \tag{10}$$

where $Ladder_f$ can be Multilayer Perceptrons (MLP) [51], Attention Networks [48], Transformer Blocks (with self attention) [72], or any other types of neural networks. Note that $Ladder_1$ takes only the ϕ_{th} representations $\boldsymbol{h}_\phi$ from the LLM backbone as the input, whereas subsequent $Ladder_f$ relies on the representation $\boldsymbol{h}_{f\cdot\phi}$ as well as the output from previous ladder.

Gate Network. For each domain, the target features to be learned comprise (i) common features shared across all domains and (ii) domain-specific features. Specifically, the ladder network's output captures the domain-specific characteristics, while the final hidden states of the LLM backbone capture the commonalities across all domains. Therefore, each domain's predictions need to leverage the ladder and LLM backbone outputs. To facilitate this integration, we design a gate network that follows the ladder network. The gate network is based on a dynamic weight assignment mechanism, aiming to adaptively balance domain-specific and common features.

Firstly, we concatenate the outputs of the last ladder and the final transformer layer of the LLM backbone:

$$\boldsymbol{O} = \text{concat}(\boldsymbol{h}_L, \boldsymbol{lad}_F). \tag{11}$$

Subsequently, we utilize an attention pooling [17] module to compute the attention weights across the concatenated features in Equation 11. This is to adjust the propositions of the domain-specific and common features

Just Accepted

dynamically:

$$\begin{aligned} \boldsymbol{score} &= \tanh(\boldsymbol{W}_k \boldsymbol{O}) \boldsymbol{W}_q, \\ \boldsymbol{A} &= \text{softmax}(\boldsymbol{score}), \\ \boldsymbol{R} &= \boldsymbol{A}^T \boldsymbol{O}, \end{aligned} \tag{12}$$

where $\boldsymbol{W}_k$ and $\boldsymbol{W}_q$ are the learnable parameters for computing the key and query, $\boldsymbol{A}$ is the attention weights normalized from $\boldsymbol{score}$ by a softmax operation. Lastly, $\boldsymbol{R}$ represents the pooled weights, including common and specific features of a certain domain. Therefore, for each domain d_m, the corresponding gate network produces a compressed representation, denoted by $\boldsymbol{R}^{d_m}$.

Tower Network. The last module of the DSN, the tower network, will use $\boldsymbol{R}^{d_m}$ to make final predictions $\hat{\boldsymbol{y}}^{d_m}$ (or $\hat{\boldsymbol{y}}^{d_m}_{\text{predict}}$ in Fig. 2), where the ground truth label is $\boldsymbol{y}^{d_m}$ (or denoted as $\boldsymbol{y}^{d_m}_{\text{true}}$ in Fig. 2). Each domain-specific network has one tower network, which is an MLP network as follows:

$$\hat{\boldsymbol{y}}^{d_m} = \text{MLP}(\boldsymbol{R}^{d_m}; \boldsymbol{W}_\gamma^{d_m}, \boldsymbol{b}_\gamma^{d_m}), \tag{13}$$

where $\boldsymbol{W}_\gamma^{d_m}$ and $\boldsymbol{b}_\gamma^{d_m}$ denote the weights and biases of the MLP network. $\hat{\boldsymbol{y}}^{d_m}$ is the prediction of whether the user will click it on this product or not.

4.3.3 ***General Network.*** In the previous Section 4.3.2, we design distinct DSNs to model and predict within specific seen domains. For Uni-CTR to operate effectively with new, unseen domains, we have constructed a general network that captures the commonalities of all established domains. Specifically, the general network contains only a tower network and directly utilizes the last hidden states h_L of the LLM backbone via MLP network, which is defined as follows:

$$\hat{\boldsymbol{y}}^{G} = \text{MLP}(\boldsymbol{h}_L; \boldsymbol{W}_\sigma^{G}, \boldsymbol{b}_\sigma^{G}), \tag{14}$$

where $\boldsymbol{W}_\sigma^{d_m}$ and $\boldsymbol{b}_\sigma^{d_m}$ denote the weights and biases of the MLP network.

The general network can adapt to situations where no prior data exists, i.e., the data-sparse domains, serving as a zero-shot predictive model. In cold-start recommendation scenarios, this capability is crucial because it allows the network to maintain predictive accuracy without the need to target any domain-specific training data.

4.4 Masked Loss Strategy

As analyzed in Section 4.3.2, the DSNs compute predictions for all domain samples during the training phase. To ensure the independence between DSNs, we employ a masked loss strategy. During training, this strategy ensures that the gradients of each domain's samples are back-propagated to their respective DSN. During inference, it allows DSNs to make predictions solely on samples from their own domains, thereby preventing the computation of redundant predictions by irrelevant DSNs.

4.4.1 Masked Multi-domain Prediction. The prediction of Uni-CTR is based on either a DSN for a known domain or the General Network for an unknown domain. For a data sample from known domains ($d_m \in \boldsymbol{D}$), the predictions of all DSNs is denoted as $\hat{\boldsymbol{y}}^D = [\hat{y}^{d_1}, \hat{y}^{d_2}, \ldots, \hat{y}^{d_M}]$. Note that we have M number of DSNs if there are M domains. To remove irrelevant predictions, we keep only the prediction of one DSN whose domain is the same as d_m via a mask. The mask is generated as follows:

$$\boldsymbol{mask}^{d_m} = \left[I(d_1 = d_m), I(d_2 = d_m), \ldots, I(d_M = d_m) \right], \tag{15}$$

where $I(\cdot)$ is an indicator function that equals 1 if the conditional statement is true and 0 otherwise. Given a data sample from an unknown domain $d_m \notin \boldsymbol{D}$, the prediction of Uni-CTR is determined by the output $\hat{y}^G$ of the

Just Accepted

general network. Accordingly, the prediction of Uni-CTR can be formulated as follows:

$$\hat{y} = \begin{cases} \text{sum}\left(\boldsymbol{mask}^{d_m} \cdot \hat{\boldsymbol{y}}^D\right) & \text{if } d_m \in \boldsymbol{D} \\ \hat{y}^G & \text{if } d_m \notin \boldsymbol{D} \end{cases}. \tag{16}$$

4.4.2 *Masked Multi-domain Training.*

Loss Computation. The loss of each data sample comprises of: (i) domain-specific loss $\mathcal{L}^D$ calculated by the prediction of the domain d_m's DSN, and (ii) general loss $\mathcal{L}^{\text{G}}$ computed based on the general network's prediction. The overall loss can be written as:

$$\mathcal{L} = \mathcal{L}^D + \mathcal{L}^G \tag{17}$$

where the domain-specific loss $\mathcal{L}^D$ can be computed as follows:

$$\begin{aligned} \mathcal{L}^D &= \text{sum}\left(\boldsymbol{mask}^{d_m} \odot \left[\ell(\hat{y}^{d_1}, y), \ell(\hat{y}^{d_2}, y), \ldots, \ell(\hat{y}^{d_M}, y)\right]\right) \\ &= \sum_{i=1}^{M} \left(mask_i^{d_m} \cdot \ell(\hat{y}^{d_i}, y)\right) \\ &= mask_m^{d_m} \cdot \ell(\hat{y}^{d_m}, y) \\ &= \ell(\hat{y}^{d_m}, y), \end{aligned} \tag{18}$$

where y is the ground truth label of the data sample, $\ell(\cdot)$ denotes the Binary Cross-Entropy Loss (BCELoss). The $mask_i^{d_m}$ is the i-th binary element $I(d_i = d_m)$ within $\boldsymbol{mask}^{d_m}$. And the general loss $\mathcal{L}^{\text{G}}$ can be calculated as:

$$\mathcal{L}^G = \ell(\hat{y}_G, y) \tag{19}$$

Loss Back-propagation. The trainable components of Uni-CTR are the LLM backbone, DSNs, and the general network. We denote the parameters of them to be $\boldsymbol{\theta}_{\text{LLM}}$, $\boldsymbol{\theta}_{\text{DSN}}$ and $\boldsymbol{\theta}_{\text{G}}$, respectively. Given a data sample (x^{d_m}, y^{d_m}) from domain d_m, we use the following loss to update the parameters of Uni-CTR:

$$\mathcal{L} = \mathcal{L}^D\left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}\right) + \mathcal{L}^G\left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}}\right) \tag{20}$$

The gradient with respect to the LLM backbone parameters $\boldsymbol{\theta}_{\text{LLM}}$ is calculated as follows:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_{\text{LLM}}}\mathcal{L} &= \nabla_{\boldsymbol{\theta}_{\text{LLM}}}\mathcal{L}^D\left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}\right) + \nabla_{\boldsymbol{\theta}_{\text{LLM}}}\mathcal{L}^G\left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}}\right) \\ &= \frac{\partial \mathcal{L}^D\left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}\right)}{\partial \boldsymbol{\theta}_{\text{LLM}}} + \frac{\partial \mathcal{L}^G\left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}}\right)}{\partial \boldsymbol{\theta}_{\text{LLM}}} \\ &= \frac{\partial \boldsymbol{mask}^{d_m} \odot \left[\ell(\hat{y}^{d_i}, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}^{d_i}), \text{for } i = 1, 2, \ldots, M\right]}{\partial \boldsymbol{\theta}_{\text{LLM}}} + \frac{\partial \ell\left(\hat{y}_G, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}}\right)}{\partial \boldsymbol{\theta}_{\text{LLM}}} \\ &= \frac{\partial \ell(\hat{y}^{d_m}, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}^{d_m})}{\partial \boldsymbol{\theta}_{\text{LLM}}} + \frac{\partial \ell\left(\hat{y}_G, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}}\right)}{\partial \boldsymbol{\theta}_{\text{LLM}}}. \end{aligned} \tag{21}$$

Just Accepted

We then elaborate the gradient for DSN parameters $\boldsymbol{\theta}_{\text{DSN}}$. Note that $\boldsymbol{\theta}_{\text{DSN}} = [\boldsymbol{\theta}_{\text{DSN}}^{d_1}, \ldots, \boldsymbol{\theta}_{\text{DSN}}^{d_M}]$ consists of M sets of DSN parameters. The gradient of each DSN of domain $d_n, n \in \{1, 2, \ldots, M\}$ is is calculated as follows:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_{\text{DSN}}^{d_n}} \mathcal{L} &= \nabla_{\boldsymbol{\theta}_{\text{DSN}}^{d_n}} \mathcal{L}^D \left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}} \right) + \nabla_{\boldsymbol{\theta}_{\text{DSN}}^{d_n}} \mathcal{L}^G \left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}} \right) \\
&= \frac{\partial \mathcal{L}^D \left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}} \right)}{\partial \boldsymbol{\theta}_{\text{DSN}}^{d_n}} \\
&= \frac{\partial \boldsymbol{mask}^{d_m} \odot \left[\ell(\hat{y}^{d_i}, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}^{d_i}), \text{for } i = 1, 2, \ldots, M \right]}{\partial \boldsymbol{\theta}_{\text{DSN}}^{d_n}} \\
&= \begin{cases} 0 & \text{if } n \neq m \\ \frac{\partial \ell(\hat{y}^{d_m}, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}}^{d_m})}{\partial \boldsymbol{\theta}_{\text{DSN}}^{d_m}}. & \text{if } n = m \end{cases},
\end{aligned} \tag{22}$$

which indicates that only parameters of DSN d_m can be updated due to the mask we defined in Equation 15. The parameters of all DSNs other than d_m remain unchanged, ensuring the decoupling of gradient updates across various domains. Therefore, each DSN can independently model the characteristics of each domain without affecting other DSNs.

Finally, the gradient of the loss with respect to the general network parameters $\boldsymbol{\theta}_{\text{G}}$ is computed as follows:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_{\text{G}}} \mathcal{L} &= \nabla_{\boldsymbol{\theta}_{\text{G}}} \mathcal{L}^D \left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{DSN}} \right) + \nabla_{\boldsymbol{\theta}_{\text{G}}} \mathcal{L}^G \left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}} \right) \\
&= \frac{\partial \mathcal{L}^G \left(x^{d_m}; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}} \right)}{\partial \boldsymbol{\theta}_{\text{G}}} \\
&= \frac{\partial \ell \left(\hat{y}_G, y; \boldsymbol{\theta}_{\text{LLM}}, \boldsymbol{\theta}_{\text{G}} \right)}{\partial \boldsymbol{\theta}_{\text{G}}}.
\end{aligned} \tag{23}$$

This gradient ensures that the general network can be trained based on data samples from all domains. Hence, the general network can effectively learn different cross-domain features, and thereby, it can be generalized to unseen domains for accurate zero shot prediction.

In summary, our proposed masked loss strategy ensures that the domain-specific networks are effectively tailored to their respective tasks, while the LLM backbone and the general network learn the domains' commonalities. This architecture enables (i) DSNs to be pluggable and scalable and make accurate predictions on known domains and (ii) the general network to robustly predict unseen domain samples.

4.5 Horizontal Comparison and Discussion

In this section, we conduct a comparative analysis of Uni-CTR against existing MDCTR systems (as summarized in Table 1), emphasizing the advancement and contribution of our proposed method.

The Shared Bottom model [4] utilizes a neural network as the shared bottom layer to capture the common information across different domains while incorporating multiple expert networks in the upper layers to model the specific characteristics of each domain. The structure of Uni-CTR appears similar to the shared bottom structure. However, Uni-CTR differs in two aspects. Firstly, Uni-CTR integrates diverse semantic representations from different layers of the LLM backbone, whereas Shared Bottom only utilizes the representation from the final layer. Secondly, Uni-CTR incorporates a General Network, enabling it to model shared information across all domains. This feature empowers Uni-CTR to be a zero-shot predictor and perform well in unknown domains.

Both MMOE [45] and PLE [67] models target to address the imbalance between different domains. They use multiple experts and gate networks to alleviate the seesaw phenomenon. However, their components are tightly

Just Accepted

Table 1. Comparison of our methodology with several previous research studies. "Semantics" represents the ability of the model to deal with complex semantics. "Balance" represents the ability of the model to deal with see-saw problems. "Generalization" represents the ability of the model to deal with unknown new domains. "Scalability" represents the scalability of the model's structure.

MDCTR Model	Semantics	Balance	Generalization	Scalability
Shared Bottom [4]	×	×	×	✓
MMOE [45], PLE [67]	×	✓	×	×
STAR [61]	×	✓	×	✓
Uni-CTR	✓	✓	✓	✓

coupled, making them hard to scale. Specifically, a new model needs to be built from scratch when incorporating a new domain. Similarly, when a domain becomes obsolete due to business changes, removing the corresponding expert network and tower structure is challenging due to this coupling. In comparison, all DSNs of Uni-CTR are pluggable and decoupled, allowing the model to incorporate new or remove existing domains easily.

The parameter-sharing mechanism in STAR [61] offers a solution with fundamental scalability and balance capabilities. However, it neither understands complex semantics nor works under unseen domains. Moreover, the element-wise computation in STAR requires all DSNs to mirror the size and structure of the shared network. Therefore, STAR can suffer from significant structural latency and complexity when applied to many domains. On the contrary, the parameters of each DSN are significantly less than the parameters of the shared networks in Uni-CTR. The lightweight design of Uni-CTR's DSN makes it easier to be trained, and faster in inference.

5 EXPERIMENTS

In this section, we first elaborate our experimental settings (*e.g.,* datasets, evaluation metrics, baselines, and implementation details) in Section 5.1. Then, we evaluate and discuss of performance and efficiency of Uni-CTR framework against baselines from different perspectives, denoted by six research questions (RQ). In Section 5.2 (RQ1), we present the main results of the performance of Uni-CTR and baseline models. Followed by Section 5.3 (RQ2) to elaborate Uni-CTR's capability in zero-shot prediction under unseen domains. Section 5.4 (RQ3) studies the scaling law issue, investigating the impact of using different LLMs of varying size as the backbone of Uni-CTR. The scalability of Uni-CTR to incorporate a new domain is illustrated in Section 5.5 (RQ4). We visualize the representations of Uni-CTR in Section 5.6 (RQ5), to show the different roles of LLM and DSN in the latent space. In Section 5.7 (RQ6), we perform ablation studies on (i) different combinations of semantic features as input, and (ii) different modules of Uni-CTR framework. In summary, we aim to address the following RQs:

- RQ1 (§ 5.2): Can Uni-CTR outperform existing SOTA baselines in MDCTR prediction across multiple domains?
- RQ2 (§ 5.3): Can Uni-CTR operate effectively in unseen domains and make accurate predictions in a zero-shot manner?
- RQ3 (§ 5.4): Does the scaling law regarding model size apply to Uni-CTR? Does a larger LLM guarantee better performance?
- RQ4 (§ 5.5): Can Uni-CTR scale to incorporate a new domain by simply adding and fine-tuning a new domain-specific network with other parts frozen?
- RQ5 (§ 5.6): What are the functionalities of the LLM and DSNs, and their impacts on overall model performance?
- RQ6 (§ 5.7): How do the main components of Uni-CTR, *i.e.,* semantic modeling prompt, LLM backbone, and DSNs, contribute to Uni-CTR's overall performance?

Just Accepted

5.1 Experimental Settings

5.1.1 Datasets. We utilize the **Amazon Review Data (2018)** [52], a widely used dataset in CTR prediction. It contains records of user interactions with products on the Amazon shopping site from 1996 to 2008. Following previous work [39, 74], we select five categories (Fashion, Musical Instruments, Gift Cards, Digital Music, All Beauty) of products as distinct domains for our experiments.

The statistics regarding the number of users, products, and samples (user-product interactions) for each domain are compiled and presented in Table 2. From Table 2, we can observe that the Fashion, Musical Instruments, and Digital Music domains are comparatively data-rich, as evidenced by the relatively large number of users, products and samples in these domains. In contrast, the Gift Cards and All Beauty domains are data-sparse, indicating insufficient training data. These five domains enable us to comprehensively evaluate our model's performance, reflecting its robustness and adaptability to different types of user-product interactions.

Table 2. Statistics of the Amazon Review dataset across different domains.

Domains	Users	Products	Samples (Interactions)	Percentage (%)
Fashion	749,233	186,189	883,636	19.64
Musical Instruments	903,060	112,132	1,512,530	33.62
Gift Cards	128,873	1,547	147,194	3.27
Digital Music	127,174	66,010	1,584,082	35.22
All Beauty	319,335	32,486	371,345	8.25
Total	2,096,583	397,834	4,498,787	100

The Percentage column represents the proportion of samples (interactions) in each domain relative to the total number of samples across all five domains.

In our experiment, we utilize features including domain names, user IDs, user click history, product IDs, product names, descriptions, and prices. We take the products with ratings above 3 as positive examples and those with ratings equal to or below 3 as negative examples. We follow Li et al. [38, 39], Wang et al. [74] to split the 80%, 10%, and 10% of the dataset for training, validation, and testing, respectively. In classical multi-domain datasets such as Ali-CCP[1] [46] and Ali-Mama[2] [15, 88], features are anonymized. Specifically, all feature values are labeled as IDs, with related natural languages masked. Since LLM-based recommendation relies heavily on the semantic features extracted by the LLM backbone, those ID-based features are unsuitable in our experiment setting. Thus, we exclude Ali-CCP and Ali-Mama from our experiments.

5.1.2 Evaluation Metrics. To evaluate Uni-CTR and other baselines, we use the area under the ROC curve (**AUC**) metric. The ROC (Receiver Operating Characteristic) plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The AUC quantifies the overall ability of the model to discriminate between the positive and negative classes in all confidence levels. Mathematically, it is represented as:

$$AUC = \int_0^1 TPR(FPR^{-1}(u))du, \tag{24}$$

where TPR is the True Positive Rate, FPR^{-1} is the inverse of the False Positive Rate, and u is the threshold. An AUC value of 1 indicates perfect discrimination, while a value of 0.5 indicates that the performance is equivalent to a random guess. Higher AUC values indicate better performance of the model.

[1]https://tianchi.aliyun.com/dataset/408
[2]https://tianchi.aliyun.com/dataset/56

Just Accepted

Additionally, we follow previous work [39, 80, 87] using **RelaImpr** metric to quantify the relative performance improvement over the baseline models. The RelaImpr is calculated as:

$$RelaImpr = (\frac{AUC(measure\ model) - 0.5}{AUC(base\ model) - 0.5} - 1) \times 100\%. \tag{25}$$

5.1.3 Baselines. To rigorously assess the performance of Uni-CTR, we compare it against a diverse array of baseline models. These models are mainly categorized into two groups: Single-domain and Multi-domain CTR models.

In the overall comparison of models, we included all the listed baselines to provide a comprehensive evaluation. In subsequent experiments, due to the need to ensure timely completion of the study, we selected the most representative models so that we can focus on comparisons while maintaining the robustness and reliability of our findings.

Single-Domain Models.

- **PNN.** Qu et al. [54] introduce a product layer, which employs product operations to capture interactions across different feature categories.
- **DCN.** Deep & Cross Network [75] proposes a cross-network explicitly tailored to model bounded-degree feature interactions balancing expressiveness with computational efficiency.
- **DeepFM.** DeepFM [20] integrates Factorization Machines (FM) for low-order feature interactions with deep neural networks for high-order feature interactions.
- **xDeepFM.** xDeepFM [41] innovates by introducing the Compressed Interaction Network (CIN) to capture high-order feature interactions efficiently.
- **DIEN.** The Deep Interest Evolution Network (DIEN) [86] introduces a two-layer structure, the attention-based Interest Extractor Layer and the Interest Evolving Layer, to capture and evolve user interests over time.
- **AutoInt.** AutoInt [62] uses multi-head self-attention networks to capture hierarchical input feature interactions.
- **FiBiNET.** Huang et al. [27] integrates the Squeeze-Excitation network (SENET) mechanism to assess feature importance dynamically and employs a bilinear function to capture complex feature interactions.
- **IntTower.** IntTower [38] is a two-tower model for pre-ranking systems, enhancing interaction between user and item features to improve prediction accuracy and inference efficiency in large-scale applications.
- **TALLRec.** TALLRec [1] is an efficient framework for fine-tuning LLMs with minimal data to significantly enhance their recommendation capabilities and cross-domain generalization. This model is trained using purely semantic information.

Multi-Domain Models.

- **Shared Bottom.** Shared Bottom [4] employs a neural network architecture to extract shared features across various tasks. Additionally, it utilizes a specialized network layer at the top to model unique characteristics specific to each task.
- **MMOE.** MMOE, introduced by Ma et al. [45], is characterized by a shared network of multiple expert submodels and a central gating mechanism that provides implicit connections across diverse tasks with different label spaces.
- **PLE.** Different from MMoE, Progressive Layered Extraction [67] divides experts into task-common and task-specific experts while extending the model from a single-layer network to multiple layers of experts.
- **STAR.** STAR [61] presents a star-shaped structure centered on a shared network, where the outputs of the shared model are multiplied by the domain-specific outputs.

Just Accepted

Table 3. Performance comparison of different models. The boldface denotes the highest score, and the underline indicates the best result of all baselines. ⋆ represents significance level p-value < 0.05 of comparing Uni-CTR (BackBone Sheared-LLama) with the best baselines.

Category	Models	Fashion		Musical Instruments		Gift Cards	
		AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr
Single-domain	PNN	0.6979	27.49%	0.6859	38.19%	0.5959	134.20%
	DCN	0.6985	27.10%	<u>0.6893</u>	35.71%	0.6126	99.47%
	DeepFM	0.6982	27.30%	0.6880	36.65%	0.5937	139.70%
	xDeepFM	<u>0.7031</u>	24.22%	0.6892	35.78%	0.6121	100.36%
	DIEN	0.6995	26.47%	0.6881	36.58%	0.6105	103.26%
	AutoInt	0.7003	25.96%	0.6888	36.07%	0.5976	130.12%
	FiBiNET	0.6770	42.54%	0.6878	36.79%	0.6120	100.54%
	IntTower	0.6988	26.91%	0.6888	36.07%	0.6100	104.18%
	TALLRec	0.6116	126.08%	0.6174	118.82%	0.6035	117.04%
Multi-domain	Shared Bottom	0.6946	29.65%	0.6875	37.01%	0.5907	147.63%
	MMOE	0.6907	32.30%	0.6857	38.34%	0.6104	103.44%
	PLE	0.6842	36.97%	0.6813	41.70%	<u>0.6375</u>	63.35%
	STAR	0.6874	34.63%	0.6831	40.31%	0.6242	80.84%
	SAR-Net	0.6824	38.32%	0.6763	45.72%	0.6055	112.89%
	DFFM	0.6973	27.88%	0.6856	38.42%	0.6324	69.64%
LLM-based Multi-domain	Uni-CTR	**0.7523⋆**	-	**0.7569⋆**	-	**0.7246⋆**	-

It is worth noting that an AUC increase of 0.001 can be considered a significant improvement in CTR prediction [27, 38, 63, 87].

- **SAR-Net.** SAR-Net [60] utilizes attention modules to learn users' cross-scenario interests and employs a scenario-specific linear transformation layer, followed by debias expert networks consisting of scenario-specific and shared experts.
- **DFFM.** DFFM [21] incorporates domain-related information into the parameters of the feature interaction and user behavior modules, allowing for domain-specific learning of these two aspects.

5.1.4 *Implementation Details.*

Implementation of Uni-CTR. For the LLM backbone in Table 3, we use a larger model, Sheared-LLaMA [78], configuration for comparison to demonstrate the upper limit of our model's performance. It comprises 1.3 billion parameters and 24 transformer layers. This configuration highlights the potential of Uni-CTR when utilizing a high-capacity backbone. For other experiments, such as verifying the model zero-shot capability, scalability, we use a smaller model, $\text{DeBERTaV3}_{\text{large}}$ [23], which is mainly due to the consideration of saving computational resources. Additionally, we conduct performance evaluations with various model sizes to understand the scaling law between model complexity and performance. For each domain-specific network (DSN), we employ four ladder layers, each consisting of a small transformer encoder block. For the tower network in both the DSNs and the general network, we use three perceptron layers with dimensions of $\{512 \times 256 \times 128\}$, which help in making final predictions for each domain.

Just Accepted

Implementation of Baselines. For single-domain baselines, the dimension of hidden layers of all MLP classifiers as $\{512 \times 256 \times 128\}$, and other settings can be seen in our open source code. For the multi-domain baselines in Table 3, we summarize their key configurations as follows:

- Shared Bottom [4]: The shared bottom layers have dimensions of $\{1024 \times 1024\}$, whereas the dimensions of the domain-specific layers are $\{1024 \times 512 \times 256\}$.
- MMOE [45]: The dimensions of tower and expert network are $\{1024 \times 1024\}$ and $\{1024 \times 512\}$, respectively. Note that we empirically configure to deploy 3 experts for best performance.
- PLE [67]: The extraction network of PLE consists 1 shared expert and 2 domain-specific experts for each domain. The dimensions of tower and expert network are $\{1024 \times 1024\}$ and $\{1024 \times 512\}$, respectively.
- STAR [61]: The hidden state dimensions of MLPs in auxiliary and star topology networks are $\{1024 \times 512 \times 256\}$.
- SAR-Net [60]: contains 2 shared and 6 specific Debias experts, while the hidden state dimension of each expert is $\{1024 \times 1024\}$. The dimension of MLP classifiers is $\{1024 \times 512\}$.
- DFFM [21]: The dimension of the MLP used for Domain Facilitated Feature Interaction (DFFI) is $\{1024 \times 512 \times 256\}$.
- TALLRec [1]: We follows the same experimental setup as described in the original paper, utilizing the LLaMA-7B [70] model as the base model. The input prompt only uses the text with semantic information, without IDs. The model with Alpaca-LoRA [68] is further fine-tuned on our dataset through Rec-Tuning.

Zero-Shot Setting. In the zero-shot experiment, we compare our Uni-CTR with six single-domain and four multi-domain models. All the models are trained on the three training domains (Fashion, Musical Instruments, and Gift Cards), and tested on the left two domains (Digital Music, All Beauty). 1) For each single-domain baseline, we train three models with an identical design on the three training domains separately. Subsequently, we use these three models to predict the 4^{th} (unknown) domain, resulting in three sets of results. Finally, we choose the best result as the final result of this single-domain baseline. 2) For multi-domain baselines, we train the models on the three training domains. Although these baselines are implemented with multi-domain classifiers, each for a specific domain, they lack a general network structure specifically designed for out-of-domain prediction. To mitigate this issue, we use all three classifiers to predict the 4^{th} domain, resulting in three sets of results. Similar to the approach used for single-domain baselines, we choose the best result as the final result. 3) Uni-CTR diverges from the baseline models by utilizing a general network for out-of-domain prediction. The general network is trained across all three aforementioned domains. Note that the predictions from three DSNs of Uni-CTR are not considered in the out-of-domain scenario.

Optimization and Training. For Uni-CTR, we employ 8 Tesla V100 GPUs with a batch size of 128. To reduce overfitting, we set the dropout [64] rate to 0.3 and utilize L2 regularization [34]. The AdamW [33] optimizer is used for Uni-CTR, and we adopt a Cyclic Learning Rate (CyclicLR) scheme to fluctuate the learning rate between the range of $[1 \times 10^{-6}, 8 \times 10^{-5}]$ To accelerate the training of the LLM backbone, we adopt LoRA [25] with a low rank of 8 and an alpha value of 32. Regarding the optimizer and hyper-parameter selection of our baselines, we follow the default settings mentioned in their original papers, if applicable. Otherwise, we employ the Adam [33] optimizer with the learning rate of 1×10^{-3} and adjust individually for other hyper-parameters. We try our best to reproduce their works to obtain the best results, ensuring fair comparisons.

5.2 Main Results (RQ1)

In this subsection, we compare the performance of Uni-CTR against other single-domain and multi-domain models. The results are summarized in Table 3. From the table, several observations can be obtained:

- **Insights from Single-Domain Models and Existing Multi-domain Models.** The performance of ID-based single-domain models, such as DeepFM and xDeepFM, is surpassed by existing multi-domain models in the

Just Accepted

Fashion and Musical Instruments domains. This demonstrates the effectiveness of single-domain models in data-rich domains. However, single-domain models are significantly inferior to multi-domain models in data-sparse domains such as Gift Cards. This suggests that the joint modeling of multiple domains is beneficial for improving the performance of models in sparse domains. For the single-domain model TALLRec using LLM, it does not incorporate ID information in its design. Consequently, in data-rich domains, its performance is inferior to other traditional single-domain models that can learn collaborative signals from IDs. Nevertheless, TALLRec still shows some effectiveness, indicating that pure semantic information, while not sufficient for CTR prediction, can still enhance existing ID-based recommendation systems. Moreover, traditional multi-domain models still suffer from a serious seesaw problem, where data-rich domains (Fashion and Musical Instruments) try to dominate the model training, leading to degraded performance in both domains.

- **Superiority of Uni-CTR.** Uni-CTR achieved substantial gains in performance across three domains, exhibiting relative improvements of **24.22%**, **35.71%**, and **63.35%** with respect to the AUC metric [3] on Fashion, Musical Instruments, and Gift Cards, respectively. We attribute this to the powerful semantic understanding of the LLM and the powerful characteristic modeling capabilities of DSNs across various domains. Additionally, Uni-CTR effectively addresses the seesaw problem caused by data sparsity. In the Gift Cards domain, Uni-CTR achieves a significant improvement with a margin of 63.35%, substantially surpassing the performance gains observed in the other two data-rich domains. We attribute this enhancement to the pre-existing world knowledge embedded in the LLMs. This pre-trained semantic knowledge effectively compensates for the seesaw problem caused by sparse data in the third domain. Furthermore, the observed performance indicates that the inherent world knowledge and semantic understanding capabilities of the LLMs can provide significant cold-start capabilities in domains where data sparsity may hinder model performance. In the subsequent section, this also inspires us to explore the "zero-shot" capabilities of our model, assessing its potential utility and effectiveness in previously unseen domains.

5.3 Zero-Shot (Cold Start) Prediction (RQ2)

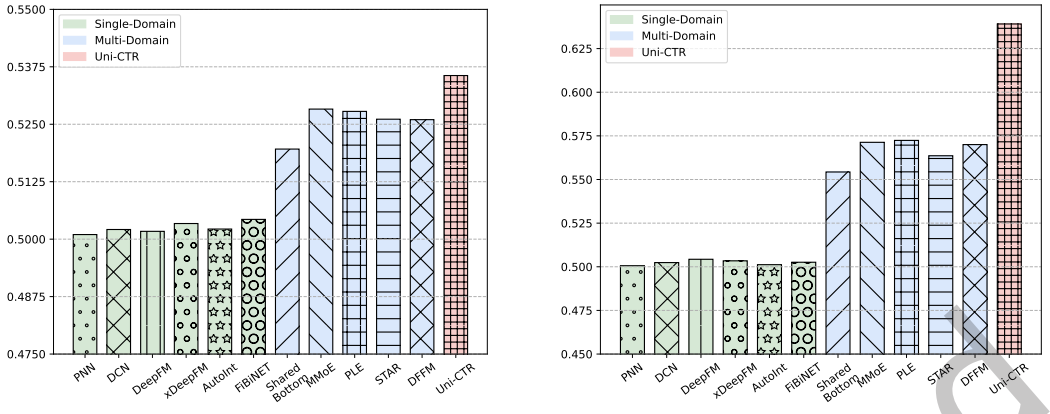
In real-world industrial recommendation systems, it is common for new business domains to emerge, often accompanied by new, unknown items. This typical situation is referred to as the cold-start problem in recommendation systems. In such cases, we lack sufficient training data to train the model. Consequently, the capability of existing models to cope with cold-start problems is of utmost importance. Therefore, this section investigates the model's performance under a zero-shot setting. The experiment is set up in Section 5.1.4. As illustrated in Fig. 3, the single-domain models demonstrate the limited capacity to generalize to new domains, as evidenced by their low AUC values of around 0.51 or below. These models are limited to understanding and predicting only within the data distribution of the domains in which they are trained, resulting in poor performance in zero-shot prediction.

For multi-domain models, we observe a marked enhancement in zero-shot prediction capabilities. We hypothesize that this improvement stems from the model's ability to leverage common information across multiple domains during training, which allows it to generalize to previously unseen domains.

Our Uni-CTR model outperforms all the baseline models, with a notable improvement exceeding **6%** points compared to traditional multi-domain models. This superior performance can be attributed to two main factors: (i) incorporating world knowledge embedded within the LLM aids in bolstering the multi-domain model's performance under a cold-start setting, and (ii) the General Network in Uni-CTR effectively distills common knowledge across multiple domains, ensuring robust generalization capabilities when confronted with new domains.

[3] It is worth noting that an AUC increase of 0.001 can be considered a significant improvement in CTR prediction

Just Accepted



(a) Digital Music (b) All Beauty

Fig. 3. Comparative performance of zero-shot prediction on traditional models and Uni-CTR on the unseen domain.

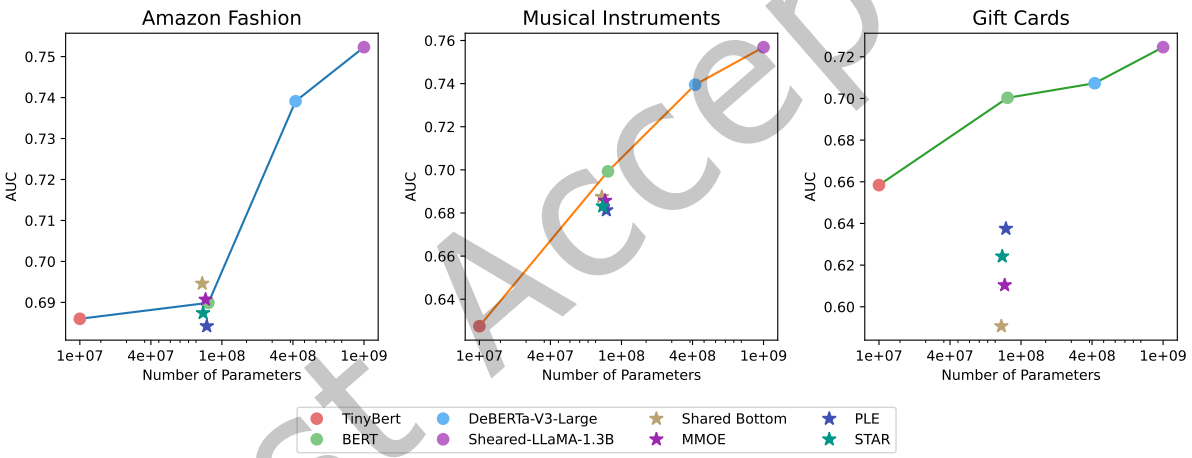


Fig. 4. Performance comparison of different language model backbones.

Just Accepted

5.4 LLM Scale Up (RQ3)

The scaling law [32] is a pivotal principle in training LLMs, describing how various aspects of a model and its training process change with scale, such as model size, dataset size, and computational resources. Specifically, the more parameters a model has and the more extensive training data it is fed, the better performance it exhibits. However, inference latency is also paramount in recommendation systems in the industry. For the CTR prediction task, the inference latency per request is typically constrained to within 10 milliseconds. Therefore, in this subsection, we examine the impact of parameter counts in the LLM backbone on model performance. We employ four language models as backbones: TinyBERT [30], BERT [12], DeBERTaV3$_{\text{large}}$ [23], and Sheared-LLama [78] with respective parameter counts of 14M, 110M, 340M, and 1.3 billion.

The experimental results are summarized in Fig. 4, from which we obtain some observations: 1) With the increasing size of LLMs, a notable enhancement in performance is observed, which indicates that the scaling laws are equally applicable to Uni-CTR. Furthermore, the LLM backbone of Uni-CTR primarily captures the commonalities across domains, which provides a potential avenue for enhancing the effectiveness of traditional multi-domain CTR models in the future. 2) Uni-CTR, based on a BERT Backbone with 110M parameters, has already surpassed traditional multi-domain models. This illustrates the powerful semantic comprehension and world knowledge embedded within language models, leading to significant performance gains. Thus, our model provides a viable approach for application within the industry recommender systems.

5.5 Scalability (RQ4)

Table 4. Performance comparison of different models when scaled to a new domain.

Category	Models	Scalability	Digital Music		All Beauty	
			AUC	RelaImpr	AUC	RelaImpr
Single-domain	PNN	×	0.5904	26.11%	0.6143	32.96%
	DCN	×	0.5919	24.05%	0.6142	33.09%
	DeepFM	×	0.5917	24.32%	0.6121	35.64%
	xDeepFM	×	0.5957	19.12%	0.6176	29.29%
	AutoInt	×	0.5913	24.86%	0.6162	30.83%
	FiBiNET	×	0.5832	37.02%	0.6103	37.84%
Multi-domain	Shared Bottom	✓	0.5975	16.92%	0.6275	19.22%
	STAR	✓	0.6038	9.830%	0.6291	17.74%
	DFFM	×	0.5964	18.25%	0.6124	35.21%
	MMOE, PLE	×	-	-	-	-
LLM-based Multi-domain	Uni-CTR	✓	**0.6140**	-	**0.6520**	-

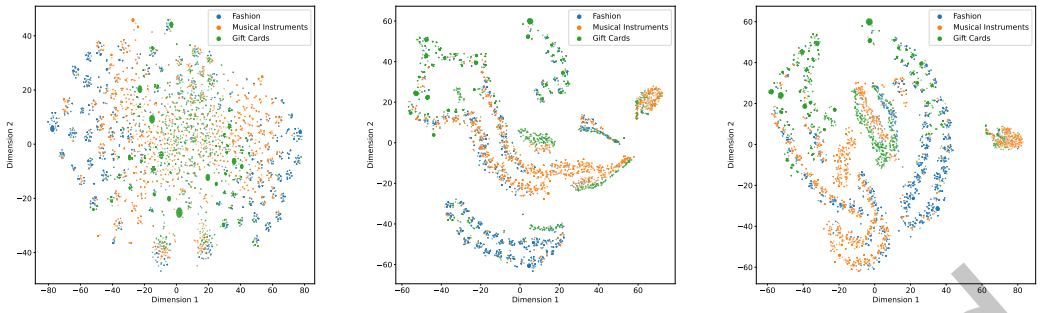
To evaluate the scalability of the model, we freeze the weights of Uni-CTR which has already been trained on three training domains. We then add two additional DSNs for fine-tuning the Uni-CTR on new domains, Digital Music and All Beauty. During training, only the parameters of the newly added DSN are updated. For single-domain models, we retrain it on the new domains separately. As for a scalable model like STAR, we add new network structures to scale it, and the training method is consistent with Uni-CTR. The results are presented in Table 4.

We can observe that by merely training an additional Domain-Specific Network (DSN), the performance of Uni-CTR is improved by 9.830% compared to the multi-domain STAR model, and at least 19.12% compared to the fully retrained single-domain baseline model. This evidence confirms that the Uni-CTR has successfully identified and assimilated the commonalities across different domains during its prior training. Such an understanding of these commonalities facilitates the adaptation of Uni-CTR to new domains, utilizing this prior knowledge as a foundational base. Therefore, even with the original parameters of Uni-CTR frozen, the newly added DSN demonstrates impressive performance. This efficiency not only conserves significant computational resources but also substantially reduces the time required for model training.

5.6 Visualization (RQ5)

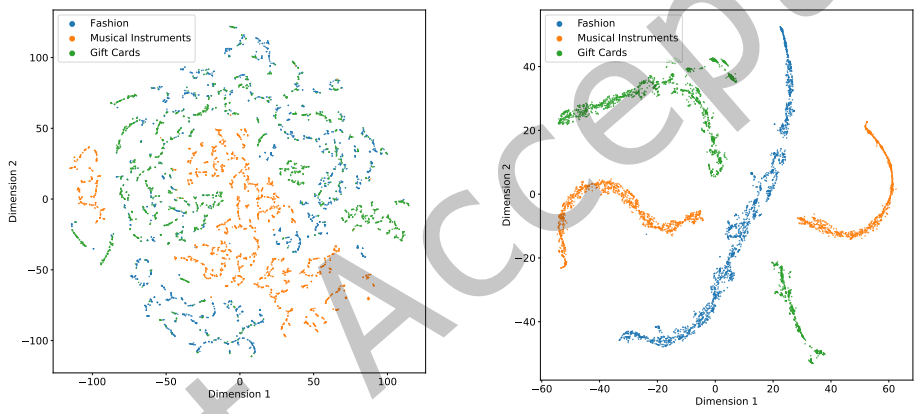
In this subsection, we use t-distributed Stochastic Neighbor Embedding (t-SNE) [71] to visualize representations of LLMs as well as representations of DSNs in latent space to explore their role in modeling multi-domain commonalities and characteristics.

Just Accepted



(a) $\boldsymbol{h}_0$ (b) $\boldsymbol{h}_2$ (c) $\boldsymbol{h}_4$

Fig. 5. Visualization of the representations of different layers of the LLM.



(a) Untrained DSN (b) Trained DSN

Fig. 6. Visualization of the representations of DSNs.

5.6.1 Visualization of LLM Representations. As depicted in Fig. 5, we visualize the representations from the 0^{th}, 2^{nd}, and 4^{th} layers of the LLM, denoted as h_0, h_2, and h_4, respectively. These representations are illustrated as colored dots, with each color corresponding to a different training domain.

From Fig. 5, we can observe two significant results. 1) The representations at different transformer layers of the LLM are distinctly distributed in the latent space, indicating that different layers of the LLM capture semantic representations at various levels. Such finding is in good agreement with the results of previous studies [29, 58]. Therefore, it is necessary to incorporate ladder networks at different LLM layers to model semantic information at multiple levels, a point further substantiated in Section 5.7. 2) As illustrated in Fig. 5a, representations from various domains at the lower layers of LLM are completely aggregated. While at higher layers, as shown in Fig. 5b and Fig. 5c, the representations of different domains gradually start to separate. This demonstrates that

Just Accepted

the LLM is capable of not only modeling the commonalities among multiple domains but also capturing the coarse-grained characteristics of each domain, which helps the DSN network further extract information about the characteristics of the domains.

5.6.2 Visualization of DSN Representations. In order to validate the role of DSNs, we extract the representations from the penultimate layer of the tower network within DSN for each domain, comparing their states before and after training, as shown in Fig. 6. From Fig. 6a, we can observe that the distributions of the representations of the three domains are mixed in the latent space before the DSNs are trained. This indicates that untrained DSNs are incapable of effectively distinguishing the representations of different domains, i.e., they cannot accurately model the characteristics of each domain. After training, as shown in Fig. 6b, the representations of different domains in the latent space are well-separated. This shift demonstrates that DSNs gain the ability to distinguish the representations of various domains and accurately capture the characteristics of each domain through training.

5.7 Ablation Study (RQ6)

To clarify the role of different components in the Uni-CTR model, we conduct ablation studies and the results are summarized in Table 5 and Table 6.

Table 5. Ablation Study Results about the input text of Uni-CTR (Backbone $\text{DeBERTaV3}_{\text{large}}$).

Input Text	Fashion	Musical Instruments	Gift Cards
Full Prompt with History ID	0.7054	0.7017	0.6825
Full Prompt	0.7047	0.7008	0.6825
Only Feature ID and Feature Name	0.6960	0.6960	0.6749
Only Feature ID	0.6951	0.6835	0.6605

5.7.1 Impact of Prompt. To assess the importance of prompt semantics in LLMs, we explore the impact of different prompts on model performance, as detailed in Table 5. The 'Full Prompt with history ID' format includes additional feature IDs of the products clicked in the users' history to maximise the utilization of all information available in the data. The 'Full Prompt' format, the original design for Uni-CTR discussed in Section 4.2, utilizes complete semantic information. The 'Only Feature ID and Feature Name' format combines feature names and feature ID into a single string, removing other background semantic information such as the introduction about the task. The 'Only Feature ID' format further simplifies this by retaining only feature ID values. We compared the results after training the same 5 epochs using Uni-CTR with $\text{DeBERTaV3}_{\text{large}}$ as the backbone LLM. Our findings indicate a marked decline in performance with the reduction of semantic information in the prompts.

The analysis of the results reveals the significance of prompt design in LLMs. In all three domains (Fashion, Musical Instruments, and Gift Cards), the fully semantic 'Full Prompt' format consistently outperformed the other two. Specifically, the performance dropped by 1.23% and 1.36% in the 'Ony Feature ID and Feature Name' and 'Only Feature ID' formats, respectively, for Fashion domain. Moreover, the most pronounced decline was observed in the Gift Cards domain compared to the other two domains. This significant drop can be attributed to the relatively smaller size of the dataset in this domain. The relatively small data size necessitates a higher reliance on rich semantic information to compensate for the lack of extensive data points, making the impact of prompt design more pronounced. In addition, extra history IDs only brings weak improvements in data-intensive domains (Fashion, Musical Instruments) comparing the 'Full Prompt with History ID' and 'Full Prompt' format. This is attributed to the fact that ID-based recommendation relies on historical interaction data between users and products. This paradigm only works well when there is a large amount of interaction data between users

Just Accepted

and products. However, the Amazon dataset is so sparse that the added historical ID information still lacks collaborative signal via users with the ID of the product to be predicted. These results emphasize the critical role of semantic information in enhancing LLMs' understanding and prediction accuracy. The performance of the model decreases as the semantic information decreases, suggesting that LLM relies heavily on semantic information to make accurate predictions. Thus, while simplifying the input format can improve computational efficiency, it clearly affects the performance of the model. Considering these findings, optimizing prompt design is a crucial aspect when deploying LLMs for MDCTR prediction.

Table 6. Ablation Study Results of Uni-CTR (Backbone $\text{DeBERTaV3}_{\text{large}}$).

Model	Fashion	Musical Instruments	Gift Cards
Uni-CTR	0.7391	0.7395	0.7073
w/o ladder	0.7084	0.6975	0.6723
w/o LLM	0.6954	0.6923	0.6100
MMOE(340M)	0.7038	0.7005	0.6712
STAR(340M)	0.7107	0.7016	0.6775

Table 7. Performance comparison of Uni-CTR using domain-specific networks and the general network across three training categories.

Backbone LLM	Prediction Network	Fashion	Musical Instruments	Gift Cards
$\text{DeBERTaV3}_{\text{large}}$	General Network	0.6974	0.7035	0.6991
	Domain-specific Networks	0.7391	0.7395	0.7073
Sheared LLaMA	General Network	0.7244	0.7246	0.7044
	Domain-specific Networks	**0.7523**	**0.7569**	**0.7246**

5.7.2 Impact of Ladder Network and LLM. In order to clarify the role of domain-specific networks (DSNs) as well as the LLM backbone in the Uni-CTR model, we conduct the following ablation study, and the results are summarized in Table 6 and Table 7. 1) We remove the ladder network, restricting the architecture to only utilize the LLM and the three Tower layers for both training and prediction. We observe a considerable degradation in performance, indicative of the essential role that the ladder network plays in leveraging multi-level semantic information from LLM. This suggests that the extraction of characteristic information pertinent to distinct domains benefits from the intricate semantics provided by the ladder network, and relying solely on the LLM is insufficient. 2) We remove the LLM, i.e., we replace the LLM with a DNN with the same number of layers, and the inputs are changed to the traditional IDs. we can observe that when the LLM backbone is lost, not only does the model show a huge loss of performance on all domains, but also the drop is significant on data-sparse domains (Gifts Cards). This shows that LLM plays the most crucial role in improving the performance on data-sparse domains. 3) To verify whether it is the increase in the number of parameters that brings about the performance improvement, we increase the number of parameters of the models of MMOE and PLE to the same size as that of Uni-CTR (340M). The results show that even with the same number of parameters, the performance of the traditional recommendation model is still much weaker than that of Uni-CTR, but better than that of Uni-CTR without LLM or ladder layer, which indicates that both the ladder layer and LLM backbone contribute to the performance of Uni-CTR. 4) To further verify the effectiveness of DSNs, we compared the performance of DSNs

Just Accepted

Table 8. The overall performance of models trained on the industrial dataset.

Category	Models	Domain 0		Domain 1	
		AUC	RelaImpr	AUC	RelaImpr
Single-domain	PNN	0.6735	37.58%	0.6199	56.88%
	DCN	0.6722	38.62%	0.6243	51.33%
	DeepFM	0.6743	36.95%	0.6223	53.80%
	xDeepFM	0.6738	37.34%	0.6226	53.43%
	AutoInt	0.6788	33.50%	0.6214	54.94%
	FiBiNET	0.6780	34.10%	0.6146	64.14%
Multi-domain	MMoE	0.7045	16.72%	0.6640	14.70%
	PLE	0.7019	18.23%	0.6706	10.26%
	STAR	0.7000	19.35%	0.6638	14.84%
LLM-based Multi-domain	Uni-CTR	**0.7387**	-	**0.6881**	-

and the general network across three training domains, as shown in Table 7. The experimental results indicate that the performance of the general network is significantly lower compared to the domain-specific networks. This clearly shows that the absence of domain-specific characteristics adversely affects the model's performance. These findings further demonstrate the necessity of designing a DSN for each domain to capture unique characteristics and improve overall performance.

6 INDUSTRIAL EXPERIMENTS

This section outlines the practical application and empirical evaluation of Uni-CTR in a real-world industrial setting. In preparation for the experiment, we gather and sample one month of user behavior data from a large-scale industrial recommender system. This platform generates millions of user logs daily, providing a substantial and diverse collection of data on user interactions and preferences.

6.1 Model Performance of Industry Platform

In this real-world application, users are divided into two domains based on business requirements, referred to as *Domain 0* and *Domain 1*. The single-domain models are trained on each domain individually, while the multi-domain models and our proposed Uni-CTR are trained conjointly on both domains.

The results are summarized in Table 8. From the table, we observe the following phenomena:

1) In this industrial scenario, multi-domain models outperform single-domain models by a wide margin, which can be attributed to the fact that industrial datasets have higher data sparsity as well as larger data sizes as compared to the Amazon review dataset. Consequently, single-domain models lack sufficient information within each domain, while multi-domain models are better equipped to uncover the domain commonalities, leading to superior performance. In contrast, smaller datasets like the Amazon review dataset provide enough data for single-domain models to perform well without needing the additional commonalities that multi-domain models offer, making the advantage of multi-domain learning less pronounced in such contexts.

2) Trained on real-world industrial datasets, Uni-CTR outperforms SOTA multi-domain models, achieving an impressive relative improvement in AUC of over **10.26%**. This significant performance gain can be attributed to the adoption of text-based input, which not only enhances the model's flexibility but also enriches its semantic

Just Accepted

Table 9. Inference Time per batch and per sample for different models.

Category	Models	Batch Size	per Batch (ms)	per Sample (ms)
Single-domain	PNN	2048	3.126	0.0015
	DCN	2048	3.971	0.0019
	DeepFM	2048	4.225	0.0020
	xDeepFM	2048	4.221	0.0020
	AutoInt	2048	3.972	0.0019
	FiBiNET	2048	14.396	0.0070
Multi-domain	Shared Bottom	2048	2.401	0.0011
	MMoE	2048	1.916	0.0009
	PLE	2048	2.007	0.0009
	STAR	2048	2.142	0.0010
LLM-based Multi-domain	Uni-CTR (TinyBERT)	64	6.522	0.1019
	UniCTR (DeBERTaV3$_{\text{large}}$)	64	348.5	5.4466

understanding. Such rich semantic understanding proves to be a crucial factor in the superior performance of Uni-CTR.

6.2 Model Inference Acceleration

In industrial recommender systems, the online model serving latency is subject to a strict constraint, typically set at around 1 to 2 milliseconds for a single instance. As a result, ensuring high service efficiency holds paramount importance for CTR models. However, for applications using LLMs, latency [39] is an intractable problem because of the complex attention mechanism and the excessive depth of the transformer layers.

To provide a comprehensive analysis, we compared the inference latency of Uni-CTR with several other models using an RTX 4090 (24GB) in the experimental environment. The results are summarized in Table 9. This analysis highlights that while LLM-based models like Uni-CTR are indeed slower than traditional models, they still operate within an acceptable range from 0.1 to 5 milliseconds for industrial applications. However, the increased latency is offset by the significant performance improvements, making them competitive for deployment in real-time recommender systems.

For further inference acceleration, we export the trained model (with backbone DeBERTaV3$_{\text{large}}$) to .onnx[4] format, which enables us to perform model inference utilizing a static graph paradigm. We then quantize it using FP16 precision with the assistance of the TensorRT tool[5]. With a batch size of 32 and a sequence length of 256, the Uni-CTR inference latency on a single Tesla V100 GPU is 80ms. The average per-sample latency is around 2ms, and the loss in AUC is below 0.01, which is still significantly better than existing traditional multi-domain recommendation models. This latency is perfectly acceptable for Uni-CTR to be used in the rank stage for industrial recommender systems.

7 CONCLUSION

In this paper, we propose Uni-CTR, a unified framework for multi-domain CTR prediction. It comprises an LLM backbone plugged with multiple domain-specific networks and a general network. The introduced LLM

[4]https://onnx.ai/
[5]https://github.com/NVIDIA/TensorRT

Just Accepted

backbone learns common features across various domains from the designed prompts with its powerful semantic understanding. After, the domain-specific networks receive layer-wise representations from the transformer layers of the LLM to capture characteristics inherent to each specific domain. Simultaneously, the general network learns common patterns across all the domains to enable zero-shot prediction for unseen domains. In the extensive experiments conducted on both public and industrial datasets, the Uni-CTR model outperforms existing single-domain and multi-domain models, effectively mitigating the "seesaw phenomenon" and improving generalization to new domains. Future research will continue to investigate the enhancement in input modalities for multi-domain CTR prediction.

ACKNOWLEDGMENTS

This research was partially supported by Huawei (Huawei Innovation Research Program), Research Impact Fund (No.R1015-23), APRC - CityU New Research Initiatives (No.9610565, Start-up Grant for New Faculty of City University of Hong Kong), CityU - HKIDS Early Career Research Grant (No.9360163), Hong Kong ITC Innovation and Technology Fund Midstream Research Programme for Universities Project (No.ITS/034/22MS), Hong Kong Environmental and Conservation Fund (No. 88/2022), and SIRG - CityU Strategic Interdisciplinary Research Grant (No.7020046, No.7020074).

REFERENCES

[1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 1007–1014. https://doi.org/10.1145/3604915.3608857

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., virtual, 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[3] Rich Caruana. 1997. Multitask Learning. *Mach. Learn.* 28, 1 (1997), 41–75. https://doi.org/10.1023/A:1007379606734

[4] Rich Caruana. 1997. Multitask Learning. *Mach. Learn.* 28, 1 (1997), 41–75. https://doi.org/10.1023/A:1007379606734

[5] Jianxin Chang, Chenbin Zhang, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. 2023. PEPNet: Parameter and Embedding Personalized Network for Infusing with Personalized Prior Information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, Long Beach, CA, USA, 3795–3804. https://doi.org/10.1145/3580305.3599884

[6] Olivier Chapelle and S. Sathiya Keerthi. 2010. Efficient algorithms for ranking with SVMs. *Inf. Retr.* 13, 3 (2010), 201–215. https://doi.org/10.1007/S10791-009-9109-9

[7] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep CTR Prediction in Display Advertising. In *Proceedings of the 24th ACM International Conference on Multimedia* (Amsterdam, The Netherlands) *(MM '16)*. Association for Computing Machinery, New York, NY, USA, 811–820. https://doi.org/10.1145/2964284.2964325

[8] Xu Chen, Zida Cheng, Shuai Xiao, Xiaoyi Zeng, and Weilin Huang. 2023. Cross-domain Augmentation Networks for Click-Through Rate Prediction. arXiv:2305.03953 [cs.IR]

[9] David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 20, 2 (1958), 215–232.

[10] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. arXiv:2205.08084 [cs.IR]

[11] Weiwei Deng, Xiaoliang Ling, Yang Qi, Tunzi Tan, Eren Manavoglu, and Qi Zhang. 2018. Ad Click Prediction in Sequence with Long Short-Term Memory Networks: an Externality-aware Model. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, Ann Arbor, MI, USA, 1065–1068. https://doi.org/10.1145/3209978.3210071

Just Accepted

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, MN, USA, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[13] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 320–335. https://doi.org/10.18653/V1/2022.ACL-LONG.26

[14] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. 2020. An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers Artif. Intell.* 3 (2020), 4. https://doi.org/10.3389/FRAI.2020.00004

[15] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction. arXiv:1704.05194 [stat.ML]

[16] Heiko Gebauer, Marco Paiola, Nicola Saccani, and Mario Rapaccini. 2021. Digital servitization: Crossing the perspectives of digitization and servitization. *Industrial Marketing Management* 93 (2021), 382–388. https://doi.org/10.1016/j.indmarman.2020.05.011

[17] Rohit Girdhar and Deva Ramanan. 2017. Attentional Pooling for Action Recognition. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., Long Beach, CA, USA. https://proceedings.neurips.cc/paper_files/paper/2017/file/67c6a1e7ce56d3d6fa748ab6d9af3fd7-Paper.pdf

[18] Rafael Glauber and Angelo C. Loula. 2019. Collaborative Filtering vs. Content-Based Filtering: differences and similarities. *CoRR* abs/1912.08932 (2019). arXiv:1912.08932 http://arxiv.org/abs/1912.08932

[19] Yuqi Gong, Xichen Ding, Yehui Su, Kaiming Shen, Zhongyi Liu, and Guannan Zhang. 2023. An Unified Search and Recommendation Foundation Model for Cold-Start Scenario. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, Birmingham, United Kingdom, 4595–4601. https://doi.org/10.1145/3583780.3614657

[20] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, Melbourne, Australia, 1725–1731. https://doi.org/10.24963/IJCAI.2017/239

[21] Wei Guo, Chenxu Zhu, Fan Yan, Bo Chen, Weiwen Liu, Huifeng Guo, Hongkun Zheng, Yong Liu, and Ruiming Tang. 2023. DFFM: Domain Facilitated Feature Modeling for CTR Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, Birmingham, United Kingdom, 4602–4608. https://doi.org/10.1145/3583780.3615469

[22] 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:2011.11938 [cs.AI]

[23] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, Kigali, Rwanda. https://openreview.net/pdf?id=sE7-XhLxHA

[24] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-Enhanced Bert with Disentangled Attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, Virtual Event, Austria. https://openreview.net/forum?id=XPZIaotutsD

[25] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, Virtual Event. https://openreview.net/forum?id=nZeVKeeFYf9

[26] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, Torino, Italy, 667–676. https://doi.org/10.1145/3269206.3271684

[27] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, Copenhagen, Denmark, 169–177. https://doi.org/10.1145/3298689.3347043

[28] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language?. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*,

Just Accepted

Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 3651–3657. https://doi.org/10.18653/V1/P19-1356

[29] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 3651–3657. https://doi.org/10.18653/v1/P19-1356

[30] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

[31] Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells (Eds.). ACM, Boston, MA, USA, 43–50. https://doi.org/10.1145/2959100.2959134

[32] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361 [cs.LG]

[33] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). OpenReview.net, San Diego, CA, USA. http://arxiv.org/abs/1412.6980

[34] Anders Krogh and John Hertz. 1991. A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*, J. Moody, S. Hanson, and R.P. Lippmann (Eds.), Vol. 4. Morgan-Kaufmann, Denver, Colorado, USA. https://proceedings.neurips.cc/paper_files/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf

[35] Rohit Kumar, Sneha Manjunath Naik, Vani D Naik, Smita Shiralli, Sunil V.G, and Moula Husain. 2015. Predicting clicks: CTR estimation of advertisements using Logistic Regression classifier. In *2015 IEEE International Advance Computing Conference (IACC)*. IEEE, Bangalore, India, 1134–1138. https://doi.org/10.1109/IADCC.2015.7154880

[36] Hao Li, Huichuan Duan, Yuanjie Zheng, Qianqian Wang, and Yu Wang. 2020. A CTR prediction model based on user interest via attention mechanism. *Appl. Intell.* 50, 4 (2020), 1192–1203. https://doi.org/10.1007/S10489-019-01571-9

[37] Pengcheng Li, Runze Li, Qing Da, Anxiang Zeng, and Lijun Zhang. 2020. Improving Multi-Scenario Learning to Rank in E-commerce by Exploiting Task Relationships in the Label Space. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, Virtual Event, Ireland, 2605–2612. https://doi.org/10.1145/3340531.3412713

[38] Xiangyang Li, Bo Chen, Huifeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, Jinxing Liu, Zhenhua Dong, and Ruiming Tang. 2022. IntTower: The Next Generation of Two-Tower Model for Pre-Ranking System. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, Atlanta, GA, USA, 3292–3301. https://doi.org/10.1145/3511808.3557072

[39] Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. CTRL: Connect Collaborative and Language Model for CTR Prediction. arXiv:2306.02841 [cs.IR]

[40] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. HAMUR: Hyper Adapter for Multi-Domain Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, Birmingham, United Kingdom, 1268–1277. https://doi.org/10.1145/3583780.3615137

[41] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, London, UK, 1754–1763. https://doi.org/10.1145/3219819.3220023

[42] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. 2015. A Convolutional Click Prediction Model. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (Melbourne, Australia) *(CIKM '15)*. Association for Computing Machinery, New York, NY, USA, 1743–1746. https://doi.org/10.1145/2806416.2806603

[43] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

[44] Yuhao Luo, Shiwei Ma, Mingjun Nie, Changping Peng, Zhangang Lin, Jingping Shao, and Qianfang Xu. 2024. Domain-Aware Cross-Attention for Cross-domain Recommendation. arXiv:2401.11705 [cs.IR]

[45] 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, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, London, UK, 1930–1939. https://doi.org/10.1145/3219819.3220007

[46] 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*

Under Accepted

Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, Ann Arbor, MI, USA, 1137–1140. https://doi.org/10.1145/3209978.3210104

[47] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, Melbourne, Australia, 2464–2470. https://doi.org/10.24963/IJCAI.2017/343

[48] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent Models of Visual Attention. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). Montreal, Quebec, Canada, 2204–2212. https://proceedings.neurips.cc/paper/2014/hash/09c6c3783b4a70054da74f2538ed47c6-Abstract.html

[49] Aashiq Muhamed, Iman Keivanloo, Sujan Perera, James Mracek, Yi Xu, Qingjun Cui, Santosh Rajagopalan, Belinda Zeng, and Trishul Chilimbi. 2021. CTR-BERT: Cost-effective knowledge distillation for billion-parameter teacher models. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*.

[50] Aashiq Muhamed, Jaspreet Singh, Shuai Zheng, Iman Keivanloo, Sujan Perera, James Mracek, Yi Xu, Qingjun Cui, Santosh Rajagopalan, Belinda Zeng, and Trishul Chilimbi. 2022. DCAF-BERT: A Distilled Cachable Adaptable Factorized Model For Improved Ads CTR Prediction. In *Companion Proceedings of the Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 110–115. https://doi.org/10.1145/3487553.3524206

[51] Fionn Murtagh. 1990. Multilayer perceptrons for classification and regression. *Neurocomputing* 2, 5 (1990), 183–197. https://doi.org/10.1016/0925-2312(91)90023-5

[52] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 188–197. https://doi.org/10.18653/V1/D19-1018

[53] XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* 63, 10 (2020), 1872–1897. https://doi.org/10.1007/s11431-020-1647-3

[54] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-Based Neural Networks for User Response Prediction. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu (Eds.). IEEE Computer Society, Barcelona, Spain, 1149–1154. https://doi.org/10.1109/ICDM.2016.0151

[55] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu (Eds.). IEEE Computer Society, Sydney, Australia, 995–1000. https://doi.org/10.1109/ICDM.2010.127

[56] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461. https://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf

[57] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, Banff, Alberta, Canada, 521–530. https://doi.org/10.1145/1242572.1242643

[58] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics* 8 (2021), 842–866.

[59] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko (Eds.). ACM, 285–295. https://doi.org/10.1145/371920.372071

[60] 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 *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, Virtual Event, Queensland, Australia, 4094–4103. https://doi.org/10.1145/3459637.3481948

[61] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, and Xiaoqiang Zhu. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, Queensland, Australia, 4104–4113. https://doi.org/10.1145/3459637.3481941

Just Accepted

[62] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, Beijing, China, 1161–1170. https://doi.org/10.1145/3357384.3357925

[63] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, Beijing, China, 1161–1170. https://doi.org/10.1145/3357384.3357925

[64] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958. https://doi.org/10.5555/2627435.2670313

[65] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. LST: Ladder Side-Tuning for Parameter and Memory Efficient Transfer Learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). New Orleans, LA, USA. http://papers.nips.cc/paper_files/paper/2022/hash/54801e196796134a2b0ae5e8adef502f-Abstract-Conference.html

[66] Shulong Tan, Meifang Li, Weijie Zhao, Yandan Zheng, Xin Pei, and Ping Li. 2021. Multi-Task and Multi-Scene Unified Ranking Model for Online Advertising. In *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021*, Yixin Chen, Heiko Ludwig, Yicheng Tu, Usama M. Fayyad, Xingquan Zhu, Xiaohua Hu, Suren Byna, Xiong Liu, Jianping Zhang, Shirui Pan, Vagelis Papalexakis, Jianwu Wang, Alfredo Cuzzocrea, and Carlos Ordonez (Eds.). IEEE, Orlando, FL, USA, 2046–2051. https://doi.org/10.1109/BIGDATA52589.2021.9671920

[67] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, London, UK, 269–278. https://doi.org/10.1145/3383313.3412236

[68] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[69] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[70] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971

[71] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). Curran Associates Inc., Long Beach, CA, USA, 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[73] Dong Wang, Kavé Salamatian, Yunqing Xia, Weiwei Deng, and Qi Zhang. 2023. BERT4CTR: An Efficient Framework to Combine Pre-trained Language Model with Non-textual Features for CTR Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, Long Beach, CA, USA, 5039–5050. https://doi.org/10.1145/3580305.3599780

[74] Hangyu Wang, Jianghao Lin, Xiangyang Li, Bo Chen, Chenxu Zhu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2023. FLIP: Towards Fine-grained Alignment between ID-based Models and Pretrained Language Models for CTR Prediction. arXiv:2310.19453 [cs.IR]

[75] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, Halifax, NS, Canada, 12:1–12:7. https://doi.org/10.1145/3124749.3124754

[76] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* 69 (2017), 29–39. https://doi.org/10.1016/J.ESWA.2016.09.040

[77] Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. arXiv:2306.10933 [cs.IR]

[78] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning. arXiv:2310.06694 [cs.CL]

Unpublished

[79] Jian Mou Xusen Cheng and Xiangbin Yan. 2021. Sharing economy enabled digital platforms for development. *Information Technology for Development* 27, 4 (2021), 635–644. https://doi.org/10.1080/02681102.2021.1971831 arXiv:https://doi.org/10.1080/02681102.2021.1971831

[80] Ling Yan, Wu-Jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled Group Lasso for Web-Scale CTR Prediction in Display Advertising. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014 (JMLR Workshop and Conference Proceedings, Vol. 32)*. JMLR.org, Beijing, China, 802–810. http://proceedings.mlr.press/v32/yan14.html

[81] Xuanhua Yang, Xiaoyu Peng, Penghui Wei, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. AdaSparse: Learning Adaptively Sparse Structures for Multi-Domain Click-Through Rate Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, Atlanta, GA, USA, 4635–4639. https://doi.org/10.1145/3511808.3557541

[82] Yanwu Yang and Panyu Zhai. 2022. Click-through rate prediction in online advertising: A literature review. *Inf. Process. Manag.* 59, 2 (2022), 102853. https://doi.org/10.1016/J.IPM.2021.102853

[83] 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* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1368–1376. https://doi.org/10.1145/3488560.3498479

[84] Wei Zhang, Pengye Zhang, Bo Zhang, Xingxing Wang, and Dong Wang. 2023. A Collaborative Transfer Learning Framework for Cross-domain Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, Long Beach, CA, USA, 5576–5585. https://doi.org/10.1145/3580305.3599758

[85] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL]

[86] 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 *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, Honolulu, Hawaii, USA, 5941–5948. https://doi.org/10.1609/AAAI.V33I01.33015941

[87] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, Beijing, China, 1059–1068. https://doi.org/10.1145/3219819.3219823

[88] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, London, UK, 1059–1068. https://doi.org/10.1145/3219819.3219823

[89] 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 *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, Madrid, Spain, 1535–1544. https://doi.org/10.1145/3477495.3531942

Just Accepted