FLIP: Fine-grained Alignment between ID-based Models and Pretrained Language Models for CTR Prediction

Hangyu Wang*
hangyuwang@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Jianghao Lin*
chiangel@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Xiangyang Li
lixiangyang34@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

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

Chenxu Zhu
zhuchenxu1@huawei.com
Huawei Noah's Ark Lab
Shanghai, China

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

Weinan Zhang†
wnzhang@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Yong Yu
yyu@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

Click-through rate (CTR) prediction plays as a core function module in various personalized online services. The traditional ID-based models for CTR prediction take as inputs the one-hot encoded ID features of *tabular modality*, which capture the collaborative signals via feature interaction modeling. But the one-hot encoding discards the semantic information included in the textual features. Recently, the emergence of Pretrained Language Models (PLMs) has given rise to another paradigm, which takes as inputs the sentences of *textual modality* obtained by hard prompt templates and adopts PLMs to extract the semantic knowledge. However, PLMs often face challenges in capturing field-wise collaborative signals and distinguishing features with subtle textual differences. In this paper, to leverage the benefits of both paradigms and meanwhile overcome their limitations, we propose to conduct Fine-grained feature-level ALignment between ID-based Models and Pretrained Language Models (FLIP) for CTR prediction. Unlike most methods that solely rely on global views through instance-level contrastive learning, we design a novel jointly masked tabular/language modeling task to learn fine-grained alignment between tabular IDs and word tokens. Specifically, the masked data of one modality (*i.e.*, IDs and tokens) has to be recovered with the help of the other modality, which establishes the feature-level interaction and alignment via sufficient mutual information extraction between dual modalities. Moreover, we propose to jointly finetune the ID-based model and PLM by adaptively combining the output of both models, thus achieving superior performance in downstream CTR prediction tasks. Extensive experiments on three real-world datasets demonstrate that FLIP outperforms SOTA baselines, and is highly compatible with various ID-based models and PLMs. The code is available[1][2].

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Fine-grained Alignment, Pretrained Language Model, CTR prediction, Recommender Systems

ACM Reference Format:
Hangyu Wang, Jianghao Lin, Xiangyang Li, Bo Chen, Chenxu Zhu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. FLIP: Fine-grained Alignment between ID-based Models and Pretrained Language Models for CTR Prediction. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3640457.3688106

1 INTRODUCTION

Click-through rate (CTR) prediction is the core component of various personalized online services (*e.g.*, web search [12, 16, 39], recommender systems [20, 77]). It aims to estimate a user's click probability towards each target item, given a particular context [40, 78, 84]. Recently, the emergence of Pretrained Language Models (PLMs) [54] has facilitated the acquisition of extensive knowledge and enhanced reasoning abilities, hence introducing a novel paradigm for predicting CTR directly through natural language [38].

On the one hand, the traditional **ID-based models** for CTR prediction adopt the one-hot encoding to convert the input data

*Both authors contributed equally to this research.
†Corresponding author.

RecSys '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688106

[1]PyTorch version: https://github.com/justarter/FLIP.
[2]MindSpore version: https://github.com/mindspore-lab/models/tree/master/research/huawei-noah/FLIP.

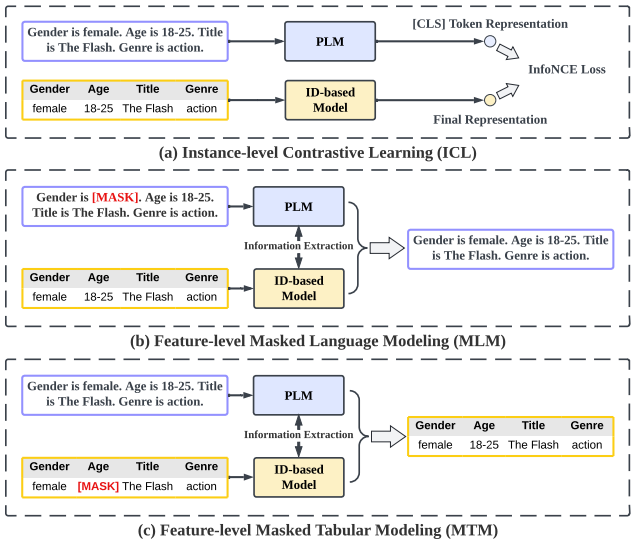


Figure 1: Three cross-modal pretraining tasks. Task (a) provides coarse-grained instance-level alignment via contrastive learning, while tasks (b) and (c) achieve fine-grained feature-level alignment through jointly masked modality modeling.

into ID features, which we refer to as **tabular data modality**. An example is shown as follows:

$$\underbrace{[0,0,1,\cdots,0]}_{\texttt{UserID=02}}\ \underbrace{[0,1]}_{\texttt{Gender=Male}}\ \underbrace{[0,1,\cdots,0]}_{\texttt{ItemID=01}}\ \underbrace{[0,1,\cdots,0]}_{\texttt{Genre=Action}} \quad (1)$$

Deriving from POLY2 [5] and FM [61], the key idea of these models is to capture the complex high-order feature interaction patterns across multiple fields by different operators (*e.g.*, product [55, 74], convolution [42, 80], and attention [65, 79]). On the other hand, the development of PLMs has ushered in another modeling paradigm that utilizes the **PLM** as a text encoder or recommender directly. The input data is first transformed into textual sentences via hard prompt templates, which could be referred to as **textual data modality**. In this way, the semantic information among original data is well preserved as natural languages (*e.g.*, gender feature as text "male" instead of ID code "[0,1]"). Then, PLMs [13, 46] encode and understand the textual sentences, and thus turn CTR prediction into either a binary text classification problem [43, 50] or a sequence-to-sequence task [18, 19]. However, both of the above models (*i.e.*, ID-based and PLMs) possess inherent limitations.

ID-based models utilize the one-hot encoding that discards the semantic information included in the textual features and fails to capture the semantic correlation between feature descriptions [35, 37]. In addition, ID-based models rely heavily on the user interactions and may struggle in scenarios with sparse interactions [63]. The aforementioned issues can be effectively alleviated by PLMs. PLMs excel in understanding the context and meaning behind textual features, and use their knowledge and reasoning capabilities to achieve robust performance in sparse-interaction situations [75, 86].

Furthermore, PLMs also have certain limitations. PLMs struggle to understand field-wise collaborative signals because their input data is formulated as textual sentences, which are broken down into subword tokens, thus splitting the field-wise features [2, 60]. Additionally, PLMs may fail to recognize subtle differences between different feature descriptions [56, 64] because it is hard to distinguish features that exhibit little textual variations in natural language [23, 83] (*e.g.*, in terms of the movie name, "The Room" and "Room" are two very similar movies literally). Fortunately, the ID-based model can perceive field-wise collaborative signals with various model structures and distinguish each different feature with the distinctive ID encodings.

To this end, it is natural to bridge two paradigms to leverage the benefits of both modalities while overcoming their limitations. Moreover, the fine-grained feature-level alignment between tabular IDs and word tokens is critical, enabling ID-based models to perceive semantic information corresponding to each feature ID and allowing PLMs to clearly distinguish features with similar text but different IDs. However, most existing methods [35, 60] align both modalities by instance-level contrastive learning, as shown in Figure 1(a). They only rely on the global view and lack supervision to encourage fine-grained alignment between IDs and tokens, potentially causing representation degeneration problems [17, 29, 53].

In this paper, we propose to conduct Fine-grained Feature-level ALignment between ID-based Models and Pretrained Language Models (**FLIP**) for CTR prediction. FLIP is a model-agnostic framework that adopts the common pretrain-finetune scheme [13, 40]. For *pretraining* objectives, as illustrated in Figure 1(b) and 1(c), we propose to build jointly masked tabular and language modeling, where the masked features (*i.e.*, IDs or tokens of specific features) of one modality are recovered with the help of another modality. This is motivated by the fact that both tabular and textual data convey almost the same information of the original raw data, but only in different formats. In order to accomplish the masked feature reconstruction, each single model (ID-based model or PLM) is required to seek and exploit the valuable knowledge embedded in the other model that corresponds to the masked features, thereby achieving fine-grained feature-level cross-modal interactions. Then, as for *finetuning*, we propose a simple yet effective adaptive finetuning approach to combine the predictions from both models for downstream CTR estimation.

The main contributions of this paper are as follows:

- We highlight the importance of utilizing fine-grained feature-level alignment between tabular IDs and word tokens, in order to explore the potential of enhancing the performance of existing recommender systems.
- We propose a model-agnostic framework FLIP, where the jointly masked tabular and language modeling tasks are involved. In these tasks, the masked features of one modality have to be recovered with the help of another modality, thus learning fine-grained feature-level cross-modal interactions. FLIP also employs an adaptive finetuning approach to combine the predictions from the ID-based model and PLM for improving performance.
- Extensive experiments on three real-world public datasets demonstrate the superiority of FLIP, compared with existing baseline models. Moreover, we validate the model compatibility of FLIP in terms of both ID-based models and PLMs.

2 PRELIMINARIES

2.1 ID-based Models for CTR Prediction

The traditional CTR prediction is modeled as a binary classification task [20, 55], whose dataset is presented as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $y_i \in \{1, 0\}$ is the label indicating user's actual click behavior, and $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,F}]$ is categorical tabular data with F different fields. The goal of CTR prediction is to estimate the click probability $P(y_i = 1|\mathbf{x}_i)$ based on input $\mathbf{x}_i$.

Generally, ID-based models adopt the "Embedding & Feature Interaction" paradigm [36, 65]: (1) the input $\mathbf{x}_i$ is first transformed into one-hot vectors, which are then mapped to low-dimensional embeddings via an embedding layer. (2) Next, Feature Interaction (FI) Layer is used to process the embeddings, compute complex feature interactions and generate a dense representation $\mathbf{v}_i$. (3) Finally, the prediction layer (usually a MLP module) estimates the click probability $\hat{y}_i \in [0, 1]$ based on the dense representation $\mathbf{v}_i$. The ID-based model is trained with the binary cross-entropy (BCE) loss in an end-to-end manner:

$$\mathcal{L}_{BCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \left(\hat{y}_i \right) + (1 - y_i) \log \left(1 - \hat{y}_i \right) \right] \quad (2)$$

2.2 PLMs for CTR Prediction

As PLMs have shown remarkable success in a wide range of tasks due to their extensive knowledge and reasoning capabilities [3, 46], researchers now tend to leverage their proficiency in natural language understanding and world knowledge modeling to solve the CTR prediction task [38, 44].

Different from ID-based models, (1) PLMs first transform the input data $\mathbf{x}_i$ into the textual sentence $\mathbf{x}_i^{text}$ via hard prompt templates. (2) Then the textual sentence $\mathbf{x}_i^{text}$ is converted into meaningful lexical tokens through tokenization, which are embedded into a vector space. (3) Next, PLMs utilize their transformer-based neural networks to process the token embeddings obtained from the previous step, generating the contextually coherent representation $\mathbf{w}_i$. (4) Finally, we can either add a randomly initialized classification head (usually MLP) on the representation $\mathbf{w}_i$ to perform binary classification and predict the click label $y_i \in \{0, 1\}$ [35, 43], or add a language modeling head to do causal language modeling tasks and predict the likelihood of generating the next keyword (*e.g.*, "yes" or "no") through a verbalizer [2, 18].

3 METHODOLOGY

3.1 Overview of FLIP

Figure 2 depicts the architecture of FLIP, which consists of three stages: **modality transformation**, **modality alignment pretraining** and **adaptive finetuning**. Firstly, FLIP transforms the raw data from tabular modality into textual modality. Then, in the modality alignment pretraining, we employ the jointly masked language/tabular modeling task to learn fine-grained modality alignments. Lastly, we propose a simple yet effective adaptive finetuning strategy to further enhance the performance on CTR prediction.

Hereinafter, we omit the detailed structure of PLMs and ID-based models, since FLIP serves as a model-agnostic framework and is compatible with various backbone models.

3.2 Modality Transformation

Standard PLMs take sequences of words as inputs [13, 57]. Consequently, the modality transformation aims to convert the tabular data $\mathbf{x}_i^{tab}$ into textual data $\mathbf{x}_i^{text}$ via hard prompt templates. Previous works [24, 35, 43] have suggested that sophisticated templates (*e.g.*, with more vivid descriptions) for textual data construction might mislead the PLM and make it fail to grasp the key information in the texts. Hence, we adopt the following simple yet effective transformation template:

$$\begin{aligned} t_{i,f} &= [m_f \oplus \text{“}is\text{”} \oplus v_{i,f} \oplus \text{“.”}], \ f \in \{1, \cdots, F\} \\ \mathbf{x}_i^{text} &= [t_{i,1} \oplus t_{i,2} \oplus \cdots \oplus t_{i,F}], \end{aligned} \quad (3)$$

where m_f is the name of f-th field (*e.g.*, `gender`), $v_{i,f}$ denotes the feature value of f-th field for input x_i^{tab} (*e.g.*, `female`), and $\oplus$ indicates the concatenation operator.

An illustrative example is given in Figure 2 (Stage 1), where we first construct a descriptive sentence $t_{i,f}$ for each feature field, and then concatenate them to obtain the final textual data. In this way, we preserve the semantic knowledge of both field names and feature values with minimal preprocessing and no information loss. Both tabular data and textual data can be considered to contain almost the same information of raw data, albeit in different modalities.

3.3 Modality Alignment Pretraining

As shown in Figure 2 (Stage 2), after obtaining the paired text-tabular data $(\mathbf{x}_i^{text}, \mathbf{x}_i^{tab})$ from the same raw input, we first perform field-level data masking to obtain the corrupted version of input pair, $(\hat{\mathbf{x}}_i^{text}, \hat{\mathbf{x}}_i^{tab})$. Then, the PLM h_{PLM} and ID-based model h_{ID} encode the input pair to obtain the dense representations $(\mathbf{w}_i, \hat{\mathbf{w}}_i)$ and $(\mathbf{v}_i, \hat{\mathbf{v}}_i)$ for textual and tabular modalities, respectively. Next, we apply three different pretraining objectives to achieve both feature-level and instance-level alignments between PLMs and ID-based models:

- *Feature-level Masked Language Modeling* (MLM) requires the model to recover the original tokens from the corrupted textual context with the help of complete tabular data.
- *Feature-level Masked Tabular Modeling* (MTM) requires the model to recover the original feature IDs from the corrupted tabular context with the help of complete textual data.
- *Instance-level Contrastive Learning* (ICL) draws positive samples together (*i.e.*, different modalities of the same input) and pushes apart negative sample pairs.

The jointly masked modality modeling task learns fine-grained modality interactions by the way of mask-and-predict, while ICL aligns the modalities from the perspective of the global consistency.

3.3.1 ***Field-level Data Masking.*** We first perform the field-level data masking strategy for textual and tabular data, respectively.

For *textual data*, we propose to mask tokens at the field level. We first randomly select a certain ratio r_{text} of the feature fields to be corrupted, and then mask all the consecutive tokens that constitute the entire text of the corresponding feature value. We denote the index set of masked tokens as $\mathcal{I}^{text}$. Note that this is quite different from the random token-level masking strategy for common language models (*e.g.*, BERT [13]).

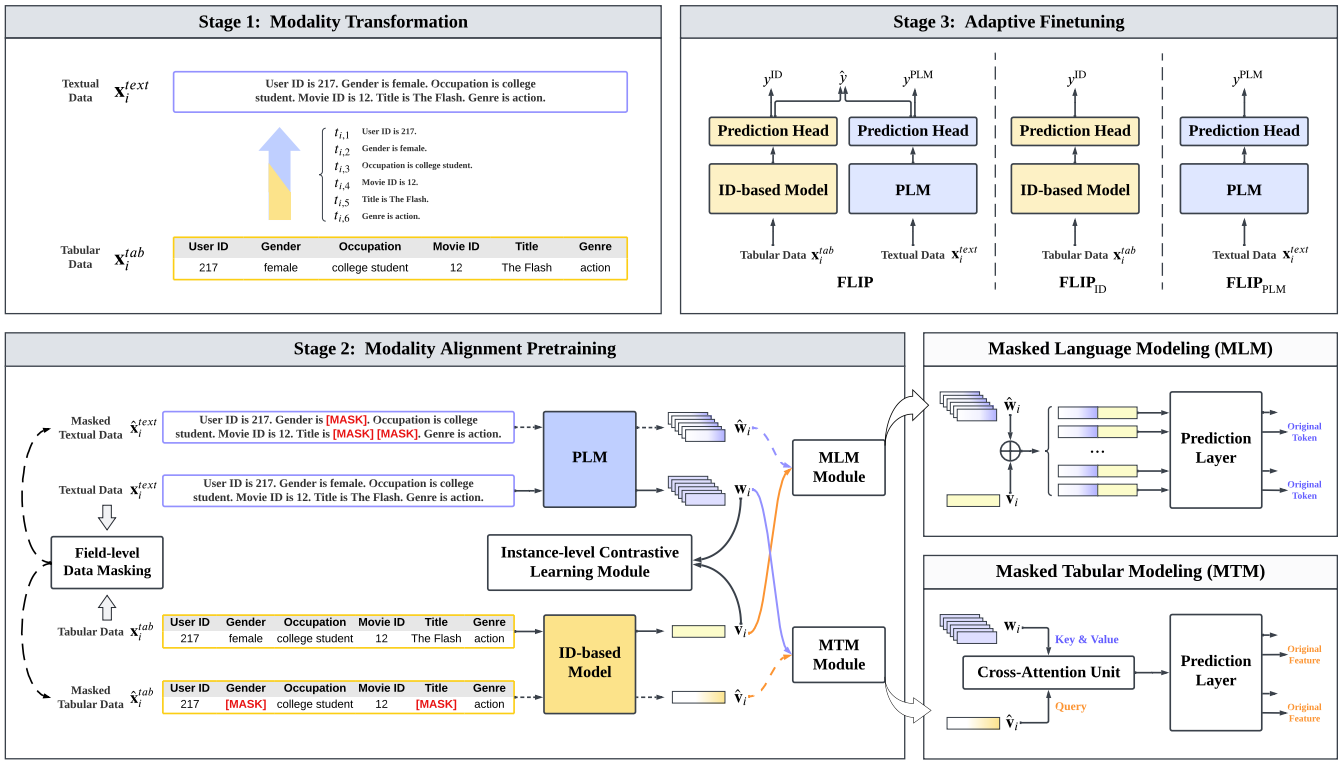


Figure 2: The overall framework of our proposed FLIP.

For example, suppose the sentence tokenization from `occupation` field is ["occupation", "is", "college", "student"]. The outcome of field-level masking should be ["occupation", "is", `[MASK]`, `[MASK]`]. But the outcome of token-level masking might be [`[MASK]`, "is", "college", "student"] or ["occupation", "is", "college", `[MASK]`]. Obviously, token-level masked tokens can be easily inferred solely based on the textual context, thus limiting the cross-modal interactions and losing the generalization ability [45].

For *tabular data*, following previous works [40, 72], we uniformly sample a certain ratio r_{tab} of the fields, and then replace the corresponding feature with an additional `<MASK>` feature, which is set as a special feature in the embedding table of ID-based model. The `<MASK>` feature is not field-specific, but shared by all feature fields to prevent introducing prior knowledge about the masked field [40]. Similarly, we denote the index set of masked fields as $\mathcal{I}^{tab}$.

After the field-level data masking, we obtain the corrupted version for both modalities, *i.e.*, $(\hat{\mathbf{x}}_i^{text}, \hat{\mathbf{x}}_i^{tab})$. It is worth noting that the selected fields to be masked do not necessarily have to be the same for textual and tabular modalities, since they serve as two independent parallel workflows.

3.3.2 ***Data Encoding.*** We employ the PLM h_{PLM} and ID-based model h_{ID} to encode the input pairs from textual and tabular modalities, respectively:

$$\begin{aligned} \mathbf{w}_i &= h_{\text{PLM}}(\mathbf{x}_i^{text}), \quad & \hat{\mathbf{w}}_i &= h_{\text{PLM}}(\hat{\mathbf{x}}_i^{text}), \\ \mathbf{v}_i &= h_{\text{ID}}(\mathbf{x}_i^{tab}), & \hat{\mathbf{v}}_i &= h_{\text{ID}}(\hat{\mathbf{x}}_i^{tab}). \end{aligned} \tag{4}$$

Here, $\mathbf{w}_i = [w_{i,l}]_{l=1}^{L} \in \mathbb{R}^{L \times D_{text}}$ is the set of hidden states from the last layer of the PLM, where L is the number of tokens for $\mathbf{x}_i^{text}$ and D_{text} is the hidden size of the PLM. Note that $w_{i,1}$ is the `[CLS]` token vector that represents the overall textual input. $\hat{\mathbf{w}}_i = [\hat{w}_{i,l}]_{l=1}^{L} \in \mathbb{R}^{L \times D_{text}}$ also satisfies the notations above. $\mathbf{v}_i, \hat{\mathbf{v}}_i \in \mathbb{R}^{D_{tab}}$ are the representations produced by ID-based model.

3.3.3 ***Masked Language Modeling.*** As shown in Figure 2 (Stage 2), the MLM module takes the text-tabular pair $(\hat{\mathbf{w}}_i, \mathbf{v}_i)$ as input, and attempts to recover the masked tokens. We denote the index set of masked tokens as $\mathcal{I}^{text}$. For each masked token with index $l \in \mathcal{I}^{text}$, we concatenate the corresponding token vector $\hat{w}_{i,l}$ with the "reference answer" $\mathbf{v}_i$, and then feed them through the prediction layer to obtain the estimated distribution:

$$q_{i,l} = g_{\text{PLM}}([\hat{w}_{i,l} \oplus \mathbf{v}_i]) \in \mathbb{R}^V, \tag{5}$$

where $\oplus$ is the concatenation operation, V is the vocabulary size, and g_{PLM} is a two-layer MLP. Finally, similar to common masked language modeling [13], we leverage the cross-entropy loss for pretraining optimization:

$$\mathcal{L}_i^{MLM} = \frac{1}{|\mathcal{I}^{text}|} \sum\nolimits_{l \in \mathcal{I}^{text}} \text{CrossEntropy}\left(q_{i,l},\ x_{i,l}^{text}\right), \tag{6}$$

where $x_{i,l}^{text}$ is the original l-th token.

3.3.4 ***Masked Tabular Modeling.*** Likewise, the MTM module takes the text-tabular pair $(\mathbf{w}_i, \hat{\mathbf{v}}_i)$ as input, and aims to recover the masked features. To dynamically capture the essential knowledge from the corresponding tokens of the masked features, we design the cross-attention unit to aggregate the tabular representation $\hat{\mathbf{v}}_i$

with the "reference answer" $\mathbf{w}_i$:

$$\mathbf{u}_i = \text{Softmax}\left(\frac{\widehat{\mathbf{v}}_i \mathbf{Q} \mathbf{w}_i^T}{\sqrt{D_{text}}}\right)\mathbf{w}_i, \quad \mathbf{u}_i \in \mathbb{R}^L \tag{7}$$

where $\mathbf{Q} \in \mathbb{R}^{D_{tab} \times D_{text}}$ is the trainable cross-modal attention matrix, and $\sqrt{D_{text}}$ is the scaling factor [66].

For each masked feature with index $f \in \mathcal{I}^{tab}$, we maintain an independent MLP network $g_{\text{ID}}^{(f)}$ followed by a softmax function to compute the distribution $p_{i,f} \in \mathbb{R}^M$ over the candidate features:

$$\begin{aligned} \mathbf{c}_{i,f} &= g_{\text{ID}}^{(f)}(\mathbf{u}_i), \quad \mathbf{c}_{i,f} \in \mathbb{R}^M, \\ p_{i,f,j} &= \frac{\exp(c_{i,f,j})}{\sum_{k=1}^{M} \exp(c_{i,f,k})}, \quad j = 1, \cdots, M, \end{aligned} \tag{8}$$

where M is the size of the entire feature space. Finally, we adopt the cross-entropy loss on all the masked features:

$$\mathcal{L}_i^{MTM} = \frac{1}{|\mathcal{I}^{tab}|} \sum\nolimits_{f \in \mathcal{I}^{tab}} \text{CrossEntropy}(p_{i,f},\ x_{i,f}^{tab}), \tag{9}$$

where $x_{i,f}^{tab}$ is the original feature of f-th field.

However, the loss above is actually impractical and inefficient since it has to calculate the softmax function over the entire feature space in Eq. 8, where M is usually at million level for real-world applications. To this end, we adopt noise contrastive estimation (NCE) [21, 40, 49]. NCE transforms a multi-class classification task into a binary classification task, where the model is required to distinguish positive features (*i.e.*, masked features) from noise features. Specifically, for each masked feature $x_{i,f}^{tab}$ of f-th field, we sample K noise features from the entire feature space according to their frequency distribution in the training set. Then, we utilize the binary cross-entropy (BCE) loss for MTM optimization:

$$\mathcal{L}_i^{MTM} = -\frac{1}{|\mathcal{I}^{tab}|} \sum_{f \in \mathcal{I}^{tab}} \left(\log \sigma(c_{i,f,t}) + \sum_{k=1}^{K} \log(1 - \sigma(c_{i,f,k}))\right) \tag{10}$$

where σ is the sigmoid function, and t, k are the indices of the positive and noise features respectively.

3.3.5 ***Instance-level Contrastive Learning (ICL).*** In addition to the feature-level alignment through the masked modality modeling, we also introduce contrastive learning to explicitly learn instance-level consistency between two modalities. The contrastive objective draws the representations of matched text-tabular pairs together and pushes apart those non-matched pairs [30, 56].

Here, we utilize the `[CLS]` token vector $w_{i,1}$ to represent the textual input $\mathbf{x}_i^{text}$. For dimensional consistency, we employ two separate linear layers to project the `[CLS]` token vector $w_{i,1}$ and tabular representation $\mathbf{v}_i$ into d-dimensional vectors, z_i^{text} and z_i^{tab}, respectively. Next, we adopt InfoNCE [51] to compute the ICL loss:

$$\begin{aligned} \mathcal{L}^{ICL} = -\frac{1}{2B} \sum_{i=1}^{B} \Bigg[&\log\left(\frac{\exp(\text{sim}(z_i^{text}, z_i^{tab})/\tau)}{\sum_j \exp(\text{sim}(z_i^{text}, z_j^{tab})/\tau)}\right) \\ &+ \log\left(\frac{\exp(\text{sim}(z_i^{tab}, z_i^{text})/\tau)}{\sum_j \exp(\text{sim}(z_i^{tab}, z_j^{text})/\tau)}\right)\Bigg] \end{aligned} \tag{11}$$

where B is the batch size, τ is the temperature hyperparameter, and the similarity function $\text{sim}(\cdot)$ is measured by dot product.

Finally, by putting the three objectives together, the overall loss for the modality alignment pretraining stage is:

$$\mathcal{L}^{pretrain} = \frac{1}{B} \sum\nolimits_{i=1}^{B} \left(\mathcal{L}_i^{MLM} + \mathcal{L}_i^{MTM}\right) + \mathcal{L}^{ICL}. \tag{12}$$

3.4 Adaptive Finetuning

After the pretraining stage, the PLM and ID-based model have learned fine-grained multimodal representations. As depicted in Figure 2 (Stage 3), in this stage, we adaptively finetune the two models jointly on the downstream CTR prediction task with supervised click signal to achieve superior performance.

FLIP places a randomly initialized linear layer on the ID-based model, and another layer upon the PLM, so that these two models can output the estimated probability $\hat{y}_i^{\text{ID}}$ and $\hat{y}_i^{\text{PLM}}$ respectively.

$$\begin{aligned} \hat{y}_i^{\text{ID}} &= \sigma(\text{Linear}_{\text{ID}}(\mathbf{v}_i)), \\ \hat{y}_i^{\text{PLM}} &= \sigma(\text{Linear}_{\text{PLM}}(\mathbf{w}_i)) \end{aligned} \tag{13}$$

And the final click probability is estimated by a weighted sum of outputs from both models:

$$\hat{y}_i = \sigma\left(\alpha \times \text{Linear}_{\text{ID}}(\mathbf{v}_i) + (1 - \alpha) \times \text{Linear}_{\text{PLM}}(\mathbf{w}_i)\right), \tag{14}$$

where $\alpha \in$ is a learnable parameter to adaptively balance the outcomes from two models. To avoid performance collapse mentioned in previous works [4, 62], we apply BCE objectives over the jointly estimated click probability $\hat{y}$, as well as the solely estimated click probability $\hat{y}_i^{\text{ID}}$ and $\hat{y}_i^{\text{PLM}}$ for model optimization:

$$\mathcal{L}^{finetune} = \mathcal{L}_{BCE}(y, \hat{y}) + \mathcal{L}_{BCE}(y, \hat{y}^{\text{ID}}) + \mathcal{L}_{BCE}(y, \hat{y}^{\text{PLM}}) \tag{15}$$

To delve deeper into the influence of fine-grained alignment on the single model, we define two variants: $\textbf{FLIP}_{\text{ID}}$ and $\textbf{FLIP}_{\text{PLM}}$. The former solely finetunes the ID-based model with loss $\mathcal{L}_{BCE}(y, \hat{y}^{\text{ID}})$, while the latter solely finetunes the PLM with loss $\mathcal{L}_{BCE}(y, \hat{y}^{\text{PLM}})$.

It is worth noting that FLIP is expected to achieve the superior performance since it explicitly combines the predictions from two models, while FLIP_{ID} and FLIP_{PLM} could more clearly reveal the effect of fine-grained alignment on a single model.

4 EXPERIMENT

4.1 Experiment Setup

4.1.1 Datasets. We conduct experiments on three real-world public datasets: MovieLens-1M, BookCrossing, GoodReads. All of the selected datasets contain user and item information. The information is unencrypted original text, thus preserving the real semantic information.

- **MovieLens-1M** [22] is a movie recommendation dataset with user-movie ratings ranging from 1 to 5. Following the previous work [35, 65], We consider samples with ratings greater than 3 as positive, samples with ratings less than 3 as negative, and remove samples with ratings equal to 3 (*i.e.*, neutral).
- **BookCrossing** [88] is a book recommendation dataset and possesses user-book ratings ranging from 0 to 10. We consider samples with scores greater than 5 as positive, and the rest as negative.

Table 1: Statistics of processed datasets.

Dataset	#Samples	#Fields	#Features
MovieLens-1M	739,012	8	16,849
BookCrossing	1,031,171	8	722,235
GoodReads	20,122,040	15	4,565,429

- **GoodReads** [67, 68] is a book recommendation dataset which contains user-book ratings ranging from 1 to 5. We take samples with ratings greater than 3 as positive, and the rest as negative.

Following previous works [33, 35], we sort all samples in chronological order and take the first 90% samples as the training set and the remaining as the testing set. The training set for the pretraining and finetuning stage are the same [40]. All our experimental results are obtained on the testing set. The statistics of the processed datasets are shown in Table 1.

4.1.2 Evaluation Metrics. We use commonly adopted metrics, AUC (Area Under the ROC Curve) and Logloss (binary cross-entropy loss) as the evaluation metrics. Notably, a slightly higher AUC or a lower Logloss (*e.g.*, **0.001**) can be considered as a significant improvement in CTR prediction [20, 34, 36].

4.1.3 Baselines. The baseline methods can be mainly classified into three categories: (1) **ID-based** models: AFM [79], PNN [55], Wide&Deep [9], DCN [73], DeepFM [20], xDeepFM [36], AFN [10], AutoInt [65] and DCNv2 [74], (2) **PLM-based** models: CTR-BERT [50], P5 [18] and PTab [43], (3) **ID+PLM** models that combine ID-based model and PLM: CTRL [35], MoRec [81].

4.1.4 Implementation Details. All of our experiments are performed on 8 NVIDIA Tesla V100 GPUs with PyTorch [52]. In the modality alignment pretraining stage, we set the text and tabular mask ratio r_{text} and r_{tab} both to 15%. Unless specified otherwise, we adopt TinyBERT [26] as the PLM, and DCNv2 [74] as the ID-based model. During pretraining, the model is trained for 30 epochs with the AdamW [47] optimizer and a batch size of 1024. The learning rate is initialized as 5e-5 followed by a cosine decay strategy. The number of noise features K is 25 in Eq. 10. The temperature τ is 0.7 in Eq. 11. In the adaptive finetuning stage, we adopt the Adam [27] optimizer with learning rate selected from {1e-5,5e-5,1e-4,5e-4,1e-3,5e-3}. The finetuning batch size is 256 for MovieLens-1M and BookCrossing, and is 2048 for GoodReads.

For all ID-based models in the baselines or FLIP framework, the embedding size is fixed to 32, and the size of DNN layers is [300,300,128]. The structure parameters and the size of PLMs in baselines are set according to their original papers. We also apply grid search to baseline methods for optimal performance.

4.2 Performance Comparison

We compare the recommendation performance of FLIP with three categories of baselines. We also include variants FLIP_{ID} and FLIP_{PLM} to reveal the effect of fine-grained alignment on a single model. The results are shown in Table 2, from which we can observe that:

- FLIP outperforms all baselines from all three categories (*i.e.*, ID-based, PLM-based, and ID+PLM) significantly, which confirms the excellence of our proposed fine-grained feature-level alignment and adaptive finetuning approach for CTR prediction.
- FLIP_{ID} surpasses all ID-based baselines, while FLIP_{PLM} outperforms all PLM-based baselines. These phenomena prove that fine-grained modality alignment can leverage the benefits of both models and boost their own performance.
- CTRL and MoRec generally outperform other baselines due to their integration of ID-based models and PLMs. However, they either overlook the fine-grained alignment between IDs and tokens or ignore cross-modal interactions, thereby degrading the performance. FLIP addresses these shortcomings by bridging ID-based models and PLMs through jointly masked tabular/language modeling, which enables fine-grained feature-level interactions between dual modalities, thus resulting in superior performance.

4.3 Compatibility Analysis

FLIP serves as a model-agnostic framework that is compatible with various backbone models. In this section, we investigate the model compatibility in terms of different ID-based models and PLMs.

4.3.1 Compatibility with ID-based models. We apply FLIP to three different ID-based models, including DeepFM, AutoInt and DCNv2, while keeping TinyBERT as the PLM. The results are listed in Table 3. We can obtain the following observations:

- First and foremost, FLIP_{ID} consistently surpasses the corresponding vanilla ID-based model by a large margin without altering the model structure or increasing the inference cost. This indicates that ID-based models of various structures can all acquire useful semantic information from PLMs through the fine-grained alignment pretraining, thereby improving performance.
- By jointly tuning the ID-based model and PLM, FLIP achieves the best performance across various ID-based backbone models significantly, demonstrating the superior compatibility of FLIP in terms of ID-based models.

4.3.2 Compatibility with PLMs. Similarly, we keep DCNv2 as the ID-based model, and select PLMs of different sizes, including TinyBERT (14.5M), RoBERTa-Base (125M) [46] , and RoBERTa-Large (355M) [46]. The results are in Table 4, from which we find that:

- As the size of PLM grows, the performance of FLIP_{PLM} continuously increases and even achieves a better AUC 0.7972 on BookCrossing with RoBERTa-Large compared with the vanilla DCNv2. A larger model size would lead to larger model capacity and better language understanding ability, thus benefiting the final predictive performance.
- While increasing the PLM's size is expected to yield more notable performance improvements, the advantages of scaling up gradually taper off. For instance, the improvement from RoBERTa-Base to RoBERTa-large is significantly smaller than the improvement from TinyBERT to RoBERTa-Base.
- FLIP and FLIP_{ID} outperform the DCNv2 model consistently and significantly, highlighting FLIP's ability to adapt seamlessly to different PLM sizes and architectures.

4.4 Ablation Study

We conduct ablation experiments for better understanding the contributions of different components in our proposed FLIP.

Firstly, we evaluate the impact of pretraining objectives (*i.e.*, *MLM*, *MTM*, *ICL*) by eliminating them from the pretraining stage.

Table 2: The overall performance of different models from three categories (*i.e.*, ID-based, PLM-based, and ID+PLM). For each type of models, the best result is given in bold, and the second-best value is underlined. *Rel.Impr* denotes the relative AUC improvement rate of our method against each baseline within each category. The symbol "$*$" indicates statistically significant improvement of FLIP over the best baseline with p-value < 0.001.

	Model	MovieLens-1M			BookCrossing			GoodReads		
		AUC	Logloss	Rel.Impr	AUC	Logloss	Rel.Impr	AUC	Logloss	Rel.Impr
ID-based	AFM	0.8449	0.3950	1.79%	0.7946	0.5116	1.06%	0.7630	0.5160	1.96%
	PNN	0.8546	0.3946	0.63%	0.7956	0.5131	0.93%	0.7725	0.5055	0.70%
	Wide&Deep	0.8509	0.3957	1.07%	0.7951	0.5116	1.00%	0.7684	0.5090	1.24%
	DCN	0.8509	0.4056	1.07%	0.7957	0.5108	0.92%	0.7693	0.5086	1.12%
	DeepFM	0.8539	0.3905	0.71%	0.7947	0.5122	1.04%	0.7671	0.5138	1.41%
	xDeepFM	0.8454	0.3934	1.72%	0.7953	0.5108	0.97%	0.7720	0.5079	0.77%
	AFN	0.8525	0.3868	0.88%	0.7932	0.5139	1.24%	0.7654	0.5118	1.64%
	AutoInt	0.8509	0.4013	1.07%	0.7953	0.5118	0.97%	0.7716	0.5071	0.82%
	DCNv2	0.8548	0.3893	0.61%	0.7956	0.5103	0.93%	0.7724	0.5057	0.72%
	FLIP$_{\text{ID}}$ (Ours)	**0.8600***	**0.3802***	-	**0.8030***	**0.5043***	-	**0.7779***	**0.5014***	-
PLM-based	CTR-BERT	0.8304	0.4131	1.88%	0.7795	0.5300	1.65%	0.7385	0.5316	1.23%
	P5	0.8304	0.4173	1.88%	0.7801	0.5261	1.58%	0.7365	0.5336	1.51%
	PTab	0.8426	0.4195	0.41%	0.7880	0.5384	0.56%	0.7456	0.5268	0.27%
	FLIP$_{\text{PLM}}$ (Ours)	**0.8460***	**0.4127***	-	**0.7924***	**0.5304***	-	**0.7476***	**0.5255***	-
ID+PLM	CTRL	0.8572	0.3838	0.57%	0.7985	0.5101	0.95%	0.7741	0.5045	0.59%
	MoRec	0.8561	0.3896	0.70%	0.7990	0.5087	0.89%	0.7731	0.5085	0.72%
	FLIP (Ours)	**0.8621***	**0.3788***	-	**0.8061***	**0.5004***	-	**0.7787***	**0.5001***	-

Table 3: The compatibility w.r.t. different ID-based models. The PLM is fixed as TinyBERT. *N/A* means to train the vanilla ID-based model from scratch. For each ID-based model, the best result is in bold, and the second-best is underlined.

ID-based Model	Finetuning Strategy	MovieLens-1M		BookCrossing	
		AUC	Logloss	AUC	Logloss
DeepFM	N/A	0.8539	0.3905	0.7947	0.5122
	FLIP$_{\text{ID}}$	0.8600	**0.3752**	0.8021	0.5083
	FLIP$_{\text{PLM}}$	0.8445	0.4132	0.7892	0.5324
	FLIP	**0.8615**	0.3758	**0.8033**	**0.5031**
AutoInt	N/A	0.8509	0.4013	0.7943	0.5118
	FLIP$_{\text{ID}}$	0.8583	0.3827	0.7992	0.5092
	FLIP$_{\text{PLM}}$	0.8453	0.4126	0.7909	0.5338
	FLIP	**0.8600**	**0.3807**	**0.8011**	**0.5050**
DCNv2	N/A	0.8548	0.3893	0.7956	0.5103
	FLIP$_{\text{ID}}$	0.8600	0.3802	0.8030	0.5043
	FLIP$_{\text{PLM}}$	0.8460	0.4127	0.7924	0.5304
	FLIP	**0.8621**	**0.3788**	**0.8061**	**0.5004**

Table 4: The compatibility w.r.t. different PLMs. The ID-based model is fixed as DCNv2. For each type of PLM, the best result is in bold, and the second-best is underlined.

PLM	Finetuning Strategy	MovieLens-1M		BookCrossing	
		AUC	Logloss	AUC	Logloss
DCNv2		0.8548	0.3893	0.7956	0.5103
TinyBERT	FLIP$_{\text{ID}}$	0.8600	0.3802	0.8030	0.5043
	FLIP$_{\text{PLM}}$	0.8460	0.4127	0.7924	0.5304
	FLIP	**0.8621**	**0.3788**	**0.8061**	**0.5004**
RoBERTa-Base	FLIP$_{\text{ID}}$	0.8601	0.3784	0.8038	0.5030
	FLIP$_{\text{PLM}}$	0.8499	0.4053	0.7961	0.5292
	FLIP	**0.8634**	**0.3770**	**0.8083**	**0.4995**
RoBERTa-Large	FLIP$_{\text{ID}}$	0.8603	0.3774	0.8036	0.5035
	FLIP$_{\text{PLM}}$	0.8506	0.4045	0.7972	0.5280
	FLIP	**0.8650**	**0.3757**	**0.8092**	**0.4986**

Table 5: The results of ablation study.

Model Variant	MovieLens-1M		BookCrossing	
	AUC	Logloss	AUC	Logloss
FLIP	**0.8621**	**0.3788**	**0.8061**	**0.5004**
w/o *MLM*	0.8610	0.3791	0.8042	0.5035
w/o *MTM*	0.8615	0.3806	0.8053	0.5032
w/o *ICL*	0.8618	0.3790	0.8039	0.5026
w/o *MLM&MTM*	0.8598	0.3815	0.8020	0.5044
w/o *MLM&MTM&ICL*	0.8593	0.3810	0.8008	0.5061
w/o *Field-level Masking*	0.8605	0.3785	0.8054	0.5015
w/o *Joint Reconstruction*	0.8618	0.3820	0.8050	0.5036

Note that removing all three objectives means that the pretraining stage does not exist. The results are reported in Table 5.

- As we can observe, the optimal performance is achieved when three losses are deployed simultaneously, and removing each loss will degrade performance, while eliminating all losses results in the lowest performance. These phenomena demonstrate that each component contributes to the final performance.
- Removing ICL (w/o *ICL*) obtains better performance than removing MLM&MTM (w/o *MLM&MTM*), indicating that joint modeling for MLM&MTM tasks can learn meaningful cross-modal alignments even without ICL.

Next, to further investigate the effect of jointly masked modality modeling, we design the following two variants:

- **w/o *Field-level Masking***: We replace the field-level masking for textual data with the common random token-level masking, as discussed in Section 3.3.1.

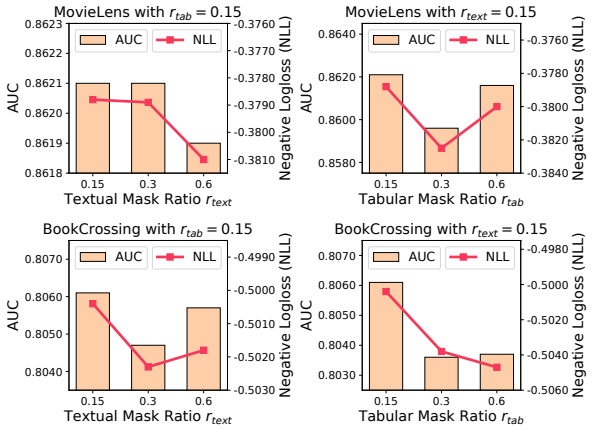


Figure 3: The hyperparameter study on textual mask ratio r_{text} (left column) and tabular mask ratio r_{tab} (right column) on MovieLens-1M (top) and BookCrossing (bottom) datasets.

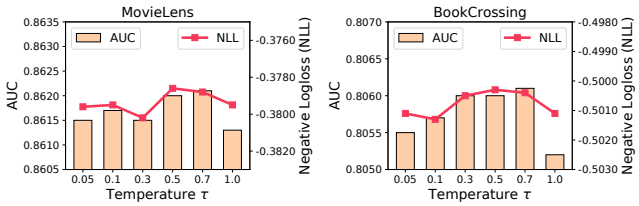


Figure 4: The hyperparameter study on the temperature τ.

- **w/o *Joint Reconstruction***: The reconstruction of one masked modality depends only on itself and no longer relies on the help from the other modality.

The results are shown in Table 5. The performance drops, especially on the Logloss metric, when we either remove the field-level masking strategy or eliminate the joint reconstruction. Such a phenomenon demonstrates the importance of fine-grained feature-level alignment, which ensures the feature-level communication and interaction between ID-based models and PLMs for dual modalities.

4.5 Hyperparameter Study

4.5.1 The Impact of Mask Ratio r_{text} and r_{tab}. Since there are two independent mask ratio for textual and tabular modalities, we fix the mask ratio on the one side and alter the mask ratio on the other side from $\{0.15, 0.3, 0.6\}$. The results are in Figure 3, from which we observe that the best performance is generally achieved when both mask ratios are relatively small (*i.e.*, 0.15). The reason is that excessive masking might lead to ambiguity in the target modality data, which hurts the model pretraining [13, 46]. So we set the r_{text} and r_{tab} both to 15%.

4.5.2 The Impact of Temperature Hyperparameter τ. The temperature τ controls the sharpness of estimated distribution. We select τ from $\{0.05, 0.1, 0.3, 0.5, 0.7, 1.0\}$, and report the results in Figure 4. When the temperature gradually grows, the performance increases first and then decreases, with the optimal temperature choice between 0.5 and 0.7. As suggested in previous works [8, 69], a too small temperature would only concentrate on the nearest sample pairs with top scores, and a too large temperature might lead to a flat distribution where all negative sample pairs receive almost the same amount of punishment. Both of them will hurt the final performance [35, 69]. Therefore, we choose $\tau = 0.7$ in our approach.

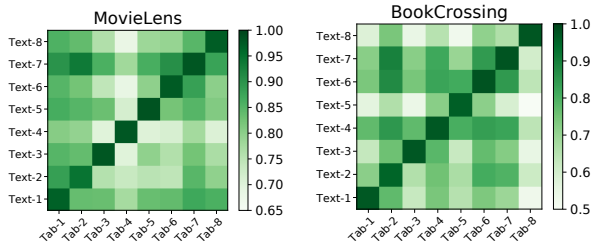
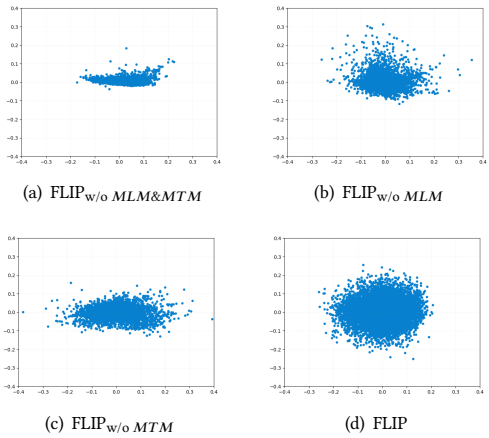


Figure 5: Visualization of similarities between the sample representations of masked textual and tabular data. "Text-f" and "Tab-f" denote that we mask the f-th field of the input data of textual or tabular modalities, respectively.



(a) $\text{FLIP}_{\text{w/o } MLM\&MTM}$ (b) $\text{FLIP}_{\text{w/o } MLM}$

(c) $\text{FLIP}_{\text{w/o } MTM}$ (d) FLIP

Figure 6: The visualization of feature ID embeddings learned by different model variants on MovieLens-1M. We use SVD to project the feature embedding matrix into 2D data.

4.6 Analysis on Fine-grained Alignment

4.6.1 Feature-level Alignment. We conduct case studies to further explore the fine-grained feature-level alignment established during pretraining. For a dual-modality input pair $(\mathbf{x}_i^{text}, \mathbf{x}_i^{tab})$, we first perform field-level data masking to mask out each field respectively, resulting in F pairs of corrupted inputs $\{(\mathbf{x}_{i,(f)}^{text}, \mathbf{x}_{i,(f)}^{tab})\}_{f=1}^{F}$, where $\mathbf{x}_{i,(f)}^{text}$ and $\mathbf{x}_{i,(f)}^{tab}$ denote that we solely mask the f-field of the textual or tabular data. Then we employ the PLM and ID-based model to encode them into F pairs of normalized sample representations $\{(\mathbf{z}_{i,(f)}^{text}, \mathbf{z}_{i,(f)}^{tab})\}_{f=1}^{F}$. Next, we compute the mutual similarity scores (measured by dot product) over each cross-modal representation pair, and visualize the heat map in Figure 5.

We can observe that the similarity score varies a lot for cross-modal input pairs with different masked fields, and the top-similar score is achieved for pairs with the same masked field (*i.e.*, on the diagonal). This indicates that FLIP can perceive the changes among field-level features for both modalities, and further maintain a one-to-one feature-level correspondence between the two modalities.

4.6.2 Visualization. we investigate the impact of the fine-grained alignment on the feature ID embedding learning. Following previous work [53], we adopt SVD [28] decomposition to project the learned ID embeddings from the ID-based model into 2D data. In Figure 6, we visualize the ID embeddings learned by four variants $\text{FLIP}_{\text{w/o } MLM\&MTM}$, $\text{FLIP}_{\text{w/o } MLM}$, $\text{FLIP}_{\text{w/o } MTM}$ and FLIP. Note that $\text{FLIP}_{\text{w/o } MLM\&MTM}$ indicates the instance-level contrastive learning (ICL) variant without the MLM and MTM objectives. We have the following observations:

- The feature ID embeddings learned by $\text{FLIP}_{\text{w/o } MLM\&MTM}$ (*i.e.*, ICL) collapse into a narrow cone, suffering from severe representation degeneration problem. In contrast, FLIP obtains feature ID embeddings with more distributed latent patterns, thus better capturing the feature diversity. This highlights the effectiveness of fine-grained alignment in promoting representation learning and mitigating the representation degeneration.
- Comparing FLIP with $\text{FLIP}_{\text{w/o } MLM}$ or $\text{FLIP}_{\text{w/o } MTM}$, we find that removing either MLM or MTM objective makes the learned embeddings more indistinguishable, demonstrating the necessity of dual alignments between modalities for learning effective representations.

5 RELATED WORK

5.1 ID-based Models for CTR Prediction

CTR prediction serves as a core function module in personalized online services, including online advertising, recommender systems, etc [82]. ID-based models follow the common design paradigm: embedding layer, feature interaction (FI) layer, and prediction layer. These models take as input one-hot features of tabular data modality, and employ various interaction functions to capture collaborative signals among features. Due to the significance of FI in CTR prediction, numerous studies focus on designing novel structures for the FI layer to capture more informative and complex feature interactions. Wide&Deep [9] combines a wide network and a deep network to achieve the advantages of both. DCN [73] and DCNv2 [74] improve Wide&Deep by replacing the wide part with a cross network to learn explicit high-order feature interactions. DeepFM [20] combines DNN and FM, and xDeepFM [36] extends DeepFM by using a compressed interaction network (CIN) to capture feature interactions in a vector-wise way. Furthermore, explicit or implicit interaction operators are designed to improve performance, such as the productive operator [55], the logarithmic operator [10] and the attention operator [65, 79].

5.2 PLMs for CTR Prediction

Pretrained Language Models (PLMs) have demonstrated exceptional success in a wide range of tasks, owing to their extensive knowledge and strong reasoning abilities [3, 41, 57, 58]. Inspired by these achievements, the application of PLMs for recommender systems has received more attention [6, 7, 14, 15, 32, 38, 44, 70, 76, 87, 87]. Different from one-hot encoding in ID-based models, this line of research needs to convert the raw data into textual modality, thus retaining the original semantic information of features.

Recent efforts to adapt PLMs for CTR prediction have yielded several breakthroughs [2, 11, 25, 31, 48, 71, 75, 81, 85]. For instance, CTR-BERT [50] leverages a two-tower structure with a user BERT and item BERT for final CTR prediction. PTab [43] pretrains a BERT model by masked language modeling, and then finetunes it on downstream tasks. P5 [18] uniformly converts different recommendation tasks into text generation tasks with T5 backbone [59]. However, PLMs encounter challenges in capturing field-wise collaborative signals and discerning features with subtle textual differences. Recent works [35, 60, 81] have tried to address this problem by adding text features into the ID-based model or integrating ID information into the PLM. However, they either overlook cross-modal interactions or depend solely on coarse-grained ICL, which is insufficient to capture fine-grained feature-level modality interactions.

Therefore, we propose to conduct fine-grained feature-level alignment between ID-based models and PLMs via the jointly masked tabular/language modeling, which learns fine-grained interactions between tabular IDs and word tokens by the way of mask-and-predict. In the finetuning stage, we also propose to jointly tune both models, and thus leverage the benefits of both textual and tabular modalities to achieve superior CTR prediction performance.

6 CONCLUSION

In this paper, we propose FLIP, a model-agnostic framework that achieves the fine-grained alignment between ID-based models and PLMs. We view tabular data and transformed textual data as dual modalities, and design a novel fine-grained modality alignment pretraining task. Specifically, the joint reconstruction for masked language/tabular modeling (with specially designed masking strategies) and cross-modal contrastive learning are employed to accomplish feature-level and instance-level alignments, respectively. Furthermore, we propose to jointly finetune the ID-based model and PLM to achieve superior performance by adaptively combining the outputs of both models. Extensive experiments show that FLIP outperforms state-of-the-art baselines on three real-world datasets, and is highly compatible with various ID-based models and PLMs.

ACKNOWLEDGMENTS

The Shanghai Jiao Tong University team is partially supported by National Natural Science Foundation of China (62177033, 62076161) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). The work is also sponsored by Huawei Innovation Research Program. We thank MindSpore [1] for the partial support of this work.

REFERENCES

[1] 2020. MindSpore. https://www.mindspore.cn/
[2] 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. *arXiv preprint arXiv:2305.00447* (2023).
[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
[4] Peter Bühlmann. 2012. Bagging, boosting and ensemble methods. *Handbook of computational statistics: Concepts and methods* (2012), 985–1022.
[5] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. 2010. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research* 11, 4 (2010).
[6] Junyi Chen. 2023. A Survey on Large Language Models for Personalized and Explainable Recommendations. *arXiv e-prints*, Article arXiv:2311.12338 (Nov. 2023), arXiv:2311.12338 pages. arXiv:2311.12338 [cs.IR]
[7] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2023. When large language

models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376* (2023).
[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
[9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
[10] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3609–3616.
[11] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).
[12] Xinyi Dai, Jianghao Lin, Weinan Zhang, Shuai Li, Weiwen Liu, Ruiming Tang, Xiuqiang He, Jianye Hao, Jun Wang, and Yong Yu. 2021. An adversarial imitation click model for information retrieval. In *Proceedings of the Web Conference 2021*. 1809–1820.
[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[14] Kounianhua Du, Jizheng Chen, Jianghao Lin, Yunjia Xi, Hangyu Wang, Xinyi Dai, Bo Chen, Ruiming Tang, and Weinan Zhang. 2024. DisCo: Towards Harmonious Disentanglement and Collaboration between Tabular and Semantic Space for Recommendation. *arXiv preprint arXiv:2406.00011* (2024).
[15] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046* (2023).
[16] Lingyue Fu, Jianghao Lin, Weiwen Liu, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. An F-shape Click Model for Information Retrieval on Multi-block Mobile Pages. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1057–1065.
[17] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009* (2019).
[18] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.
[19] Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. 2023. VIP5: Towards Multimodal Foundation Models for Recommendation. *arXiv preprint arXiv:2305.14302* (2023).
[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 26th International Joint Conference on Artificial Intelligence*. 1725–1731.
[21] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 297–304.
[22] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.
[23] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging Large Language Models for Sequential Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1096–1102.
[24] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 5549–5581.
[25] Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, and Yongfeng Zhang. 2023. UP5: Unbiased Foundation Model for Fairness-aware Recommendation. *arXiv preprint arXiv:2305.12090* (2023).
[26] 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*. 4163–4174.
[27] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[28] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*. ACM, 426–434.
[29] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864* (2020).
[30] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems* 34 (2021), 9694–9705.
[31] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. *arXiv preprint arXiv:2305.13731* (2023).
[32] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023. Large Language Models for Generative Recommendation: A Survey and Visionary Discussions. *arXiv preprint arXiv:2309.01157* (2023).
[33] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, et al. 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*. 3292–3301.
[34] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, et al. 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*. 3292–3301.
[35] Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. CTRL: Connect Tabular and Language Model for CTR Prediction. *arXiv preprint arXiv:2306.02841* (2023).
[36] 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 *KDD*. 1754–1763.
[37] Jianghao Lin, Bo Chen, Hangyu Wang, Yunjia Xi, Yanru Qu, Xinyi Dai, Kangning Zhang, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. ClickPrompt: CTR Models are Strong Prompt Generators for Adapting Language Models to CTR Prediction. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 3319–3330.
[38] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817* (2023).
[39] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1259–1268.
[40] Jianghao Lin, Yanru Qu, Wei Guo, Xinyi Dai, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. MAP: A Model-agnostic Pretraining Framework for Click-through Rate Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1379–1389.
[41] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3497–3508.
[42] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. 2019. Feature generation by convolutional neural network for click-through rate prediction. In *WWW*. 1119–1129.
[43] Guang Liu, Jie Yang, and Ledell Wu. 2022. PTab: Using the Pre-trained Language Model for Modeling Tabular Data. *arXiv preprint arXiv:2209.08060* (2022).
[44] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems. *arXiv preprint arXiv:2302.03735* (2023).
[45] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[46] 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 preprint arXiv:1907.11692* (2019).
[47] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).
[48] Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559* (2023).
[49] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[50] 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*.
[51] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances*

in neural information processing systems 32 (2019).
[53] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.
[54] 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.
[55] 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 *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1149–1154.
[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
[57] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).
[58] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[59] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.
[60] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation Learning with Large Language Models for Recommendation. *arXiv preprint arXiv:2310.15950* (2023).
[61] Steffen Rendle. 2010. Factorization machines. In *ICDM*.
[62] Robert E Schapire. 2013. Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*. Springer, 37–52.
[63] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 253–260.
[64] Yaya Shi, Xu Yang, Haiyang Xu, Chunfeng Yuan, Bing Li, Weiming Hu, and Zheng-Jun Zha. 2022. EMScore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17929–17938.
[65] 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*. 1161–1170.
[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[67] Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*. 86–94.
[68] Mengting Wan, Rishabh Misra, Ndapandula Nakashole, and Julian McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2605–2610.
[69] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2495–2504.
[70] Hangyu Wang, Jianghao Lin, Bo Chen, Yang Yang, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards Efficient and Effective Unlearning of Large Language Models for Recommendation. *arXiv preprint arXiv:2403.03536* (2024).
[71] Lei Wang and Ee-Peng Lim. 2023. Zero-Shot Next-Item Recommendation using Large Pretrained Language Models. *arXiv preprint arXiv:2304.03153* (2023).
[72] Peng Wang, Jiang Xu, Chunyi Liu, Hao Feng, Zang Li, and Jieping Ye. 2020. Masked-field Pre-training for User Intent Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2789–2796.
[73] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
[74] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*. 1785–1797.
[75] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. LLMRec: Large Language Models with Graph Augmentation for Recommendation. *arXiv preprint arXiv:2311.00423* (2023).
[76] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A Survey on Large Language Models for Recommendation. *arXiv preprint arXiv:2305.19860* (2023).
[77] Yunjia Xi, Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Rui Zhang, Ruiming Tang, and Yong Yu. 2023. A Bird's-eye View of Reranking: from List Level to Page Level. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1075–1083.
[78] Yunjia Xi, Weiwen Liu, Jianghao Lin, 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 preprint arXiv:2306.10933* (2023).
[79] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3119–3125.
[80] Xin Xin, Bo Chen, Xiangnan He, Dong Wang, Yue Ding, and Joemon M Jose. 2019. CFM: Convolutional Factorization Machines for Context-Aware Recommendation.. In *IJCAI*, Vol. 19. 3926–3932.
[81] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835* (2023).
[82] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435* (2017).
[83] Wenxuan Zhang, Hongzhi Liu, Yingpeng Du, Chen Zhu, Yang Song, Hengshu Zhu, and Zhonghai Wu. 2023. Bridging the Information Gap Between Domain-Specific Model and General LLM for Personalized Recommendation. *arXiv preprint arXiv:2311.03778* (2023).
[84] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. *IJCAI* (2021).
[85] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488* (2023).
[86] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).
[87] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107* (2023).
[88] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. 22–32.