CTRL: Connect Collaborative and Language Model for CTR Prediction

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

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

Lu Hou
houlu3@huawei.com
China Huawei Noah's Ark Lab

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

ABSTRACT

Traditional click-through rate (CTR) prediction models convert the tabular data into one-hot vectors and leverage the collaborative relations among features for inferring user's preference over items. This modeling paradigm discards essential semantic information. Though some works like P5 and M6-Rec have explored the potential of using Pre-trained Language Models (PLMs) to extract semantic signals for CTR prediction, they are computationally expensive and suffer from low efficiency. Besides, the beneficial collaborative relations are not considered, hindering the recommendation performance. To solve these problems, in this paper, we propose a novel framework **CTRL**, which is industrial friendly and model-agnostic with superior inference efficiency. Specifically, the original tabular data is first converted into textual data. Both tabular data and converted textual data are regarded as two different modalities and are separately fed into the collaborative CTR model and pre-trained language model. A cross-modal knowledge alignment procedure is performed to fine-grained align and integrate the collaborative and semantic signals, and the lightweight collaborative model can be deployed online for efficient serving after fine-tuned with supervised signals. Experimental results on three public datasets show that CTRL outperforms the state-of-the-art (SOTA) CTR models significantly. Moreover, we further verify its effectiveness on a large-scale industrial recommender system.

ACM Reference Format:
Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. CTRL: Connect Collaborative and Language Model for CTR Prediction . In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Click-through rate (CTR) prediction is an important task for recommender systems and online advertising [15, 42], where users' willingness to click on items is predicted based on the historical behavior data. The estimated CTR is leveraged to determine whether an item can be displayed to the user. Consequently, accurate CTR prediction service is critical to improving user experience, product sales, and advertising platform revenue [69].

For the CTR prediction task, historical data is organized in the form of tabular data. During the evolution of recommendation models, from the early Matrix Factorization (MF) [30], to shallow machine learning era models like Logistic Regression (LR) [7] and Factorization Machine (FM) [51], and continuing to the deep neural models such as DeepFM [17] and DIN [71], **collaborative signals** have always been the core of recommendation modeling, which leverages the feature co-occurrences and label signals for inferring user preferences. After encoding the tabular features into one-hot features [20], the co-occurrence relations (i.e., interactions) of the features are captured by various human-designed operations (e.g., inner product [17, 48], outer product [35, 61], non-linear layer [6, 68], etc.). By modeling these collaborative signals explicitly or implicitly, the relevance between users and items can be inferred.

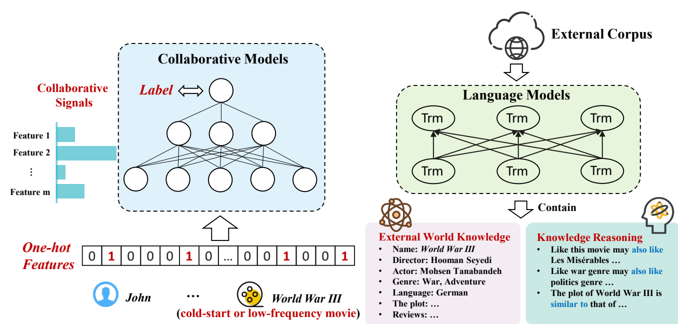


Figure 1: The external world knowledge and reasoning capabilities of pre-trained language models facilitate recommendations.

However, the collaborative based modeling paradigm discards the semantic information among the original features due to the one-hot feature encoding process. Therefore, for cold-start scenarios or low-frequency long-tailed features, the recommendation performance is unsatisfactory, limited by the inadequate collaborative relations [41]. For example, in Figure 1, when inferring the click probability of user `John` over a cold start movie `World War III`, the inadequate collaborative signals in historical data may impede accuracy recommendation. Recently, some works are proposed to address this drawback by involving Pre-trained Language Models (PLMs) to model **semantic signals**, such as P5 [14], M6-Rec [8], CTR-BERT [44], TALLRec [1], PALR [5]. These works feed the original textual features directly into the language models for recommendation, rather than using one-hot encoded features. On the one hand, the linguistic and semantic knowledge in PLMs helps to extracting the semantic information within the original textual features [37]. On the other hand, the *external world knowledge* such

*Co-first authors with equal contributions.

Conference'17, July 2017, Washington, DC, USA
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

as the director, actors, even story plot and reviews for the movie `World War III`, as well as *knowledge reasoning capability* in Large Language Models (LLMs) provide general knowledge beyond training data and scenarios [70], thus enlightening a new technological path for recommender systems.

Although remarkable progress has been achieved, the existing semantic signals based solutions suffer from several shortcomings: 1) Making predictions based on semantics merely without traditional collaborative modeling can be suboptimal [14], because the feature co-occurrence patterns and user-item interactions are indispensable indicators for personalized recommendation [17], which are not yet well equipped for PLMs [38, 70]. 2) Online inferences of language models are computationally expensive due to their complex structures. To adhere to low-latency constraints, massive computational resources and engineering optimizations are involved, hindering large-scale industrial applications [8, 14].

Therefore, incorporating PLMs into recommendation systems to capture semantic signal confronts two major challenges:

- How to combine the collaborative signals with semantic signals to boost the performance of recommendation?
- How to ensure efficient online inference without involving extensive engineering optimizations?

To solve these two challenges above, inspired by the recent works in contrastive learning, we propose a novel framework to **C**onnec**t** Collabo**r**ative and **L**anguage Model (**CTRL**) for CTR prediction, which consists of two stages: **Cross-modal Knowledge Alignment** stage, and **Supervised Fine-tuning** stage. Specifically, the raw tabular data is first converted into textual data by human-designed prompts, which can be understood by language models. Then, the original tabular data and generative textual data are regraded as different modalities and fed into the collaborative CTR model and pre-trained language model, respectively. We execute a cross-modal knowledge alignment procedure, meticulously aligning and integrating collaborative signals with semantic signals. Finally, the collaborative CTR model is fine-tuned on the downstream task with supervised signals. During the online inference, only the lightweight fine-tuned CTR model is pushed for serving without the language model, thus ensuring efficient inference.

Our main contributions are summarized as follows:

- We first propose a novel training framework CTRL, which is capable of aligning signals from collaborative and language models, introducing semantic knowledge into the collaborative models.
- Through extensive experiments, we demonstrate that the incorporation of semantic knowledge significantly enhances the performance of collaborative models on CTR task.
- CTRL is industrial friendly, model-agnostic and can adapt with any collaborative models and PLMs, including LLMs. Moreover, the high inference efficiency is also retained, facilitating its application in industrial scenarios.
- In experiments conducted on three publicly available datasets from real-world industrial scenarios, CTRL achieved SOTA performance. Moreover, we further verify its effectiveness on a large-scale industry recommender systems.

2 RELATED WORK

2.1 Collaborative Models for Recommendation

During the evolution of recommendation models, from the early matrix factorization (MF) [30], to shallow machine learning era models like Logistic Regression (LR) [7] and Factorization Machine (FM) [51], to the deep neural models [17, 71], collaborative signals have always been the core of recommendation modeling. These collaborative based models convert the tabular features into one-hot features and leverage various interaction functions to extract feature co-occurrence relations (a.k.a. feature interactions).

Different human-designed interaction functions are proposed to improve the modeling ability of collaborative signals. Wide&Deep [6] and FNN [68] deploy the non-linear layers to extract implicit high-order interactions. PNN [48] and DeepFM [17] leverage the inner product to capture pairwise interactions with stacked and parallel structure, respectively. CFM [61] and FGCNN [35] use the convolution operation to identify the local feature interaction patterns. DCN [59] and EDCN [3] deploy cross layers to model bit-wise feature interactions, while xDeepFM [34] extends to vector-wise level with a compressed interaction layer. Moreover, some AutoML-based CTR models are proposed to search suitable feature interactions and interaction functions, such as AIM [72], and AutoFeature [28].

Though collaborative based recommendation models have been achieved significant progress, they cannot capture the semantic information of the original features, thereby hindering the prediction effect in some scenarios such as cold-start or low-frequency long-tailed features.

2.2 Semantic Models for Recommendation

Transformer-based language models, such as BERT [9], GPT-3 [2], and T5 [50], have emerged as foundational architectures in the realm of Natural Language Processing (NLP). Typically, these models undergo pre-training on voluminous web text data and subsequent fine-tuning on downstream tasks [57]. Their dominance across various NLP subdomains, such as text classification [33, 43], sentiment analysis [21, 62], intelligent dialogue [14, 46], and style transfer [23, 32], is primarily attributed to their robust capabilities for knowledge reasoning and transfer. Nevertheless, since recommender systems mainly employ tabular data, which is heterogeneous with text data, making it difficult to apply the language model straightforwardly to the recommendation task.

In recent times, innovative research trends have surfaced, exploring the viability of language models in recommendation tasks. One such development is Alibaba's M6-Rec [8], which translates users' purchasing intents into prompts, then utilizes the M6 pre-trained model for training, inference, and deployment. Another model, P5 [14], serves as a generative model tailored for recommendations, underpinning all downstream recommendation tasks into a text generation task and utilizing the T5 [50] model for training and prediction. P-Tab [37] introduces a recommendation methodology based on discriminative language models, also translating tabular data into prompts, pre-training these prompts with a Masked Language Model objective, and finally fine-tuning on downstream tasks. Concurrently, Amazon's CTR-BERT [44], a two-tower structure comprising two BERT models, encodes user and item text information respectively. More recently, a considerable upsurge in scholarly works has been observed, leveraging Large Language Models (LLMs) for recommendation systems [1, 22, 54, 66, 67]. For instance, a study by Baidu [54] investigates the possibility of using LLM for re-ranking within a search context. Similarly, RecLLM [66] addresses the issue of fairness in the application of LLMs within recommendation systems. Besides, WeChat has introduced InstructRec [67], primarily employing instruction tuning to integrate LLMs into various recommendation tasks.

However, although the above semantic-based recommendation models have exposed the possibility of application in recommender systems, they have two fatal drawbacks: 1) Discarding the superior experience accumulation in collaborative modeling presented in Section 2.1 and making prediction with semantics only may be suboptimal [14] and hinder the performance for cold-start scenarios or low-frequency long-tailed features. 2) Due to the huge number of parameters of the language models, it is quite arduous for language models to meet the low latency requirements of recommender systems, making the online deployment much more challenging. Instead, our proposed CTRL overcomes these two shortcomings by combining both collaborative and semantic signals via two-stage training paradigm.

3 PRELIMINARY

In this section, we present the collaborative based deep CTR model and reveal the deficiencies in modeling semantic information. The CTR prediction is a supervised binary classification task, whose dataset consists of several instances $(\mathbf{x}, y)$. Label $y \in \{0, 1\}$ indicates user's actual click action. Feature $\mathbf{x}$ is multi-fields tabular feature that contains important information about the relations between users and items, including user profiles (e.g., gender, occupation), item features (e.g., category, price) as well as contextual information (e.g., time, location) [16]. Based on the instances, the traditional deep CTR models leverage the collaborative signals to estimate the probability $P(y = 1|\mathbf{x})$ for each instance.

The existing collaborative based CTR models first encode the tabular features into one-hot features, and then model the feature co-occurrence relations by various human-designed operations. Specifically, the multi-field tabular features are transformed into the high-dimensional sparse features via field-wise one-hot encoding [20]. For example, the feature (Gender=`Female`, Occupation=`Doctor`, Genre=`Sci-Fi`, ..., City=`Hong Kong`) of an instance can be represented as a one-hot vector:

$$\mathbf{x} = \underbrace{[0, 1]}_{\text{Gender}} \underbrace{[0, 0, 1, \ldots, 0]}_{\text{Occupation}} \underbrace{[0, 1, 0, \ldots, 0]}_{\text{Genre}} \ldots \underbrace{[0, 0, 1, \ldots, 0]}_{\text{City}}. \quad (1)$$

Generally, deep CTR models follow an "Embedding & Feature interaction" paradigm [3, 16]. The high-dimensional sparse one-hot vector is mapped into a low-dimensional dense space via an embedding layer with embedding look-up operation. Specifically, for the i-th feature, the corresponding feature embedding $\mathbf{e}_i$ can be obtained via $\mathbf{e}_i = \mathbf{E}_i \mathbf{x}_i$, where $\mathbf{E}_i$ is the embedding matrix. Following, feature interaction layers are proposed to capture the explicit or implicit feature co-occurrence relations. Massive effort has been made in designing specific interaction functions, such as product [17, 48], cross layer [3, 34, 59], non-linear layer [6, 68], and attention layer [71]. Finally, the predictive CTR score $\hat{y}$ is obtained via an output layer and optimized with the ground-truth label y through the widely-used Binary Cross Entropy (BCE).

As we can observe, collaborative based CTR models leverage the one-hot encoding to convert the original tabular data into one-hot vectors as E.q.(1), discarding the semantic information among the feature fields and values[1]. By doing this, the feature semantics is lost and the only signals that can be used for prediction are the feature co-occurrence relations, which is suboptimal when the relations are weak in some scenarios such as cold-start or low-frequency long-tailed features. Therefore, introducing the language model to capture the essential semantic information is conducive to compensating the information gaps and improving the performance.

4 METHOD

As depicted in the Figure 3, the proposed CTRL is a two-stage training paradigm. The first stage is **Cross-modal Knowledge Alignment**, which feeds paired tabular data and textual data from two modalities into the collaborative model and the language model respectively, and then aligns them with the contrastive learning objective. The second stage is the **Supervised Fine-tuning** stage, where the collaborative model is fine-tuned on the downstream task with supervised signals.

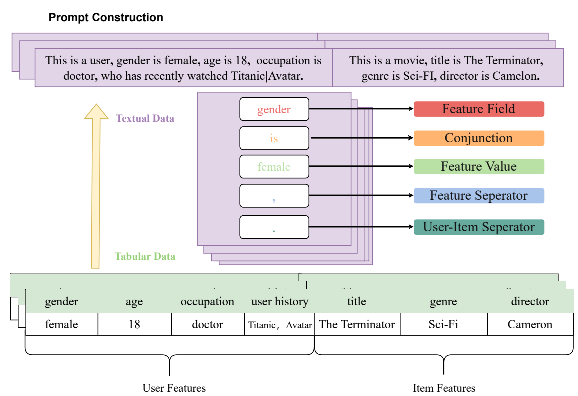


Figure 2: The overall process of prompt construction.

4.1 Prompt Construction

Before introducing the two-stage training paradigm, we first present the prompt construction process. As illustrated in Figure 2, to obtain textual prompt data, we design prompt templates to transform the tabular data into textual data for each training instance. As mentioned in previous work [8, 14], a proper prompt should contain sufficient semantic information about the user and the item. For example, user's profiles such as age, identity, interests, and behaviors can be summarized in a single sentence. Besides, item's description sentence can be organized with the features such as color, quality, and shape. For this purpose, we design the following template to construct the prompts:

This is a user, gender is female, age is 18, occupation is doctor, who has recently watched Titanic|Avatar.

[1]We use "feature field" to represent a class of features following [16] and "feature value" to represent a certain value in a specific field. For example, `occupation` is a "feature field" and `doctor` is one of the "feature value".

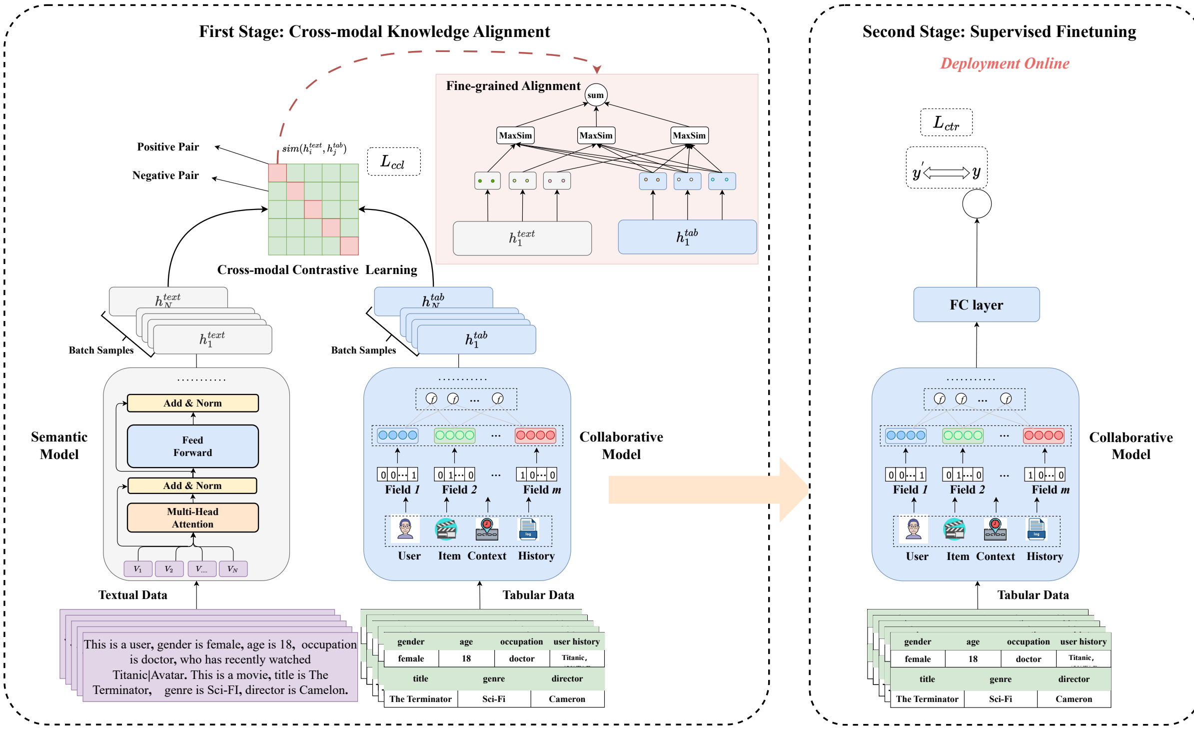


Figure 3: An intuitive illustration of the CTRL, which is a two-stage framework, where in the first stage, cross-modal contrastive learning is used to fine-grained align knowledge of the two modalities. In the second stage, the lightweight collaborative model is fine-tuned on downstream tasks. Red square represents a positive pair in the batch, while green square represents a negative pair.

> This is a movie, title is The Terminator, genre is Sci-FI, director is Camelon.

In this prompt, the first sentence "`This is a user, gender is female, age is 18, occupation is doctor, who has recently watched Titanic|Avatar.`" describes the user-side features, including his/her profiles such as age, gender, occupation, and history behaviors, etc. The following sentence "`This is a movie, title is The Terminator, genre is Sci-FI, director is Camelon.`" describes the item-side features such as title, category, director, etc. In the practical implementation, we use the period **"."** to separate the user-side and item-side descriptions, the comma **","** to separate each feature, and vertical bar **"|"** to separate each user's historical behavior[2]. We also explore the effect of different prompts, of which results are presented in Section 5.7.2.

4.2 Cross-modal Knowledge Alignment

As mentioned before, existing collaborative-based recommendation models [52, 59] leverage the feature co-occurrence relations to infer users' preferences over items, facilitating the evolution of recommendation. Besides, the pre-trained language models [9] specialize in capturing the semantic signals of recommendation scenarios with the linguistic and external world knowledge [14]. In order to combine the modeling capabilities of both collaborative-based models and pre-trained language models, as well as ensure efficient online inference, CTRL proposes an implicit information integration method via contrastive learning [4, 13], where cross-modal knowledge (i.e., tabular and textual information) between collaborative and semantic space is aligned.

4.2.1 Cross-modal Contrastive Learning. The cross-modal contrastive procedure is presented in Figure 3. First, the collaborative model and semantic model (a.k.a., pre-trained language model) are utilized to encode the tabular and textual data for obtaining the corresponding representations, respectively. Specifically, let $\mathcal{M}_{col}$ denotes collaborative model, and $\mathcal{M}_{sem}$ denotes semantic model, for a instance $\mathbf{x}$, $\mathbf{x}^{tab}$ denotes the tabular form, and $\mathbf{x}^{text}$ denotes the textual form of the same instance that is obtained after the prompt construction process. The instance representations under collaborative and semantic space can be presented as $\mathcal{M}_{col}(\mathbf{x}^{tab})$ and $\mathcal{M}_{sem}(\mathbf{x}^{text})$, respectively. To convert the unequal length representations into a same dimension, a linear projection layer is designed and the transformed instance representations can be obtained as follows:

$$\mathbf{h}^{tab} = \mathcal{M}_{col}(\mathbf{x}^{tab})\mathbf{W}^{tab} + \mathbf{b}^{tab}, \tag{2}$$

$$\mathbf{h}^{text} = \mathcal{M}_{sem}(\mathbf{x}^{text})\mathbf{W}^{text} + \mathbf{b}^{text}, \tag{3}$$

where $\mathbf{h}^{tab}$ and $\mathbf{h}^{text}$ are the transformed collaborative and semantic representations for the same instance $\mathbf{x}$, $\mathbf{W}^{tab}, \mathbf{W}^{text}$ and $\mathbf{b}^{tab}, \mathbf{b}^{text}$ are the transform matrices and biases of the linear projection layers.

Then, the contrastive learning is used to align the instance representations under different latent space, which is proved effective in both unimodal [4, 13] and cross-modal [49] representation learning. The assumption behind is that, under a distance metric, the correlated representations should be constrained to be close, and vice

[2]Note that this step is performed in the data process pipeline, and generating millions of textual prompts only takes a few seconds with parallel computing. For datasets with hundreds of features, a subset of significant features are selected to generate prompts.

versa should be far away. We employ InfoNCE [18] with in-batch negative sampling to align two representations under collaborative and semantic space for each instance. Denote $\mathbf{h}_k^{text}, \mathbf{h}_k^{tab}$ are the representations of two modals for the k-th instance, the textual-to-tabular contrastive loss can be formulated as:

$$\mathcal{L}^{textual2tabular} = -\frac{1}{N}\sum_{k=1}^{N} log \frac{exp(sim(\mathbf{h}_k^{text}, \mathbf{h}_k^{tab})/\tau)}{\sum_{j=1}^{N} exp(sim(\mathbf{h}_k^{text}, \mathbf{h}_j^{tab})/\tau)}, \tag{4}$$

where τ is a temperature coefficient and N is the number of instances in a batch. Besides, function $sim(\cdot,\cdot)$ measures the similarity between two vectors, which is calculated by:

$$sim(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i^\top \mathbf{h}_j}{\|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\|}. \tag{5}$$

In order to avoid spatial bias towards collaborative modal, motivated by the Jensen–Shannon (J-S) divergence [11], we also design a tabular-to-textual contrastive loss for uniformly aligning into a multimodal space, which is shown as:

$$\mathcal{L}^{tabular2textual} = -\frac{1}{N}\sum_{k=1}^{N} log \frac{exp(sim(\mathbf{h}_k^{tab}, \mathbf{h}_k^{text})/\tau)}{\sum_{j=1}^{N} exp(sim(\mathbf{h}_k^{tab}, \mathbf{h}_j^{text})/\tau)}. \tag{6}$$

Finally, the cross-modal contrastive learning loss $\mathcal{L}_{ccl}$ is defined as the average of $\mathcal{L}^{textual2tabular}$ and $\mathcal{L}^{tabular2textual}$, and all the parameters including collaborative model $\mathcal{M}_{col}$ and semantic model $\mathcal{M}_{sem}$ are trained.

$$\mathcal{L}_{ccl} = \frac{1}{2}(\mathcal{L}^{textual2tabular} + \mathcal{L}^{tabular2textual}). \tag{7}$$

4.2.2 Fine-grained Alignment. As mentioned above, CTRL leverages the cross-modal contrastive learning to perform knowledge alignment, where the quality of alignment is measured by the similarity function E.q.(5). However, this approach models the global similarities merely and ignores fine-grained information alignment between the two modalities $\mathbf{h}^{tab}$ and $\mathbf{h}^{text}$. To address this issue, CTRL adopts a fine-grained cross-modal alignment method.

Specifically, both collaborative and semantic representations $\mathbf{h}^{tab}$ and $\mathbf{h}^{text}$ are first transformed into M sub-spaces to extract informative knowledge from different aspects. Taking the collaborative representation $\mathbf{h}^{tab}$ as example, the m-th sub-representation $\mathbf{h}_m^{tab}$ is denoted as:

$$\mathbf{h}_m^{tab} = \mathbf{W}_m^{tab}\mathbf{h}^{tab} + \mathbf{b}_m^{tab}, \qquad m = 1, 2, \ldots, M, \tag{8}$$

where $\mathbf{W}_m^{tab}$ and $\mathbf{b}_m^{tab}$ are the transform matrix and bias vector for the m-th sub-space, respectively. Similarly, the m-th sub-representation for semantic representation is denoted as $\mathbf{h}_m^{text}$.

Then, the fine-grained alignment is performed by calculating the similarity score, which is conducted as a sum of maximum similarity over all sub-representations, shown as:

$$sim(\mathbf{h}_i, \mathbf{h}_j) = \sum_{m_i=1}^{M} \max_{m_j \in 1,2,\ldots,M} \{(\mathbf{h}_{i,m_i})^T \mathbf{h}_{j,m_j}\}, \tag{9}$$

where $\mathbf{h}_{i,m}$ is the m-th sub-representation for representation $\mathbf{h}_i$. By modeling fine-grained similarity over the cross-modal spaces, CTRL allows for more detailed alignment within instance representations to better integrate knowledge.

4.3 Supervised Fine-tuning

After the cross-modal knowledge alignment stage, the collaborative knowledge and semantic knowledge are aligned and aggregated in a hybrid representation space, where the relations between features is mutually strengthened. In this stage, CTRL further fine-tunes the collaborative models on different downstream tasks (CTR prediction task in this paper) with supervised signals.

At the top of the collaborative model, we add an extra linear layer with random initialization, acting as the output layer for final prediction $\hat{y}$. The widely-used Binary Cross Entropy (BCE) loss is deployed to measure the classification accuracy between the prediction score $\hat{y}$ and the ground-truth label y, which is defined as follows:

$$\mathcal{L}_{ctr} = -\frac{1}{N}\sum_{k=1}^{N}(y_k log(\hat{y}_k) + (1 - y_k) log(1 - \hat{y}_k)), \tag{10}$$

where y_k and $\hat{y}_k$ are the ground-truth label and the model prediction score of the k-th instance. After the supervised fine-tuning stage, only the lightweight collaborative model will be deployed online for serving, thus ensuring efficient online inference.

4.4 Discussion

4.4.1 Semantics Discussion. Involving external semantic information for enhancing recommendation performance has been explored for a long time. In addition to the pre-trained language models (PLMs) we discuss here, knowledge graph (KG) [60] and pre-trained text/image embeddings [36] are also commonly used. 1) In comparison with the KG-based methods, PLMs-based models do not rely on the construction of knowledge graph, which is time-consuming and resource-consuming. Besides, plenty of important features are not included in KG, such as user-related features and ID features. Instead, CTRL deploys the PLMs to model semantic relations in the form of natural language, which is more economical and efficient. Furthermore, knowledge reasoning capability in PLMs [2] is a unique advantage. 2) Compared with the pre-trained text/image embeddings methods, our proposed CTRL overcomes the fatal problem that the learned representations are not in the same latent space with the collaborative models, resulting in significant information collapse, which has been well solved by CTRL via the fine-grained cross-modal knowledge alignment.

4.4.2 Efficiency Discussion. As is known to us, the training process of PLMs is extremely time-consuming, especially for large language models (LLMs) such as GLM [10], LLaMA [55], GPT-3 [2], making it difficult to update models with the latest data without customized engineering optimizations and hindering industrial applications. However, recommendation models need to fit latest data to learn users' changing preferences, thus encountering the dilemma. CTRL overcomes this issue with a two-stage training procedure, which is industrial friendly with high training and inference efficiency. During the cross-modal knowledge alignment stage, only the relations between features $\mathbf{x}$ is modeled without involving supervised signal y, where the distribution $P(\mathbf{X})$ is relatively stable [47]. Therefore, the model update frequency of the first stage can be reduced so that the two-stage model update process can be organized in the form of pipeline (single update in the first stage with multiple updates in the second stage), thus ensuring the high training efficiency.

Table 1: Basic statistics of datasets.

Dataset	Users	Items	User Field	Item Field	Samples
MovieLens-1M	6,040	3,952	5	3	1,000,000
Amazon(Fashion)	749,232	196,637	2	4	883,636
Alibaba	1,061,768	785,597	9	6	26,557,961

As for online serving, only the lightweight collaborative model will be deployed, retaining efficient online inference as traditional recommendation models.

5 EXPERIMENTS

In this section, we describe the experiments in detail, including the experimental settings, comparison with the SOTA baseline models and the corresponding analysis. Through experiments such as performance comparisons and efficiency studies, we aim to answer the following research questions about our proposed CTRL framework:

- RQ1: How does CTRL perform compared to the SOTA models?
- RQ2: Can CTRL meet the requirement of low inference latency, which is vital for industrial recommender systems?
- RQ3: How well CTRL aligns the collaborative and semantic spaces, which reflects the effect of knowledge integration.
- RQ4: Does CTRL have sufficient compatibility? To what extent does it affect the performance when applying with different semantics and collaborative models, including LLMs?
- RQ5: Can CTRL be applied in large-scale industrial scenarios?

5.1 Experimental Setting

5.1.1 Datasets. In the experiment, we deploy three large-scale public datasets, which are MovieLens, Amazon (Fashion), and Taobao, whose statistics are summarized in Table 1. **MovieLens Dataset**[3] is a movie recommendation dataset and following previous work [52], we consider samples with ratings less than 3 as negative, samples with scores greater than 3 as positive, and remove neutral samples, i.e., rating equal to 3. **Amazon Dataset**[4] [45] is a widely-used benchmark dataset [48, 64, 65, 71] and our experiment uses a subset Fashion following [71]. We take the items with rating of greater than 3 as positive and the rest as negative. **Alibaba Dataset**[5] [12] is a Taobao ad click dataset. For the MovieLens and Amazon datasets, following previous work [31], we divide the train, validation, and test sets by user interaction time in the ratio of 8:1:1. For the Alibaba dataset, we divide the datasets according to the official implementation [71], and the data from the previous seven days are used as the training and validation samples with 9:1 ratio, and the data from the eighth day are used for test.

5.1.2 Evaluation Metrics. Following previous work [25, 52, 71], we use two popular metrics to evaluate the performance. The area under the ROC curve (**AUC**) measures the probability that the model will assign a higher score to a randomly selected positive item than to a randomly selected negative item. **Logloss** is a widely-used metric in binary classification to measure the distance between two distributions. As acknowledge by many studies [24, 52, 71], an improvement of **0.001** in AUC (↑) or Logloss (↓) can be regarded as significant because it will bring a large increase in the online revenue. **RelaImpr** metric [71] measures the relative improvement with respect to base model, which is defined as follows:

$$RelaImpr = (\frac{AUC(measure\ model) - 0.5}{AUC(base\ model) - 0.5} - 1) \times 100\%. \quad (11)$$

Besides, the two-tailed unpaired t-test is performed to detect a significant difference between CTRL and the best baseline.

5.1.3 Competing Models. We compare CTRL with the following models, which are classified into two classes: 1) Collaborative Models and 2) Semantic Models.

1) **Collaborative Models**: **Wide&Deep** [6] has been widely-used in industry, which contains wide part and deep part, where wide part handles the manually designed cross product features while deep part automatically extracts nonlinear relations among features. **DeepFM** [17] imposes a Factorization Machine as "wide" module in Wide&Deep saving feature engineering jobs. **DCN** [59] modifies the wide part of the Wide&Deep model with a cross network to better learn high-order feature interaction. **AutoInt** [52] employs Multi-head Self-Attention to automatically build high-order features, which acts as a strong collaborative-based baseline. Additionally, several strong baselines are considered, including **PNN** [48], **xDeepFM** [34], and **FiBiNet** [24].

2) **Semantic Models**: **P5** [14] is a semantic-based recommendation model that converts various recommendation tasks into text generation tasks by prompt learning, which uses T5 [50] as the base model. **CTR-BERT** [44] is a semantic two-tower model proposed by Amazon, which adopts two-tower BERT [9] and feeds the semantic information of user and item separately to get the prediction score. **P-Tab** [37] conducts MLM pre-training task on the training set, followed by fine-tuning on downstream score prediction tasks.

5.1.4 Implementation Details. For prompt construction process, only one type of prompt is used and the comparisons are presented in Section 5.7.2. In the first stage, we utilize AutoInt [52] as the collaborative model and RoBERTa [39] as the semantic model by default, as discriminative language models are more efficient at text representation extraction than generative models like GPT under the same parameter scale [58]. Additionally, we also evaluated the performance of the **LLM** model like ChatGLM, with the results summarized in Table 4. The mean pooling results of last hidden states are used as the semantic information representation. For the projection layer, we compress the collaborative representation and the semantic representation to 128 dimensions. Besides, the batch size of the cross-modal knowledge alignment stage is set to 6400 and the temperature coefficient is set to 0.7. The AdamW [40] optimizer is used and the initial learning rate is set to 1×10^{-5}, which is accompanied by a warm-up mechanism [19] to 5×10^{-4}. In the second stage, the learning rate of the downstream fine-tuning task is set to 0.001 with Adam [29] optimizer, and batch size is set to 2048. Moreover, Batch Normalization [26] and Dropout [53] is also applied to avoid overfitting. The feature embedding dimension d for all models are set to 32 empirically. Besides, for all collaborative models, we set the number of hidden layers L as 3 and the number of hidden units as $[256, 128, 64]$. To ensure a fair comparison, other hyperparameters such as training epochs are adjusted individually for all models to obtain the best results.

[3]https://grouplens.org/datasets/MovieLens/1m/
[4]https://jmcauley.ucsd.edu/data/amazon/
[5]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

Table 2: Performance comparison of different models. Boldface denotes the highest score and underline indicates the best result of all baselines. ⋆ represents significance level p-value < 0.05 of comparing CTRL with the best baselines. RelaImpr denotes the relative AUC improvement rate of CTRL against each baseline.

Category	Model	MovieLens			Amazon			Alibaba		
		AUC	Logloss	RelaImpr	AUC	Logloss	RelaImpr	AUC	Logloss	RelaImpr
Collaborative Models	Wide&Deep	0.8261	0.4248	3.52%	0.6968	0.4645	5.30%	0.6272	0.1943	5.19%
	DeepFM	0.8268	0.4219	3.30%	0.6969	0.4645	5.33%	0.6280	0.1951	4.53%
	DCN	<u>0.8313</u>	<u>0.4165</u>	1.90%	0.6999	0.4642	3.75%	<u>0.6281</u>	0.1949	4.45%
	PNN	0.8269	0.4220	3.27%	0.6979	0.4657	4.80%	0.6271	0.1956	5.27%
	AutoInt	0.8290	0.4178	2.61%	<u>0.7012</u>	<u>0.4632</u>	3.08%	0.6279	<u>0.1948</u>	4.61%
	FiBiNet	0.8196	0.4188	5.63%	0.7003	0.4704	3.54%	0.6270	0.1951	5.35%
	xDeepFM	0.8296	0.4178	2.43%	0.7009	0.4642	3.23%	0.6272	0.1959	5.19%
Semantic Models	P5	0.7583	0.4912	30.70%	0.6923	0.4608	7.85%	0.6034	0.3592	29.40%
	CTR-BERT	0.7650	0.4944	27.40%	0.6934	0.4629	7.24%	0.6005	0.3620	33.13%
	P-Tab	0.8031	0.4612	11.38%	0.6942	0.4625	6.80%	0.6112	0.3584	20.32%
CTRL		**0.8376**⋆	**0.4025**⋆	-	**0.7074**⋆	**0.4577**⋆	-	**0.6338**⋆	**0.1890**⋆	-

* **It is worth noting that an AUC increase of 0.001 can be considered significant improvement in CTR prediction [24, 31, 52, 71].**

5.2 Performance Comparison (RQ1)

We compare the overall performance with some SOTA collaborative and semantic models, whose results are summarized in Table 2. From which, we obtain the following observations: 1) CTRL outperforms all the SOTA baselines including semantic and collaborative models over three datasets by a significant margin, showing superior prediction capabilities and proving the effectiveness of the paradigm of combining collaborative and semantic signals. 2) In comparison to the best collaborative model, our proposed CTRL achieves a improvement in AUC of **1.90%**, **3.08%**, and **4.45%** on the three datasets respectively, which effectively demonstrates that integrating semantic knowledge into collaborative models contributes to boost performance. We attribute the significant improvements to the external world knowledge and knowledge reasoning capability in PLMs [70]. 3) The performance of existing semantic models is lower than that of collaborative models, indicating that collaborative signals and co-occurrence relations are crucial for recommender systems, and relying solely on semantic modeling is difficult to surpass the existing collaborative-based modeling scheme[14, 37, 44]. Instead, our proposed CTRL integrates the advantages of both by combining collaborative signals with semantic signals for recommendation. This approach is likely to be a key path for the future development of recommender systems.

5.3 Serving Efficiency (RQ2)

In industrial recommender systems, online model serving has strict limit, e.g., 10~20 milliseconds. Therefore, high service efficiency is essential for CTR models. In this section, we compare the model parameters and inference time of different CTR models over the Alibaba and Amazon datasets, shown in Table 3.

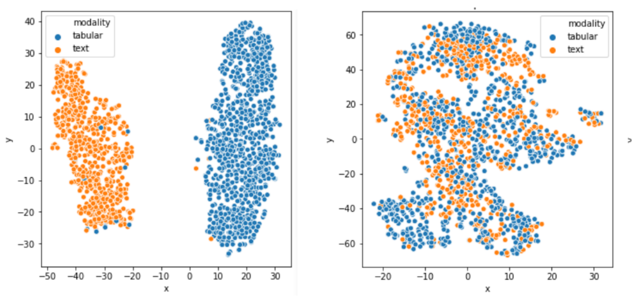
We can observe that existing collaborative-based CTR models have fewer model parameters and higher inference efficiency in comparison with semantic-based models. Moreover, the majority of parameters for the collaborative-based models are concentrated in the embedding layer while the hidden network has very few parameters, thus benefiting the online serving. On the contrary, the semantic-based models (e.g., P5 and CTR-BERT), have a larger number of parameters and lower inference efficiency due to the complex Transformer-based structures, hindering the industrial applications. Instead, for the CTRL with AutoInt as skeleton models, both model parameters and inference time are the same as the original AutoInt model, which is thanks to the decoupled training framework (semantic model is not required for online inference) and ensures the high online serving efficiency.

Table 3: Inference efficiency comparison of different models in terms of Model Inference Parameters and Inference Time over testing set with single V100 GPU. As for CTRL, only the collaborative model is need for online serving, so the number of model parameters is the same as backbone AutoInt.

	Alibaba		Amazon	
Model	Params	Inf Time	Params	Inf Time
DeepFM	8.82×10^7	18s	3.45×10^7	0.58s
DCN	8.84×10^7	19s	3.46×10^7	0.59s
AutoInt	8.82×10^7	19s	3.45×10^7	0.59s
P5	2.23×10^8	10832s	1.10×10^8	440s
CTR-BERT	1.10×10^8	4083s	1.10×10^8	144s
CTRL(ours)	8.82×10^7	19s	3.45×10^7	0.59s

5.4 Visualization of Modal Alignment (RQ3)

To study in depth the distribution of tabular representations and textual representations in the latent space before and after the cross-modal knowledge alignment, we visualize the representations in the MovieLens dataset by projecting them into a two-dimensional space using t-SNE [56], shown in Figure 4. The two colored points represent the tabular and textual representations, respectively. We can observe that, before the cross-modal knowledge alignment, the representations of the two modalities are distributed in two separate spaces and are essentially unrelated, while mapped into a unified multimodal space after the alignment. This phenomenon substantiates that CTRL aligns the space of two modalities (i.e., tabular and textual), thus injecting the semantic information and external general knowledge into the collaborative model.



(a) Before Alignment (b) After Alignment

Figure 4: Visualization of the tabular and textual representations before and after the cross-modal knowledge alignment.

5.5 Compatibility Study (RQ4)

5.5.1 Compatibility for semantic models. Specifically, for semantic models, we compare four pre-trained language models with different sizes: TinyBERT [27] with 14.5M parameters ($\text{CTRL}_{\text{TinyBERT}}$), BERT-Base [9] with 110M parameters ($\text{CTRL}_{\text{BERT}}$), RoBERTa [39] with 110M parameters ($\text{CTRL}_{\text{RoBERTa}}$), and BERT-Large with 336M parameters ($\text{CTRL}_{\text{Large}}$). Moreover, we have introduced a novel LLM model, ChatGLM [10], with 6B parameters ($\text{CTRL}_{\text{ChatGLM}}$). For $\text{CTRL}_{\text{ChatGLM}}$, during the training process, we freeze the majority of the parameters and only retained the parameters of the last layer. The experimental results are summarized in Table 4, from which we obtain some observations: 1) In comparison with the backbone model AutoInt, CTRL with different pre-trained language models achieves consistent and significant improvement, where AUC increases by **3.22%** and **3.63%** for $\text{CTRL}_{\text{ChatGLM}}$, demonstrating the effectiveness of semantics modeling and model compatibility. 2) Among the four CTRL variants ($\text{CTRL}_{\text{TinyBERT}}$, $\text{CTRL}_{\text{BERT}}$, and $\text{CTRL}_{\text{BERTLarge}}$, $\text{CTRL}_{\text{ChatGLM}}$), despite a substantial number of parameters being frozen in ChatGLM, $\text{CTRL}_{\text{ChatGLM}}$ achieves optimal performance. This phenomenon indicating that enlarging the size of the language model can imbue the collaborative model with a wealth of worldly knowledge. Furthermore, even when the parameter scale of the language model is elevated to the billion-level, it continues to make a positive contribution to the collaborative model. 3) It can be observed that while the parameter size of ChatGLM is several times that of BERTLarge, the gains are mild. Therefore, when conducting modality alignment, it is only necessary to select language models of moderate scale, such as RoBERTa. 4) Using only TinyBert can lead to a 0.005 increase in AUC, indicating that we can use lightweight pre-trained language models to accelerate model training. 4) $\text{CTRL}_{\text{RoBERTa}}$ has a better performance in the case of equal number of parameters compared to $\text{CTRL}_{\text{BERT}}$. We hypothesize that this improvement is due to RoBERTa possessing a broader range of world knowledge and a more robust capability for semantic modeling compared to BERT. This indirectly underscores the advantages of increased knowledge in facilitating the knowledge alignment process in collaborative models.

Table 4: Model compatibility study with different semantic models.

	MovieLens		Amazon	
Model	AUC	Logloss	AUC	Logloss
AutoInt (backbone)	0.8290	0.4178	0.7012	0.4632
$\text{CTRL}_{\text{TinyBERT}}$ (14.5M)	0.8347	0.4137	0.7053	0.4612
$\text{CTRL}_{\text{BERT}}$ (110M)	0.8363	0.4114	0.7062	0.4609
$\text{CTRL}_{\text{RoBERTa}}$ (110M)	0.8376	0.4105	0.7074	0.4607
$\text{CTRL}_{\text{BERTLarge}}$ (336M)	0.8380	0.4090	0.7076	0.4604
$\text{CTRL}_{\text{ChatGLM}}$ (6B)	**0.8396**	**0.4070**	**0.7085**	**0.4587**

Table 5: Model compatibility study with different collaborative models. The semantic model is set to RoBERTa.

	MovieLens		Amazon	
Model	AUC	Logloss	AUC	Logloss
Wide&Deep	0.8261	0.4348	0.6966	0.4645
$\text{CTRL}_{\text{Wide\&Deep}}$	0.8304	0.4135	0.7001	0.4624
DeepFM	0.8268	0.4219	0.6965	0.4646
$\text{CTRL}_{\text{DeepFM}}$	0.8305	0.4136	0.7004	0.4625
DCN	0.8313	0.4165	0.6999	0.4642
CTRL_{DCN}	0.8365	0.4029	0.7055	0.4615
AutoInt	0.8290	0.4178	0.7012	0.4632
$\text{CTRL}_{\text{AutoInt}}$	0.8376	0.4025	0.7063	0.4582

5.5.2 Compatibility for collaborative models. Besides, we apply CTRL to different collaborative models, including Wide&Deep, DeepFM, DCN, and AutoInt. From Table 5, we can observe that CTRL achieves remarkable improvements with different collaborative models consistently. The average improvements over RelaImpr metric are **1.31%** for Wide&Deep, **1.13%** for DeepFM, **1.57%** for DCN, and **2.61%** for AutoInt respectively, which demonstrates the effectiveness and model compatibility.

5.6 Application in Industry System (RQ5)

In this section, we deploy CTRL in a large-scale industrial recommender system to verify its effectiveness. We collect and sample one month of user behavior data from a large-scale recommendation platform, where millions of user logs are generated daily. More than **30** distinct features are used, including user profile features (e.g., department), user behavior features (e.g., list of items clicked by the user), item original features (e.g., item title) and statistical features (e.g., the number of clicks on the item), as well as contextual features (e.g., time). We compare CTRL model (backbone model AutoInt and RoBERTa) with SOTA models. For the semantic models, we choose CTR-BERT and P5, while for the collaborative models, we choose DeepFM, AutoInt, and DCN, which are widely-applied in large-scale recommender systems.

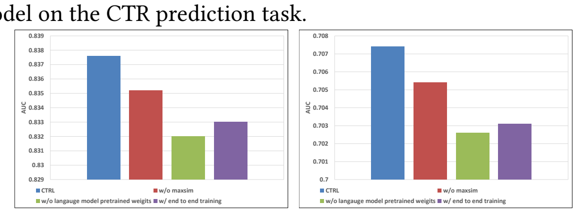
The performance results are presented in Table 6. It is evident that CTRL outperforms the baseline models significantly in terms of AUC and Logloss, thereby demonstrating its superior performance. By incorporating the modeling capabilities of both the semantic and collaborative models, CTRL achieves a significant performance improvement over both collaborative models and semantic models. Moreover, according to the results in Table 3, CTRL would not increase any serving latency compared to the backbone collaborative model, which is an industrial friendly framework with high accuracy and low inference latency.

Table 6: Industrial recommender system performance comparison.

Category	Model	AUC	Logloss	RelaImpr
Collaborative	DeepFM	0.6547	0.1801	8.79%
	AutoInt	<u>0.6586</u>	<u>0.1713</u>	6.12%
	DCN	0.6558	0.1757	8.02%
Semantic	CTR-BERT	0.6484	0.1923	13.41%
	P5	0.6472	0.1974	14.33%
CTRL		**0.6683***	**0.1606***	-

5.7 Ablation Study and Hyperparameter Analysis

5.7.1 Ablation Study Analysis. In this section, we conduct ablation experiments to better understand the importance of different components. We first replace the maxsim similarity with cosine similarity; then we remove the the pre-trained language model weights. Moreover, we also investigate the impact of end-to-end training, which combines the two-stage process into a single stage(i.e., cross-modal knowledge alignment and CTR prediction tasks are trained together). From Figure 5, we observe the following results: 1) When we remove the weights of the pre-trained language model, the loss in model performance is quite significant. This demonstrates that the primary source of improvement in the collaborative model's performance is attributed to the world knowledge and semantic modeling capabilities of the language model, rather than solely due to contrastive learning. 2) After replacing cosine similarity with maxsim similarity, there is a degradation in the model performance. This indicates that fine-grained alignment facilitates the collaborative model in learning semantic representations. 3) We observe that the performance of end-to-end training is inferior to the pre-training and fine-tuning paradigm of CTRL. We conjecture that this may be due to the multi-objective setting in end to end training paradigm, which hampers the performance of the collaborative model on the CTR prediction task.

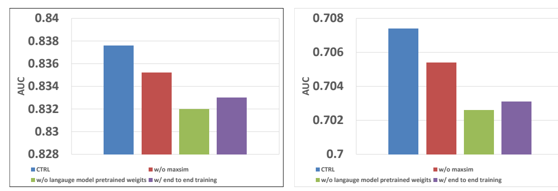


(a) Movielens (b) Amazon

Figure 5: The results of the ablation study.

5.7.2 Prompt Analysis. In this subsection, we explore the impact of different prompts construction methods on training CTRL. We believe that this exploration will inspire future work on how to better construct prompts. Below are several rules for constructing prompts: 1) Transform user and item features into natural language text that can be easily understood; 2) Remove auxiliary text descriptions and connect feature fields and values with "-" directly; 3) Remove the feature fields and transform all the feature values into a single phrase; 4) Mask the feature fields with a meaningless unified word "Field"; 5) Replace the separator "-" with separator ":".

We pre-train CTRL on these prompts and then fine-tune on the CTR prediction task with the collaborative model, whose results are shown in Figure 6. From Figure 6, we can obtain the following observations: 1) Prompt-1 performs significantly better than all prompts, which indicates that constructing prompts in the form of natural language is beneficial for modeling. 2) The performance of Prompt-3 is weaker than Prompt-2, which confirms the importance of semantic information of feature fields, the lack of which will degrade the performance of the model remarkably. Meanwhile, the performance of Prompt-3 is weaker than Prompt-4, indicating that prompt with rules is stronger than prompt without rules. 3) The performance of Prompt-2 and Prompt-5 are similar, suggesting that the difference of connectives between feature field and feature value has little effect on the performance. Based on these findings, we can identify the following characteristics of designing a good prompt: 1) including feature field such as age, gender, etc.; 2) having fluent and grammatically correct sentences and containing as much semantic information as possible.



(a) Movielens (b) Amazon

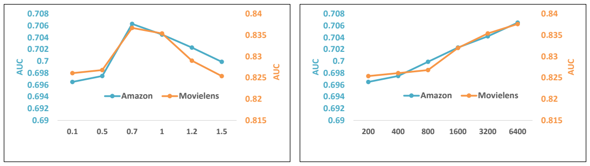
Figure 6: Performance on MovieLens and Amazon datasets in terms of different prompts.

5.7.3 The Impact of Contrastive Learning Temperature Coefficient. To explore the effect of different temperature parameters in the cross-modal knowledge alignment contrastive learning, we implement experiments on MovieLens and Amazon datesets, and the results are in Figure 7(a). From the results we can get the following observations: 1) The temperature coefficient in contrastive learning has an obvious impact on the performance. As the temperature coefficient increases, the performance will have a tendency to improve first and then decrease, indicating that increasing coefficient within a certain range is beneficial to improve the performance. 2) For both MovieLens and Amazon datasets, the optimal temperature coefficient is below 1 in our experiments, which has also been verified in previous work [49, 63].

5.7.4 The Impact of Contrastive Learning Batch Size. We also explore the impact of different batch sizes, and the results are shown in Figure 7(b). We can observe that as the batch size increases, the performance is also improved on both datasets, which indicates that increasing the batch size during the contrastive learning pre-training is conducive to achieving better cross-modal knowledge alignment effect and improving the prediciton accuracy.

6 CONCLUSION

In this paper, we reveal the importance of both collaborative and semantic signals for CTR prediction and present CTRL, an industrial friendly and model-agnostic framework with high inference efficiency. CTRL treats the tabular data and converted textual data



(a) Temperature Coefficient (b) Batch sizes

Figure 7: Influence of different contrastive learning temperature coefficient and batch sizes.

as two modalities and leverages the contrastive learning for fine-grained knowledge alignment and integration. Finally, the lightweight collaborative model can be deployed online for efficient serving after fine-tuned with supervised signals. Our experiments demonstrate that CTRL outperforms state-of-the-art collaborative and semantic models while maintaining good inference efficiency. Future work includes exploring the application on other downstream tasks, such as sequence recommendation and explainable recommendation.

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. arXiv preprint arXiv:2305.00447 (2023).

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. https://doi.org/10.48550/ARXIV.2005.14165

[3] Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. 2021. Enhancing Explicit and Implicit Feature Interactions via Information Sharing for Parallel Deep CTR Models. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 3757–3766.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In Proceedings of ICML. PMLR, 1597–1607.

[5] Zheng Chen. 2023. PALR: Personalization Aware LLMs for Recommendation. arXiv preprint arXiv:2305.07622 (2023).

[6] 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.

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

[8] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. https://doi.org/10.48550/ARXIV.2205.08084

[9] 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).

[10] 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). 320–335.

[11] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In International symposium onInformation theory, 2004. ISIT 2004. Proceedings. IEEE, 31.

[12] 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 preprint arXiv:1704.05194 (2017).

[13] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Proceedings of EMNLP. 6894–6910.

[14] 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.

[15] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress.

[16] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2910–2918.

[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017).

[18] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of AISTATS. JMLR Workshop and Conference Proceedings, 297–304.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 770–778.

[20] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In Proceedings of the eighth international workshop on data mining for online advertising. 1–9.

[21] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using bert. In Proceedings of the 22nd nordic conference on computational linguistics. 187–196.

[22] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large Language Models are Zero-Shot Rankers for Recommender Systems. arXiv preprint arXiv:2305.08845 (2023).

[23] Zhiqiang Hu, Roy Ka-Wei Lee, Charu C Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. ACM SIGKDD Explorations Newsletter 24, 1 (2022), 14–45.

[24] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET. In Proceedings of the 13th ACM Conference on Recommender Systems. ACM. https://doi.org/10.1145/3298689.3347043

[25] 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. 169–177.

[26] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning. PMLR, 448–456.

[27] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351 (2019).

[28] Farhan Khawar, Xu Hang, Ruiming Tang, Bin Liu, Zhenguo Li, and Xiuqiang He. 2020. Autofeature: Searching for feature interactions and their architectures for click-through rate prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 625–634.

[29] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/ARXIV.1412.6980

[30] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401, 6755 (1999), 788–791.

[31] 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.

[32] Xiangyang Li, Xiang Long, Yu Xia, and Sujian Li. 2022. Low Resource Style Transfer via Domain Adaptive Meta Learning. arXiv preprint arXiv:2205.12475 (2022).

[33] Xiangyang Li, Yu Xia, Xiang Long, Zheng Li, and Sujian Li. 2021. Exploring text-transformers in aaai 2021 shared task: Covid-19 fake news detection in english. In Combating Online Hostile Posts in Regional Languages during Emergency Situation: First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers 1. Springer, 106–115.

[34] 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. 1754–1763.

[35] 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 The World Wide Web Conference. 1119–1129.

[36] Fan Liu, Huilin Chen, Zhiyong Cheng, Anan Liu, Liqiang Nie, and Mohan Kankanhalli. 2022. Disentangled Multimodal Representation Learning for Recommendation. IEEE Transactions on Multimedia (2022).

[37] 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).

[38] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. arXiv preprint arXiv:2304.10149

(2023).
[39] 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).
[40] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017).
[41] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1563–1573.
[42] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 1222–1230.
[43] Marcin Michał Mirończuk and Jarosław Protasiewicz. 2018. A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications 106 (2018), 36–54.
[44] 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.
[45] Jianmo Ni, Jiacheng Li, and Julian 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). 188–197.
[46] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155 (2022).
[47] Jiarui Qin, Jiachen Zhu, Bo Chen, Zhirong Liu, Weiwen Liu, Ruiming Tang, Rui Zhang, Yong Yu, and Weinan Zhang. 2022. RankFlow: Joint Optimization of Multi-Stage Cascade Ranking Systems as Flows. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 814–824.
[48] 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.
[49] 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.
[50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://doi.org/10.48550/ARXIV.1910.10683
[51] Steffen Rendle. 2010. Factorization machines. In 2010 IEEE International conference on data mining. IEEE, 995–1000.
[52] 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.
[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1 (2014), 1929–1958.
[54] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. arXiv preprint arXiv:2304.09542 (2023).
[55] 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]
[56] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).
[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems 30 (2017).
[58] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. arXiv:1905.00537 [cs.CL]
[59] 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.
[60] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 950–958.
[61] 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.
[62] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint arXiv:1904.02232 (2019).
[63] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. FILIP: fine-grained interactive language-image pre-training. arXiv preprint arXiv:2111.07783 (2021).
[64] Yantao Yu, Weipeng Wang, Zhoutian Feng, and Daiyue Xue. 2021. A Dual Augmented Two-tower Model for Online Large-scale Recommendation. (2021).
[65] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation.. In IJCAI. 4213–4219.
[66] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. arXiv preprint arXiv:2305.07609 (2023).
[67] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. arXiv preprint arXiv:2305.07001 (2023).
[68] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In European conference on information retrieval. Springer, 45–57.
[69] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 1077–1086.
[70] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language models as recommender systems: Evaluations and limitations. (2021).
[71] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 1059–1068.
[72] Chenxu Zhu, Bo Chen, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2021. AIM: Automatic Interaction Machine for Click-Through Rate Prediction. IEEE Transactions on Knowledge and Data Engineering (2021).